

# Multiagentensimulation der Wissensweitergabe in Organisationen am Beispiel von Individualsoftwareherstellern

Dissertation zur Erlangung des Dr. rer. pol.



Martin-Luther-Universität Halle-Wittenberg

Juristische und Wirtschaftswissenschaftliche Fakultät

Institut für Wirtschaftsinformatik und Operations Research



Vorgelegt von: René Peinl

Wohnhaft in: Neutauperlitzer Weg 35  
95028 Hof

Telefon: 09281/840177

Abgabe am: 16.01.2008

Verteidigung am: 03.11.2008

Betreuer: Prof. Dr. Ronald Maier  
Prof. Dr. Ralf Peters

**urn:nbn:de:gbv:3-000014779**

[<http://nbn-resolving.de/urn/resolver.pl?urn=nbn%3Ade%3Agbv%3A3-000014779>]



---

## Inhaltsverzeichnis

Inhaltsverzeichnis .....	III
Abkürzungsverzeichnis .....	X
Abbildungsverzeichnis .....	XII
Tabellenverzeichnis .....	XV
1 Einführung .....	1
1.1 Problemstellung .....	1
1.2 Zielsetzung .....	2
1.3 Vorgehen .....	3
1.4 Überblick über die Arbeit .....	5
2 Wissensmanagement .....	7
2.1 Wissen .....	7
2.1.1 Wurzeln und Semiotik .....	8
2.1.2 Merkmale .....	10
2.1.3 Definitionen für Wissen .....	13
2.1.4 Verwandte Begriffe .....	14
2.1.5 Zusammenfassung .....	16
2.2 Wissensweitergabe und weitere Wissensprozesse .....	17
2.2.1 Überblick über Wissensprozesse .....	18
2.2.2 Beschreibung der Wissensprozesse .....	24
2.2.3 Wissensweitergabe im Detail .....	27
2.2.4 Ein Modell der Wissensweitergabe .....	33
2.3 Wissensmanagement in der Softwareentwicklung .....	36
2.3.1 WM als strategische Aufgabe .....	37
2.3.2 WM als technische Aufgabe .....	39
2.3.3 WM als personenorientierte Aufgabe .....	43
2.3.4 WM als organisatorische Aufgabe .....	44

## Inhaltsverzeichnis

---

2.3.5	Wissensintensive Tätigkeiten und Organisationen .....	46
2.3.6	Softwareentwicklung als Anwendungsdomäne .....	49
2.4	Wissensmanagement-Instrumente für Softwareentwickler .....	54
2.4.1	WM-Instrumente in der Literatur .....	57
2.4.2	Beschreibung der identifizierten Instrumente .....	59
2.4.3	Eigene Vorschläge .....	65
2.4.4	Fazit .....	68
2.5	Erfolgsmessung im Wissensmanagement .....	68
2.5.1	Messansätze .....	69
2.5.2	Integration der Ansätze .....	72
2.5.3	Erfolgsmessung der Wissensweitergabe .....	73
2.6	Zusammenfassung .....	75
3	Erklärungsansätze für Wissensweitergabe und Arbeitsverhalten .....	76
3.1	Einflussfaktoren für Wissensweitergabe .....	76
3.2	Gelegenheits- und Nutzungskontext .....	86
3.2.1	Arbeitssituation .....	88
3.2.2	Nutzung in Geschäftsprozessen .....	92
3.2.3	Diskussion .....	94
3.3	Quellen- und Empfängerkontext .....	95
3.3.1	Entscheiden .....	96
3.3.2	Motive und Motivation .....	101
3.3.3	Expertise .....	108
3.3.4	Lernen .....	113
3.3.5	Gedächtnis .....	121
3.3.6	Wahrnehmen .....	126
3.3.7	Emotion .....	129
3.3.8	Diskussion .....	134
3.4	Inhaltskontext .....	135
3.4.1	Art des Wissens .....	136

---

3.4.2	Eigenschaften des Wissens.....	139
3.4.3	Diskussion .....	143
3.5	Übermittlungskontext .....	144
3.5.1	Kommunikation .....	144
3.5.2	Technologievermittelte Kommunikation .....	150
3.5.3	Diskussion .....	154
3.6	Beziehungskontext .....	155
3.6.1	Soziale Netzwerke .....	156
3.6.2	Normen.....	161
3.6.3	Macht .....	165
3.6.4	Reputation .....	168
3.6.5	Vertrauen.....	171
3.6.6	Diskussion .....	175
3.7	Organisationskontext .....	176
3.7.1	Organisationsstruktur.....	178
3.7.2	Gruppen- und Teamarbeit .....	183
3.7.3	Projektarbeit .....	188
3.7.4	Personalführung und -entwicklung.....	192
3.7.5	Diskussion .....	194
3.8	Domänenmodell der Wissensweitergabe.....	195
4	Multiagentensimulation in den Sozial- und Wirtschaftswissenschaften.....	203
4.1	Computersimulation als Forschungsansatz.....	203
4.1.1	Klassische Simulationsansätze .....	205
4.1.2	Grundlagen der Simulation .....	209
4.1.3	Social Simulation.....	211
4.1.4	Multiagentensimulation .....	213
4.1.5	Vorgehen bei der Entwicklung von Simulationsmodellen.....	215
4.2	Softwareagenten in Multiagentensystemen .....	218
4.2.1	Definition und Eigenschaften.....	218

---

## Inhaltsverzeichnis

---

4.2.2	Wissensbasis .....	221
4.2.3	Multiagentensysteme (MAS) .....	222
4.2.4	Kommunikation.....	225
4.2.5	Interaktion, Koordination und Kooperation.....	228
4.3	Agentenarchitekturen für die Simulation von Menschen.....	230
4.3.1	Generische Architekturen .....	231
4.3.2	Anforderungen an Agentenarchitekturen .....	234
4.3.3	BDI: Belief, Desire, Intention.....	236
4.3.4	InteRRaP: Integration of Reactivity and Rational Planning.....	237
4.3.5	PECS: Physis, Emotion, Cognition, Status.....	238
4.3.6	ACT-R: Adaptive Control of Thought - Rational .....	240
4.3.7	SOAR: States, Operators And Reasoning .....	242
4.3.8	BRAHMS.....	244
4.3.9	H-CogAff.....	245
4.3.10	MicroPSI.....	247
4.3.11	Ergebnisse des Vergleichs .....	248
4.4	Verwandte Arbeiten .....	250
4.4.1	Simulation von Organisatorischem Lernen .....	253
4.4.2	Sozial- und Wirtschaftssimulationen als Computerspiele .....	255
4.4.3	Simulation von Wissensflüssen und Wissensweitergabe .....	259
4.4.4	Simulation von Wissensmanagement-Strategien.....	266
4.4.5	Simulation sozialer Netzwerke .....	266
4.4.6	Simulation von Teamarbeit.....	268
4.4.7	Simulation von Konsumentenverhalten.....	269
4.4.8	Simulation von Arbeitsverhalten und Arbeitseinsatz.....	270
4.4.9	Simulation von Softwareentwicklungsprozessen.....	272
4.5	Konzeptmodell der Wissensweitergabe .....	276
4.5.1	Architekturüberblick.....	278
4.5.2	Wissensrepräsentation .....	282

---

4.5.3	Umgebung .....	284
5	SimKnowledge.....	285
5.1	Agent-oriented Software Engineering .....	285
5.2	Überblick.....	287
5.3	Beschreibung nach der MASSIVE Methode .....	290
5.3.1	Aufgabensicht.....	290
5.3.2	Umgebungssicht .....	291
5.3.3	Rollensicht.....	292
5.3.4	Interaktionssicht.....	293
5.3.5	Gesellschaftssicht .....	295
5.3.6	Architektursicht .....	296
5.3.7	Systemsicht.....	308
5.4	Auswahl der Simulationssprache und des Frameworks .....	311
5.4.1	Anforderungen an MABS-Frameworks.....	312
5.4.2	Überblick über MABS-Frameworks.....	315
5.4.3	Vergleich ausgewählter MABS Frameworks.....	319
5.4.4	Fazit.....	323
5.5	Besonderheiten der Implementierung.....	324
5.5.1	Generics für das Gedächtnis.....	324
5.5.2	Agentenaktionen als innere Klassen .....	325
5.5.3	Nutzenbasierte Aktionsauswahl mit Reflection .....	326
5.5.4	Agenten als Thread .....	327
6	Simulationsstudien .....	328
6.1	Parametrisierung.....	328
6.1.1	Studien zu Skills .....	328
6.1.2	Gehaltsstudien .....	332
6.1.3	Dauer-Aufwands-Relationen.....	336
6.1.4	Fallstudie itCampus .....	338
6.2	Kalibrierung .....	342

---

## Inhaltsverzeichnis

---

6.2.1	Projektarbeit ohne Lernen.....	342
6.2.2	Erweiterung um Lernen bei der Arbeit (learning by doing) .....	343
6.2.3	Erweiterung um Lernen am PC.....	344
6.2.4	Erweiterung um Vergessen.....	344
6.2.5	Erweiterung um Lernen von Kollegen .....	345
6.2.6	Strategie zur Auswahl der Projektmitarbeiter.....	346
6.2.7	Erweiterung um Kenntnis der Teamkollegen .....	347
6.2.8	Erweiterung um Rückgabe eines Arbeitspakets an den Projektmanager ..	349
6.2.9	Zusammenfassung der Designentscheidungen .....	349
6.3	Sensitivitätsanalysen .....	351
6.4	Verifikation und Validierung.....	356
6.5	Studie 1: Skillmanagement.....	359
6.5.1	Experiment 1.1: Zuordnung der Arbeitspakete nach Kenntnis des PM....	360
6.5.2	Experiment 1.2: Zuordnung mit Skillmanagement .....	360
6.5.3	Experiment 1.3: Skillmanagement auch für Projektmitarbeiter.....	362
6.5.4	Ergebnisse.....	362
6.6	Studie 2: Dokumentation von Lernerfahrungen .....	366
6.6.1	Experiment 2.1: Keine dokumenten-orientierten WM-Maßnahmen .....	367
6.6.2	Experiment 2.2: persönliche Lernerfahrungen (Wissensdokumentation) .	367
6.6.3	Experiment 2.3: Lernerfahrungen im Team (Projektreview) .....	368
6.6.4	Ergebnisse.....	370
6.7	Diskussion der Ergebnisse.....	374
6.7.1	Erfahrungen mit der Entwicklung der Simulation .....	374
6.7.2	Agentenarchitektur zur Beherrschung der Komplexität.....	374
6.7.3	Technische und methodische Defizite.....	375
6.7.4	Inhaltliche Erfahrungen .....	376
6.7.5	Limitationen des Modells und mögliche Weiterentwicklungen .....	377
7	Zusammenfassung und Ausblick.....	382
7.1	Zusammenfassung.....	382

---

7.2	Ausblick .....	385
7.3	Fazit .....	387
Literatur.....		388
Anhang .....		419
Anhang A:	Einflussfaktoren für Wissensweitergabe nach Quelle .....	419
Anhang B:	UML-Diagramme .....	425
Anhang C:	Ergebnisse des Interviews mit dem Geschäftsführer von itCampus.....	433
Anhang D:	Fragebogen für itCampus .....	436
Anhang E:	Liste der Skills aus dem Skillmanagementsystem von itCampus .....	451

## Abkürzungsverzeichnis

AOSE	Agentenorientierte Softwareentwicklung
ABSS	Agent-based Social Simulation
ACL	Agent Communication Language
AR	Authority Ranking (Relation Modell von Fiske)
AWS	Anwendungssystem
BBL	Bahavior-Based Layer der InterRaP Agentenarchitektur
BDI	Belief Desire Intention
BPMN	Business Process Modeling Notation
CEO	Chief Executive Officer
CMS	Content Management System
COT	Computational Organization Theory
CPL	Cooperative Planning Layer der InterRaP Agentenarchitektur
CS	Communal Sharing (Relation Modell von Fiske)
CSV	Comma Separated Values
DMS	Document Management System
DPS	Distributed Problem Solving
EKI	Enterprise Knowledge Infrastructure
EM	Equality Matching (Relation Modell von Fiske)
EPML	Event-Driven Process Chain Markup Language
F&E	Forschung und Entwicklung
GIS	Geografisches Informationssystem
GPL	Gnu General Public License
HRM	Human Resource Management
IC	Intellectual Capital
IKT	Informations- und Kommunikations-Technologie
IS	Informationssystem
IT	Informationstechnologie

IuK	Informations- und Kommunikations-
KI	Künstliche Intelligenz
KIF	Knowledge Interchange Format
KQML	Knowledge Query and Manipulation Language
LPL	Local Planning Layer der InterRaP Agentenarchitektur
MABS	Multi-Agent Based Simulation (Multiagentensimulation)
MAS	Multi-Agent System (Multiagentensystem)
MP	Market Pricing (Relation Modell von Fiske)
MS	Microsoft
NDM	Naturalistic Decision Making
PECS	Physis Emotion Cognition Status
PM	Projektmanager
RSS	Really Simple Syndication, auch RDF Site Summary oder Rich Site Summary
SAAS	Social Aspects of Agent Systems
SE	Software Engineering
SkMS	Skillmanagementsystem
SVG	Scalable Vector Graphics
US	United States
VKI	Verteilte Künstliche Intelligenz
VoIP	Voice over Internet Protocol
WM	Wissensmanagement
WMS	Wissensmanagementsystem
XPDL	XML Process Definition Language

## Abbildungsverzeichnis

Abbildung 1-1: Kapitelüberblick (eigene Darstellung) .....	6
Abbildung 2-1: Bausteine des Wissensmanagement (Probst et al. 1998, S. 56) .....	19
Abbildung 2-2: Wissenskonzessionsprozesse nach (Nonaka, Takeuchi 1995) .....	20
Abbildung 2-3: Bestandteile von Wissensarbeit nach (Efimova 2004) .....	22
Abbildung 2-4: Modell der Wissensweitergabe (eigene Darstellung) .....	33
Abbildung 2-5: Architektur einer Enterprise Knowledge Infrastructure (Maier, Peinl 2005a) .....	41
Abbildung 2-6: Einflussfaktoren auf Kosten und Qualität von Software (Krishnan 1998)	52
Abbildung 2-7: Wissensmanagement-Instrumente nach (Maier et al. 2005, S. 42) .....	56
Abbildung 2-8: Startpunkte für die Erfolgsmessung im Wissensmanagement (eigene Darstellung).....	73
Abbildung 3-1: Inhaltstheorien der Motivation im Überblick (Weinert 1998, S. 157).....	103
Abbildung 3-2: Motivationsmodell von Porter und Lawler nach (Weinert 1998, S. 163)	105
Abbildung 3-3: Konzeptuelles Modell der Einflussfaktoren für Wissensweitergabe nach (Boer et al. 2002b, S. 14).....	107
Abbildung 3-4: Elemente der Aktivitätentheorie nach (Thompson 2004, S. 585) .....	118
Abbildung 3-5: Zusammenhänge von Aktivitäten, Aktionen und Operationen in Anlehnung an (Hädrich, Maier 2004, S. 8).....	119
Abbildung 3-6: Zusammenhang zwischen Umgebung, Lernen und Wissen (Rich 2002, S. 5).....	120
Abbildung 3-7: Emotionen in sozialen Austauschbeziehungen nach (Lawler, Thye 1999, S. 223).....	130
Abbildung 3-8: Zusammenhang zwischen Anforderungen, Fähigkeiten und Flow (adaptiert von Csikszentmihalyi 2004, S. 93).....	133
Abbildung 3-9: Beispiel für einen Skillbaum in der Softwareentwicklung (eigene Darstellung).....	138
Abbildung 3-10: Wissensreifungsprozess nach (Maier, Schmidt 2007) .....	142
Abbildung 3-11: Kommunikationsmodell nach (Reichwald 1993, S. 2176) .....	146
Abbildung 3-12: Kommunikationsmodell nach (Herrmann, Kienle 2004, S. 57).....	147
Abbildung 3-13: Medienauswahl in wissensintensiven Prozessen (Wong, Dalmadge 2004, S. 3).....	152
Abbildung 3-14: Organisatorische Einflussfaktoren für Softwareentwicklung (Hanne, Neu 2004, S. 6).....	178

---

Abbildung 3-15: Verbindung von Personalwesen und Leistung (vgl. Paauwe 2004, S. 57) .....	192
Abbildung 3-16: Domänenmodell der Wissensweitergabe grafisch (eigene Darstellung)	196
Abbildung 4-1: Historische Entwicklung von Simulationstechniken nach (Troitzsch 1997, S. 42) .....	206
Abbildung 4-2: Verschiedene Formen von Nachbarschaft nach (Klügl 2001, S. 55) .....	209
Abbildung 4-3: Abgrenzung der Forschungsgebiete nach (Davidsson 2002) .....	214
Abbildung 4-4: Vorgehen bei der Entwicklung einer Simulation nach (Troitzsch 1990, S. 6) .....	215
Abbildung 4-5: Vorgehensmodell MABS nach (Drogoul et al. 2002, S. 9) .....	216
Abbildung 4-6: Grafische Darstellung eines MAS nach (Drogoul et al. 2002, S. 7) .....	223
Abbildung 4-7: Klassifikation von Agentenarchitekturen nach (Klügl 2001, S. 22) .....	230
Abbildung 4-8: Reflexbasierter Agent (nach Russel, Norvig 2003, S. 47) .....	232
Abbildung 4-9: Modellbasierter Agent (nach Russel, Norvig 2003, S. 49) .....	232
Abbildung 4-10: Modell-basierter Agent mit Zielen (nach Russel, Norvig 2003, S. 50) ..	233
Abbildung 4-11: Nutzen-basierter Agent (nach Russel, Norvig 2003, S. 52) .....	233
Abbildung 4-12: BDI Verarbeitungsmodell nach (Rao, Georgeff 1995, S. 7) .....	236
Abbildung 4-13: Agentenarchitektur InterRaP nach (Fischer et al. 1994, S. 3) .....	238
Abbildung 4-14: PECS Agenten-Architektur (Urban 2000, S. 93) .....	239
Abbildung 4-15: ACT-R Architektur nach (Anderson et al. 2004, S. 1037) .....	241
Abbildung 4-16: SOAR Architektur nach (Laird et al. 1987) .....	242
Abbildung 4-17: H-CogAff Architektur nach (Sloman et al. 2004, S. 32) .....	246
Abbildung 4-18: MicroPSI Agenten-Architektur nach (Bach 2003, S. 16) .....	248
Abbildung 4-19: Aktivitäts-, Wissens-, Produktivitäts- und Gesamtmodell nach (Hanakawa et al. 2002) .....	274
Abbildung 4-20: Modell der Softwareentwicklung nach (Hanne, Neu 2004) .....	275
Abbildung 4-21: Zusammenhang zwischen Entwicklungsphasen und Ergebnis (Kirk, Tempero 2004, S. 5) .....	278
Abbildung 4-22: SimKnowledge Agentenarchitektur (eigene Darstellung) .....	279
Abbildung 5-1: Genealogie einiger AOSE Methoden in Anlehnung an (Henderson-Sellers, Giorgine 2006, S. 7) .....	285
Abbildung 5-2: Wirkungszusammenhänge in SimKnowledge (eigene Darstellung) .....	289
Abbildung 5-3: Soziales Netzwerk in SimKnowledge generiert mit AGNA (eigene Darstellung) .....	295
Abbildung 5-4: Ablauf aus Sicht eines Projekts (eigene Darstellung) .....	298
Abbildung 5-5: Zustandsübergangsdiagramm für WorkerAgent (eigene Darstellung) ....	299

---

## Abbildungsverzeichnis

---

Abbildung 5-6: Datenmodell von SimKnowledge (eigene Darstellung) .....	301
Abbildung 5-7: Klassendiagramm der Agentenklassenhierarchie (eigene Darstellung)...	302
Abbildung 5-8: InterRAP Detailmodell (Fischer et al. 1994).....	303
Abbildung 5-9: Screenshot der SimKnowledge Benutzerschnittstelle (eigene Darstellung) .....	310
Abbildung 5-10: Vor- und Nachteile von Alternativen bei Simulationssoftware nach (Brooks, Robinson 2001, S. 57).....	311
Abbildung 6-1: Zusammenhänge zwischen den Wissensbereichen (Hanakawa et al. 2002, S. 385).....	331
Abbildung 6-2: Langzeitvergleich der Gehälter im Software-Engineering nach IG-Metall .....	334
Abbildung 6-3: Zusammenhänge zwischen Skill und Projektdauer (Moore 1999, S. 6_19) .....	337
Abbildung 6-4: soziales Netzwerk nach 61 Tagen mit Kenntnis der Teamkollegen (eigene Darstellung).....	348
Abbildung 6-5: Überblick über die Modellerweiterungen (eigene Darstellung).....	350
Abbildung A-1: Einflüsse für Wissensweitergabe (Goh 2002, S. 28) .....	422
Abbildung A-2: Modell der Wissensweitergabe (Riempp 2003, S. 6).....	422
Abbildung A-3: Einflussfaktoren für Wissensweitergabe (Cummings, Teng 2003, S. 40) .....	423
Abbildung A-4: Einflussfaktoren für Wissensweitergabe nach (Becker, Knudsen 2003 S. 15).....	423
Abbildung B-5: Klassenübersicht der Agentenklassen .....	425
Abbildung B-6: Klassenübersicht Projekt-Workpackage-Skill .....	425
Abbildung B-7: Klassenübersicht Simulationsumgebung und Wahrnehmung .....	425
Abbildung B-8: Klassendiagramm SimulationEngine und Parameter.....	426
Abbildung B-9: Klassendiagramm der Agentenklassen.....	427
Abbildung B-10: Klassendiagramm der Anzeigeelemente .....	428
Abbildung B-11: Klassendiagramm Projekt-Workpackage-Skill.....	429
Abbildung B-12: Klassendiagramm der Lernumgebung.....	429
Abbildung B-13: Klassendiagramm der externen Einflüsse.....	430
Abbildung B-14: Klassendiagramm der DatabaseManager-Klasse.....	430
Abbildung B-15: Klassendiagramm der Wahrnehmungsklassen .....	431
Abbildung B-16: Klassendiagramm der Umgebungsobjekte .....	432

## Tabellenverzeichnis

Tabelle 2-1: Zusammenfassung der Sicht auf Wissen.....	17
Tabelle 2-2: Übersicht der WM-Instrumente .....	58
Tabelle 3-1: Wissensweitergabesituationen nach (Berends et al. 2004).....	79
Tabelle 3-2: Einflussfaktoren im Gelegenheitskontext.....	87
Tabelle 3-3: Einflussfaktoren im Quellkontext .....	95
Tabelle 3-4: Einflussfaktoren im Empfängerkontext.....	96
Tabelle 3-5: Relation Model Theorie, Anreize und Organisationskultur nach Geißler und Fiske .....	106
Tabelle 3-6: Vorbedingungen und Folgen von Flow-Erlebnissen.....	133
Tabelle 3-7: Informationshärte nach (Watson 2003, S. 35) .....	140
Tabelle 3-8: Charakterisierung verschiedener Medien nach (Dennis, Valacich 1999, S. 3) .....	151
Tabelle 3-9: Gründe für die Nutzung von Medien (Mäki et al. 2004, S. 6).....	153
Tabelle 3-10: Vor- und Nachteile von Kommunikationsmedien (Mäki et al. 2004, S. 7).....	153
Tabelle 3-11: Einflussfaktoren im Beziehungskontext und deren Kateogrisierung .....	156
Tabelle 3-12: Einflussfaktoren im Organisationskontext .....	176
Tabelle 4-1: Anforderungen an soziale Agente.....	234
Tabelle 4-2: Zusammenfassende Übersicht der Agentenarchitekturen .....	249
Tabelle 4-3: Zusammenfassende Übersicht der Schichtenarchitekturen .....	250
Tabelle 4-4: Überblick über verwandte Arbeiten .....	251
Tabelle 4-5: Zustandsvariablen im Spiel Die Sims .....	256
Tabelle 4-6: Zustandsvariablen im Spiel Tropico .....	259
Tabelle 4-7: Meta-Matrix nach (Carley 2001, S. 3), (Carley 2002b, S. 211), (Carley, Tsvetovat 2004, S. 24) .....	283
Tabelle 5-1: Sichten in MASSIVE .....	288
Tabelle 5-2: Sprechakte in SimKnowledge.....	294
Tabelle 5-3: Zusammenfassung des Vergleichs der MABS-Frameworks #1 .....	319
Tabelle 5-4: Zusammenfassung des Vergleichs der MABS-Frameworks #2 .....	321
Tabelle 6-1: Wissensgebiete mit großer praktischer Bedeutung (nach Lethbridge 1999).....	329
Tabelle 6-2: Gehälter 2003 und 2004 nach IG-Metall.....	333
Tabelle 6-3: Gehälter 2001 und 2002 nach c't .....	335
Tabelle 6-4: Gehälter 2003 bis 2005 nach c't .....	335

## Tabellenverzeichnis

---

Tabelle 6-5: Simulationsergebnisse nach 31 Tagen im Modell mit Lernen von Kollegen	346
Tabelle 6-6: Ergebnisse der Sensitivitätsanalysen .....	355
Tabelle 6-7: Vergleich der Skillwerte von itCampus mit den Simulationsergebnissen....	358
Tabelle 6-8: Vergleich der Skillwerte von itCampus mit den Simulationsergebnissen....	359
Tabelle 6-9: Häufigkeitsverteilung der Kontakte in der Simulation .....	359
Tabelle 6-10: Ergebnisse der Skillmanagement-Studie Teil 1 .....	363
Tabelle 6-11: Ergebnisse der Skillmanagement-Studie Teil 2 .....	364
Tabelle 6-12: Veränderung der Lernergebnisse im Zeitverlauf.....	365
Tabelle 6-13: Vergleich der Wissensweitergabepflege bei verschiedener Anzahl Agenten .....	366
Tabelle 6-14: Ergebnisse der Studie Wissensdokumentation Teil 1 .....	370
Tabelle 6-15: Ergebnisse der Studie Wissensdokumentation Teil 2.....	371
Tabelle 6-16: Verteilung der Arbeitszeit auf einzelne Tätigkeiten in 1080 Tagen Arbeitszeit.....	371
Tabelle 6-17: Entwicklung des dokumentierten Wissens im Zeitverlauf.....	373

# 1 Einführung

Im Folgenden wird die Problemstellung skizziert, welche die Motivation für die Arbeit darstellt. Anschließend wird darauf aufbauend die Zielsetzung definiert und ein dazu passendes Vorgehen abgeleitet. Den Abschluss der Einleitung bildet ein Überblick über die Arbeit.

## 1.1 Problemstellung

In den letzten 30 Jahren des 20. Jahrhunderts wurde die Geschäftswelt durch komplexere Produkte und Dienstleistungen, kürzere Produktlebenszyklen und die Zunahme geistiger im Vergleich zu körperlicher Arbeit grundlegend verwandelt (Riempp 2005, S. 6f). Dies führt dazu, dass Wissen zunehmend als wichtigste Ressource im Unternehmen angesehen werden muss (Maier 2004, S. 1). Im Forschungsgebiet Wissensmanagement (WM) werden neue Herausforderungen untersucht, die durch die Verschiebung des Arbeitsschwerpunkts von traditioneller industrieller Fertigung und Büroarbeit hin zur Wissensarbeit entstehen. Wissensarbeit zeichnet sich v.a. durch schwach strukturierte Abläufe und großen Bedarf an Wissen aus (vgl. Maier, Sametinger 2002). Um das Untersuchungsobjekt begrifflich abzugrenzen spricht die Literatur von Wissensarbeitern. Zur Unterstützung der Wissensarbeiter werden dort eine Reihe von WM-Maßnahmen diskutiert (vgl. Sveiby, Lloyd 1987; Probst et al. 1998; Holsapple 2003). Dabei handelt es sich um organisatorische Maßnahmen (Nonaka, Takeuchi 1995; Schüppel 1997), die Einführung von Softwaresystemen (Maier 2004) oder idealerweise eine Kombination von beiden (Eppler, Sukowski 2000). Ein Beispiel hierfür sind Expertenverzeichnisse, mit deren Hilfe die Wissensarbeiter sowohl ihre persönliche Kompetenzentwicklung planen können (in Zusammenarbeit mit der Personalabteilung), als auch andere Experten identifizieren können, die ihnen bei konkreten Fragestellungen weiterhelfen oder die ein Projektteam ergänzen können. Einige Arbeiten beschäftigen sich auch mit der Entwicklung von Strategien für das WM (Hansen et al. 1999), untersuchen empirisch welche Auswirkungen die Einführung von WM-Maßnahmen im Unternehmen haben, oder auf welchem Stand die Unternehmen sich derzeit befinden (Maier 2002). Die eigentlichen Prozesse der Wissensgenerierung, -verteilung und -anwendung, ebenfalls auf theoretischer Ebene diskutiert (Wiig 1993; Davenport et al. 1996), sind jedoch bisher nicht genauer betrachtet worden.

Die Fähigkeit einer Organisation am Markt erfolgreich agieren zu können wird in der Literatur in erster Linie dem in der Organisation verfügbaren Wissen zugeschrieben, wobei auch darüber hinaus gehende Begriffe wie Kompetenzen, Fähigkeiten, oder Innovationskraft verwendet werden (Argote, Ingram 2000, S. 156). Obwohl dafür Wissensweitergabe zwischen Mitarbeitern einer Abteilung und auch über Abteilungsgrenzen und Standorte hinaus notwendig ist, werden die der Wissensweitergabe zugrunde liegenden Mechanismen bisher nicht untersucht (Argote et al. 2000, S. 4), insbesondere nicht auf der Ebene der menschlichen Interaktion (Argote, Ingram 2000, S. 156). Daher fehlt es in Unternehmen bisher auch an fundierten Auswahlkriterien, welche WM-Maßnahme in einer konkreten Unternehmenssituation den größten Effekt hat, um Wissensweitergabe zu fördern und darüber hinaus den Unternehmenserfolg zu steigern.

### **1.2 Zielsetzung**

Ziel der Arbeit ist es, ausgehend von einer gegebenen Unternehmenssituation und einer Reihe von potenziell anzuwendenden WM-Maßnahmen (z.B. Einführung eines Skill-Management-Systems oder Einführung von wissensorientierten Projektnachbesprechungen) Anhaltspunkte zu liefern, wie sich der Einsatz dieser Maßnahmen auf das Unternehmen auswirkt. Insbesondere soll untersucht werden, wie sich die WM-Situation des Unternehmens entwickelt (z.B. Wissensstand, Wissensverteilung, Kommunikation, Nutzung von WM-Instrumenten) und wie sich ceteris paribus betriebswirtschaftliche Unternehmenskennzahlen (z.B. Umsatz und Gewinn) dazu verhalten. Dadurch sollen die WM-Maßnahmen gefunden werden, die am besten geeignet sind, den wirtschaftlichen Erfolg nachhaltig zu erhöhen.

Dieses Gesamtziel kann in mehrere Teilziele zerlegt werden. So muss zuerst ein geeignetes Instrument gewählt und gezeigt werden, warum dieses Instrument sich besonders eignet, um den angestrebten Erkenntnisgewinn zu erreichen. Für das in dieser Arbeit gewählte Instrument Simulation ist es anschließend nötig ein Modell zu konzipieren, welches den relevanten Ausschnitt des Realsystems überschaubar komplex aber hinreichend detailliert abbildet. Danach muss das Modell operationalisiert und in ein ausführbares Simulationssystem überführt werden. Es müssen Simulationsstudien und -experimente definiert werden, mit denen sich im Simulationssystem Sachverhalte aus dem Realsystem nachstellen lassen, so dass durch Auswertung der gewonnenen Daten und Übertragung der Ergebnisse auf das Realsystem unter Berücksichtigung der vorgenommenen Vereinfachungen und Annahmen die beschriebene Fragestellung beantwortet werden kann.

### 1.3 Vorgehen

Kilduff stellt bei der Untersuchung empirischer Arbeiten zur Wissensweitergabe fest, dass es eine strukturelle Lücke zwischen psychologischen Arbeiten gibt, die sozialwissenschaftliche Erkenntnisse vernachlässigen, und Arbeiten, die sich auf soziale Netzwerke konzentrieren und psychologische Aspekte unberücksichtigt lassen (Kilduff, Tsai 2003, S. 70). Er sieht sich selbst als organisatorischen Netzwerkforscher in der Pflicht diese Lücke zu schließen, um die Forschungsfragen umfassend beantworten zu können.

Dieser Grundsatz stellt auch eine der Maximen hinter der vorliegenden Arbeit dar. Als Wirtschaftsinformatiker ist man besonders gefordert einen interdisziplinären Forschungsansatz zu verfolgen. In dieser Arbeit wird darunter nicht nur eine Integration von Ansätzen der Wirtschaftswissenschaften und Informatik verstanden, sondern darüber hinaus das Bemühen, die relevanten Erkenntnisse der Soziologie, Psychologie und Pädagogik zu erkennen, zu verstehen und auf das eigene Forschungsgebiet anzuwenden.

Dabei wird die Analyse bewusst nicht auf die Wissensweitergabe im engeren Sinne eingegrenzt sondern auf die damit in enger Beziehung stehenden Konzepte Arbeitsverhalten, Entscheidungsfindung sowie soziale Beziehungen zwischen Personen ausgedehnt. Die grundlegende Hypothese hinter diesem Vorgehen ist, dass so wie Wissen eng mit dem Kontext verknüpft ist, in dem es entsteht und in dem es in die Wissensbasis eines Menschen eingegliedert wird (Maier 2004, S. 68; Reinhardt 2002, S. 206), auch Wissensweitergabe in enger Verbindung mit dem Kontext steht, in dem sie auftritt. Daher wird bei der Simulation der Wissensweitergabe eine isolierte Betrachtung unter Ausschluss von sozialem und Arbeitsumfeld als nicht sinnvoll erachtet.

Die Arbeit soll zu Beginn durch eine Literaturanalyse die Problemstellung aus der Sicht des Wissensmanagements heraus deutlich machen. Dabei wird v.a. auf aktuelle Bücher und Zeitschriftenartikel zurückgegriffen, die bereits Zusammenstellungen verschiedener Perspektiven auf das gesamte Gebiet Wissensmanagement oder die relevanten Teilaspekte Wissen und Wissensarbeit enthalten, z.B. (Reinhardt 2002; Holsapple 2003; Mertins et al. 2003; Maier 2004). Die Diskussion um WM-Instrumente und die Erfolgsmessung von WM-Initiativen ist hingegen noch nicht weit genug fortgeschritten, um ausführliche Zusammenfassungen hervorgebracht zu haben. Somit wird hier weitgehend auf neuere Aufsätze aus Fachzeitschriften und Konferenzbeiträge zurückgegriffen, z.B. (Neumann et al. 2002; Jennex, Olfman 2004), bzw. Bücher und Artikel untersucht, die Teilaspekte des Themas behandeln, z.B. (Disterer 2002; Wenger 2002; Asprey, Middleton 2003).

Anschließend soll durch Aufarbeiten der Literatur über Agententechnologie, Multiagentensysteme, insbesondere Multiagentensimulation (MABS, Multi Agent-Based Simulation) sowie verwandter Forschungsgebiete wie Social Simulation gezeigt werden, dass MABS ein geeignetes Mittel zur Analyse der beschriebenen Fragestellungen ist. In diesem noch relativ jungen Forschungsgebiet existieren nur einige wenige fundierte Grundlagen- oder Standardwerke, die den Stand der Forschung beschreiben (z.B. Russell, Norvig 2003; Ferber 2001; Wooldridge 2002). Daher stützt sich die Literaturanalyse in diesem Bereich hauptsächlich auf häufig zitierte technische Berichte (z.B. Rao, Georgeff 1991), Konferenzbeiträge (z.B. Marietto et al. 2002), Artikel in Fachzeitschriften, (z.B. Wooldridge, Jennings 1995) und Dissertationen (z.B. Müller 1996).

Aufbauend auf Konzepten und Theorien der Soziologie, Psychologie, Pädagogik, Informatik, Wirtschaftsinformatik, sowie der Betriebs- und Volkswirtschaftslehre soll dann ein Modell entwickelt und als Multiagentensimulation implementiert werden, das eine hinreichend gute Abbildung des untersuchten Ausschnitts des Realsystems darstellt. Die dazu durchgeführte interdisziplinäre Literaturstudie stützt sich im Gegensatz zu den Recherchen über Softwareagenten weitgehend auf etablierte Standardwerke (z.B. March, Simon 1958), Lehrbücher (z.B. Jost 2000; Gebert, Rosenstiel 2002), sowie wegweisende Aufsätze (z.B. Galbraith 1974) und wird nur zu einem geringen Teil durch aktuellere Aufsätze ergänzt (z.B. Zafirovski 2003) da in diesen Bereichen durch die geringeren Vorkenntnisse des Autors kein Anspruch auf Vollständigkeit und Auswertung der neuesten Erkenntnisse erhoben werden kann. Das Studium ähnlicher Simulationsprojekte (z.B. Martinez-Miranda, Aldea 2002; Carley, Tsvetovat 2004; Clancey et al. 2004; Nissen, Levitt 2004) trägt wesentlich dazu bei relevante Theorien zu identifizieren, sowie zulässige Annahmen und Vereinfachungen zu erkennen.

Ein operationalisierter Teil des so entstandenen umfassenden Modells über Wissensweitergabe in Organisationen wird anschließend als Prototyp eines Simulationssystems mit Java implementiert. Durch Nachstellen von in der Literatur dokumentierten empirischen Untersuchungen, (z.B. Berends et al. 2004; Joshi et al. 2004) mit Hilfe des Prototypen und Vergleich der so erzielten Ergebnisse mit den empirisch gewonnenen Daten soll die Validität des Modells verifiziert werden, bevor das Modell entsprechend einiger ausgewählter Szenarien parametrisiert wird, wobei für jedes Szenario mehrere Simulationsläufe durchgeführt werden, um die Zufallseinflüsse statistisch auszumitteln. Die Ergebnisse dieser Szenarien sollen mit Hilfe von sozialer Netzwerkanalyse und Unter-

suchung des Wissensstandes aller Mitarbeiter unter Berücksichtigung des jeweils vom simulierten Unternehmen erzielten Gewinns verglichen werden. Die Ergebnisse dieser Untersuchung werden anschließend auf Nachvollziehbarkeit und logische Konsistenz überprüft sowie vor dem Hintergrund der Literatur diskutiert. Abschließend sei noch erwähnt, dass die Literaturlauswahl durch die Verfügbarkeit der Quellen beeinflusst war, wodurch frei im Internet verfügbare Quellen tendenziell gegenüber anderen bevorzugt werden.

## **1.4 Überblick über die Arbeit**

Nach dieser Einführung in die Arbeit werden in Kapitel zwei die Grundlagen von Wissensmanagement erläutert, was zur Identifikation des Forschungsbedarfs dient. Dort wird auch der Untersuchungsgegenstand definiert und wichtige Begriffe in dessen Umfeld erläutert. Als Ergebnis des zweiten Kapitels wird aus bestehenden Modellen ein eigenes Modell der Wissensweitergabe entwickelt, das als Rahmen für die spätere Simulation dient.

Das Kapitel drei stellt eine multiperspektivische und interdisziplinäre Sicht auf die Wissensweitergabe dar. Dort werden sowohl theoretische als auch empirische Forschungsergebnisse dargestellt, die einen Erklärungsbeitrag zum Untersuchungsgegenstand Wissensweitergabe liefern. Diese werden diskutiert und auf Ihre Kompatibilität hin untersucht, so dass am Ende des dritten Kapitels ein Domänenmodell steht, das die zueinander passenden Theorien und Befunde auf einer grobgranularen Ebene integriert.

Im Kapitel vier wird auf diesem Domänenmodell aufbauend eine geeignete Abbildung in ein konzeptionelles Multiagentenmodell geschaffen. Dazu werden Agentenarchitekturen analysiert und die Erfahrungen aus verwandten Arbeiten eingebracht. Daraus wird schließlich ein Konzeptmodell entworfen, das eine eigene Agentenarchitektur als Adaption des InterRaP Modells beinhaltet und das Domänenmodell konkretisiert.

Dieses Konzeptmodell wird im fünften Kapitel in ein Simulationsprogramm überführt. Dazu werden einige wichtige Entscheidungen der Implementierung vorgestellt und begründet. Die Umsetzung wird anhand von UML Diagrammen und kurzen Code-Ausschnitten illustriert.

Der Abschluss des fünften Kapitels stellt eine Art Wendepunkt in der Arbeit dar, da damit die zunehmende Abstraktion und Operationalisierung des Untersuchungsgegenstands abgeschlossen ist. Dies wird in auch Abbildung 1-1 durch die Wahl der Symbole illustriert. Sie stellen die abnehmende Komplexität und zunehmende Abstraktion vom realen System

bis hin zum Simulationssystem dar. Wie in der Abbildung durch die Zuordnung der Kapitel zum Vorgehensmodell deutlich wird liegt der Schwerpunkt der Arbeit auf der Modellierung und Implementierung. Dies ist nötig, da für die Experimente auf kein fertiges Simulationssystem aufgesetzt werden konnte sondern ein eigenes von Grund auf neu entwickelt werden musste.

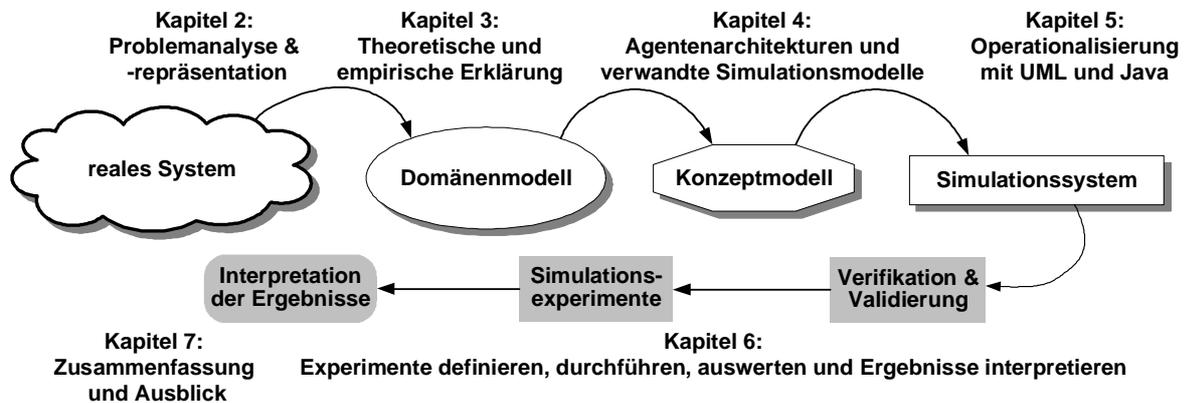


Abbildung 1-1: Kapitelübersicht (eigene Darstellung)

Das Kapitel sechs beginnt mit einem schrittweisen Aufbau der Komplexität des Modells und Sensitivitätsanalysen. Daran schließt sich die Verifikation und Validierung des Simulationsmodells und –systems an. Dies dient dazu die Korrektheit der Simulation und eine hinreichend gute Abbildung des Untersuchungsgegenstands sicherzustellen. Anschließend werden die eigentlichen Simulationsexperimente definiert, durchgeführt, die Ergebnisse dokumentiert und interpretiert.

Den Abschluss bildet das Kapitel sieben, in dem die gewonnenen Erfahrungen noch einmal zusammengefasst, Rückschlüsse auf die betriebliche Praxis gezogen werden und ein Ausblick auf mögliche Weiterentwicklungen gegeben wird.

## 2 Wissensmanagement

Das Fachgebiet Wissensmanagement (WM) wurde in den letzten fünfzehn Jahren in unterschiedlichen Disziplinen bearbeitet und hat über die Zeit hinweg zwischen Bedeutungslosigkeit, scharfer Ablehnung und Hype unterschiedliche Resonanz in Wissenschaft und Praxis erfahren (Riempp 2005, S. 6). Wissen wurde einerseits als wertvolle Ressource in Unternehmen erkannt, andererseits jedoch z.B. der Begriff Wissensmanagement als irreführend und falsch abgelehnt (Wilson 2002). Zum Teil wurde Wissensmanagement aber auch nur als zu teuer und im Vergleich dazu wenig hilfreich für die betriebliche Praxis eingeschätzt (ibid.). In der vorliegenden Arbeit wird der Begriff Wissensmanagement als Klammer für die Diskussion einiger grundlegender Begriffe und Phänomene im Zusammenhang mit der Wissensweitergabe gewählt, ohne die z.T. immer noch geführte Diskussion über die Sinnhaftigkeit des Begriffs neu aufzugreifen.

Das Kapitel beginnt mit dem Begriff Wissen, dessen Bedeutung deutlich gemacht werden muss, bevor man sich mit Wissensweitergabe beschäftigen kann. Anschließend werden verschiedene dynamische Aspekte des Wissens beleuchtet, die unter dem Begriff Wissensprozesse zusammengefasst werden. Wissensweitergabe als einer der Wissensprozesse wird dabei besonders ausführlich behandelt und deren Aspekte in einem Modell dargestellt. Anschließend wird die organisatorische Einordnung der Wissensweitergabe in den betrieblichen Rahmen des Wissensmanagements geschildert, bevor WM-Instrumente diskutiert werden, die gezielt zur Verbesserung des Umgangs mit Wissen in Organisationen eingesetzt werden können. Um deren Effektivität zu beurteilen, muss der Erfolg des Einsatzes eines Instruments gemessen werden können. Verschiedene Ansätze dazu werden im vorletzten Abschnitt des Kapitels vorgestellt, bevor es mit einer Zusammenfassung beschlossen wird.

### 2.1 Wissen

Bevor man sich mit Wissensweitergabe beschäftigt ist es nötig darzustellen, welche Interpretation des Begriffs Wissen der Betrachtung zugrunde liegt. Die Beschäftigung mit dem Begriff Wissen geht bis in die Antike auf die griechischen Philosophen zurück und auch heute noch sind die verschiedenen Definitionen ein Spiegelbild unterschiedlicher philosophischer Strömungen.

### 2.1.1 Wurzeln und Semiotik

Eine Reihe dieser Strömungen wurden von Maier gesammelt (Maier 2004, S. 58f). Ein grundlegender Unterschied besteht zwischen den Ansichten des *Positivismus*, dass Wissen durch Beobachtung einer objektiven Realität gewonnen werden kann, und dem *Konstruktivismus*, der davon ausgeht, dass Wissen nur in den Köpfen der Menschen existiert und das Beobachtbare nicht zwingend eine objektive Realität richtig widerspiegelt (ibid.). Der *Pragmatismus* entspannt den Konflikt um eine objektive Realität oder universelle Wahrheit, indem er auf die Begrenztheit der menschlichen Wahrnehmung verweist und von einer „lokalen Realität“ ausgeht, die von den Menschen beobachtbar ist. Somit kann Wissen als intersubjektiv überprüfbar innerhalb der lokalen Realität angesehen werden. Diese Arbeit folgt den Argumenten von Alavi und Leidner, die pragmatisch feststellen, dass eine Auseinandersetzung mit der Frage nach einer universellen Wahrheit die Forschung nach einer wissensbasierten Organisationstheorie nicht voranbringt und es vielmehr sinnvoll ist, sich mit den verschiedenen Sichten auf Wissen in der Ökonomie und IT zu beschäftigen (Alavi, Leidner 2001, S. 109; siehe auch Riempp 2003, S. 3). Eine ähnliche, wenn auch deutlich konstruktivistischere Sichtweise, vertritt Nooteboom, wonach Wissen aus der Interaktion mit der physischen und der sozialen Umgebung entsteht, woraus aber trotzdem keine objektive Wahrheit sondern nur eine bessere intersubjektive Überprüfbarkeit von Wissen über physische als über soziale Dinge resultiert (Nooteboom 2002, S. 24f).

Nimmt man eine andere Perspektive ein, so muss zwischen Daten, Informationen und Wissen unterschieden werden. Die Semiotik liefert die Grundlage für diese Differenzierung. Danach werden Zeichen durch Syntax zu Daten verbunden. Durch Einbetten in eine Semantik erhalten die Daten Bedeutung für den Menschen und werden zur Information. Interpretiert der Mensch diese Informationen (Pragmatik), so erhält er handlungsrelevantes Wissen (Abecker et al. 1999, S. 747f). Das Wissen unterscheidet sich also von Informationen durch die enge Bindung an einen Kontext und an einen Menschen (Graefe 2003, S. 135). Eine ähnliche, aber leicht abweichende Darstellung findet sich bei Probst et al. Demnach sind Informationen um Kontext angereicherte Daten, während Wissen durch Vernetzung von Informationen entsteht (Probst et al. 1998, S. 34ff). Trotzdem spielen Daten und Informationen eine wichtige Rolle im Zusammenhang mit Wissensweitergabe, insbesondere in der IT, da im Computer letztendlich alles mit binären Zeichen kodiert abgelegt wird. Aus Daten können durch Deduktion auch Informationen

abgeleitet werden, die dann beim Menschen durch Verknüpfung mit dem Vorwissen zu neuem handlungsorientiertem Wissen werden können. Ein Beispiel hierfür bietet das Fachgebiet Knowledge Discovery in Databases, das aus dem Data Mining hervorgegangen ist und sich mit dem Entdecken interessanter und relevanter Zusammenhänge in Daten (z.B. Verkaufsdaten) beschäftigt (vgl. Fayyad, Uthurusamy 1996; Wrobel 1998 für eine Übersicht; Ho, PAKDD 2005 für eine detaillierte Darstellung). Aber auch der umgekehrte Weg von Wissen zu Informationen und Daten ist möglich, wie Stenmark argumentiert (Stenmark 2002, S. 2f). Eine ausführliche Diskussion der Zusammenhänge mit einem Schwerpunkt auf dem Begriff Information findet sich in (Maier, Lehner 1995), eine neuere Zusammenstellung mit einer Übersicht über unterschiedliche Definition der Begriffe in der Wissensmanagementliteratur bei (Stenmark 2002). Eine detaillierte Gegenüberstellung von Wissen und Information in (Braf 2001) zeigt, dass das Thema auch im 21. Jahrhundert noch nicht abschließend behandelt ist.

Boer et al. identifizieren aufbauend auf den philosophischen Strömungen drei unterschiedliche Interpretationen für Wissen: (1) Wissen als Objekt definiert als begründete wahre Überzeugung (justified true belief), (2) Wissen in den Köpfen der Menschen definiert als das, was gewusst wird und (3) Wissen als soziales Phänomen definiert als die soziale Praxis des Wissens (Boer et al. 2002a, S. 2). Basierend auf den Überlegungen zu Daten, Informationen und Wissen folgt diese Arbeit der Auffassung, dass Wissen nur in den Köpfen der Menschen existiert. Sobald das Wissen expliziert und damit vom Menschen getrennt wird, handelt es sich um Information im Sinne der Semiotik.

Andererseits ist Frank der Ansicht, dass eine multiperspektivische Sicht auf Wissen im Wissensmanagement sinnvoller ist als ein striktes Folgen einer Grundanschauung (Frank 2006, S. 15). So kann es dem Pragmatismus folgend trotzdem sinnvoll sein im Falle von handlungsorientierten, kontextualisierten Informationen, die als Buch oder elektronisches Dokument vorliegen, von kodifiziertem Wissen zu sprechen, insbesondere wenn sie die korrekte Verwendung von Maschinen beschreiben, die durch Menschen geschaffen wurden. Ebenso ist nicht von der Hand zu weisen, dass ein geteiltes Verständnis von der Realität, wie sie im Soziologie-des-Wissens-Ansatz propagiert wird, für die erfolgreiche Wissensweitergabe eine wichtige Voraussetzung darstellt. Riempp stellt z.B. fest, dass es unter Menschen die Fähigkeit und die Neigung zur Bildung eines sozialen Konsensus über „gültige“ mentale Modelle gibt. Ihnen liegen Abstimmungsprozesse und Vereinbarungen über Ansichten und Vorstellungen bezüglich der Realität zugrunde, die innerhalb von

abgegrenzten Gruppen zu Übereinstimmungen in Teilbereichen der mentalen Modelle der beteiligten Individuen führen. Dies bildet die Grundlage gemeinschaftlichen und koordinierten Handelns (Riempp 2003, S. 2f). Solch eine multiperspektivische Sicht verfolgt z.B. auch Blackler mit seiner Einteilung in *embrained*, *embodied*, *encultured* und *encoded knowledge* (Blackler 1995; Meyer, Sugiyama 2007).

Darüber hinaus wird Wissen in der Literatur auch oft auf verschiedenen Ebenen diskutiert (z.B. Argote et al. 2000; Huysman, de Wit 2003). So wird häufig nicht nur Individuen, sondern auch Gruppen und Organisationen eigenes Wissen zugesprochen (Stenmark 2002, S. 1). Wie Stenmark aber richtig anmerkt, bestehen beide größeren Einheiten letztlich aus Individuen und nur die können Wissen weitergeben, so dass sich eine Untersuchung auf die Individuen konzentrieren muss. Unpersönlichen Entitäten und Makrophänomenen Konzepte zuzuschreiben, die streng genommen nur auf Personen anwendbar sind, ist eine oft verwendete Metapher und ein legitimes Mittels zur Komplexitätsreduktion (vgl. Gershenson 2002, Absätze 3.5ff). Eine Mikrobetrachtung erscheint jedoch erforderlich, da die Wissensweitergabe zuerst einmal auf der Mikroebene stattfindet (Wiig 2003, S. 20). Wie Conte betont, sind soziale Strukturen und Systeme zwar zunächst Artefakte, die aus den (Inter-)Aktionen von Individuen hervorgehen, haben wenn sie erst einmal existieren aber auch einen Rückkoppelungseffekt auf das Verhalten der Individuen (Conte et al. 2001, S. 202). Um wichtige Aspekte der Makroebene nicht zu vernachlässigen (z.B. Arbeitsteilung und Gruppenprozesse), müssen solche Rückkopplungseffekte entsprechende Berücksichtigung finden und werden in der vorliegenden Arbeit ebenfalls behandelt.

### **2.1.2 Merkmale**

Über diese grundlegenden Perspektiven von Wissen hinaus, werden in der Literatur unterschiedliche Arten von Wissen und deren Eigenschaften diskutiert. Sammlungen solcher Wissensarten finden sich u.a. bei (Maier 2004, S. 63ff) und bei (Reinhardt 2002, S. 138ff). Im Folgenden werden einige der Unterscheidungen dargestellt, die für die Arbeit wesentlich sind.

Polanyi stellt fest, dass jedes Wissen auch nicht explizierbare Bestandteile hat, denen sich die Person z.T. gar nicht bewusst ist (Polanyi 1966, S. 24) und prägt damit die Unterscheidung zwischen explizitem Wissen und implizitem (*tacit*, wörtlich: schweigsamen). Implizites Wissen lässt sich nur unvollständig formalisieren, ist schwer kommunizier- und teilbar. Es beinhaltet sowohl kognitive Elemente als auch subjektive Einsichten, Wahrnehmungen, Intuition, Erfahrung, Gefühle, Wertvorstellungen und Ideale

(Abecker et al. 1999, S. 749). „Explizites Wissen ist beschreibbares, formalisierbares, zeitlich stabiles Wissen, welches standardisiert, strukturiert und methodisch in sprachlicher Form in Dokumentationen, Datenbanken, Patenten, Produktbeschreibungen, Formeln, aber auch in Systemen, Prozessen oder Technologien angelegt werden kann“ (Bullinger et al. 1997, S. 8). Diese Unterscheidung findet in der Literatur großes Echo, wird jedoch unterschiedlich interpretiert. Eine Auslegung sieht implizites Wissen als einen Typ von Wissen (z.B. das Wissen wie man künstlerisch malt, oder Haare schneidet), das grundsätzlich schwer explizierbar ist (Ramesh, Tiwana 1999, S. 215), bzw. nicht in expliziter Form vorliegt (Carley, Schreiber 2002, S. 3), während es anderes Wissen gibt (z.B. Wissen über ökonomische Zusammenhänge oder Programmiersprachen), das leicht explizierbar ist oder schon explizit gemacht wurde. Wie aus den Beispielen schon ersichtlich wird, besteht in dieser Auslegung eine starke Ähnlichkeit zwischen implizitem Wissen und embodied knowledge einerseits und explizitem Wissen und embrained knowledge andererseits. Eine andere Interpretation, der auch diese Arbeit folgt, sieht die Explizierbarkeit von Wissen als kontinuierliche Eigenschaft des Wissens, wobei manches Wissen zu einem größeren Anteil und anderes Wissen zu einem geringeren Anteil explizierbar ist (vgl. Jimes, Lucardie 2003, S. 24f). Letztere Auffassung scheint eher im Sinne Polanyis zu sein, während erstere scheinbar auf eine Interpretation der frühen Arbeiten Nonakas zurückzuführen ist (Nonaka, Takeuchi 1995), die trotz dessen späterer Einschränkungen im Konzept des Ba (Nonaka, Konno 1998) nach wie vor in der Literatur verbreitet ist (Wilson 2002, S. 149f; siehe auch Meyer, Sugiyama 2007, S. 18).

Nach Sveiby, der sich eng an Polanyi anlehnt, sind Wissen vier Merkmale zuzuordnen: Wissen ist implizit, handlungsorientiert, stützt sich auf Regeln und verändert sich fortwährend (Sveiby 1997, S. 29ff). Eine umfangreichere Liste von Merkmalen findet sich bei (Maier, Lehner 1995, S. 208), die in (Maier 2004, S. 63f) nochmals erweitert wurde. Die folgende Aufzählung stellt die dort aufgeführten Merkmale dar, die für die vorliegende Arbeit relevant sind:

- Inhalt oder Verwendung von Wissen
  - Abstraktion: oberflächliches Alltagswissen ↔ tief greifendes Fachwissen
  - Generalisierung: spezifisches ↔ allgemeines Wissen
- Wissensträger
  - Verhältnis zur Person: implizites ↔ explizites Wissen

- Existenz: Wissen ↔ Vergessenes
- Organisation
  - Relevanz: im Organisationskontext relevantes ↔ irrelevantes Wissen
  - Wissensträger: individuelles ↔ kollektives Wissen
  - Reichweite: bereichsübergreifendes ↔ bereichsspezifisches Wissen
- Rechtssystem
  - Sicherheit: patentiertes, geschütztes privates ↔ ungesichertes öffentliches Wissen
  - Eigentümer: organisations-internes ↔ externes Wissen
- Informations- und Kommunikationssysteme (kodifiziertes Wissen)
  - Zugriff: zugreifbares ↔ nicht zugreifbares Wissen
  - Medium: elektronisches ↔ papierbasiertes Wissen
  - Kodifizierbarkeit: kodifizierbares ↔ schwer kodifizierbares Wissen
- Wissenslebenszyklus
  - Neuheit: neues ↔ existierendes Wissen
  - Aktualität: aktuelles ↔ obsoletes Wissen
- Geschäftsprozess
  - Verhältnis zum Geschäftsprozess: Wissen über den, in dem, oder aus dem Prozess

Die vorliegende Arbeit fokussiert dabei tief greifendes, spezifisches, organisations-internes Fachwissen, das relevant, bereichsübergreifend, existierend, aktuell und zumindest teilweise explizierbar ist. Dieser Fokus wurde festgelegt, weil diese Merkmale nach Meinung des Autors das Wissen auszeichnen, welches für Organisationen am relevantesten ist und Ihnen Wettbewerbsvorteile sichern kann, während z.B. Alltagswissen oder veraltetes Wissen im Organisationskontext kaum relevant ist. Bzgl. des Verhältnisses zum Geschäftsprozess werden die Möglichkeiten Wissen in dem Prozess und aus dem Prozess fokussiert. Wissen über die Prozesse wird als für alle Mitarbeiter gleichermaßen gegeben angenommen. Der Sicherheitsaspekt erscheint zwar wichtig, wird im weiteren Verlauf der Arbeit aber nicht weiter betrachtet, da es die Wissensweitergabe innerhalb der Organisation nicht beeinflusst, ob das Wissen patentiert ist oder nicht. Für alle anderen Merkmale werden alle möglichen Ausprägungen berücksichtigt.

Die auf den britischen Philosophen Gilbert Ryle zurückgehende Unterscheidung zwischen Faktenwissen und Handlungswissen wird in der Literatur weitgehend berücksichtigt (Schindler 2001, S. 29). Als Synonyme für Faktenwissen identifiziert Schindler Tatsachenwissen, Kenntnisse, deklaratives Wissen, know-what und know-that, während Handlungswissen auch als prozedurales Wissen oder know-how diskutiert wird. In der vorliegenden Arbeit werden beide Aspekte berücksichtigt, da nur beides zusammen angewandt zum Lösen von Arbeitsaufgaben in Organisationen führt.

Eine weitere wichtige Unterscheidung besteht zwischen individuellem Wissen und organisatorischem Wissen (Meyer, Sugiyama 2007, S. 18; Güldenbergs 2001). Individuelles Wissen ist die Voraussetzung für organisatorisches Wissen. Es kann in organisatorische Strukturen wie Methoden, Modelle, Dokumentation und Kultur eingehen und dadurch organisatorisches Wissen werden (ibid.). Dies wird dann auch als organisatorische Wissensbasis bezeichnet.

### **2.1.3 Definitionen für Wissen**

Im Folgenden sollen einige Definitionen des Begriffs Wissen dargestellt werden, um die Diskussion des Konzeptes Wissens abzurunden. Eine knappe Beschreibung liefert Romhardt, der Wissen als „interpretierte Beobachtungen“ aus Sicht der Systemtheorie definiert (Romhardt 1996, S. 8; vgl. auch Willke 2001, S. 10f). Nach Probst et al. bezeichnet Wissen „die Gesamtheit der Kenntnisse und Fähigkeiten, die Individuen zur Lösung von Problemen einsetzen. Dies umfasst sowohl theoretische Erkenntnisse als auch praktische Alltagsregeln und Handlungsanweisungen. Wissen stützt sich auf Daten und Informationen, ist im Gegensatz zu diesen jedoch immer an Personen gebunden. Es wird von Individuen konstruiert und repräsentiert deren Erwartungen über Ursache-Wirkungs-Zusammenhänge“ (Probst et al. 1998, S. 44). Diese Definition beinhaltet viele, vom Autor als wichtig erachtete Merkmale, wie die Personengebundenheit, Daten und Informationen als Basis, Handlungsorientierung, Regelmäßigkeit und Problemorientierung. Ein Kritikpunkt ist der Bezug auf die nicht näher bezeichneten „Kenntnisse und Fähigkeiten“, so dass trotz umfangreicher Definition Unklarheiten bleiben. Maier geht mit seiner Definition einen Schritt weiter und definiert wie folgt: „Wissen umfasst alle kognitiven Erwartungen – Beobachtungen die bedeutungsvoll organisiert, akkumuliert und durch Erfahrung, Kommunikation oder Inferenz in einen Kontext eingebettet wurden – die ein Individuum oder ein organisatorischer Akteur benutzt um Situationen zu interpretieren, Handlungen auszuführen, Verhalten zu zeigen und Probleme zu lösen, unabhängig davon,

ob diese Erwartungen rational sind, oder absichtsvoll eingesetzt werden“ (Maier 2004, S. 73)<sup>1</sup>. Diese Definition ist sehr umfassend und soll als Grundlage dieser Arbeit dienen. Sie betont die Einbettung in einen Kontext und den Handlungsbezug. Trotz der Notwendigkeit einer griffigen Definition erscheint dem Autor die Attributierung verschiedener Merkmale zur Charakterisierung des Verständnisses von Wissen insgesamt wichtiger für eine Abgrenzung als eine Definition in Satzform.

Einen bisher nicht behandelten Aspekt bringt die Betrachtung von Wissen als Ressource. So wird in einigen Quellen Wissen als Produktionsfaktor neben den klassischen Ressourcen der Volkswirtschaftslehre Arbeit, Kapital und Boden, bzw. neben den Ressourcen der Betriebswirtschaftslehre Arbeit, Werkstoffe und Betriebsmittel gesehen (vgl. z.B. Schindler 2001, S. 35f; Reinhardt 2002). Diese Betrachtungsweise motiviert auch die Forderung nach Wissensmanagement, also einem systematischen Umgang mit der Ressource Wissen, so wie es auch für andere Ressource entsprechende Managementtechniken gibt (Schindler 2001, S. 36; für eine Zusammenfassung der einzelnen Aspekte und Unterscheidungen siehe auch Maier 2004, S. 75).

Es bleibt abschließend noch zu erwähnen, dass Wissen in der Informatik, insbesondere im Bereich der Künstlichen Intelligenz auch sehr technisch und völlig losgelöst vom Menschen betrachtet wird. Dort wird bei Expertensystemen oder auch Agentensystemen von einer Wissensbasis gesprochen, die meist aus Regeln und Fakten besteht (o.V. 2005). Auch im Zusammenhang mit Ontologien wird oft von Wissen gesprochen (z.B. bei der Wissensrepräsentation Gómez-Pérez et al. 2004, S. 47). Dieser Wissensbegriff wird aber in der vorliegenden Arbeit nicht weiter betrachtet.

### **2.1.4 Verwandte Begriffe**

Trotz des sehr breit und unterschiedlich interpretierten Wissensbegriffs finden sich in der Literatur auch noch eine Reihe weiterer Begriffe, die teilweise synonym verwendet werden z.T. aber auch über den Wissensbegriff hinausgehen oder auf eine bestimmte Art von Wissen abzielen.

---

<sup>1</sup> Im Original: „Knowledge comprises all cognitive expectancies – observations that have been meaningfully organized, accumulated and embedded in a context through experience, communication or inference – that an individual or organizational actor uses to interpret situations and to generate activities, behavior and solutions no matter whether these expectancies are rational or used intentionally.”

In der Organisationspsychologie unterscheidet man zwischen Fähigkeiten und Fertigkeiten. *Fähigkeiten (capabilities)* sind grundlegende, aufgabenübergreifende Begabungen etwas zu tun oder zu erlernen. *Fertigkeiten (skills)* sind dagegen aufgabenspezifische Qualifikationen (Dunnette 1976). Beide Begriffe zielen aber auf die erfolgreiche Anwendung von Wissen und gehen daher über den Wissensbegriff hinaus. Des Weiteren besteht ein Zusammenhang der Begriffe zu bestimmten Wissensarten. Mit Fähigkeit wird eher Allgemeinwissen verbunden, während Fertigkeit eher mit Fachwissen in Verbindung steht. Sveiby definiert Fähigkeit als die „Kunst des Wissens, wie etwas gemacht wird, die praktisches Können erfordert – physisch wie mental – und hauptsächlich durch Training und Praxis erworben wird. Fertigkeiten beinhalten Wissen über Verfahrensweisen und Kommunikationsfähigkeiten“ (Sveiby 1997, S. 35). Der Begriff der Fertigkeiten ist deswegen von Bedeutung für die Wissensweitergabe, weil er im Zusammenhang mit Skillmanagementsystemen (SkMS) am häufigsten erhoben wird (vgl. Abschnitt 2.5). Beispiele für relevante Bereiche für Fertigkeiten in der Softwareentwicklung sind Programmiersprachen, Softwareentwurf, Datenbanken oder Netzwerktechnik (Lethbridge 1999).

Jost hingegen verwendet die Begriffe Fähigkeit und Fertigkeit synonym und unterscheidet physische und psychische Fähigkeiten. Letztere lassen sich weiter unterteilen in kognitive, soziale und psychologische Fähigkeiten (Jost 2000, S. 61). Bemerkenswerterweise ist bei keiner der drei psychischen Fähigkeiten nach Jost ein besonderer Zusammenhang zu explizitem Wissen zu erkennen. Vielmehr ist die kognitive Fähigkeit eng mit der *Intelligenz* gekoppelt. Als soziale Fähigkeiten werden Teamfähigkeit und Hilfsbereitschaft genannt und die psychologische Fähigkeit wird auf geistige Beanspruchbarkeit in Stresssituationen zurückgeführt. Der Begriff Intelligenz wird oft im medizinischen oder pädagogischen Kontext gebraucht und stellt stark auf die vererbten kognitiven Möglichkeiten ab (Blöink 2006). Der Duden definiert Intelligenz als Fähigkeit (des Menschen) zum abstrakten Denken (6. Auflage). In jüngerer Zeit wird der Begriff über die klassischen Intelligenzquotient und Intelligenztest hinaus jedoch auch weiter gefasst und unter dem Stichwort „multiple Intelligenzen“ auf angrenzende Bereiche ausgedehnt (Neisser et al. 1996).

*Erfahrung* wird durch Reflektieren vergangener Fehler und Erfolge erworben (Sveiby 1997, S. 35). Mit Erfahrung wird meist schwer explizierbares Wissen verbunden. Sie kann demnach als Teilaspekt des Wissens bezeichnet werden.

*Kompetenz* kann als Verbindung von explizitem Wissen, Fertigkeiten, Erfahrung, Wertvorstellungen und dem sozialen Netzwerk aufgefasst werden (Sveiby 1997, S. 35f), wobei sich das soziale Netzwerk aus Meta-Wissen über die Kompetenzen anderer Mitarbeiter und sozialen Fähigkeiten zusammensetzt. Für Riempp zeigt sich Kompetenz durch Kennen, Können und angemessenes Entscheiden (Riempp 2003, S. 3).

*Expertise* sieht Sveiby in engem Zusammenhang mit dem Expertenbegriff und spricht erst dann von Expertise, wenn jemand so große Fertigkeiten besitzt, dass er nicht nur alle relevanten Verfahrensweisen kennt und sie für sich selbst modifiziert, sondern auch eigene, bessere entwickelt (Sveiby 1997, S. 38). Nach Hinds und Pfeffer hingegen zeichnet sich Expertise durch eine abstrakte, konzeptionelle Wissensrepräsentation aus (Hinds, Pfeffer 2003, S. 5). Neuweg destilliert aus dem Expertise Modell von Dreyfus und Dreyfus (Dreyfus, Dreyfus 1986) fünf Kriterien, durch die sich Expertise auszeichnet (Neuweg 1999, S. 297ff):

1. Implizite Integration von neuem Wissen in die vorhandene Wissensbasis und die damit verbundene Fähigkeit zum Umgang mit unstrukturierten Problemen
2. Qualitativer Wandel des Wissens vom regelgeleiteten Know-that zum erfahrungsbasierten Know-how
3. Eine kontextangepasste Urteilskraft, die es ermöglicht eine Situation rasch zu erkennen und aufgrund von Erfahrungen mit ähnlichen Situationen ein Muster zu identifizieren und dadurch Erfolg versprechende Handlungen vorzuschlagen.
4. Expertise kann nur schrittweise erreicht werden über den anfänglich analytischen und den anschließenden planerischen bis hin zum intuitiven Lösungsmodus.
5. Abkehr von regelbasiertem Wissen zugunsten von holistischem, situativem Verstehen.

Der Begriff *intellektuelles Kapital* stellt die Perspektive der Organisation auf ihre Wissensressourcen dar und folgt damit einer ressourcen-basierten Interpretation des Wissens. Bekannt wurde der Begriff durch die Firma Skandia, die ihren Investoren neben der Bilanz auch eine Übersicht über das intellektuelle Kapital in Anlehnung an eine Balanced Scorecard verfügbar machte (Wiig 1997).

### **2.1.5 Zusammenfassung**

Insgesamt hat die vorliegende Arbeit den folgenden Blick auf Wissen. Grundsätzlich existiert Wissen nur in Personen. Da es jedoch für Wissensweitergabe (siehe nächstes

Kapitel) zwischenzeitlich als Informationen vorliegt und ständig Konversionsprozesse zwischen Wissen und Informationen stattfinden ist auch die Perspektive, die Wissen als Objekt sieht, nützlich für die Arbeit. Bis zu einem gewissen Grad ist es möglich Wissen in Systeme einzubetten, so dass diese scheinbar intelligente Entscheidungen treffen und Handlungen vollziehen können (z.B. ein Schachcomputer). Schließlich ist Wissensweitergabe hauptsächlich ein sozialer Prozess, wodurch auch die Perspektive von Wissen in sozialen Systemen eine Bedeutung erhält. Eine Übersicht der in der später vorgestellten Simulation berücksichtigten Wissensarten findet sich in Tabelle 2-1.

*Tabelle 2-1: Zusammenfassung der Sicht auf Wissen*

<b>Wissensarten</b>	<b>Beschreibung</b>
tacit vs. explicit knowing (Polanyi 1966)	Nur explizites und explizierbares Wissen wird berücksichtigt.
declarative vs. procedural knowledge and meta-knowledge (Anderson 1976, Squire 1987, Fayol 1994)	Meta-Wissen wird explizit berücksichtigt (Wissensquellen kennen), prozedurales Wissen wird als gleichwertig für alle Mitarbeiter angenommen (Anwendung von Faktenwissen) und deklaratives Wissen ist der Untersuchungsgegenstand.
fact knowledge (about things) vs. episodic knowledge (about events) and procedural knowledge (about relationships) (Heideloff, Baitsch 1998)	Faktenwissen ist der Untersuchungsgegenstand. Episodenwissen wird für Adaption benutzt (der Kollege X hat mir geholfen, also helfe ich ihm auch) und prozedurales Wissen ist modellinhärent (Agenten „kennen“ die Konsequenzen ihres Handelns)
Fachwissen vs. Allgemeinwissen vs. Wissen über Zustände von (immateriellen) Objekten	Für die Wissensweitergabe ist in erster Linie das Fachwissen interessant. Aber auch das Wissen über Zustände von Objekten wird untersucht.

## 2.2 Wissensweitergabe und weitere Wissensprozesse

Mehrere Autoren unterscheiden verschiedene elementare Tätigkeiten im Zusammenhang mit Wissen und bezeichnen diese als Wissensprozesse (Staab et al. 2003), Kernprozesse des Wissensmanagement (Probst et al. 1998), Wissensverarbeitungsaktivitäten (Holsapple, Joshi 2002), WM-Aktivitäten (Remus 2002) oder WM-Prozesse (Riempp 2003). Oft werden sie zusätzlich in einem Modell angeordnet, das die Beziehungen zwischen diesen einzelnen Prozessen darstellt und die Autoren sprechen dann von einem Lebenszyklus-

oder Kreislaufmodell des Wissens, obwohl kein Kreislauf im eigentlichen Sinne existiert, da es viele Rücksprünge und Querverbindungen gibt (Remus 2002, S. 125f).

### 2.2.1 Überblick über Wissensprozesse

Trotz dieser Begriffsvielfalt konstatieren Hoffmann et al., dass die Literatur insgesamt recht einheitlich der Definition von Bach folgt (Hoffmann et al. 2003, S. 161), wonach Wissensprozesse eigenständige Unterstützungsprozesse zur Sammlung, Aufbereitung, Verteilung und Pflege von Wissen, aber auch der gesamten Wissensbasis sind, die von den Geschäftsprozessen unterschieden werden können (Bach 1999). In der Praxis sind Wissens- und Geschäftsprozesse, z.B. bei der Wissensverteilung, miteinander verzahnt, weil Geschäftsprozesse Aktivitäten der Generierung, Speicherung, Verteilung oder Anwendung von Wissen beinhalten (Hoffmann et al. 2003, S. 161). Weiterführende Betrachtungen zum Verhältnis von Wissensprozessen zu Geschäftsprozessen und speziell deren Modellierung findet sich in (Strohmaier 2005).

Im Folgenden werden einige der Modelle aus dem deutschen und angloamerikanischen Raum vorgestellt, die am häufigsten in der Literatur zitiert werden. Es werden die dort genannten Wissensprozesse diskutiert und schließlich eine genauere Analyse des Wissensprozesses Wissensweitergabe durchgeführt.

**Probst, Raub, Romhardt:** Eines der bekanntesten Modelle im deutschen Sprachraum sind die „Bausteine des Wissensmanagements“ (Probst et al. 1998, S. 51ff). Es identifiziert sechs als „Kernprozesse des Wissensmanagements“ bezeichnete Wissensprozesse und erweitert diese durch Wissensbewertung und die Einführung von Wissenszielen zu den genannten Bausteinen (vgl. Abbildung 2-1). Die Analyse der Wissensprozesse erfolgt dabei aus Sicht des Gesamtunternehmens. *Wissensidentifikation* stellt darauf ab, Meta-Wissen darüber zu generieren, welches Wissen die Mitarbeiter besitzen, bzw. welches kodifizierte Wissen in Dokumenten vorliegt. *Wissenserwerb* bezeichnet das „Einkaufen“ von Wissen aus unternehmensexternen Quellen. Dabei spielen Beziehungen zu Kunden, Lieferanten und Partnern ebenso eine Rolle wie die Rekrutierung von neuen Mitarbeitern oder das Engagieren von Beratern und Trainern. *Wissensentwicklung* ist die Bemühung unternehmensintern neues Wissen zu generieren. Dabei steht zwar die Forschungs- und Entwicklungsabteilung im Vordergrund, die Autoren betonen aber, dass alle Abteilungen zur Wissensentwicklung beitragen sollen. *Wissensverteilung* ist der Prozess der Verbreitung bereits vorhandenen Wissens innerhalb des Unternehmens. Dabei muss nicht jeder alles wissen, sondern es muss gezielt die nötige Information zu den richtigen

Mitarbeitern geleitet werden. *Wissensnutzung* ist die Anwendung des Wissens im Unternehmenskontext. Sie wird von den Autoren als ultimatives Ziel des Wissensmanagements beschrieben. *Wissensbewahrung* zielt schließlich darauf ab, einmal gewonnenes Wissen auch in Zukunft nutzen zu können. Hier stellen die Autoren hauptsächlich auf elektronische Speicherung in Form von Informationen ab. Gezielte Personalmaßnahmen werden nicht erwähnt.

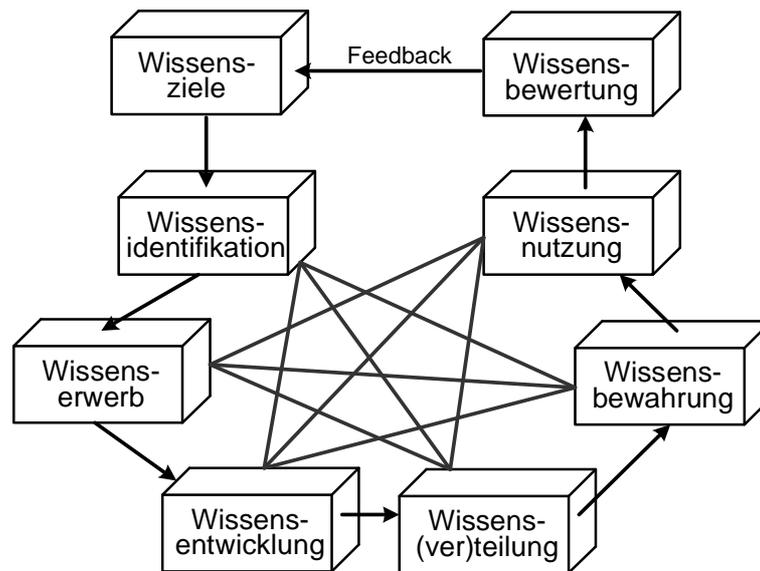


Abbildung 2-1: Bausteine des Wissensmanagement (Probst et al. 1998, S. 56)

Neben diesen Kernprozessen sehen Probst et al. noch „pragmatische Ergänzungen“ des Konzeptes vor und führen *Wissensziele* als richtungsweisendes Strategieinstrument ein, deren Erreichung durch *Wissensbewertung* ermittelt wird. Abbildung 2-1 zeigt die geschilderten Bausteine und deren Verbindung untereinander. Probst et al. postulieren einen Kreislauf ausgehend von den Wissenszielen, die zur Identifikation von benötigtem Wissen, dessen Erwerb und Weiterentwicklung bis hin zur Verteilung, Bewahrung und Nutzung des Wissens führen. Aus der Bewertung der Aktivitäten ergeben sich neue Wissensziele und der Kreislauf beginnt erneut.

**Riempp** unterstellt keinen Kreislauf oder Lebenszyklus, sondern benennt nur die elementaren WM-Prozesse (Riempp 2003, S. 4). Dies sind:

- *Lokalisieren und Erfassen* von bestehendem Wissen, welches in Form von impliziten und explizitem Wissen der Mitarbeiter (bei Riempp: Kompetenz) vorliegt oder von dem es Abbildungsversuche in der Gestalt von Informationsobjekten mit Inhalt und Kontext gibt.

- *Austausch* von Wissen sowie Verteilung von Informationsobjekten oder Hinweisen auf Kompetenzträger.
- *Entwickeln* von aktuell oder künftig benötigtem Wissen, beispielsweise durch Entdecken, Lernen und Zusammenarbeit.
- *Nutzung* von Wissen als zentraler Zweck des Wissensmanagements, z. B. in Geschäftsprozessen der Produktentwicklung oder des Customer Relationship Managements (CRM).

Obwohl sie einen etwas anderen Fokus haben sollen hier auch noch die Arbeiten von Nonaka und Takeuchi sowie von Schultze erwähnt werden, da sie interessante neue Perspektiven auf Wissensprozesse werfen.

**Nonaka und Takeuchi** untersuchen die Prozesse, in denen Wissen von explizitem zu implizitem und umgekehrt umgewandelt und dabei auch weitergegeben wird (Nonaka, Takeuchi 1995). Sie definieren dazu die vier Wissenskonversionsprozesse Sozialisation, Externalisierung, Kombination und Internalisierung (siehe Abbildung 2-2).

		<b>Endpunkt</b>	
		Implizites Wissen	Explizites Wissen
A u s g a n g s p u n k t	Implizites Wissen	<b>Sozialisation</b> Erfahrungen teilen: Training on the Job, Brainstorming Informelles Zusammensitzen Ergebnis: Sympathetisches Wissen	<b>Externalisierung</b> Intuitive, subjektive Erfahrung kommunizieren: Metaphern, Analogien, physische Modelle Ergebnis: Konzeptuelles Wissen
	Explizites Wissen	<b>Internalisierung</b> Neuerfindung des Rades vermeiden: Handbücher, Diagramme, Learning by Doing, Geschichten, Referate Ergebnis: Operatives Wissen	<b>Kombination</b> Wissen austauschen, zusammenführen: Text/Bild/Ton- Dokumente, Diskussion, Formelle Ausbildung Ergebnis: Systemisches Wissen

Abbildung 2-2: Wissenskonversionsprozesse nach (Nonaka, Takeuchi 1995)

Dabei muss berücksichtigt werden, dass Nonaka und Takeuchi eine etwas andere Auffassung von implizitem Wissen haben, was im Abschnitt 2.1 schon angeklungen ist. Implizites Wissen ist demnach kein verborgenes Wissen, dessen sich der Wissensträger nicht bewusst ist, sondern vielmehr Wissen, dass in der Organisation noch nicht explizit gemacht wurde, also nur im Kopf einer, oder einer weniger Personen vorhanden ist. *Externalisierung* führt dazu, dass dieses implizite Wissen für die Organisation zugänglich

wird. Durch die *Kombination* von Wissen aus mehreren Bereichen oder von mehreren Quellen kann neues Wissen entwickelt werden. *Internalisierung* dient dem Verinnerlichen von explizit vorliegendem Wissen, so dass operativ gearbeitet werden kann. *Sozialisation* dient schließlich dem Austausch von implizitem Wissen zwischen Personen.

*Schultze* beobachtet Wissensprozesse in ethnographischen Studien. Sie identifiziert vier so genannte *informing practices*, sinngemäß etwa Arbeitspraktiken zur Informationsverarbeitung. Dies sind Ausdrücken (*expressing*), Beobachten (*monitoring*), Übertragen (*translating*) und Vernetzen (*networking*) (Schultze 2003, S. 50f). *Ausdrücken* ist dabei die Praktik des Überführens von implizitem Wissen in explizit vorliegendes Wissen. Dies kann entweder mündlich durch Kommunikation erfolgen oder aber schriftlich durch Dokumentation des Wissens. Diese Praktik ist demnach der Externalisierung von Nonaka recht ähnlich. *Beobachten* hat im Gegensatz dazu keine Entsprechung bei Nonaka oder anderen Autoren, die Wissensprozesse beschreiben. Es beschreibt die Tätigkeiten des kontinuierlichen Verfolgens der Entwicklung eines oder mehrerer Themengebiete in Zeitschriften, dem eigenen Unternehmen und anderen Quellen. Ein wichtiger Aspekt ist dabei insbesondere das Schaffen von Awareness bzgl. neuer Details eines Themengebiets. Dies ist die Voraussetzung für tiefer gehendes Lernen und wird in der übrigen Literatur zu wenig beachtet. Das *Übertragen* von Wissen bezeichnet den Prozess der mentalen, aber auch praktischen Anwendung von Wissen aus einem Kontext auf eine aktuelle Problemstellung. Damit besteht hier wieder eine gewisse Ähnlichkeit zum Kombinationsprozess von Nonaka, bzw. der Wissensnutzung von Probst et al. Es wird jedoch ein anderer Aspekt in den Vordergrund gestellt. *Vernetzen* schließlich kann aus zwei Perspektiven gesehen werden. Zum einen das Vernetzen von verschiedenen Wissensquellen oder Wissensbausteinen zu einem großen Ganzen, also die Integration verschiedener Informationen. Der andere Aspekt des Vernetzens ist das Vernetzen von Personen mit Wissen, die als Team neues Wissen generieren können und durch Kombination ihres Wissens Probleme lösen können. Insbesondere dieser soziale Aspekt ist dabei interessant und wird in vielen Quellen zu wenig beachtet.

*Efimova*: Zu ähnlichen Überlegungen kommt Efimova bei der Analyse von Weblogs von Wissensarbeitern. Sie identifiziert ebenfalls vier wesentliche Bestandteile von Wissensarbeit, die sich durch die Überschneidungen der Bereiche Individuen, Communities und Ideen (was explizitem Wissen entspricht) ergeben (siehe Abbildung 2-3).

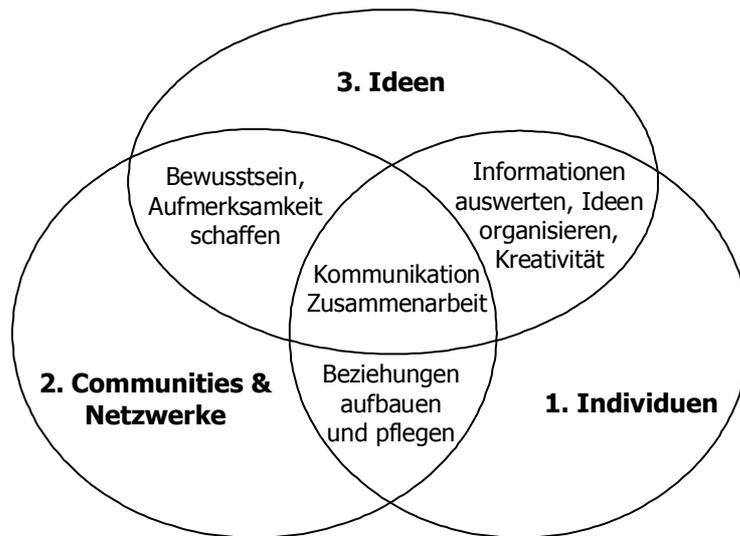


Abbildung 2-3: Bestandteile von Wissensarbeit nach (Efimova 2004)

*Aufmerksamkeit schaffen* entspricht dem Beobachten von Schultze, *Beziehungen aufbauen und pflegen* entspricht dem sozialen Aspekt des Vernetzens bei Schultze. *Informationen auswerten* und *Ideen organisieren* ist eine konkretere Darstellung des Übertragens von Schultze. *Kommunikation und Zusammenarbeit* schließlich sind viel mehr als ein Wissensprozess und können sowohl Wissensweitergabe als gemeinschaftliche Wissensentwicklung beinhalten.

**Literaturreviews:** Nissen et al. untersuchen in ihrer Arbeit eine Reihe verschiedener Wissens-Lebenszyklusmodelle aus dem angloamerikanischen Raum und erstellen daraus ein zusammenfassendes Modell, das insgesamt sechs Wissensprozesse umfasst (vgl. Nissen et al. 2000, S. 30). Schindler sammelt ebenfalls eine Reihe von WM-Prozessmodellen, versucht allerdings nicht, diese zu einem gemeinsamen Modell zusammenzufassen (Schindler 2001, S. 53f).

- (Nissen 1999): Erfassen, Organisieren, Formalisieren, Verteilen, Anwenden
- (Despres, Chauvel 1999): Erzeugen, Zusammenführen, Speichern, Weitergeben, Wiederverwenden, Weiterentwickeln
- Gartner Group<sup>2</sup>: Erzeugen, Organisieren, Erfassen, Zugreifen, Anwenden
- (Davenport, Prusak 1998): Generieren, Kodifizieren, Übertragen

---

<sup>2</sup> Die Quelle konnte nicht ausfindig gemacht werden. Die Quellenangabe bei Nissen et al. lautet: Gartner Group (1998). "Knowledge Management Scenario," conference presentation, SYM8KnowMan1098Kharris. Im Text wird dann aber Gartner Group 1999 zitiert (S. 30).

- (Nissen et al. 2000): Erzeugen, Organisieren, Formalisieren, Verteilen, Anwenden, Weiterentwickeln
- (Eppler et al. 1999, S. 4ff): Identifikation, Evaluation, Allokation, Anwendung
- (O'Dell, Grayson 1998, S. 6f): Identifizieren, Organisieren, Weitergeben, Adaptieren, Anwenden, Erzeugen, Sammeln
- (Pawlowski 1999, S. 115f): Identifikation, Generierung und Erneuerung, Diffusion, Integration, Transfer
- (Schüppel 1997, S. 191f): Produktion, Reproduktion, Distribution, Verwertung, Logistik
- (Zack 1999, S. 45): Erzeugen, Auffinden, Erfassen, Weitergeben
- (Maier 2004, S. 75): Erzeugen, Identifizieren, Formalisieren, Organisieren, Weitergeben, Verteilen, Verfeinern, Anwenden, Feedback Geben.
- (Maier 2004, S. 134): Erzeugen (record, develop, research, accumulate), Identifizieren, individuelles Lernen, Weitergeben, Institutionalisieren, Anwenden, Feedback Geben, neu Anordnen (repackage, reproduce), Verkaufen (license, sell, consult), (intraorganisationales) Verbreiten (communicate, disseminate).

Interessant ist bei Maier vor allem das Nebeneinander von Weitergeben und Verteilen, das darauf hinweisen könnte, dass der Verbreitung vom Wissen im Unternehmen eine große Bedeutung zukommt. Weiterhin fällt auf, dass Maier im Rahmen des Modells für Aufgaben und Flüsse im Wissensmanagement (Maier 2004, S. 134) seine vorherige Liste erweitert und verändert.

Noch ausführlicher befasst sich Remus mit Wissensprozessen in seiner Dissertationsschrift, die sowohl deutsche, als auch englischsprachige Quellen berücksichtigt. Er kommt zu insgesamt zehn verschiedenen WM-Aktivitäten und beschreibt jede einzelne ausführlich (Remus 2002, S. 127).

1. Wissen identifizieren (transparent machen)
2. Wissen suchen (navigieren, zugreifen)
3. Wissen entwickeln (aufnehmen, erfassen)
4. Wissen erwerben
5. Wissen bewerten (evaluieren)

6. Wissen aufbereiten (organisieren, klassifizieren, integrieren, einbetten, erfassen, kombinieren, kartieren, formalisieren, publizieren)
7. Wissen weiterentwickeln (verbessern)
8. Wissen bewahren (speichern, kodifizieren, sichern)
9. Wissen verteilen (teilen, übertragen, zirkulieren, transferieren, publizieren)
10. Wissen anwenden (nutzen, wieder verwenden, einbetten)

Trotz der guten Zusammenstellung der Prozesse aus der Literatur bei Remus scheint es kleinere Inkonsistenzen zu geben. Es ist z.B. nicht ersichtlich, warum Erfassen sowohl zur Wissensaufbereitung zählt, wie auch zur Entwicklung. Ebenso wird Einbetten bei der Anwendung von Wissen und beim Aufbereiten angeführt. Andererseits ist zu beachten, dass die einzelnen Prozesse oft nicht isoliert zu betrachten sind und nur eine geringe Trennschärfe aufweisen, da sie untereinander in Wechselwirkung stehen (Schindler 2001, S. 55).

Einen anderen Ansatz zur Sammlung von Wissensprozessen verfolgen Holsapple und Joshi. Sie sprechen von Wissensverarbeitungsaktivitäten statt von Wissensprozessen und lassen eine Unterteilung in Haupt- und Unterprozesse zu (Holsapple, Joshi 2002, S. 482ff).

1. Wissen erwerben (identifizieren, erfassen, organisieren, weitergeben)
2. Wissen auswählen (identifizieren, erfassen, organisieren, weitergeben)
3. Wissen internalisieren (bewerten, Ziele auswählen, strukturieren, zum Ziel liefern)
4. Wissen erzeugen (monitoring, bewerten, produzieren, weitergeben)
5. Wissen externalisieren (Ziel festlegen, produzieren, weitergeben)

Diese Ergebnisse einer Delphi Studie müssen vor dem Hintergrund des organisatorischen Lernens betrachtet werden. Bei der Internalisierung z.B. ist nicht gemeint, dass ein Individuum das Wissen internalisiert, sondern dass das Wissen in die organisatorische Wissensbasis eingeht. Es existieren auch eine Reihe weiterer Publikationen, die sich mit Wissensprozessen auseinandersetzen (z.B. Hoffmann et al. 2003). Diese bringen aber für die Identifikation verschiedener Wissensprozesse keine neuen Beiträge, was als Indiz für die Stabilität des Themas in der Literatur gewertet werden kann.

### **2.2.2 Beschreibung der Wissensprozesse**

Die Beschreibung der einzelnen Wissensprozesse in (Remus 2002, S. 128ff) wird hier kurz wiedergegeben, um einen Rahmen für die Beschäftigung mit Wissensweitergabe zu haben,

die im nächsten Abschnitt erfolgt. Die Entscheidung für Remus als Grundlage erfolgte aufgrund der umfangreichen und umfassenden Diskussion des Themas dort, die auf vielen anderen der vorgestellten Quellen aufsetzt und vielfach darüber hinaus geht.

**Identifikation** (S. 128): Ziel der Identifikation des Wissens ist es nicht nur transparent zu machen, wer welches Wissen besitzt, sondern auch wer welches Wissen benötigt, welches Wissen fehlt und wo welches Wissen angewendet werden kann. Es kann auch zwischen personeller und struktureller Transparenz unterschieden werden, wobei ersteres z.B. durch ein Skillmanagementsystem oder eine Wissensträgerkarte, letzteres durch eine Wissensstrukturkarte oder eine Ontologie/Taxonomie unterstützt werden kann.

**Entwicklung** (S. 129): Die Entwicklung neuen Wissens basiert auf der individuellen Ebene auf Kreativität einerseits und Problemlösekompetenz andererseits. Der Wissensentwicklungsprozess kann nicht systematisch geplant werden, Motivation, Inspiration und evtl. sogar Zufall spielen eine Rolle. Obwohl Wissen nach der oben genannten Definition (siehe Abschnitt 2.1) nur von Individuen entwickelt werden kann, sind Gruppenprozesse oft wesentlich für die Entstehung neuen Wissens. Diese von Remus zusammengetragenen Aspekte müssen noch um die Betrachtung ergänzt werden, wie speziell im Sinne von problem- und situationsbezogen das neu entwickelte Wissen ist. Je allgemeiner – und damit von einem singulären Problem abstrahiert – es ist, desto höherwertiger kann es eingeschätzt werden. Das Wissen über die Architektur des Informationssystems eines Kunden wäre z.B. Wissen über eine recht spezielle Problemlösung, wogegen die Entwicklung eines Referenzmodells für eine ganze Branche zusätzliche Abstraktionsleistungen und das Einbeziehen einer viel größeren Zahl an Basisinformationen erfordert. Eine weitere betrachtenswerte Unterscheidung ist der Neuheitsgrad des neu entwickelten Wissens. Ist das Wissen für das Individuum neu, aber innerhalb der Organisation schon anderen Mitarbeitern bekannt, für die Organisation neu, aber innerhalb der Branche schon bekannt, oder weltweit neu? Die Antwort auf diese Frage entscheidet zusammen mit dem Nutzen durch dessen Anwendung über den Wert des neuen Wissens.

**Erwerb** (S. 130): Als Ersatz oder Ergänzung der eigenständigen Entwicklung neuen Wissens kann Wissen auch von externen Quellen erworben werden. Dabei ist zu beachten, dass externes Wissen auf den Anwendungsfall in der Organisation adaptiert werden muss. Wissen kann in Form von Personen, Kommunikation und (in Dokumenten) kodifiziertem Wissen oder als (in Produkten) eingebettetes Wissen erworben werden.

**Bewertung** (S. 131): Die Bewertung von Wissen kann zum Einen im Sinne von (Probst et al. 1998) als Erfolgsmessung von WM-Aktivitäten verstanden werden, andererseits aber auch als Bewertung von einzelnen Wissenselementen, also z.B. kodifiziertem Wissen in Dokumenten in einem Review-Prozess, oder auch Wissen der Mitarbeiter welches in Mitarbeitergesprächen oder sogar in Assessment-Centern oder anderen Prüfungssituationen evaluiert werden kann (siehe dazu auch Abschnitt 2.5).

**Aufbereitung** (S. 131f): Zur Aufbereitung des Wissens zählt Remus eine Reihe wertsteigernder Aktivitäten. Diese beziehen sich allerdings ausschließlich auf kodifiziertes Wissen (strukturieren, kontextualisieren, mit Metadaten anreichern) oder weisen starke Überschneidungen mit den Prozessen Wissen weiterentwickeln (z.B. kombinieren) und Wissen bewahren (z.B. kodifizieren, formalisieren) auf.

**Bewahrung** (S. 132f): Im Gegensatz zu Probst et al. erwähnt Remus ausdrücklich personenorientierte Maßnahmen wie die Bindung von Wissensträgern mit Anreizsystemen, oder die Unterstützung von Wissensaustausch in informellen Netzwerken als Mittel zur Bewahrung von Wissen. Darüber hinaus weist auch er auf die unterstützenden Informationssysteme hin, die zur Speicherung kodifizierten Wissens verwendet werden können. Ein weiterer Aspekt der Bewahrung ist die Sicherung von Wissen, was sowohl die rechtliche Seite betrifft (z.B. mittels Patenten), als auch den Zugriffsschutz (z.B. mittels Rechtevergabe). Der Schutz vor Verlust wichtiger Dokumente (z.B. Backup, oder Tresore) muss ebenfalls hierzu gerechnet werden.

**Verteilung** (S. 133f): Verteilung kann im Sinne einer aktiven Steuerung durch das Management verstanden werden, aber auch dem Gedanken des Teilens folgend als freiwillig durch Mitarbeiter initiiertes Prozess. Die Wahl des richtigen Mediums für die Verteilung kodifizierten Wissens ist ebenso wichtig, wie die Etablierung von Instrumenten zur Förderung von Wissensaustausch, wie z.B. das Einrichten von Communities of Practice. Auch das Unterstützen des Verständnisses für die Notwendigkeit von Wissensverteilung und damit die Entwicklung einer offenen Unternehmenskultur, in der Wissensverteilung positiv sanktioniert wird ist wichtig.

**Suche** (S. 134): Die Abgrenzung zwischen Suche und Identifikation von Wissen wird bei Remus nicht ganz klar. Vermutlich soll die Identifikation eher dazu dienen, Meta-Wissen über Experten, Dokumente und Wissensstrukturen zu kodifizieren, wobei die Suche auf dieses explizite Meta-Wissen zurückgreifen kann, um Wissensquellen für spezielle Probleme zu finden. Suchtechnologien einerseits, sowie die Etablierung von neuen Rollen

wie dem Knowledge Broker, der gezielt Recherchen durchführen und Wissensquellen vermitteln kann, werden als Unterstützungsmöglichkeiten genannt.

**Anwendung** (S. 134f): Auch Remus betont die Anwendung oder Nutzung des Wissens innerhalb eines Geschäftsprozesses als letztendliches Ziel des Wissensmanagements. Dabei ist der Anwendungskontext entscheidend. Erfolgreiches Anwenden des Wissens wird erst durch Einbetten des Wissens in diesen Kontext möglich, der sich möglicherweise vom Entstehungskontext oder auch dem Kontext, in dem es im Kopf des Mitarbeiters eingebettet ist, unterscheidet (vgl. dazu auch Singley, Anderson 1989).

**Weiterentwicklung** (S. 135): Die Weiterentwicklung von Wissen schließt sich idealer Weise direkt an die Anwendung an, indem über die erzielten Erfolge reflektiert und das dadurch neu gewonnene Wissen zusammen mit dem vorherigen und evtl. weiteren Wissen aus anderen Wissensquellen kombiniert und neu expliziert wird. Des Weiteren nennt Remus hier auch noch einmal Aktivitäten, die schon im Zusammenhang mit der Aufbereitung zur Sprache kamen, z.B. Strukturierung, und Verbessern des Zugangs. Auch das Löschen nicht mehr relevanter Informationen wird unter Wissensweiterentwicklung subsumiert.

Durch die Diskussion wird klar, dass trotz der großen Übereinstimmung bei der Nennung einzelner Wissensprozesse, die von Remus vorgenommene Einteilung noch nicht abschließend sein kann, da es zu viele Überschneidungen und mehrereindeutige Zuordnungen gibt. Auf den Versuch einer Weiterentwicklung des Gesamtmodells wird hier aber zugunsten einer tiefer gehenden Beschäftigung mit der Wissensweitergabe als dem im Fokus stehenden Wissensprozess verzichtet.

### **2.2.3 Wissensweitergabe im Detail**

Die Wissensweitergabe als zentraler Untersuchungsgegenstand soll im Folgenden noch näher analysiert werden. Der Begriff selbst wird in der Literatur uneinheitlich verwendet und es existieren zahlreiche synonym verwendete Termini. Im angloamerikanischen Raum finden sich die Begriffe knowledge sharing (z.B. Berends et al. 2004), knowledge diffusion (Cowan, Jonard 2004), knowledge dissemination (Song et al. 2003), knowledge distribution (Carley 2002a), knowledge exchange (Thomas-Hunt et al. 2003) und knowledge transfer (Tsai 2001). Knowledge flow (Dervisoglua, Berberb 2004) ist ebenfalls ein häufig anzutreffender Begriff, der als Synonym zu Wissensweitergabe betrachtet wird. Im Deutschen Sprachraum finden sich diese Begriffe in mehr oder weniger wörtlicher Entsprechung wieder, nämlich Wissens(ver)teilung, Wissensstreuung, Wissenszirkulation,

Wissenstransfer und Wissensübertragung (Remus 2002, S. 133; Weggeman 1999, S. 237). Darüber hinaus prägen Eppler und Reinhardt den bewusst provokant gewählten Begriff der Wissenskommunikation (Eppler, Reinhardt 2004).

Hansen definiert Wissensweitergabe als einen Prozess der aus der Suche nach einer passenden Wissensquelle und dem Transfer des Wissens selbst besteht. Die Suche beinhaltet dabei die Identifikation des benötigten Wissens und die Identifikation einer geeigneten Wissensquelle. Der Transfer beinhaltet die „Bewegung“ des Wissens und das Verinnerlichen beim Empfänger (Hansen 1999, S. 82). Boer et al. betonen die Bedeutung des sozialen Umfelds (sociality) und schließen daraus, dass knowledge sharing als ein sozialer Prozess aufgefasst werden kann, durch den Individuen versuchen ein gemeinsames Verständnis der Realität zu erzeugen indem sie verschiedene Kombinationen von Zeichen (z.B. Sprache, Gesten, Zeichnungen) und Werkzeugen (z.B. physische Objekte, Kommunikationstechnologie, mentale Modelle) einsetzen (Boer et al. 2002b, S. 3)<sup>3</sup>. Diese Definition lässt erkennen, dass ein Verständnis von Wissen als sozialem Prozess zugrunde liegt, was bei Boer et al. auch explizit erwähnt wird. Der Autor ist aber der Auffassung, dass Wissensweitergabe sehr wohl als Versuch des Etablierens eines gemeinsamen Verständnisses aufgefasst werden kann, ohne im Umkehrschluss folgern zu müssen, dass Wissen ausschließlich ein sozial konstruiertes Phänomen darstellt. Eine isolierte Person kann Wissen besitzen und neues erzeugen, indem sie gezielte Experimente in ihrer Umgebung durchführt und aus den Beobachtungen Regeln ableitet. Auch ohne Weitergabe des Wissens, also die soziale Komponente, ist das Wissen schon vorhanden.

Wissensdiffusion hat eher eine passive Konnotation, die betont, dass Wissen sich im Laufe der Zeit auch ohne gezielten Eingriff ausbreitet. Der Begriff wird jedoch z.T. auch verwendet, um die aktive Weitergabe von Wissen zu bezeichnen (Reinhardt 2004). Nach Cowan und Jonard verbreitet sich Wissen durch Austauschbeziehungen zwischen Paaren von [organisatorischen] Agenten (Cowan, Jonard 2004, S. 1558)<sup>4</sup>.

---

<sup>3</sup> Im Original: “knowledge sharing should consequently be considered to be a social process through which individuals try to establish a shared understanding about reality, by using diverse combinations of signs (e.g., language, gestures, illustrations) and tools (e.g., physical objects, communication technologies, mental models)”

<sup>4</sup> Im Original: “knowledge diffuses through barter exchange among pairs of agents”

In Song et al. wird Wissensverteilung (dissemination) definiert als eine Interaktion, in der das Wissen eines Individuums genutzt wird, um das Wissen eines Anderen zu vergrößern, und dadurch über die Organisation verbreitet wird (Song et al. 2003, S. 168f)<sup>5</sup>.

Probst et al. verwenden den Begriff Wissens(ver)teilung zur Beschreibung des Kernprozesses des Wissensmanagement, der dafür sorgen soll, „isoliert vorhandene Informationen oder Erfahrungen für die gesamte Organisation nutzbar zu machen“ (Probst et al. 1998, S. 53).

Wissenstransfer in Organisationen ist der Prozess durch den eine Einheit (z.B. ein Individuum, eine Gruppe, Abteilung oder ein Bereich) durch die Erfahrung einer Anderen beeinflusst wird (Argote et al. 2000, S. 3)<sup>6</sup>. In der kognitiven Psychologie hingegen wird Wissenstransfer homonym verwendet um die Anwendung von Wissen, das in einem Kontext gewonnen wurde in einem anderen Kontext zu bezeichnen (Singley, Anderson 1989, S. 1).

Wissenskommunikation wird von Eppler und Reinhard gleich mehrfach definiert, zum Einen als „die (meist) absichtsvolle, interaktive Konstruktion und Vermittlung von Erkenntnis und Fertigkeit auf der verbalen und nonverbalen Ebene“ (Eppler, Reinhardt 2004, S. 2), aber auch als „Transfer (und die Erweiterung) von ‚know-why‘, ‚know-what‘ und ‚know-how‘ durch unmittelbare (‚face-to-face‘) oder medienbasierte (‚virtuelle‘) Interaktion“ (ibid. S. 3). Schließlich hat für die Autoren „Wissenskommunikation dann stattgefunden, wenn eine Erkenntnis, Erfahrung oder Fertigkeit von einer Person aufgrund von Kommunikation adäquat rekonstruiert werden konnte“ (ibid. S. 1).

Auf Grundlage der Trennbarkeit von Wissen und Person, kann Wissensweitergabe auch als Transfer von Wissensobjekten definiert werden (Boer et al. 2002a, S. 2). Eine Reihe weiterer Definitionen, die jedoch keine wesentlichen neuen Aspekte einbringen findet sich bei (Becker, Knudsen 2003 ). Dort wird das Fazit gezogen, dass eine Analyse der Wissensweitergabe die Intention des Transfers berücksichtigen sollte, die Anwendung des Wissens durch den Empfänger als finaler Schritt des Wissensweitergabeprozesses angesehen werden sollte und die drei Faktoren Vorgeschichte, Transfermechanismus und

---

<sup>5</sup> Im Original: “knowledge dissemination [is] defined as an interaction in which the knowledge of one individual is used to enlarge the knowledge of another and is disseminated throughout the organization”

<sup>6</sup> “Knowledge transfer in organizations is the process through which one unit (e.g., individual, group, department, division) is affected by the experience of another”

Ergebnis des Wissensweitergabeprozesses untersucht werden sollten, um ein umfassendes Verständnis davon zu erlangen (ibid. S. 14). Auch Hendriks berücksichtigt in seinem fünfstufigen Wissensweitergabeprozess sowohl vor- als auch nachgelagerte Schritte. Er startet mit der Aufmerksamkeit für ein Wissensdefizit oder –ungleichgewicht, geht dann weiter mit dem Bereitstellen des Wissens durch die Wissensquelle, den eigentlichen Transfer, das Erhalten des Wissens durch den Empfänger und die Anwendung des Wissens (Hendriks 2004, S. 6).

Hier wird jedoch analog den oben diskutierten Wissensprozessen (siehe Abschnitt 2.2.2) die Auffassung vertreten, dass die Anwendung des Wissens zwar das finale Ziel der Wissensweitergabe ist, der Prozess der Wissensweitergabe aber schon vorher abgeschlossen ist. Es wird daher eine enge Prozessdefinition der Wissensweitergabe verwendet, während die Vorgeschichte und Anwendung des Wissens, ebenso wie die Intention weiter unten in einem umfassenderen Modell der Wissensweitergabe Berücksichtigung finden.

Wissensweitergabe wird also definiert als der Prozess, in dem eine Person (die Quelle) (1) sich entscheidet, Wissen weiterzugeben, (2) das relevante Wissen erinnert, (3) es in Form von (kontextualisierten) Informationen expliziert, (4) die Informationen direkt oder indirekt zu einer anderen Person (Empfänger) weiterleitet, (5) welche die Informationen wahrnimmt, (6) und sie im gegebenen Kontext interpretiert, so dass das Wissen rekonstruiert und in die Wissensbasis der Person integriert wird. Im abschließenden Prozessschritt (7) wird das neu gewonnene Wissen durch den Empfänger evaluiert, wovon der Einfluss des Wissens auf dessen zukünftige Handlungen maßgeblich abhängt.

Der Definition liegen das Modell der Wissensweitergabe zugrunde, das Maier auf Grundlage der Wissensmodelle von Lehner und Hildebrandt entwickelt hat (Maier 2004, S. 68f; Lehner et al. 1995, S. 257) und die Definition von Wissen, nach der es an Menschen gebunden ist und Wissensweitergabe als Austausch von Informationen mit dem Ziel der Wissensgewinnung zu verstehen ist (Boer et al. 2002b, S. 2). Ähnliche, aber einfacher gehaltene Modelle finden sich auch bei (Meredith, Burstein 2000, S. 5ff) und (Nissen 2002, S. 253f).

Obwohl es streng genommen nur Informationen sind, die ausgetauscht werden, wird in dieser Arbeit von Wissensweitergabe gesprochen um zu betonen, dass kontextualisierte und handlungsorientierte Informationen ausgetauscht werden müssen, um beim Empfänger

eine Rekonstruktion des Wissens zu ermöglichen, womit der Informationsaustausch erst sinnvoll wird.

1. Der Prozess beginnt mit der Entscheidung, Wissen in der aktuellen Situation an einen bestimmten oder unbestimmten Empfänger weiterzugeben. Eine Wissensweitergabeaktivität kann bereits in dieser frühen Phase scheitern, z.B. weil keine gute soziale Beziehung zum Empfänger besteht.
2. Die Person erinnert den bestimmten Teil des Wissens, den sie weitergeben möchte. Dieser Schritt wird eigens aufgeführt, da das Erinnern nicht nur dazu führt, dass dieses Wissen wieder „aktiviert“ wird und dadurch Vergessen verlangsamt oder verhindert wird, sondern nach neueren kognitionswissenschaftlichen Theorien dieses Erinnern auch zu einer veränderten Interpretation auf der Seite des Senders führen kann, da Wissen nicht abgelegt und wieder abgerufen werden kann, sondern auch auf Senderseite ein Rekonstruktionsprozess stattfindet (Cohen 1998, S. 30ff).
3. Explikation ist der Prozessschritt, der personengebundenes Wissen in kontextualisierte Informationen umwandelt, die dann durch andere Personen wahrgenommen werden können. Je geringer der Anteil an „verborgenem Wissen“, desto leichter lässt sich das Wissen explizieren. Der implizite Anteil des Wissens kann nur schwer oder gar nicht expliziert werden. Oft ist das Vorführen der Anwendung die einzige Möglichkeit das Wissen zu beschreiben, wobei dies keine Explikation im engeren Sinne darstellt. Explizites Wissen lässt sich dagegen auch mündlich oder schriftlich weitergeben. Der Prozessschritt Explikation ist zwar ähnlich aber nicht zwangsweise identisch mit dem Wissenskonversationsprozess bei Nonaka, da ein unterschiedliches Verständnis von Wissen zugrunde liegt.
4. Liegt die Information vor, kann sie über ein Medium an den Empfänger übermittelt werden. Mündliche Übermittlung nutzt Luft als Medium, schriftliche Papier. In beiden Fällen können auch elektronische Medien eingesetzt werden. Eine direkte Übermittlung besteht, wenn der Empfänger direkt adressiert wird, z.B. durch direkte Ansprache (Face-to-Face) oder durch Schreiben einer Email. Eine indirekte Übermittlung liegt vor, wenn der Adressat beim Explizieren des Wissens noch unbekannt ist, bzw. nur eine anonyme Gruppe von Empfängern bekannt ist. Dies ist der Fall, wenn das Wissen anderen Personen zugänglich gemacht wird, z.B. durch Schreiben und Veröffentlichen eines Buches durch einen Verlag, oder eines Dokumentes im Intranet, oder auch durch Anwendung des Wissens, so dass es durch

andere beobachtet werden kann. Der Schritt der Übermittlung kann auch durch Intermediäre erfolgen und zu signifikanten Verzögerungen zwischen Explikation und der Wahrnehmung der Informationen führen.

5. Wichtige Sinne für die Wahrnehmung von Informationen im Rahmen der Wissensweitergabe sind der auditive und der visuelle Sinn. Der haptische, olfaktorische und gustatorische Sinn dagegen werden i. d. R. weniger eingesetzt um komplexe Informationen aufzunehmen (vgl. Looß 2001). In einigen Branchen kann es zu Abweichungen von dieser Faustregel kommen, z.B. kann der Geruchssinn in der chemischen Industrie für Wissensweitergabe in einigen Fällen durchaus wichtig sein. Auf solche Spezialfälle wird im Folgenden aber nicht näher eingegangen.
6. Die Interpretation wird durch den Kontext beeinflusst, in dem sich der Empfänger befindet, besonders durch seinen kognitiven Zustand. Nach der Interpretation kennt der Empfänger die Bedeutung der Information und kann deren Folgen abschätzen.
7. Schließlich wird das neu gewonnene Wissen vom Empfänger evaluiert. Vorheriges Wissen und neues Wissen müssen in Einklang miteinander gebracht und potentielle Konflikte aufgelöst werden. Das Resultat der Evaluation entscheidet maßgeblich darüber, wie stark sich das neu erworbene Wissen auf zukünftige Handlungen auswirken wird.

Falls weder die Quelle noch der Empfänger das nötige Wissen haben, um ein Problem zu lösen, sondern zusammen an der Lösung arbeiten, indem sie das Problem diskutieren, wird in dieser Arbeit nicht von (reziproker) Wissensweitergabe gesprochen, sondern von gemeinschaftlicher Wissensentwicklung. Es ist jedoch sehr schwierig, zwischen dem geschilderten Fall und dem Fall zu unterscheiden, in dem eine Wissensweitergabe-Aktivität eine umgekehrt gerichtete Aktivität mit einem ähnlichen Inhalt anstößt.

Die Definition stellt explizit auf die Wissensweitergabe auf Individuenebene ab und nimmt damit bewusst eine andere Sicht ein als andere Autoren, die es ebenfalls zulassen, dass eine Gruppe, ein Team oder eine Abteilung direkt von einer anderen Organisationseinheit lernen, wie das z.B. Argote und Ingram behaupten (Argote, Ingram 2000, S. 151). Aufgrund der Erfahrungen des Autors und der später noch vorgestellten Modellierung des Wissensweitergabeprozesses im Rahmen einer Multiagentensimulation, wo ausschließlich die Mikroebene modelliert wird, also Individuen und deren Interaktionen miteinander und der Umwelt, und die Makroebene sich emergent ergibt, folgt die vorliegende Arbeit der Meinung, dass Wissensaustausch nur auf Individuenebene erfolgt. Der Übergang von

Wissen von einer Organisation(-seinheit) auf eine andere lässt sich durch die Wissensweitergabe zwischen Individuen erklären. Gruppen, Abteilungen und Organisationen können kein Wissen weitergeben, es sei denn ihre Mitglieder tun es.

#### 2.2.4 Ein Modell der Wissensweitergabe

Basierend auf der vorgestellten Definition der Wissensweitergabe soll im Folgenden ein Modell diskutiert werden, das zur Einordnung der potentiellen Einflussfaktoren dienen kann, die in Kapitel 3 identifiziert werden. Der Kern des Modells besteht, wie in Abbildung 2-4 zu sehen ist, aus den sieben Schritten der Wissensweitergabe wie im vorherigen Abschnitt beschrieben. Wie von Becker und Knudsen gefordert, wurde die Vorgeschichte und das Ergebnis des Wissensweitergabeprozesses berücksichtigt (Becker, Knudsen 2003 S. 14). Die Vorgeschichte ist in Form einer Gelegenheit zur Wissensweitergabe abgebildet (siehe dazu auch das SEKS Modell Oliver et al. 2003; Wiig 2003). Das Ergebnis ist in Form der Nutzung des Wissens modelliert. Diese ist insbesondere im organisatorischen Kontext wichtig, da Organisationen Wissen nicht um des Wissens selbst Willen sammeln. Das Modell lehnt sich dabei stark an existierende Kontextmodelle an.

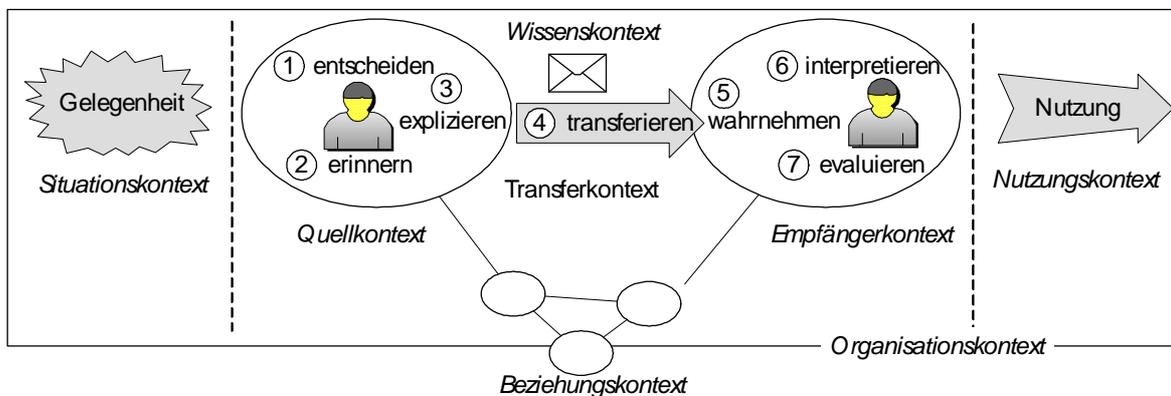


Abbildung 2-4: Modell der Wissensweitergabe (eigene Darstellung)

Quelle, Empfänger, Kanal, Nachricht und Kontext sind die fünf Basiselemente der Wissensweitergabe (Szulanski 1996). Diese Elemente finden sich auch im Wissensweitergabemodell wieder (siehe Abbildung 2-4). Quelle und Empfänger sind jeweils durch ein Personensymbol repräsentiert, der Kanal ist durch den grauen Pfeil symbolisiert, die Nachricht wird durch den Briefumschlag abgebildet. (Joshi et al. 2004) verfeinern dieses Konzept und differenzieren den Quell- und Empfängerkontext, den Wissenskontext, den Beziehungskontext und den Situationskontext. Dem gegenüber sprechen (Cummings, Teng 2003) nur von vier Kontextbereichen, dem Wissenskontext,

Beziehungskontext, Empfängerkontext und Aktivitätskontext (ibid. S. 41). Das umfassendste Modell stammt von Becker und Knudsen (2003) welches im hier vorgestellten Modell nochmals verfeinert wurde, so dass der Kontext nun in sieben Kontextbereiche aufgegliedert ist. Die Bedeutung des Kontexts für die Wissensweitergabe betonen auch (Herrmann, Kienle 2004), die als Kontextbereiche den inneren Kontext (Quell- und Empfängerkontext in Abbildung 2-4) vom äußeren Kontext trennen, der sich für Quelle und Empfänger unterscheiden kann, wobei es aber auch einen gemeinsamen Teil des äußeren Kontextbereichs gibt (ibid. S. 53). Im Folgenden werden die Elemente des Modells erläutert.

*Gelegenheit*: ein Aspekt, der in der Literatur nur selten Berücksichtigung findet, ist die Gelegenheit zur Wissensweitergabe, ohne die der Prozess nicht in Gang kommen kann. In (Berends 2002) und (Berends et al. 2004) wird zumindest anhand des Initiators der Wissensweitergabe nach Push oder Pull unterschieden, also in Weitergabeaktivitäten, die von der Quelle initiiert werden (z.B. die Lösung eines langwierigen Problems, die dem Kollegen erzählt wird) und solche, die vom Empfänger initiiert werden (z.B. das Fragen eines Kollegen nach der Lösung für ein Problem). Die Identifikation des Partners für die Wissensweitergabe ist wiederum ein eigener komplexer Prozess, der daher genau wie die Anwendung nicht als Teil der Wissensweitergabe im engeren Sinne angesehen wird. Nichts desto trotz ist er ein wichtiger Faktor, der in der Simulation später berücksichtigt werden muss. Eine Reihe von „Wissenstransfermechanismen“ wurden von Argote et al. gesammelt und können als Beispiele für Gelegenheiten zur Wissensweitergabe dienen, z.B. Personalrotation, Training, Beobachtungen, Technologietransfer, wissenschaftliche Publikationen und Präsentationen, sowie Interaktionen mit Lieferanten und Kunden (Argote et al. 2000, S. 3). Auch Eppler und Reinhardt listen einige typische Situationen auf, in denen Wissensweitergabe stattfinden kann, z.B. bei der Entscheidungsvorbereitung, wenn Fachleute ihr Wissen an Manager weitergeben, bei Weiterbildungsmaßnahmen, wenn Dozenten Wissen an die Kursteilnehmer weitergeben, oder Einarbeitung, wenn langjährige MitarbeiterInnen ihr Wissen an neue Mitarbeiter weitergeben.

*Übertragungskontext*: Die Übertragung des expliziten Wissens kann persönlich, oder anonym, synchron oder asynchron erfolgen. Es kann unterschiedliche Multiplizität haben (1-1, 1-n, m-n) und verschiedene Medien zur Übertragung nutzen (Luft, Papier, elektronische Medien). Hier wird noch einmal besonders deutlich, dass Wissensweitergabe in diesem Modell in einem weiten Sinne aufgefasst wird (vgl. dazu auch Eppler, Reinhardt

2004, insb. S. 3). Das Schreiben eines Buches, das ein anderer liest, um daraus das Wissen des Autors zu rekonstruieren, wird genauso als Wissensweitergabe betrachtet (anonym, asynchron, 1-n, Papier) wie das Beantworten der Frage eines Kollegen per Instant Message (persönlich, synchron, 1-1, elektronisches Medium) und der Austausch von Erfahrungen im Rahmen einer Projektnachbesprechung (persönlich, synchron, m-n, Luft).

*Nutzung:* Auch die Nutzung des Wissens muss nach Meinung des Autors breiter betrachtet werden, als dies in der Literatur oft getan wird. Nicht nur die Anwendung des Wissens in einem Geschäftsprozess, sondern auch die Weitergabe des Wissens wird hier als Nutzung betrachtet.

*Kontextbereiche:* Weil die Unterschiede im Kontext von Sender und Empfänger besondere Herausforderungen an die Wissensweitergabe stellen, wird Forschung, die den Kontext explizit berücksichtigt, als besonders lohnend angesehen (Argote, Ingram 2000, S. 164). Dementsprechend sollen hier die relevanten Kontextbereiche diskutiert werden. Cummings und Teng identifizieren vier relevante Kontextbereiche für die Wissensweitergabe, nämlich den Wissenskontext, den Beziehungskontext zwischen Sender und Empfänger, den Empfängerkontext und den Aktivitätenkontext (Cummings, Teng 2003, S. 41). Basierend auf dieser Erkenntnis, den Basiselementen der Wissensweitergabe nach Szulanski (siehe oben) und anderen Arbeiten definieren Joshi et al Kontextbereiche für die Quelle, den Empfänger, die Beziehung zwischen Beiden, die Situation und das Wissen (Joshi et al. 2004, S. 2). Cummings und Teng meinen die Weitergabeaktivitäten, wenn sie von einem Aktivitätenkontext sprechen, was in etwa den Charakteristika des Transfermechanismus bei Becker und Knudsen entspricht (Becker, Knudsen 2003 S. 15). Aktivitätenkontext kann aber auch im Sinne der Aktivitätentheorie interpretiert werden, die soziale Regeln, vermittelnde Artefakte und die Arbeitsteilung als Kontextvariablen zwischen dem Subjekt und dem Objekt der Aktivität sowie den anderen involvierten Akteuren definiert (siehe auch Aktivitätentheorie im Abschnitt 3.3.4) (Boer et al. 2002a, S. 4). Becker und Knudsen unterscheiden im Gegensatz dazu zwischen dem Organisationskontext und dem sozialen Netzwerk in das das Subjekt eingebettet ist, was beides Teile des Aktivitätenkontexts im Sinne der Aktivitätentheorie beinhaltet (Becker, Knudsen 2003 S. 15). Der Situationskontext bei Joshi et al. ist ebenfalls sehr ähnlich. Allerdings ist eine Aktivität im Sinne der Aktivitätentheorie eher eine länger andauernde Tätigkeit, wogegen eine Situation oft auch kurz sein kann, z.B. ein Projekttreffen. Die Situation wird daher im vorgestellten Modell der Gelegenheit zugerechnet. Eine ausführlichere Diskussion der Aktivitätentheorie und

Gelegenheiten findet sich bei (Hädrich, Maier 2004). Menschen bilden im Laufe ihrer Entwicklung individuell unterschiedliche mentale Modelle zur Ordnung ihrer sensorischen Erfahrung und als Abbild ihrer Verstehens- und Gestaltungsanstrengungen (Piaget 1971, S. 27). Deswegen ist es wichtig, diese unterschiedlichen mentalen Modelle als Teil des Empfänger- und Senderkontexts für die Wissensweitergabe zu berücksichtigen.

Die Faktoren, die Wissensweitergabe innerhalb der einzelnen Kontextbereiche beeinflussen, werden in Kapitel 4 genauer untersucht.

### **2.3 Wissensmanagement in der Softwareentwicklung**

Das Wissensmanagement bildet den Rahmen für die in dieser Arbeit untersuchten Sachverhalte. Deswegen soll an dieser Stelle ein kurzer Überblick über die Hauptentwicklungen im Wissensmanagement gegeben, sowie der Begriff selbst definiert werden.

Schüppel definiert Wissensmanagement als einen Entwurf, der alle möglichen human- und technikorientierten Interventionen und Maßnahmenpakete umfasst, die dazu geeignet sind, die von ihm definierten Wissensprozesse in einer Organisation zu optimieren (Schüppel 1997, S. 191f). Maier sieht Wissensmanagement als „Managementfunktion, die für die regelmäßige Auswahl, Implementierung und Evaluation der zielorientierten Wissensstrategien verantwortlich ist, mit dem Ziel, den Umgang mit internem und externem Wissen zu verbessern und dadurch die Leistungsfähigkeit der Organisation zu erhöhen. Die Implementierung umfasst dabei alle personenorientierten, organisatorischen und technischen Instrumente, die dafür geeignet sind, den organisationsweiten Stand der Kompetenzen, die Ausbildung und Lernfähigkeit der Mitarbeiter dynamisch zu optimieren und eine kollektive Intelligenz zu entwickeln“ (Maier 2004, S. 55).

Wie in dieser Definition anklingt, haben sich in der Literatur drei verschiedene Interventionsebenen für Wissensmanagement herauskristallisiert: Personen (v.a. durch das Human Resource Management, HRM), die Technik (v.a. IKT) und die Organisation (Schindler 2001, S. 39). Diese drei Interventionsebenen werden durch die Strategie verknüpft und finden dort ihre Entsprechung mit einer Personalisierungs-, Kodifizierungs- oder „Bridging-the-gap“-Strategie, welche auch als prozessorientierte Strategie bezeichnet wird (Maier, Remus 2003, S. 63f). Deshalb ist der hier dargestellte kurze Abriss über das Wissensmanagement in die vier Abschnitte Strategie, IKT, Personal und Organisation gegliedert.

### 2.3.1 WM als strategische Aufgabe

Eine viel zitierte Publikation, die sich mit Wissensmanagement aus strategischer Sicht beschäftigt ist Hansen et al. (1999). Dort wird in erster Linie zwischen der Personalisierungs- und der Kodifizierungsstrategie unterschieden. Personalisierung bedeutet dabei, dass WM sich in erster Linie mit dem Zusammenbringen von Experten und dem Ermöglichen von Kommunikation zwischen den Mitarbeitern beschäftigt, während Kodifizierung auf das Dokumentieren von Wissen in expliziter Form abstellt. Eine Organisation sollte sich für eine dominierende Strategie entscheiden, während die andere Strategie eine unterstützende Rolle einnimmt (Zack 1999). Man kann aber nicht beide Strategien gleichermaßen verfolgen.

In späteren Arbeiten wird diese Unterscheidung aufgenommen, aber noch um eine weitere strategische Alternative ergänzt. Die so genannte „Bridging-the-gap“-Strategie wird v.a. im Zusammenhang mit prozessorientiertem WM als sinnvolle Strategie angesehen (Maier, Remus 2003, S. 63f). Dabei ist es entscheidend, das im Geschäftsprozess benötigte Wissen zur richtigen Zeit am richtigen Ort verfügbar zu haben, unabhängig davon, ob es in dokumentierter Form oder in Form von Experten vorliegt.

Im konkreten Fall ist es für eine Organisation nicht ausreichend eine der drei Alternativen zu wählen, sondern das favorisierte Modell muss auch verfeinert, konkretisiert und an die Rahmenbedingungen in der Organisation angepasst werden. Van der Spek et al. stellen dazu einen idealtypischen Prozess zur Definition einer WM-Strategie und Einführung von WM in Organisationen vor. Er besteht aus sechs Phasen (van der Spek et al. 2003, S. 451ff):

1. Spezifizieren des Geschäftsfalls
2. Identifizieren von Wissensbereichen, die im Kontext des Geschäftsfalls relevant sind
3. Identifizieren der wichtigsten Leistungsindikatoren (Key Performance Indicators, KPI) im Kontext des Geschäftsfalls
4. Analysieren der Wissensbereiche auf ihre derzeitigen und zukünftigen Auswirkungen auf die Leistungsindikatoren
5. Bewerten der Wissensbereiche bezüglich der Mitarbeiterkenntnisse, Kodifizierung und Diffusion von Wissen
6. Entwickeln des WM-Aktionsplans

Aus dieser Darstellung geht hervor, dass sich die WM-Strategie am Kerngeschäft der Organisation und dem Ist-Zustand innerhalb der Organisation orientieren muss. Dazu ist es erforderlich Indikatoren zu definieren, die sich einerseits gut messen lassen und andererseits direkte Rückschlüsse auf die Unterstützung des Geschäftsfalls ermöglichen. Schließlich können ausgehend von diesen Informationen konkrete Aktionen festgeschrieben werden, die zu einer Verbesserung der Situation führen, oder es bei gutem Ist-Zustand ermöglichen auch in Zukunft dieses Niveau zu halten. Es ist also wichtig für eine Strategie konkret, messbar und umsetzbar zu sein, statt auf abstrakter Ebene wenig greifbare Ziele zu definieren.

Eine weitere strategische Aufgabe ist die Einführung von WM in Organisationen. Damit beschäftigen sich O'Dell et al. und identifizieren durch Analyse von neun Fallstudien fünf Phasen (O'Dell, al. 2003):

- |                               |                            |
|-------------------------------|----------------------------|
| (1) Getting Started           | (2) Explore and Experiment |
| (3) Pilots and KM Initiatives | (4) Expand and Support     |
| (5) Institutionalize KM.      |                            |

Es überrascht auf den ersten Blick, bei einer strategisch geplanten Einführung eine Experimentierphase explizit vorzusehen (Phase 2). Es ist jedoch sinnvoll, die Resonanz auf verschiedene WM-Maßnahmen vor deren umfassender Einführung im Kleinen zu testen. Dabei sollen nach O'Dell nicht nur top-down geplante Pilotprojekte eingesetzt werden (Phase 3), sondern auch gezielt von Mitarbeitern selbst initiierte Projekte evaluiert und gefördert werden (Phase 4). Die abschließende Phase 5 sieht vor, dass WM ein selbstverständlicher Teil der Organisation wird.

Maier dagegen findet bei seiner Befragung von deutschen Großunternehmen vier Szenarien, die nicht direkt mit den Phasen O'Dells korrespondieren (Maier 2004, S. 514ff): (1) WM Anfänger, (2) zentralisiert: Markt und Hierarchie, (3) dezentralisiert: Netzwerk und Community, (4) persönlich: Idee und Individuum. Die WM-Anfänger korrespondieren noch am ehesten mit O'Dells Phasen und subsumieren Organisationen die sich in der Phase eins, zwei und drei befinden. Die anderen Szenarien sind jedoch eher parallel zueinander als unterschiedliche Ausprägungen von fortgeschrittenen WM-Implementierungen zu sehen. Maier selbst sieht in erster Linie Markt und Hierarchie als Alternative zu Netzwerk und Community, während er Idee und Individuum als finales Stadium betrachtet. Markt und Hierarchie ist dabei Ausdruck einer dominierenden Kodifizierungsstrategie während Netzwerk und Community mit einer Personalisierungs-

strategie einhergeht. Idee und Individuum ist dagegen eine sehr spezielle Form des WM, die in erster Linie bei besonders expertenabhängigen Organisationen wie z.B. Krankenhäusern, Anwaltskanzleien oder Beratungsunternehmen anzutreffen ist.

### **2.3.2 WM als technische Aufgabe**

Die Diskussion über Unterstützung des Wissensmanagement durch IKT ging von einer Reihe von Systemklassen aus, die einen Bezug zu kodifizierten Wissen aufweisen. Dazu zählen unter anderem Dokumentenmanagementsysteme (DMS), Contentmanagementsysteme (CMS), Suchmaschinen, Data Warehouses und Business Intelligence Tools, Groupwaresysteme, Experten- und E-Learningsysteme (Tsui 2003, S. 7ff; Maier 2004, S. 232). Diese bieten jeweils Unterstützung für einen Teilaspekt des WM und weisen darüber hinaus auch Überschneidungen auf. Aufbauend auf diesen Systemklassen wurde schnell offensichtlich, dass die Integration der Systeme auch eine Verbesserung der Unterstützung verspricht. Im Zuge dieser Entwicklung wurden integrierte Systeme entwickelt, die unter dem Namen Wissensmanagementsystem (WMS) firmieren.

Ein WMS „ist ein IuK-System im Sinne eines Anwendungssystems oder einer IKT-Plattform, das Funktionen für die kontextualisierte Behandlung von explizitem und implizitem Wissen im ganzen Unternehmen, oder dem Teil davon, das von einer WM-Initiative betroffen ist, kombiniert und integriert.“ Es unterstützt Netzwerke von Wissensarbeitern bei der Ausübung von Tätigkeiten in Wissensprozessen mit dem Ziel der Unterstützung von organisatorischem Lernen und Effektivität (Maier 2004, S. 83). Beispiele für solche Produkte sind Open Text Livelink, die Empolis e:Corporate Knowledge Suite, EMC Documentum oder Hummingbird Enterprise KM<sup>7</sup>. Obwohl sich diese Produkte bemühen, eine umfassende Unterstützung für WM zu bieten, sind sie doch eher den integrativen WMS zuzuordnen, die hauptsächlich auf kodifiziertes Wissen abstellen. Diese Systeme werden in jüngerer Zeit auch unter dem Begriff Enterprise Content Management subsumiert, womit die Verschmelzung von DMS und CMS bezeichnet wird (Asprey, Middleton 2003; Laugero, Globe 2002). Im Bereich der interaktiven WMS finden sich v.a. umfassende Plattformen in der Nachfolge von Groupwaresystemen. Insbesondere des System Lotus Notes/Domino<sup>8</sup>, das mittlerweile von IBM entwickelt und vertrieben wird zieht nach wie vor eine große Aufmerksamkeit auf

---

<sup>7</sup> [www.opentext.com](http://www.opentext.com), [www.empolis.com](http://www.empolis.com), [www.hummingbird.com](http://www.hummingbird.com), [www.documentum.com](http://www.documentum.com)

<sup>8</sup> [www.ibm.com/software/lotus/](http://www.ibm.com/software/lotus/)

sich. Aber auch auf Basis von Microsoft Exchange Server und Microsoft Office Sharepoint Server werden Lösungen gebaut, die diesem Bereich zuzuordnen sind. Im Kielwasser der großen Aufmerksamkeit, die das Thema Wissensmanagement erfahren hat, wurden um die Jahrtausendwende eine Reihe weiterer Produkte unter dem Schlagwort WMS verkauft. Derzeit scheint aber eher eine rückläufige Entwicklung in diesem Bereich stattzufinden. Dafür werden unter dem Begriff Enterprise Knowledge Portal eine Reihe von Produkten angeboten, die einen einheitlichen Zugang zu den unterschiedlichsten Informations- und Datenquellen bieten sollen (Tsui 2003, S. 17). Beispiele hierfür sind IBM Websphere Portal, Plumtree Enterprise Portal / Collaboration und SAP Netweaver Portal<sup>9</sup>. Diese stellen aber genauer betrachtet nur die Zugangs- und teilweise auch die Integrationstechnologie bereit, nicht aber die vollständige Funktionalität, die für ein WMS benötigt wird.

Durch die zunehmende Komplexität der Anwendungen und den ökonomischen und technischen Problemen bei der Ablösung existierender Einzelsysteme durch eine neue, umfassende WMS-Plattform (vgl. Strehlitz 2005) zeichnet sich seit 2004 eine Entwicklung weg von einem umfassenden System hin zur Integration verschiedener Systeme und Bausteine ab. Nicht zuletzt aufgrund der anhaltenden Diskussion über serviceorientierte Architekturen (SOA, Alonso et al. 2004, S. 123ff) und vorbereitet durch die Entwicklungen im Bereich Enterprise Application Integration (EAI, Alonso et al. 2004, S. 67ff) gehen immer mehr Hersteller dazu über, ihre eigenen mehr oder weniger monolithischen Systeme durch ein Netz von Bausteinen abzulösen, die auf Basis eines Application Servers integriert sind. Dies bietet v.a. auch Vorteile bei der Integration von Software aus Unternehmensübernahmen, wie es in jüngerer Zeit häufig zu beobachten war<sup>10</sup>.

Ein Architekturvorschlag für ein solchermaßen modularisiertes WMS wurde aufbauend auf einer idealtypischen Architektur für WMS (Maier 2004, S. 258ff) unter dem Begriff Enterprise Knowledge Infrastructure gemacht (siehe Abbildung 2-5, vgl. auch (Maier et al. 2005, S. 76ff)).

---

<sup>9</sup> [www.ibm.com/software/genservers/portal/](http://www.ibm.com/software/genservers/portal/), <http://www.plumtree.com/products/collaboration/>, [www.sap.com/germany/solutions/netweaver/netweaverportal/](http://www.sap.com/germany/solutions/netweaver/netweaverportal/),

<sup>10</sup> z.B. Oracle mit den Übernahmen von J.D.Edwards, Peoplesoft und Siebel, sowie Innobase, Retek, Oblix und weiteren Anbietern, siehe z.B. [www.heise.de/newsticker/meldung/64716](http://www.heise.de/newsticker/meldung/64716)

Darunter wird „eine umfassende IKT-Plattform für Wissensdokumentation, Zusammenarbeit und Lernen [verstanden], die Funktionen für den strukturierten und kontextualisierten Umgang mit explizitem und implizitem, organisationsinternem und -externem Wissen kombiniert und integriert. Damit werden Netzwerke von Mitarbeitern unterstützt, mit dem Ziel der Steigerung der Produktivität von Wissensarbeit“ (Maier, Peinl 2005a).

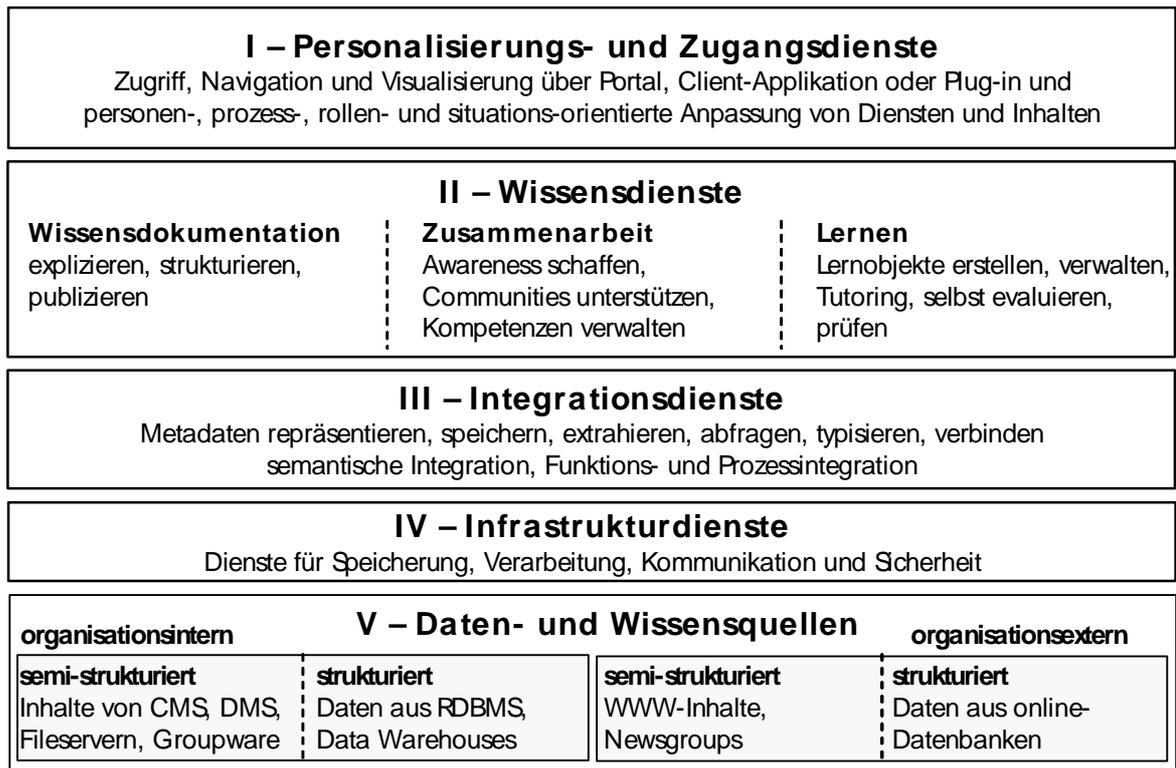


Abbildung 2-5: Architektur einer Enterprise Knowledge Infrastructure (Maier, Peinl 2005a)

Auf der untersten Ebene (V) sind dabei die *Daten- und Wissensquellen* angesiedelt, die aus den unterschiedlichsten Quellsystemen stammen und sich in strukturierte und semi-strukturierte Daten, sowie organisations-interne und externe Quellen aufteilen lassen. Je nach Organisationstyp kann einzelnen Quellen eine größere Bedeutung zukommen. Prinzipiell sind jedoch für eine umfassende Betrachtung alle Quellen relevant.

Auf der *Ebene der Infrastrukturdienste (IV)* sind Technologien angesiedelt, die für eine sichere Zugreifbarkeit der Quellen sorgen und weitere Basisdienste für die darauf aufbauenden Anwendungen zur Verfügung stellen (z.B. Versionierung für Dokumente), jedoch selbst noch keine spezifischen Funktionen für Geschäftsanwendungen besitzen (z.B. Application Server).

Die *Dienste auf der Integrationsebene (III)* bauen darauf auf und versuchen Datenelemente zueinander in Beziehung zu setzen (Daten- und semantische Integration), sowie Geschäftsprozesse durch Anwendungssystem-übergreifende Workflows elektronisch abzubilden (Funktions- und Prozessintegration). Für ersteres wird als technische Umsetzung die Nutzung von XML-basierten Standards (insb. Semantic Web Standards) und Ontologien diskutiert (Maier et al. 2005, S. 171ff). Letzteres basiert auf den Standards für Web Services und Web Service Orchestration (Maier et al. 2005, S. 203ff). Quasi als Bindeglied zwischen Daten und Anwendungssystemen (AWS) stehen die Benutzer. Deren elektronische Repräsentation in Form von Benutzerkonten, Zugangsberechtigungen und persönlichen Systemeinstellungen werden daher unter dem Begriff Benutzerintegration (in der Praxis meist als Identity Management bezeichnet) gesondert behandelt. Aus WM-Sicht kommen zu den genannten Benutzerdaten auch noch deren Skill-Profil, also das Meta-Wissen über die Kenntnisse und Fähigkeiten der Mitarbeiter, und die Zugriffshistorie, die als Grundlage für einen personalisierten Zugang zu kodifiziertem Wissen dienen können. Schließlich wird als weiterer Dienst eine einheitliche Suchfunktion angeboten.

Auf der *Ebene der Wissensdienste (II)* werden dann spezifische Funktionen zum Umgang mit Wissen als Objekt (Wissensdokumentation), Wissen in Personen (Lernen) und Wissen in sozialen Systemen (Zusammenarbeit) angeboten (vgl. auch Abschnitt 2.1). Diese Dreiteilung findet sich auch bei anderen Autoren wieder (z.B. Mandl 2003, S. 38; van der Spek et al. 2003, S. 446). Technisch gesehen finden sich dort die Nachfolger der Systemklassen Enterprise Content Management, E-Learning und Groupware wieder, ergänzt um einige zusätzliche Systemklassen wie z.B. Skillmanagement und Community-Home-Spaces (Foren). Die Zuordnung der einzelnen Systeme und Funktionen zu den drei Bereichen ist weder eindeutig noch überschneidungsfrei. Die Einteilung erscheint aber aus didaktischer Sicht trotzdem sinnvoll.

Auf der Ebene der *Personalisierungs- und Zugangsdienste (I)* werden der Zugriff auf die Systeme, aber auch die Inhalte selbst situations-, rollen-, prozess- und personenabhängig aufbereitet. Dabei spielen z.B. Limitationen der verwendeten Endgeräte (z.B. mobiler Zugriff mit PDA), Zeitpunkt und Ort des Zugriffs, sowie der Arbeitskontext und Vorkenntnisse des Benutzers eine Rolle. Personalisierung kann sowohl von Themenverantwortlichen vorgenommen werden, als auch vom Benutzer selbst.

Speziell für die Softwareentwicklung werden in der Literatur bisher weniger integrierte Lösungen angeboten sondern hauptsächlich einzelne Spezialsysteme wie Experience

Management oder Case-Based Reasoning Systeme, sowie spezialisierte WMS für die Softwareentwicklung, die auf Wiederverwendung von Softwarecode und Programmierprobleme abstellen (Dingsøyr, Conradi 2002, S. 404). CASE Werkzeuge (Computer Aided Software Engineering) können eine Ergänzung zu den genannten Systemen darstellen (Henninger 1997, S. 24). Erste Schritte in Richtung stärker integrierter Plattformen für die Softwareentwicklung sind erst in jüngster Zeit zu erkennen (z.B. Microsoft Visual Studio Team System<sup>11</sup>).

### **2.3.3 WM als personenorientierte Aufgabe**

Die personenorientierte Perspektive wird z.B. von Pädagogen (z.B. Mandl 2003), Psychologen (z.B. Reinmann, Mandl 2004) und Personalwissenschaftlern (z.B. Yahya, Goh 2002) vertreten. Dementsprechend wird die Bedeutung von Lernkonzepten und Lernprozessen (Dehnbostel, Dybowski 2000; Boud, Middleton 2003), kognitiven Prozessen (Anderson et al. 2001; Hinds, Pfeffer 2003) und sozialen Netzwerken (Erickson, Kellogg 2003; Winkler, Mandl 2003), bzw. die gezielte Planung der Personalakquisition und betrieblichen Weiterbildung, sowie die Motivation der Mitarbeiter über Anreizsysteme und ein Kontrollieren der Fluktuation (Becker 2002; Yahya, Goh 2002) betont. Tsui identifiziert als die Hauptherausforderungen für WM im HRM Bereich die Lokalisierung von Expertise, Personalentwicklung, E-Learning und Leistungsmessung (Tsui 2003, S. 21). Auch der Aufbau von Kompetenznetzwerken oder die Förderung von Communities of Practice (CoP) sind Maßnahmen innerhalb einer personenorientierten WM Initiative. Eine Community of Practice ist eine Gruppe von Personen, die sich durch gemeinsame Expertise und Leidenschaft für eine gemeinsame Unternehmung informell zusammengehörig fühlen (Wenger, Snyder 2000, S. 139). CoPs helfen dabei die Wissensmanagementstrategie umzusetzen, neue Produktfelder zu entwickeln, Best Practices zu transferieren, sowie neue Talente anzuwerben, weiterzuentwickeln und in der Organisation zu halten (S. 140f).

Das Anwerben qualifizierter Mitarbeiter wird auch als Kernaufgabe der Personalarbeit angesehen, insbesondere im IT-Bereich wo der Fachkräftemangel seit Jahren immer wieder beklagt wird (Scholz 2000, S. 14). Um die Fachkräfte anschließend auch im Unternehmen zu halten, muss laut Scholz ein motivierendes Arbeitsumfeld geschaffen werden. Dies kann erreicht werden, indem klar definierte Ziele und Visionen vermittelt,

---

<sup>11</sup> <http://msdn2.microsoft.com/de-de/teamsystem>, letzter Zugriff am 27.12.2006

herausfordernde Arbeitsaufgaben gestellt, Chancen zur individuellen Weiterbildung geboten werden und eine Führungskraft beratend zur Seite steht (ibid. S. 16f). Eine gute Entlohnung wird zwar ebenfalls als wichtig erachtet, stellt jedoch keinen Motivations-, sondern einen Hygienefaktor im Sinne Herzbergs Motivationstheorie dar (siehe Abschnitt 3.3.2).

CoPs können wegen ihres informellen Charakters und der Selbstorganisation zwar nicht direkt vom Management angeordnet werden, es ist jedoch möglich durch unterstützende Maßnahmen eine förderliche Umgebung für CoPs zu schaffen (Wenger, Snyder 2000, S. 143f). Dazu sollten strategisch wichtige CoPs identifiziert, diesen eine Infrastruktur zur Verfügung gestellt und ihr Wert nicht mit traditionellen Methoden gemessen werden. Unter Infrastruktur verstehen Wenger und Snyder u.a. auch finanzielle Mittel, da CoPs im Gegensatz zu Abteilungen kein festes Budget haben, über das sie verfügen können.

Um eine CoP gründen zu können müssen Gleichgesinnte identifiziert werden. Dazu kann Skillmanagement einen Beitrag leisten (Beck 2003). „Gelbe Seiten“, in denen jeder Mitarbeiter mit seiner Position und Zuständigkeiten verzeichnet ist, stellen einen ersten Schritt dar (ibid. S. 1383). Diese werden durch ein Skillmanagementsystem (SkMS, siehe Abschnitt 2.4.2) ergänzt, welches die Skillprofile der Mitarbeiter enthält. Sie können zum Besetzen von Projektteams verwendet, zum Besetzen neuer Stellen eingesetzt und zum Identifizieren von Wissenslücken und geeigneten Trainingsmaßnahmen zum Schließen dieser Lücken genutzt werden.

### **2.3.4 WM als organisatorische Aufgabe**

Die Organisationsstruktur eines Unternehmens besteht aus der Aufbau- und der Ablauforganisation. Bei der Aufbauorganisation unterscheiden Schnauffer et al. die Primär-, Sekundär- und die Tertiär-Organisation (Schnauffer et al. 2004, S. 17). Die Idee geht auf Nonaka zurück, der die Dreiteilung bereits in seiner Hypertextorganisation vorgeschlagen hat (Nonaka 1994). Während die Primär-Organisation eine der bewährten Organisationsformen wie Hierarchie, Stab-Linie oder Matrix annimmt und für die Erledigung von Routineaufgaben von Bedeutung ist, wird die Sekundär-Organisation von Teams gebildet, die mit der Lösung einmaliger Aufgaben befasst sind. Die Tertiär-Organisation wird schließlich von Schnauffer et al. vorgeschlagen um die Wissensbasisschicht Nonakas in den Fokus der Betrachtung zu rücken. Sie wird gebildet von Wissensnetzwerken und Communities (Schnauffer et al. 2004, S. 17). Aus WM-Sicht ist v.a. die Sekundär- und Tertiär-Organisation betrachtenswert, da dort wissensintensive

Tätigkeiten verrichtet werden, während in der Primärorganisation überwiegend Routineaufgaben bearbeitet werden. Eine weitere relevante Frage für das Wissensmanagement ist, ob es eine eigene Abteilung für Mitarbeiter gibt, die hauptsächlich mit WM-Aufgaben betraut sind und wie diese in die Primär-Organisation eingegliedert ist (Maier 2004).

Weiterhin fällt die Definition von neuen Rollen im Zusammenhang mit der Einführung von WM in den Bereich der Organisation. Eine Schlüsselrolle kommt dabei dem WM-Verantwortlichen zu (Wissensmanager oder CKO, Chief Knowledge Officer genannt, McKeen, Staples 2003). Es existieren aber noch eine ganze Reihe weiterer Rollen, die von den Mitarbeitern oft neben ihrer Hauptrolle im Geschäftsprozess wahrgenommen werden. Maier identifiziert die Rollen Themenspezialist, Wissensadministrator, Wissensbasis-Administrator, Wissensvermittler (Knowledge Broker), Boundary Spanner (frei übersetzt etwa: Verbindungsglied), WM-Sponsor, Communitymanager, Mentor, Coach, WM-Koordinator, Teilnehmer, bzw. Wissensarbeiter und WM-Partner (Dingsøyr, Conradi 2002, S. 406f; Maier 2004, S. 142ff). Im Rahmen des WM bei mittelständischen Individualsoftwareentwicklern ist ein WM-Koordinator vermutlich nicht nötig, da bei maximal 250 Mitarbeitern die Koordination zwischen den wenigen gleichzeitig laufenden Initiativen auch dezentral möglich ist.

Der zweite Bereich innerhalb der Organisation ist die Ablauforganisation. Dort stehen die Geschäftsprozesse im Fokus. Da Wissensmanagement nur dann sinnvoll ist, wenn das Wissen auch in den Geschäftsprozessen angewendet wird, hat sich der Forschungszweig des prozessorientierten WM etabliert, der die Verknüpfung von Geschäftsprozessen einerseits und Wissensprozessen andererseits untersucht (z.B. Strohmaier 2005). Der Ansatz geht auf Allweyer zurück, der 1998 die ARIS-Methode zur Modellierung von Geschäftsprozessen um wissensorientierte Aspekte ergänzt hat und damit die Grundlage für die wissensorientierte Analyse der Prozesse gelegt hat (Allweyer 1998b; Allweyer 1998a). In der Folge hat sich prozessorientiertes Wissensmanagement als eigenständiger Themenbereich etabliert und es wurden einige umfassende Arbeiten dazu veröffentlicht (z.B. Thiesse 2001; Remus 2002). Viele Aufsätze in diesem Bereich konzentrieren sich jedoch ausschließlich auf die Modellierung und machen keine konkreten Vorschläge, wie zur Prozessverbesserung vorzugehen ist (z.B. die Ansätze von Heisig, Gronau und Schauer, für einen Überblick siehe Hädrich, Maier 2004, S. 5). Erste konkrete Verbesserungsvorschläge, die auch mit Werkzeugunterstützung aufwarten können, wurden erst vor kurzem vorgestellt (Strohmaier 2005).

### 2.3.5 Wissensintensive Tätigkeiten und Organisationen

Nach diesem kurzen Überblick über die verschiedenen Aspekte des WM soll im folgenden Abschnitt über die Art der Tätigkeit und den Typ von Organisation eine weitere Eingrenzung des Untersuchungsgebiets vorgenommen werden.

Die Begriffe Wissensarbeiter und Wissensarbeit werden in der Literatur häufig verwendet, obwohl sie streng genommen nicht zutreffend sind. Seit Drucker Ende der 50er Jahre von Wissensarbeitern sprach, die ihre Arbeit durch Anwendung von Konzepten, Ideen und Theorien erledigen, die sie in systematischer Ausbildung gelernt haben, statt mit Muskelkraft oder manuellen Fähigkeiten (Drucker 1959; vgl. auch Elliman et al. 2005, S. 80), wurde der Begriff oft diskutiert. In jüngeren Publikationen wird der Begriff Wissensarbeit bevorzugt, um Diskriminierung zu vermeiden, der Begriff des Wissensarbeiters findet sich aber immer noch in zahlreichen Beiträgen. Aber auch an diesem Begriff wird zunehmend Kritik geäußert, da es eine „Nicht-Wissensarbeit“ als Gegensatz zu Wissensarbeit nicht gibt und eine Privilegierung einer bestimmten Klasse von Menschen vermieden werden soll (Schultze 2003, S. 51).

Obwohl es trotzdem als legitim erachtet wird, den Begriff der Wissensarbeit zu verwenden, um die Gemeinsamkeiten der Tätigkeiten von Arbeitern unterschiedlicher Branchen und Berufsgruppen zu beschreiben und wissenschaftlich zu untersuchen (Schultze 2003, S. 53), wird in dieser Arbeit der Begriff wissensintensive Tätigkeiten bevorzugt, der in der Literatur zunehmend Verwendung findet (z.B. Apostolou, Mentzas 2003; Hädrich, Maier 2004) und stärker darauf hinweist, dass immer nur ein Teil der gesamten Tätigkeit eines Mitarbeiters aus wissensintensiven Tätigkeiten besteht. Um möglichst nahe an den Begrifflichkeiten der Quellen zu bleiben, wird jedoch im nächsten Abschnitt nochmals der Begriff Wissensarbeit aufgegriffen. Dort soll näher erläutert werden, was darunter verstanden wird und es werden einige Abgrenzungskriterien zu anderen Arbeitsformen herangezogen, um den Begriff genauer zu beschreiben.

Schultze beleuchtet Wissensarbeit aus Sicht der Ökonomie, des Arbeitsprozesses und der Arbeitspraktik (Schultze 2003). Aus ökonomischer Perspektive betont sie v.a. die Versuche zur Einteilung von Arbeit in unterschiedliche Typen oder die Unterteilung von Berufsgruppen in solche, die mit Wissensarbeit beschäftigt sind und andere, die es nicht sind. Dabei wird v.a. die Dienstleistungsbranche hervorgehoben und betont, dass Dienstleistungen den größten Teil der Arbeitsformen ausmachen. Eine generelle Zuordnung von Dienstleistungen zu Wissensarbeit greift jedoch zu kurz, da auch sehr

strukturierte und repetitive Tätigkeiten wie Lieferdienste jeglicher Art zu den Dienstleistungen zählen. Eine auf die OECD zurückgehende Unterscheidung differenziert zwischen Informationsarbeit und anderer Arbeit und unterteilt erstere weiter in Wissensarbeit, Datenarbeit und Managementarbeit (Schultze 2003, S. 45). Wissensarbeit ist dabei der Teil, der neues Wissen produziert oder Expertise erfordert, die nicht einfach transferierbar ist. Als Beispiele für Berufsgruppen, die Wissensarbeit ausüben werden Forscher, Ingenieure und Informatiker genannt. Insgesamt ist Wissensarbeit aus ökonomischer Sicht hauptsächlich dadurch gekennzeichnet, dass es abstraktes, theoretisches Wissen voraussetzt und dass durch Wissensarbeit auch wieder neues Wissen entsteht (ibid., S. 46).

Aus Arbeitsprozessperspektive zeichnen sich Wissensarbeiter durch systematische Kenntnis wissenschaftlicher Theorie, lange formelle Ausbildung, Autonomie, ethische Regeln, Kundenorientierung und gesellschaftliche Anerkennung aus (ibid. S. 47). Aus Arbeitspraktikperspektive sind Wissensarbeiter diejenigen, die überwiegend wissensintensive Tätigkeiten verrichten. Sie müssen Wissen produzieren und reproduzieren, interpretieren es und sind als Berater oder Experte in wertsteigernde Wissenstransformationsprozesse involviert (ibid. S. 49). Schultze stellt dabei besonders auf die Verrichtung der von ihr identifizierten „informing practices“ ab, nämlich Ausdrücken (expressing), Beobachten (monitoring), Übertragen (translating) und Vernetzen (networking, siehe dazu auch S. 21).

Schon 1997 hat die Delphi Consulting Group für Informationsarbeit einen Anteil von 60% an der Erwirtschaftung des Bruttosozialprodukts der Vereinigten Staaten geschätzt und ein ebenso großer Anteil an US-Organisationen schätzt den Anteil von Wissensarbeitern in seiner Belegschaft auf 60-100% (Delphi 1997, S. 10).

Maier et al. listen ebenfalls eine Reihe von Charakteristika auf, die Wissensarbeit auszeichnet (Maier et al. 2005, S. 25). Wissensarbeit ...

- ... löst schlecht strukturierbare Probleme in komplexen Domänen mit einem hohen Grad an Variabilität und Ausnahmen.
- ... ist kreativ, erfordert die Generierung, Beschaffung, Anwendung und Verteilung von Wissen und basiert auf der Verarbeitung von Daten und Informationen.
- ... besteht aus typischen Arbeitspraktiken, wie z.B. die von Schulze identifizierten Ausdrücken, Beobachten, Übertragen und Vernetzen.

- ... benötigt primär intellektuelle Fähigkeiten und spezialisiertes Wissen statt physischen Fähigkeiten. Sie erfordert ein hohes Ausbildungsniveau, Training und Erfahrung.
- ... ist oft dezentral organisiert und verwendet neue organisatorische Formen wie Communities, erfordert ein hohes Maß an Kommunikation, Koordination und Kooperation und ist häufig mobil, flexibel und verteilt.
- ... erfordert hochwertige, flexible und personalisierbare Unterstützung durch IKT.

Sveiby sieht Wissensarbeiter als gut ausgebildete und hochqualifizierte Spezialisten (Sveiby 1997, S. 19). Schindler bezeichnet Mitarbeiter, die mit Wissensarbeit beschäftigt sind, als Experten und definiert diese als eine neu entstehende Kategorie von Mitarbeitern die hochqualifiziert und -motiviert sind, sich weitgehend selbst führen, bzw. Teil eines sich selbst führenden Teams sind. Sie werden für wissensintensive Tätigkeiten eingesetzt und agieren als Träger von (Kern-)Kompetenzen und Kundenbeziehungen (Schindler 2001, S. 35). Als Beispiele führt Schindler Ingenieure, Wissenschaftler, Berater und Architekten an, da sie Produkte oder Dienstleistungen entwerfen oder neues Wissen für die Organisation generieren.

Wie schon oben diskutiert ist diese Aussage kritisch zu bewerten, da sie dazu verleitet bestimmte Gruppen von Mitarbeitern als „elitär“ und damit im Umkehrschluss andere Mitarbeiter als weniger wertvoll einzustufen. Davon abgesehen stimmt der Autor jedoch mit Schindler in den attribuierten Eigenschaften weitgehend überein.

Nach Efimova können Wissensarbeiter als Investoren verstanden werden, die entscheiden, wann und wie viel ihres Wissens und ihrer Energie sie in eine Firma investieren (Efimova 2004, S. 3). Daraus folgt, dass Wissensarbeit als System von Aktivitäten verstanden werden muss, zu deren Durchführung Wissensarbeiter sich freiwillig entscheiden und das Management von Wissensarbeit muss die richtigen organisatorischen Bedingungen herstellen, um die Wahrscheinlichkeit von aus Organisationssicht erwünschten Entscheidungen zu erhöhen (ibid.). Die Wahrscheinlichkeit, dass Mitarbeiter sich für Wissensarbeit entscheiden, ist hoch, wenn sie die Fähigkeit, die Motivation und die Gelegenheit dazu haben (Kelloway, Barling 2000, S. 287). Erickson und Kellogg wiederum betonen die sozialen Aspekte der Wissensarbeit. Sie schlagen vor, dass Wissensarbeit nicht nur als Herausforderung für die Informationsversorgung zu betrachten ist, sondern auch als soziale Herausforderung, bei dem persönliche Interessen und soziale Beziehungen eine wichtige Rolle spielen (Erickson, Kellogg 2003, S. 300ff).

### 2.3.6 Softwareentwicklung als Anwendungsdomäne

Da Wissensweitergabe ein komplexes Phänomen ist und daher zu dessen Analyse eine Komplexitätsreduktion notwendig ist, bietet es sich bei der Untersuchung die Konzentration auf eine Anwendungsdomäne an. Das hilft nicht nur, die Komplexität zu begrenzen, sondern auch konkretere Beispiele zu finden und dadurch die Argumentation zu veranschaulichen. Zur Einordnung der Arbeit in der existierenden Literatur zur Wissensweitergabe werden im Folgenden einige Kriterien zur Klassifikation vorgestellt, bevor im Anschluss die Anwendungsdomäne genauer eingegrenzt wird.

*Inter-organisationale und intra-organisationale Weitergabe:* (Becker, Knudsen 2003 ) haben die Unterschiede zwischen diesen beiden Szenarien detailliert untersucht und kommen zu dem Schluss, dass sie zwar verschieden sind, aber nur in Bezug auf die Mechanismen der Wissensweitergabe substantielle Unterschiede festgestellt werden konnten (ibid. S. 14). Sie schlagen vor, dass beide Strömungen der Literatur voneinander lernen sollten.

*Analyseebene:* Innerhalb der Literatur zu intra-organisationaler Wissensweitergabe sind weiterhin einzelne Analyseebenen zu unterscheiden. Einige Studien konzentrieren sich auf Individuen (Boer, Berends 2003), andere analysieren Wissensweitergabe zwischen Gruppen (Cummings 2001), wieder andere untersuchen Wissensweitergabe zwischen Geschäftseinheiten (Tsai 2002). Wird die Wissensweitergabe zwischen Niederlassungen multinationaler Konzerne untersucht, so ist die Grenze zwischen intra- und inter-organisationaler Wissensweitergabe fließend.

*Weitergaberrichtung (horizontal vs. vertikal):* (Becker, Knudsen 2003 ) stellen fest, dass bisher hauptsächlich horizontale Wissensweitergabe zwischen Mitarbeitern auf derselben Hierarchieebene, oder vertikale Wissensweitergabe zwischen einer Organisation und deren Kunden oder Lieferanten untersucht wurde. Wissensweitergabe zwischen Wettbewerbern oder zwischen Mitarbeitern und deren Vorgesetzten wurde bisher aber weitgehend vernachlässigt (ibid. S. 5).

*Arbeitskontext:* Die meisten Arbeiten untersuchen Wissensweitergabe im Kontext wissensintensiver Tätigkeiten, z.B. zwischen Wissenschaftlern (Ensign, Hébert 2004), Forschern (Cummings, Teng 2003), Lehrern (Wah et al. 2005), Beratern (Cross et al. 2000) und Entwicklern (Sonne, Harmsen 2005). Einige andere Arbeiten konzentrieren sich auf den Kontext der Managementarbeit (Cross et al. 2000) oder berücksichtigen sogar Mitarbeiter, die hauptsächlich körperlicher Arbeit nachgehen (z.B. Fließbandarbeiter

Dervisoglou, Berberb 2004). Für eine Diskussion der Unterschiede zwischen Wissensarbeit und Managementarbeit wird auf (Maier 2004, S. 44f) verwiesen.

*Forschungsmethode:* Die Literatur zur Wissensweitergabe kann generell in konzeptuelle Arbeiten basierend auf einer Literaturstudie einerseits und empirische Arbeiten andererseits unterteilt werden. Erstere können weiter unterschieden werden in breit angelegte Literaturstudien vieler Einflussfaktoren (z.B. Goh 2002) und tief greifende Analysen eines einzelnen Faktors (z.B. Chowdhury 2002). Ähnlich können bei empirischen Arbeiten empirisch-induktive, die vorher formulierte Hypothesen über Fragebögen überprüfen (z.B. Cummings, Teng 2003) von empirisch-deskriptiven unterschieden werden, die Einflussfaktoren in Interviews sammeln, ohne vorher Hypothesen aufzustellen (z.B. Cross et al. 2000).

Die vorliegende Arbeit betrachtet Wissensweitergabeaktivitäten in einer einzigen, mittelständischen nationalen Organisation, die Individualsoftware entwickelt. Dieser Fokus wurde aufgrund der Arbeitserfahrung des Autors in der Entwicklung unternehmensspezifischer Softwarelösungen, der weiter unten noch diskutierten Wissensintensität dieser Branche und dem Zugang zu einem Unternehmen gewählt, in dem eine empirische Studie durchgeführt werden konnte. Organisationen mittlerer Größe eignen sich, weil eine gewisse Größe notwendig ist, damit nicht jeder Mitarbeiter jeden anderen gut kennt und damit die Wissensweitergabe erheblich vereinfacht wird und andererseits die Größe trotzdem überschaubar und damit für das Simulationsvorhaben berechenbar bleibt.

Die Analyseebene ist der individuelle Softwareentwickler. Die Wissensweitergabe wird primär zwischen Entwicklern auf gleicher Hierarchieebene untersucht, da angenommen werden kann, dass diese für das untersuchte Wissen die primäre Quelle darstellen. Mitarbeiter auf höheren Hierarchieebenen können jedoch als Quelle für Metawissen dienen und werden aus diesem Grunde ebenfalls betrachtet. Die Arbeit unternimmt zuerst eine breit angelegte Literaturstudie, die auch versucht, eine angemessene Detailtiefe innerhalb der einzelnen Bereiche zu erreichen. Anschließend wird der Simulationsansatz als Mittel zum Erkenntnisgewinn genutzt.

Softwareentwicklung ist eine wissensintensive Tätigkeit (Henninger 1997, S. 24; Robillard 1999, S. 87). Das schlägt sich auch in hohen Misserfolgsquoten nieder. So werden nach einer Studie der Standish Group über 30% der Projekte vor Fertigstellung eingestellt und in mehr als 50% der Projekte die geplanten Kosten um durchschnittlich 80% überschritten (Dingsøyr, Conradi 2002, S. 392). Um diese Situation zu verbessern werden regelmäßig

Neuerungen im Softwareentwicklungsprozess und Softwarewerkzeuge zur Unterstützung der Softwareentwicklung eingeführt sowie neue Programmiersprachen entwickelt (ibid., S. 393). Die Entwickler selbst sind also einem permanenten Veränderungs- und Verbesserungsdruck ausgesetzt, der zur Notwendigkeit der ständigen Weiterbildung führt.

Betrachtet man die oben dargestellten Charakteristika von wissensintensiven Tätigkeiten, so stellt man fest, dass viele davon auf Softwareentwicklung zutreffen. Es werden schlecht strukturierbare Probleme in komplexen Domänen mit einem hohen Grad an Variabilität und Ausnahmen gelöst. Es erfordert Kreativität ein Geschäftsproblem mit Hilfe eines neu entwickelten Softwaresystems zu lösen. Arbeitspraktiken wie Beobachten neuer Entwicklungen im Softwareentwicklungsbereich oder Übertragen von im Internet veröffentlichten Problemlösungen auf das eigene Programmierproblem sind weit verbreitet. Softwareentwicklung erfordert ausschließlich intellektuelle Fähigkeiten und nur mit kontinuierlicher Weiterbildung kann eine dauerhaft hohe Produktivität aufrechterhalten werden.

Edwards sammelt einige Argumente, warum Softwareentwicklung als Wissensarbeit zu betrachten ist, und was die besonderen Herausforderungen dieser Art der Tätigkeit sind (Edwards 2003, S. 11ff). So zeichnet sich Softwareentwicklung durch hohe Autonomie aus und benötigt eine Mischung aus breit gefächerten Fertigkeiten (z.B. Datenbankdesign) und sehr spezifischen Fertigkeiten (z.B. Performance Optimierung mit Oracle 8i auf Sun Solaris). Softwareentwickler sind ständigen Veränderungen ausgesetzt und müssen im Fehlerfall schnell auf Probleme reagieren (ibid. S. 15).

Krishnan vermutet, dass drei Größen, die sich dem Wissen der Softwareentwickler zuordnen lassen, auf Kosten und Qualität von Software wirken (siehe Abbildung 2-6).

Davon haben sich aber in der Empirie nur die Fähigkeiten des Teams und die Erfahrung in der Anwendungsdomäne bestätigt (Krishnan 1998, S. 28). Die Erfahrung mit der Programmiersprache konnte nicht als relevanter Einflussfaktor bestätigt werden, was nach Krishnan allerdings damit zusammenhängen könnte, dass von Kunden berichtete Problem meist unabhängig von der verwendeten Programmiersprache sind.

In kleinen und mittelständischen Unternehmen ist Softwareentwicklung oft in Projekten organisiert, die Mitarbeiter sind meist jung und gut ausgebildet und eine offene, motivierende Atmosphäre ist typisch (Kautz, Thaysen 2001, S. 352f).

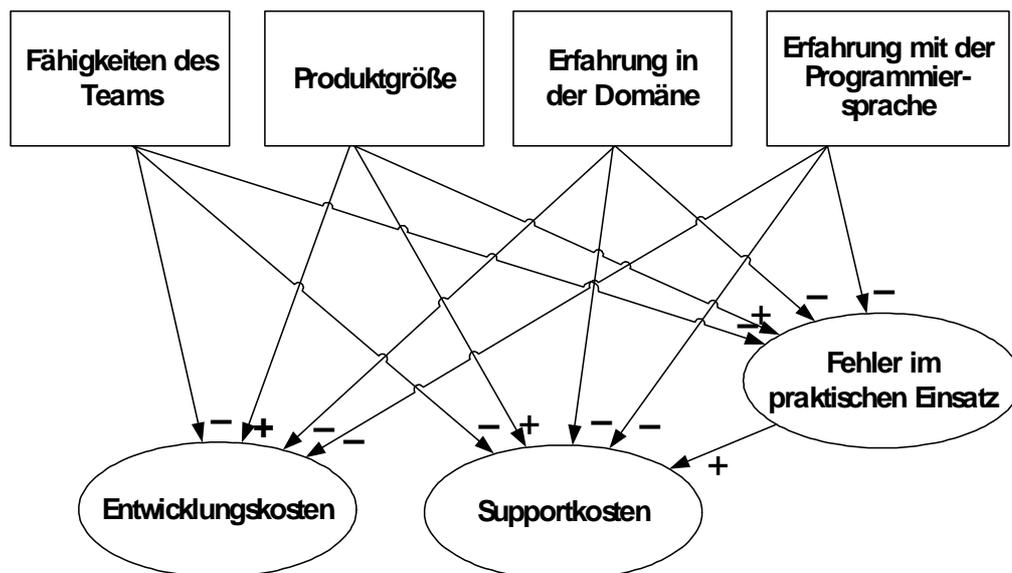


Abbildung 2-6: Einflussfaktoren auf Kosten und Qualität von Software (Krishnan 1998)

Bei einem Softwarehersteller, der Individualsoftware entwickelt, ist neben der eigentlichen Entwicklung auch der Vertrieb wissensintensiv, da Kundenbedürfnisse verstanden und technisch mögliche, sowie ökonomisch effiziente Lösungen vorgeschlagen werden müssen. Auch das Schreiben von Handbüchern oder das Schulen von Anwendern muss als wissensintensive Tätigkeit angesehen werden, da nicht nur Wissen über die Software vorhanden, sondern dieses auch so aufbereitet werden muss, dass es mit dem Vorwissen der Zielgruppe anschlussfähig ist (Arbeitspraktik: Übertragen, siehe S. 21, vgl. auch Reinhardt 2002). Die Anzahl weniger wissensintensiver Tätigkeiten wie Produktion der Datenträger, Auftragsverwaltung etc. nimmt hingegen nur einen kleinen Teil der Belegschaft in Anspruch.

In der Softwareentwicklung besteht kontinuierlicher Druck zum Senken der Entwicklungszeit und –kosten, sowie zum Erhöhen der Software- und Entscheidungsqualität (Lindvall, Rus 2003, S. 74). Weiterhin muss Wissen über neue Technologien, die Problemdomäne sowie die lokale Gesetzgebung und Richtlinien gesammelt werden. Innerhalb der Organisation besteht dadurch die Notwendigkeit zum Finden von Wissensquellen und zur Wissensweitergabe in verteilten Teams (ibid. S. 75f). Das Wissen im Softwareentwicklungsbereich entwickelt sich schnell weiter. Neue Technologien ermöglichen effizientere Entwicklung und funktionalere Produkte. Dies alles muss als Hinweis darauf gewertet werden, dass Softwareentwicklung eine wissensintensive Tätigkeit darstellt.

(Müller 2002b, S. 191f) stellt fest, dass das Können einzelner Entwickler trotz aller Versuche einer ingenieurshandwerklichen Herangehensweise immer noch einen bestimmenden Faktor für den Projekterfolg darstellt. Das Wissen gut ausgebildeter und erfahrener Mitarbeiter nimmt einen hohen Stellenwert in der Softwareentwicklung ein und muss bewahrt, entwickelt und gezielt eingesetzt werden, um den Kundenanforderungen gerecht zu werden. Dabei unterscheidet sich jedes Projekt von den vorhergehenden. Immer neue Kundenanforderungen, steigender Wettbewerbsdruck und eine rasante Entwicklung der Technologien führen zu einem hohen Lernbedarf.

Weiter merkt Müller an, dass die Entwicklung von Software in einigen Merkmalen von der Produktion physischer Güter abweicht (Müller 2002b, S. 193). Da Software entwickelt statt produziert wird lassen sich die meisten Arbeitsschritte nur begrenzt automatisieren. Sie sind in hohem Maße vom Wissen und den Fähigkeiten der Mitarbeiter abhängig. Jedes Softwareentwicklungsprojekt ist in einen eigenen Kontext eingebettet, so dass jedes Mal neue Erfahrungen gemacht werden. Die Übertragung von Erfahrungen auf neue Projekte stellt eine immer noch nicht befriedigend gelöste Herausforderung dar. Das Aufbereiten und Dokumentieren von Erfahrungen ist zwar eine wichtige Aufgabe, wird aber vom Kunden nicht bezahlt (ibid.). Mitarbeiter lernen nicht nur bei der Projektarbeit, sondern müssen sich auch darüber hinaus über aktuelle technologische Entwicklungen informieren, um innovative Lösungen kosteneffizient entwickeln zu können.

Fallstudien in sechs dänischen Unternehmen der IT-Branche ergaben, dass Projekt-basierte Teamarbeit vorherrscht, gezielte, permanente Weiterbildung ein wichtiger Faktor ist und in allen Firmen ein leistungsorientiertes Entlohnungssystem mit bis zu 25% leistungsabhängiger Bezahlung und regelmäßigen Beurteilungsgesprächen (PRP, Performance Review Process) implementiert ist (Paauwe 2004, S. 138f).

Ähnlich wie in der industriellen Forschung berichtet wird, dass die Mitarbeiter ihre Aufgaben sehr gerne verrichten und Wissenschaft und Technik auch ihre privaten Vorlieben seien (Boer, Berends 2003, S. 6), kann man nach den Erfahrungen des Autors auch in der Softwareindustrie davon ausgehen, dass die Mitarbeiter durch hohe Technikaffinität gekennzeichnet sind und Testen neuer Hard- und Software nicht nur als Arbeit angesehen wird, sondern auch z.T. den Spieltrieb befriedigt. Technikliebe verbindet die Mitarbeiter und Neuerungen in der Industrie werden oft begeistert in den Pausen diskutiert. Lesen diverser Fachzeitschriften (z.B. c't, Computerzeitung, PC Magazin) sowie von technisch orientierten Nachrichtenseiten im Internet (z.B. heise online,

Computerwoche online, tecChannel) gehört zu den üblichen Freizeitbeschäftigungen. Dies bestätigen auch die Interviews und Befragungen des Autors (siehe Abschnitt 6.1.4)

Fragestellungen für das Wissensmanagement in solchen Organisationen sind v.a., ob die Einführung von Wissensmanagement die Qualität der produzierten Software erhöht, die Entwicklungskosten senkt oder die Arbeitssituation der Mitarbeiter verbessert (Dingsøyr, Conradi 2002, S. 401). Zusammenfassend betrachtet eignet sich die Branche der Individualsoftwareentwicklung gut zur Analyse von Wissensweitergabe, da es sich bei der Softwareentwicklung um wissensintensive Tätigkeiten handelt und in der Literatur eine Reihe von Verbesserungspotentialen identifiziert wurden.

### **2.4 Wissensmanagement-Instrumente für Softwareentwickler**

Nachdem das Untersuchungsfeld jetzt abgesteckt ist, soll im Folgenden erläutert werden, welche Instrumente und Maßnahmen zur Verfügung stehen, um den Umgang mit Wissen in Softwareunternehmen zu verbessern. Der Begriff des Wissensmanagement-Instruments (WM-Instrument) wird in der Literatur immer wieder verwendet, jedoch oft nicht ausreichend definiert oder sehr einseitig aus Sicht der Technologie, Personal- oder Organisationsentwicklung heraus betrachtet (Reuter 2004, S. 2). Eine der ersten Arbeiten, die sich mit der Sammlung von Instrumenten beschäftigt ist (Bach, Homp 1998). Sie gehen dabei von den Instrumenten des strategischen Managements aus (S. 143), ohne jedoch den Begriff WM-Instrument selbst zu definieren. Mangold fokussiert dagegen in seiner Arbeit zu WM-Instrumenten ausschließlich die Technik (Mangold 1999). Koch und Mandl zielen mit ihrer pädagogischen Sicht v.a. auf personenorientierte WM-Instrumente (Koch, Mandl 1999). Moreno et al. sind Vertreter eines multi-perspektivischen Ansatzes und stellen fest, dass neben den oft diskutierten technologischen Ansätzen auch personenorientierte berücksichtigt werden müssen. Sie definieren daher WM-Instrumente als „Hilfsmittel mit oder ohne technische Unterstützung, die Vorteile des ganzen, oder eines Teils eines WM-Prozesses verstärken“ (Moreño et al. 2004, S. 4)<sup>12</sup>. In der Arbeit von Reuter (Reuter 2004), die sich stark an Roehl anlehnt (Roehl 2000), sowie der darauf aufbauenden Diskussion in (Maier, Peinl 2005b) und (Maier et al. 2005, S. 39ff) wurde versucht noch einen Schritt weiter zu gehen und den Begriff aus einer umfassenderen Sichtweise zu definieren, obwohl ein Technologiefokus unbestreitbar ist. Die Definition lautet wie folgt:

---

<sup>12</sup> Im Original: “[...] we take into account a broader, mixed concept, by considering KM practices as aids, with technological support or not, which reinforce benefits of the whole or part of KM processes. “

„WM-Instrumente sind (1) Teil einer technologiegestützten Intervention (2) in die organisatorische Wissensbasis und bestehen aus einer (3) Sammlung von organisatorischen, personenorientierten und informationstechnischen Maßnahmen, die aufeinander abgestimmt und (4) klar definiert sind, (5) absichtsvoll eingesetzt werden können um Ziele des Wissensmanagements zu erreichen, auf die (6) Verwaltung kontextualisierter Informationen abzielen und (7) unabhängig von einer bestimmten Wissensdomäne sind“ (Maier et al. 2005, S. 41)<sup>13</sup>.

(1) Die Betonung einer technologiegestützten Intervention entsteht aus der Idee der Technologie als „Enabler für das Wissensmanagement“. (2) Die organisatorische Wissensbasis besteht aus Personen als Wissensträger und kodifiziertem Wissen in Form von Dokumenten, aber auch „inkorporiertem Wissen“ in organisatorischen Abläufen und Strukturen. All diese, obgleich nicht notwendigerweise alle gleichzeitig, sollen durch das WM-Instrument positiv beeinflusst werden. (3) Dazu sind aufeinander abgestimmte Maßnahmen erforderlich, die jeweils in den Teilgebieten Organisation, Personal und IKT angesiedelt sind. (4) Die Forderung nach klarer Abgrenzung zielt darauf ab, dass eine Definition der einzusetzenden Maßnahmen erforderlich ist, um einschätzen zu können, ob ein beobachtetes oder in der Literatur diskutiertes Phänomen als WM-Instrument einzustufen ist, oder nicht. (5) Der absichtsvolle Einsatz deutet an, dass ein Instrument strategisch geplant eingesetzt werden sollte, was die Definition von zu erreichenden Zielen und Messgrößen zur Evaluation des Erfolgs beinhaltet. (6) Um eine Abgrenzung gegenüber Daten- und Informationsmanagementwerkzeugen zu erzielen, wird der Fokus auf Wissen in Form von kontextualisierten, handlungsorientierten Informationen explizit angeführt. So kann dasselbe IKT-Werkzeug abhängig von den Inhalten als Maßnahme im Rahmen eines WM-Instruments gelten oder nicht (Maier et al. 2005, S. 41). Damit genügt z.B. eine Datenbank mit Adressdaten der Definition nicht. (7) Schließlich sollte das WM-Instrument hinreichend allgemein einsetzbar sein, um in mehreren Domänen Anwendung

---

<sup>13</sup> Im Original: KM instruments (1) are part of an ICT-supported intervention (2) into an organizational knowledge base and consist of (3) a collection of organizational, human resources and ICT measures that are aligned, (4) clearly defined, (5) can be deployed purposefully in order to achieve knowledge-related goals, (6) target contextualized information as object of intervention and (7) are independent of a particular knowledge domain.

zu finden. Damit genügen z.B. Maßnahmen des Customer Relationship Managements nicht der Definition, wenn sie ausschließlich auf Wissen über Kunden abzielen.

Analog zur Definition eines WM-Instruments ist demnach eine *WM-Maßnahme* ein Teil eines WM-Instruments, der für sich genommen nicht mit anderen abgestimmt ist, sondern isoliert eingesetzt wird. Die übrigen Aspekte gelten für die WM-Maßnahme gleichermaßen.

Eine *WM-Initiative* kann als die Umsetzung des strategischen Rahmens für die Intervention in die organisatorische Wissensbasis angesehen werden. Im Rahmen einer WM-Initiative werden ein oder mehrere WM-Instrumente in der Organisation eingeführt und miteinander koordiniert. Die Einführung erfolgt dabei im Rahmen eines Projekts. Es können auch mehrere Projekte durchgeführt werden, z.B. eines je eingeführter WM-Maßnahme, die dann einer zentralen Koordination und Abstimmung bedürfen. Die WM-Initiative muss an den in der WM-Strategie festgelegten Zielen ausgerichtet sein.

In den oben genannten Arbeiten wurde weiterhin durch eine Literaturanalyse von Fallstudien eine Reihe von Vorschlägen für WM-Instrumente gesammelt und diese auf Konformität mit der Definition geprüft (Maier, Peinl 2005b; siehe auch Maier et al. 2005, S. 39ff). Daraus hervorgegangen sind die WM-Instrumente Kompetenzmanagement, persönliches Erfahrungsmanagement, Communities und Wissensnetzwerke, Wissensprozessverbesserung, Lessons learned, Good/best practices, semantisches Content Management und Wissenskarten (siehe Abbildung 2-7).

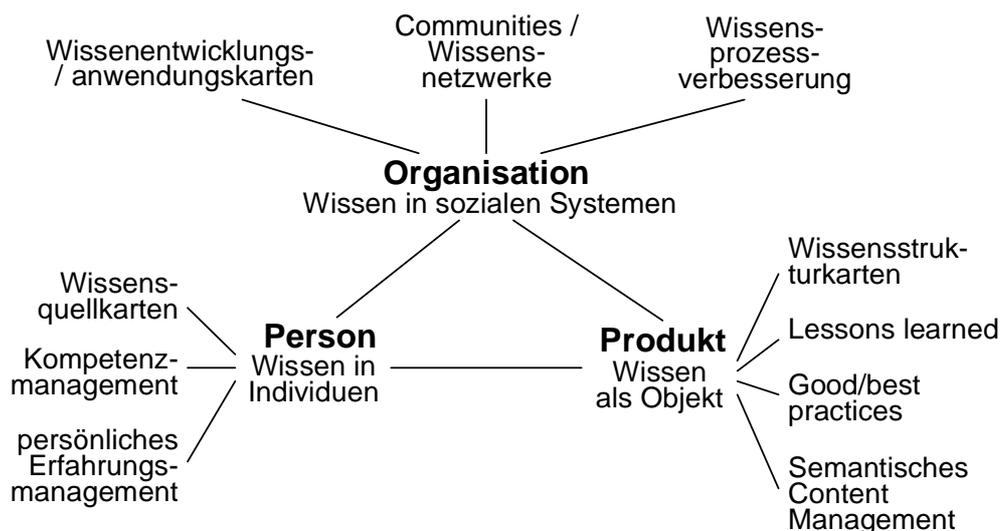


Abbildung 2-7: Wissensmanagement-Instrumente nach (Maier et al. 2005, S. 42)

Dieser Stand stellt zwar einen Fortschritt gegenüber der vorhergehenden Situation in der Literatur dar, muss jedoch als Zwischenstand angesehen werden, da sich mehrere

Probleme ergeben. Die Instrumente sind nicht als gleichwertig einzustufen, es existiert kein Kriterium zur Überprüfung der Vollständigkeit der Vorschläge und die Zuordnung auf entsprechende idealtypische WM-Maßnahmen ist bisher nicht ausreichend beschrieben.

Zum Beispiel sind Wissenskarten (siehe Eppler 2001; Eppler 2003) in den verschiedensten Ausprägungen von den Schwierigkeiten bei der Darstellung abgesehen technisch einfach aus den Daten verschiedener Systeme generierbar. Die eigentliche Aufgabe besteht also darin, die Daten in den zugrunde liegenden Systemen verfügbar zu machen und aktuell zu halten. Wissenskarten enthalten keine Daten, die für die Karten spezifisch sind und die Pflege der Karten sollte idealer Weise automatisch über die Aktualisierung der entsprechenden Basisdaten erfolgen. Sie bieten vielmehr einen alternativen Zugang zu den Wissensquellen im jeweiligen Bereich und ergänzen die Suchfunktionalität. Auch sind organisatorische und personenorientierte Maßnahmen, die Wissenskarten sinnvoll unterstützen und ergänzen, schwer zu finden. Versucht man den IT-Teil der WM-Instrumente auf die idealtypische Architektur einer unternehmensweiten Wissensinfrastruktur (Enterprise Knowledge Infrastructure, EKI) abzubilden, so fallen Wissenskarten nicht in die Ebene der Wissensdienste sondern sind vielmehr in der Ebene Zugriff/Personalisierung anzusiedeln. Auch die Ansiedelung der Wissenskarten in jeder Kategorie der in Abbildung 2-7 dargestellten Klassifikation kann als Hinweis gewertet werden, dass Wissenskarten nicht zu den anderen Instrumenten passen.

#### **2.4.1 WM-Instrumente in der Literatur**

Ein Vollständigkeitskriterium für WM-Instrumente ist schwer zu finden und es wird auch nicht als Aufgabe dieser Arbeit angesehen, eine abschließende Liste aller WM-Instrumente aufzustellen. Im Folgenden soll jedoch versucht werden über eine Systematik einige Lücken zu schließen und die Beschreibung der diskutierten Instrumente zu verbessern.

Zur Unterscheidung von WM-Instrumenten ist es erforderlich ein Kriterium zu finden, nach dem sich beurteilen lässt, ob ein Kandidat wirklich ein eigenständiges Instrument darstellt oder als Maßnahme eines umfassenderen Instruments zu bewerten ist. Dazu wird hier jeder Kandidat anhand der Dimensionen Ziel (Strategie), Inhalt (Wissenstyp), Person (Ersteller), Organisation (Prozesse) und System beschrieben. Stimmen zwei Kandidaten in mindestens drei der Dimensionen überein, so werden sie als Teile eines umfassenderen Instruments beurteilt, andernfalls sollen sie als eigenständiges Instrument gelten. Dies stellt jedoch nur einen pragmatischen Vorschlag dar, der zu Diskussion gestellt, aber noch nicht validiert worden ist.

Das Ziel kann in den Ausprägungen Personalisierung oder Kodifizierung (Hansen et al. 1999) in Kombination mit Exploration oder Exploitation (Zack 1999) definiert werden. Die Inhaltsdimension kann die Ausprägungen prozedurales Wissen, Faktenwissen und Meta-Wissen annehmen (Anderson 1976, S. 114ff). Interessant für den Inhalt ist aber auch die Unterscheidung zwischen organisations-externem und -internem Wissen, sowie ob das Wissen aus strukturierten oder semi-strukturierten Daten gewonnen wird (vgl. die Daten- und Wissensquellen der EKI). Auf der Personendimension wird zwischen Individuen, Gruppen bzw. Teams und Team-übergreifenden Zielgruppen des WM-Instruments unterschieden. Die Organisationsdimension kann die Ausprägungen der Wissensprozesse nach Schultze annehmen, also Ausdrücken, Beobachten, Übertragen und Vernetzen.

*Tabelle 2-2: Übersicht der WM-Instrumente*

	<b>Ziel</b>	<b>Inhalt</b>	<b>Person</b>	<b>Organisation</b>
<b>Kompetenzmanagement</b>	Personalisierung Exploitation	Meta-Wissen, intern	Individuum	vernetzen Sozialisierung
<b>Persönliches Erfahrungsmanagement</b>	Kodifizierung Exploration	prozedural, intern, semi-s.	Individuum	ausdrücken Externalisierung
<b>Communities, Wissensnetzwerke</b>	Personalisierung Exploration	prozedural, intern, semi-s.	Team- übergreifend	vernetzen Sozialisierung
<b>Wissensbasierte Geschäftsprozessverbesserung</b>	Kodifizierung Exploitation	prozedural, intern, semi-s.	Team	übertragen Kombination
<b>Good / Best Practices</b>	Kodifizierung Exploitation	prozedural, intern, semi-s.	Team	ausdrücken Externalisierung
<b>Lessons Learned</b>	Kodifizierung Exploitation	prozedural, intern, semi-s.	Team	ausdrücken Externalisierung
<b>Semantisches Content Management</b>	Kodifizierung Exploitation	Faktenwissen, intern, semi-s.	Team- übergreifend	ausdrücken Externalisierung

Obwohl die Zuordnung der einzelnen Ausprägungen nicht zweifelsfrei festgestellt werden kann, ist die Grundtendenz auch bei geringfügigen Variationen einzelner Einträge erkennbar. Es fehlen Instrumente, die Wissen aus strukturierten Daten fokussieren, auf externes Wissen abstellen und den Wissensprozess des Beobachtens unterstützen. Des

Weiteren sind die vorgeschlagenen Instrumente Lessons learned und Good/Best Practices anhand der Beschreibungen in den vorgeschlagenen Dimensionen identisch, so dass sie als verschiedene Maßnahmen eines umfassenderen Instruments betrachtet werden sollten.

#### **2.4.2 Beschreibung der identifizierten Instrumente**

Im folgenden Abschnitt sollen die übrigen von Maier et al. vorgeschlagenen WM-Instrumente diskutiert und idealtypische Maßnahmen identifiziert werden.

**Kompetenzmanagement** wird bei Maier et al. verstanden als WM-Instrument, das den systematischen Aufbau von Kompetenzen der Mitarbeiter unterstützt. Das beginnt mit der Erfassung von Fertigkeiten und Erfahrungen der Mitarbeiter und das Verfügbarmachen dieses Meta-Wissens für andere Mitarbeiter, die einen Experten für ein Problem suchen (Deiters et al. 1999; Benjamins, al. 2002; Lauf, Sure 2002).

Dies kann z.B. durch ein *Skillmanagementsystem* (SkMS) unterstützt werden. Skillmanagement ist die Verwaltung von Qualifikationen, Erfahrungen und Wissen der Mitarbeiter mit dem Ziel der kostenoptimalen Bereitstellung der benötigten Fertigkeiten zur richtigen Zeit am richtigen Ort (Beck 2003, S. 1384). Ein SkMS (auch als Skill Datenbank bezeichnet) soll diese Managementfunktion unterstützen, indem es eine Taxonomie oder Ontologie der relevanten Skills bereitstellt (Lauf, Sure 2002), die Erfassung und Aktualisierung von Skill-Profilen durch Mitarbeiter oder Vorgesetzte ermöglicht und Suchfunktionen bietet, mit denen Mitarbeiter mit bestimmten Skills gefunden werden können (vgl. Hiermann, Höfferer 2005, S. 3f). Weiterhin sollten Ziele für die Weiterentwicklung der Fertigkeiten in Einklang mit den Schwerpunkten des Unternehmens definiert und deren Erreichung in regelmäßigen *Reviews* (z.B. jährlich) überwacht werden. Geeignete *Schulungsmaßnahmen* bzw. die Teilnahme an entsprechenden Projekten sollten die Zielerreichung unterstützen. *Mentoring* kann eingesetzt werden, um Berufsanfängern schnell das nötige Wissen zu vermitteln und sie beim Lernen während eines Projektes zu unterstützen.

**Persönliches Erfahrungsmanagement:** Wissensintensive Tätigkeiten erfordert das Filtern einer großen Menge von Informationen, deren Verständnis und das Verbinden mehrerer Teile um neue Ideen zu generieren. Dies wird oft durch elektronische Artefakte unterstützt, so dass eine Art persönliches Informationsmanagement nötig ist, um all die Dokumente, Nachrichten und Lesezeichen zu organisieren (Efimova 2004, S. 13).

Eine Maßnahme in Form der elektronischen Unterstützung für das Festhalten von Ideen und Erfahrungen ist ein *Weblog* oder kurz Blog. Ein Weblog ist in erster Linie eine häufig aktualisierte Webseite die aus datierten Einträgen besteht, welche in umgekehrter chronologischer Reihenfolge angeordnet sind (Herring et al. 2004, S. 1). Technisch bestehen Blogs aus einer Art einfachem Content Management System, das eine Datenbank mit Texteinträgen verwaltet und diese durch Templates aufbereitet als HTML-Seiten zugreifbar macht (Efimova 2004, S. 4). Bemerkenswert ist auch die Fähigkeit zum Publizieren der Inhalte als RSS-Feed, wodurch das einfache Beobachten mehrerer Weblogs ermöglicht wird. Auch eine Vernetzung von Weblogs ist leicht möglich und wird als Ping bezeichnet. Häufig werden sie knapp als elektronisches Tagebuch charakterisiert, was den starken Bezug zum Ersteller gut zum Ausdruck bringt. Im Rahmen des persönlichen Erfahrungsmanagements können sie als „externer Wissensspeicher“, Lerntagebuch oder Instrument zur Vernetzung eingesetzt werden. Aus organisatorischer Sicht erscheint es dem Autor wichtig, dass entsprechende zeitliche Freiräume für das persönliche Erfahrungsmanagement geschaffen werden. Als personenorientierte Maßnahme sind Weiterbildungsmöglichkeiten zur strukturierten Informationsablage denkbar.

***Communities und Wissensnetzwerke:*** Zur Unterstützung des Austausches von implizitem Wissen werden in der Literatur Communities oder Wissensnetzwerke vorgeschlagen (Ferrán-Urdaneta 1999; Wenger 2000; Wenger 2002). Die beiden Konzepte unterscheiden sich dabei durch das Vorhandensein bzw. Fehlen eines zentralen steuernden Organs. Während Communities bottom-up durch Initiative einiger Mitarbeiter der Organisation entstehen und die Teilnahme vollkommen freiwillig ist, werden Wissensnetzwerke vom Management der Organisation eingeführt und Mitarbeiter können Aufgaben innerhalb des Wissensnetzwerks zugewiesen bekommen (vgl. Enkel 2002, S. 8f). Communities sind „Gruppen von Personen, die informell durch gemeinsame Expertise und eine Leidenschaft für ein gemeinsames Unterfangen zusammengehalten werden“ (Wenger, Snyder 2000, S. 139)<sup>14</sup>. Dagegen werden Wissensnetzwerke definiert als “eine Anzahl von Personen, Ressourcen und Beziehungen zwischen ihnen, die zusammengestellt werden, um Wissen durch Wissenserzeugungs- und -übertragungsprozesse zu sammeln und anzuwenden, mit

---

<sup>14</sup> Im Original: “[...] groups of people informally bound together by shared expertise and passion for a joint enterprise [...]”

dem Ziel Wert zu schaffen“ (Vassiliadis et al. 1999, S. 7)<sup>15</sup>. In beiden Fällen sollen Wissensträger miteinander verknüpft werden, die über die „normale“ Organisationshierarchie (z.B. Linie, Matrix) keine direkte Verbindung zueinander haben (Enkel 2002, S. 11). Im Rahmen der Communities sind hier besonders die Communities of Practice gemeint (für eine Übersicht über verschiedene Typen von Communities siehe Maier 2004, S. 160f), da es um den Austausch von Wissen über Arbeitsprozesse oder in Arbeitsprozessen angewandtes Wissen geht. Mitarbeitern, die in ähnlichen Arbeitspraktiken tätig sind, fällt Wissensweitergabe leichter, auch wenn sie sich nie persönlich kennen lernen (Bansler, Havn 2001, S. 654). Obwohl Communities selbstorganisierend und freiwillig sind, kann die Organisation dennoch Maßnahmen ergreifen, um das Entstehen und Überleben von Communities zu erleichtern. So können Plattformen für den Wissensaustausch, sog. *Foren* oder Blackboard-Systeme, aber auch Listserver für den Email-basierten Austausch angeboten werden, die von den Communities genutzt werden können. Auch *Wikis* und miteinander *vernetzte Weblogs* können als Softwareplattform Communities unterstützen. Diese drei Systemtypen sind letztlich sehr ähnlich und unterscheiden sich nur durch unterschiedliche Betonung einzelner Aspekte. Während Wikis vollständig themenzentriert sind und weder Autoren noch Zeitpunkte der Erstellung hervorgehoben werden (z.B. Leuf, Cunningham 2001) stehen bei Weblogs die Autoren im Mittelpunkt, Zeitpunkte dienen als Hauptgliederungskriterium und Themen werden bestenfalls über Kategorien geordnet (z.B. Efimova 2004). Bei Foren ist zunächst das Themengebiet das Gliederungskriterium, darunter kommen zeitlich geordnet die Beiträge zu den einzelnen Threads, themenbezogenen Diskussionen. Die Autorenschaft wird i. d. R. direkt ersichtlich (vgl. Waltert 2002). Aus organisatorischer Sicht kann die Definition einer *Community-Manager*-Rolle helfen, wertvolle Diskussionen in Gang zu halten und die Aufmerksamkeit des Managements auf viel diskutierte Themen zu lenken (Maier et al. 2005, S. 45; vgl. auch Maier, Remus 2002, S. 116). Als personenorientierte Maßnahme können *Treffen für besonders aktive oder erfolgreiche Communities* organisiert werden, bei denen Community-Teilnehmer die Gelegenheit erhalten sich persönlich kennen zu lernen und Themen direkt zu diskutieren. So eine Maßnahme kann nicht nur motivieren, weiter aktiv

---

<sup>15</sup> Im Original: “We use the term ”Knowledge Networking” to signify a number of people, resources and relationships among them, who are assembled in order to accumulate and use knowledge primarily by means of knowledge creation and transfer processes, for the purpose of creating value.”

an der Community teilzunehmen, sondern hilft auch besseres Verständnis zu erzeugen, da Face-to-Face Gespräche nicht vollständig durch elektronische Kommunikation ersetzt werden können.

**Wissensbasierte Prozessverbesserung:** Wissensbasierte Prozessverbesserung als Instrument des WM stellt in erster Linie darauf ab, die Ergebnisse anderer Instrumente wieder in den Prozess einfließen zu lassen. So können die Erkenntnisse aus dem Erheben von Lessons learned und Best practices ebenso wie das gewonnene Wissen aus der Analyse der Geschäfts- und Produktionsdaten zu Verbesserungszwecken eingesetzt werden. Ein weiterer Aspekt ist die Tatsache, dass die Prozessabläufe als eine Form kodifizierten Wissens in keinem anderen Instrument systematisch genug betrachtet werden. Als technische Maßnahme ist in erster Linie die Einführung eines *Systems zur Geschäftsprozessmodellierung* zu nennen. In diesem Bereich gibt es zwar etablierte Werkzeuge (z.B. ARIS, CA Process Modeler), aber eine nahtlose Integration in das Intranet ist selten vorgesehen (erste Schritte in diese Richtung geht z.B. der ARIS Web Publisher). Erst die Einigung auf herstellerunabhängige XML-basierte Austauschformate (z.B. XPDL, EPML, siehe Mendling et al. 2004), der Vorschlag für eine einheitliche Darstellungsform (BPMN, Business Process Modeling Notation) und die Unterstützung von XML-basierten Vektorgrafiken (SVG, Scalable Vector Graphics) in den verbreiteten Browsern (Internet Explorer 6 mit Adobe SVG Viewer, Firefox ab 1.5 nativ, Opera 8 nativ) ermöglichen eine standardisierte und effiziente Integration ins Intranet. Der Begriff *Process Warehouse* wird vorgeschlagen für ein zentrales Repository, das nicht nur die Modelle der Geschäftsprozesse enthält, sondern auch dokumentierte Ausnahmefälle, fallbasierte Erfahrungen, Gründe für Entscheidungen, Check-Listen, Tipps, häufig gestellte Fragen und deren Antworten, potentielle Kooperationspartner und Verbesserungsvorschläge (Maier et al. 2005, S. 45). Als organisatorische Maßnahmen sind in erster Linie das regelmäßige Anberaumen von *Treffen für die Diskussion von Verbesserungen* zu sehen, sowie ein dokumentiertes Vorgehen, wie Verbesserungen eingeführt werden. Als personenorientierte Maßnahme kann das *Sensibilisieren der Mitarbeiter* gewertet werden, Prozesse entsprechend den dokumentierten Abläufen und Best practices durchzuführen und Verbesserungsvorschläge zu kommunizieren und zu dokumentieren.

**Erfahrungsmanagement im Team:** Das systematische Erfassen, Auswerten und Anwenden von Erfahrungen aus Projekten oder Prozessdurchläufen wird als wesentlicher Bestandteil von Wissensmanagement betrachtet. Besonders in der Softwareentwicklung

wird unter dem Begriff Erfahrungsfabrik (experience factory) eine technische und soziale WM-Infrastruktur verstanden um Erfahrungen aus dem Software-Lebenszyklus, Prozessen und Produkten wieder zu verwenden (Dingsøyr, Conradi 2002, S. 398). In anderen Quellen wird auch häufig von Lessons learned gesprochen, also Lernerfahrungen (Neumann et al. 2002). Diese werden im Team diskutiert und festgehalten und erhalten somit eine gewisse organisatorische Verbindlichkeit und Legitimation (Maier et al. 2005, S. 42f). Durch diese beiden Aspekte unterscheiden sie sich von Erfahrungen eines Einzelnen die im Rahmen des persönlichen Erfahrungsmanagements verwaltet werden. Durch kontinuierliche Anwendung von Erfahrungen im Prozess und die Diskussion und Anerkennung auf höherer organisatorischer Ebene werden aus Lessons learned Good oder Best practices entwickelt. Der Erstellungsprozess kann zentral oder dezentral organisiert sein, wobei ein dezentraler Ansatz die Gefahr eines organisatorischen Engpasses senkt (Bansler, Havn 2001, S. 658). Um Best practices erfolgreich anzuwenden muss auch das Umfeld geeignet sein. Zeitdruck, fehlende Anreize und v.a. das Fehlen einer verbindenden Community of Practice zwischen Best Practice Erstellern und Anwendern werden als Hauptfaktoren für das Scheitern genannt (ibid. 660f). Beide Formen von Erfahrungen können in einem Informationssystem gespeichert werden, in dem dann Anwender fallbasiert Erfahrungen mit ähnlichen Situationen abrufen können (Henninger 1997, S. 25). Dies stellt eine wertvolle Ergänzung zum Zugriff auf allgemeines Faktenwissen dar, da sich alle Fälle auf organisationsinterne Vorgehensweisen beziehen (ibid.).

**Semantisches Content Management:** Die beiden Softwareklassen Content Management und Dokumentenmanagement wachsen unter dem Namen Enterprise Content Management (ECM) zusammen (Laugero, Globe 2002). Da im Wissensmanagement nicht nur Informationen sondern auch kodifiziertes Wissen (also handlungsorientierte, kontextualisierte Informationen) fokussiert werden, ist eine weitere Komponente nötig, um wissensintensive Tätigkeiten effektiv zu unterstützen (Asprey, Middleton 2003). Ontologien können helfen die Zusammenhänge zwischen Inhalten abzubilden, und somit Kontextinformationen für die Interpretation der Inhalte (Staab et al. 2003). Es wird daher vorgeschlagen ein solchermaßen um Kontextinformationen angereichertes ECM System als semantisches Content Management System zu bezeichnen. Neuere Ansätze gehen weg von den streng formalisierten und aufwändig zu erstellenden Ontologien und bevorzugen organisch gewachsene, dezentral aktualisierte sog. „Folksonomies“, die durch Tagging und Social Bookmarking entstehen (z.B. Golder, Huberman 2006; Xian et al. 2006). Aus

organisatorischer Sicht müssen Regelungen getroffen werden, wann welche Inhalte im semantischen CMS abgelegt werden sollen. *Themenverantwortliche* können helfen, neu eingestellte Beiträge zu sichten und evtl. um zusätzliche Metadaten anzureichern. Als personenorientierte Maßnahme können *Anreizsysteme* angeführt werden, die z.B. Punkte für Beiträge vergeben und im Intranet an prominenter Stelle die Mitarbeiter mit den meisten Punkten vorstellen, oder einmal jährlich die Punkte in Form der Finanzierung von Konferenzbesuchen oder Trainings ausbezahlen (vgl. Trittman, Brössler 2000; Lindvall, Rus 2003).

**Wissenskarten:** Eine Wissenskarte ist eine grafische Darstellung von Informationen und Beziehungen, welche die effiziente Kommunikation und das Erlernen des Wissens auf verschiedenen Detaillierungsstufen durch Betrachter mit verschiedenem Hintergrund ermöglicht<sup>16</sup>. Die in der Karte dargestellten Wissensbausteine können Text, Geschichten, Grafiken, Modelle oder Zahlen sein (Eppler 2001, S. 3). Dabei können nach den abgebildeten Inhalten verschiedene Typen von Wissenskarten unterschieden werden (ibid.). Wissensquellkarten (Wissensträgerkarten, Nohr 2000) stellen Experten und deren Wissen nach Wissensgebiet oder anderen Ordnungskriterien sortiert dar. Wissensvermögenskarten stellen eine vereinfachte Sicht auf das Wissenskapital eines Teams, einer Abteilung oder einer Organisation dar. In Wissensstrukturkarten werden die Beziehungen einzelner Konzepte einer Wissensdomäne dargestellt, so dass die Struktur explizit wird. Sie können als Visualisierung einer Taxonomie oder Ontologie betrachtet werden. Wissensanwendungskarten zeigen welches Wissen in welchen Prozessschritten oder Geschäftssituationen benötigt wird. Oft werden dazu auch Verweise auf Experten, Dokumente oder Datenbanken abgelegt, die das benötigte Wissen enthalten. Wissensentwicklungskarten werden benutzt um das Vorgehen zum schrittweisen Erwerb von Wissen einer Domäne abzubilden. Als Visualisierungstechnik können einfache grafische Mittel wie Mind maps oder komplexere wie 3D-Landschaften verwendet werden (Eppler 2001, S. 3). Sie helfen ein geteiltes Verständnis einer Domäne herzustellen oder die Suche nach Wissen unterstützen (Wexler 2001, S. 251). Mit ihrer Hilfe können auch Wissenslücken aufgedeckt werden (Nohr 2000). Wichtig für den Erfolg einer Wissenskarte

---

<sup>16</sup> Im Original: „A knowledge map is a visual display of captured information and relationships, which enables the efficient communication and learning of knowledge by observers with differing backgrounds at multiple levels of detail. The individual items of knowledge included in such a map can be text, stories, graphics, models, or numbers.“

ist die Abstimmung der Visualisierung auf den Wissenstyp und das Ziel das mit der Karte erreicht werden soll (Eppler, Burkhard 2005). Aus technischer Sicht sind Wissenskarten v.a. durch Visualisierungsalgorithmen bestimmt. Organisatorisch können neben den Erstellern der Karten auch die Rolle des Champions für die organisatorische Legitimation und Innovatoren unter den Anwendern für die Weiterentwicklung sorgen (Wexler 2001, S. 254).

### **2.4.3 Eigene Vorschläge**

Blackler unterscheidet Organisationen anhand des hauptsächlich benötigten Wissens in Experten-abhängige (expert dependent), Analytischen-abhängige (symbolic-analyst dependent), Kommunikations-intensive (communication-intensive) und solche die in erster Linie routinisiertes Wissen anwenden (knowledge-routinized) (Blackler 1995). Bei der Analyse dieser Organisationstypen nach ihrer Unterstützbarkeit durch WM-Instrumente fällt auf, dass für Analytischen-abhängige Organisationen keines der oben aufgeführten WM-Instrumente die nötige Unterstützung bereitstellt.

*Knowledge Discovery in Databases:* Hier ist ein WM-Instrument nötig, das die Auswertung strukturierter Daten unterstützt, aber auch die Verbindung zwischen diesen Daten und semi-strukturierten Dokumenten herstellt und bei der Extraktion von strukturierten Daten aus semi-strukturierten Dokumenten (z.B. HTML-Seiten mit Marktübersichten, PDF-Dokumente mit Finanzberichten) hilft. Unter den Stichwörtern Business Intelligence (z.B. Grothe, Gentsch 2000; Hannig 2002), Knowledge Warehouse (Nedeß C., Jacob U. 2000; Herschel, Jones 2005) und Text Mining (Nahm, Mooney 2002) bzw. Information Extraction (Baumgartner et al. 2001) werden IT-Werkzeuge für die Datenanalyse, die Verbindung zwischen strukturierten und semi-strukturierten Daten und die Extraktion von Daten aus Dokumenten diskutiert. Eine Diskussion der Sinnhaftigkeit einer Verbindung von strukturierten und semi-strukturierten Daten findet sich auch in (Rieger et al. 2000; Haak 2002). Für spezialisierte Bereiche innerhalb des gesamten Spektrums strukturierter Unternehmensdaten gibt es bereits Integrationsvorschläge, z.B. für Kundendaten im Rahmen des Customer Relationship Managements (CRM, Bueren et al. 2004). Moreño et al. schlagen Entscheidungsunterstützungssysteme basierend auf Data Warehousing, OLAP und Data Mining als eigenständiges WM-Instrument vor (Moreño et al. 2004, S. 4). In dieser Arbeit soll der Begriff Wissensentdeckung in Datenbanken (KDD, Knowledge Discovery in Databases) verwendet werden, da er in der Literatur zwar weitgehend synonym zu Business Intelligence verwendet wird (Cios, Kurgan 2005, S. 1),

aber eigentlich der Umfassendere ist, die Wissensorientierung stärker betont und damit der Ausrichtung des Instruments besser gerecht wird.

Ein Vergleich mit den Wissensdiensten einer EKI (siehe Abbildung 2-5) zeigt, dass auch hier die explizite Berücksichtigung von Informationen fehlt, die aus strukturierten Daten gewonnenen wurden. Bei den Daten- und Wissensquellen werden strukturierte Daten noch explizit erwähnt, auf den oberen Schichten fehlt jedoch eine Komponente zu deren systematischen Behandlung. Eine Beschreibung von KDD als WM-Instrument könnte wie folgt aussehen.

Zum systematischen Analysieren der strukturierten Daten in der Organisation sollte KDD eingeführt werden. Dazu sollte mit Hilfe eines Data Warehouse eine einheitliche Datenbasis geschaffen werden, welche die Grundlage für Analysen bildet. Mit Hilfe von Data Mining und OLAP Werkzeugen werden diese Daten dann ausgewertet. Bei den Analyseergebnissen kann man von Informationen sprechen, die dann mit semi-strukturierten Informationen aus Dokumenten angereichert werden sollten, um die Interpretation für den Benutzer zu erleichtern (vgl. Haak 2002, S. 304). Dadurch wird die Forderung nach Unterstützung von kontextualisierten Informationen, statt isolierten Daten erfüllt. Zum Herstellen einer Verbindung zwischen den Analyseergebnissen einerseits und den Dokumenten andererseits bietet es sich an, eine Ontologie zu verwenden, wie sie für die Integrationsschicht einer EKI vorgeschlagen wird. Das logische Schema des Data Warehouse stellt bereits eine einfache Form einer Ontologie dar, so dass die Verbindung erleichtert wird. Bzgl. der personenorientierten Maßnahmen ist neben der obligatorischen Schulung der Nutzer auch eine team-basierte Erarbeitung der Ontologie und des DWh Schemas zu nennen. Weiterhin können Workshops veranstaltet werden, in denen Nutzer gemeinsam komplexe Analysen vornehmen, über die Interpretation der Ergebnisse diskutieren und Vorschläge für geeignete Reaktionen auf die neu gewonnenen Erkenntnisse erarbeiten. Organisatorische Maßnahmen wären das Einrichten eines Analyse-Helpdesks oder die Definition neuer Rollen für die Datenanalyse, z.B. eines „Data Warehouse Champion“, der Datenanalyse als Methode der Prozessverbesserung in der Organisation bewirbt und die Unterstützung des Managements sicher stellt (vgl. Maier et al. 2003, S. 27). Die Unabhängigkeit von einer Wissensdomäne ist ebenfalls erfüllt, da sich die Datenanalyse für Produktionsdaten ebenso eignet wie für Verkaufszahlen oder Daten über Beraterstunden. Das unterstützte Ziel für das Wissensmanagement ist in erster Linie eine Verbesserung der Entscheidungsqualität durch Bereitstellen von konsolidierten Daten,

die aggregiert und aufbereitet zu Informationen werden, welche dann ihrerseits kontextualisiert werden. In der Individualsoftwareentwicklung, wo weder ein Massenmarkt noch Massenproduktion zu finden ist, kann die Betrachtung von quantitativen Daten dennoch nützlich sein. So stellen Fehlerdatenbanken und Messgrößen für Softwaremetriken (z.B. lines of code, Fehler pro tausend LOC) Quellen für strukturierte Daten dar, die mit Werkzeugen des KDD ausgewertet werden können (vgl. Dingsøyr, Conradi 2002, S. 409f). Diese Daten werden sich in mittelständischen Unternehmen zwar nicht zu Terabytes aufsummieren, verdienen aber trotzdem eine systematische Auswertung.

**Marktbeobachtung:** Um die identifizierte Lücke für externes Wissen und den Wissensprozess Beobachten zu füllen wird das Instrument Marktbeobachtung vorgeschlagen. Es soll Mitarbeiter dabei unterstützen, sich über externes Wissen zu informieren, es zu filtern und seine Brauchbarkeit für die Organisation einzuschätzen. Eine geeignete Maßnahme aus organisatorischer Sicht ist v.a. das Bereitstellen von Arbeitszeit, die Mitarbeiter aufwenden können, um sich z.B. im Internet auf einschlägigen Nachrichtenseiten über Marktentwicklungen informieren können. Weiterhin tragen Zeitschriftenabonnements, das Beschaffen von Büchern, oder das Ermöglichen von Konferenzteilnahmen zur Marktbeobachtung bei. Aus technischer Sicht ist neben dem obligatorischen Webbrowser die Unterstützung beim Archivieren von Inhalten und Verweisen sinnvoll. Dazu sollten die Inhalte idealerweise nicht nur für den einzelnen Mitarbeiter offline verfügbar gemacht werden sondern auch eine Kopie auf einem Server im Intranet gespeichert, eine automatische Kategorisierung vorgenommen und Interessenten, die für die Kategorie ein Informationsabonnement haben über Neuzugänge benachrichtigt werden. Somit besteht auch eine enge Anknüpfung an das semantische Content Management insbesondere für die dort angedeuteten Social Bookmarking Funktionen. RSS-Feeds können für Übersichten über die Neuerungen verwendet werden.

**Zusammenarbeitsunterstützung:** Moreño et al. schlagen weiterhin noch Groupware als technologie-orientiertes Instrument vor (Moreño et al. 2004). Im Sinne der oben angeführten Definition kann Groupware alleine sicher kein eigenständiges WM-Instrument bilden. Es stellt vielmehr die technische Unterstützung für Communities dar. Andererseits ist Groupware eher als Teamunterstützungswerkzeug entwickelt worden. Das ist als Hinweis zu werten, dass ein WM-Instrument fehlt, welches Gruppenarbeit in Teams unterstützt. Der Wissensdienst Zusammenarbeit in der EKI deutet ebenfalls darauf hin, dass so ein Instrument sinnvoll wäre. Neben den von Groupware bereitgestellten

Grundfunktionen zur Kommunikation über Email, der Verwaltung von Terminen in eigenen und Gruppenkalendern sowie der Ablage von Notizen werden in diesem Zusammenhang in den letzten Jahren vermehrt synchrone Kommunikationsdienste wie Instant Messaging, Voice-over-IP (VoIP) und Desktop-Videokonferenzen diskutiert. Dies soll dazu beitragen, in jeder Situation die optimale Kommunikationsform wählen zu können. Ein Awareness-Mechanismus hilft bei der Einschätzung (Gutwin, Greenberg 1998). Dazu wird für die Teamkollegen oder andere Mitarbeiter, mit denen häufig kommuniziert wird, ein Status angezeigt, der darüber informiert, ob er gerade online verfügbar und somit synchron erreichbar ist, nicht gestört werden will oder unterwegs ist. Noch weiter gehende Unterstützung kann geboten werden, wenn die Kommunikationsplattform weiter integriert ist und z.B. eine Fax-Integration bietet, eine Verbindung zwischen VoIP und Telefonanlage besteht oder Awareness-Informationen bei Abwesenheit aus dem Groupware-Kalender bezogen werden.

### **2.4.4 Fazit**

Die Diskussion bestehender und neuer Vorschläge für WM-Instrumente zeigt, dass dieses Themengebiet noch nicht abschließend bearbeitet wurde. Eine ausschließlich theoretische Aufarbeitung wird somit auch nicht zu einem endgültigen Ergebnis führen können. Es müssen empirische Studien durchgeführt werden, bei denen WM-Maßnahmen erhoben und Zusammenhänge aufgezeigt werden. Auf Grundlage solcher Maßnahmenkataloge können dann Vorschläge für Instrumente neu durchdacht und vervollständigt werden. Die vorgestellten Instrumente stellen jedoch eine Grundlage dar, auf deren Basis erste Analysen in der Theorie und Empfehlungen für die Praxis möglich erscheinen.

## **2.5 Erfolgsmessung im Wissensmanagement**

Nachdem nun die verschiedenen Foki des Wissensmanagement, die Anwendungsdomäne sowie mögliche Maßnahmen und Instrumente des WM vorgestellt wurden, soll im abschließenden Abschnitt des Kapitels diskutiert werden, wie die Wirksamkeit eingeführter Maßnahmen und Instrumenten gemessen werden kann.

Für die Einführung jeder Art von Informationssystemen oder organisatorischer Neuerungen ist die Messung des Erfolges der Neuerung wichtig. Return on Investment (ROI) und ähnliche hoch aggregierte Kennzahlen sind aber oft schwer zu ermitteln oder basieren auf nicht belegbaren Annahmen. Erfolgsmessung im Allgemeinen ist ein komplexer und fehleranfälliger Prozess und für die Erfolgsmessung im WM gilt dies

besonders, da Wissen durch seine immaterielle Natur schwer messbar ist (Kankanhalli, Tan 2004, S. 1; Hanley, Malafsky 2003, S. 369).

Zusätzlich zur Komplexität des Messens ist nach wie vor unklar, welche Messgrößen für die Erfolgsmessung im WM heranzuziehen sind. Kankanhalli und Tan sammeln Forschungsergebnisse aus 22 Projekten zur Erfolgsmessung, die benutzerbezogene, WMS-bezogene, projektbezogene, prozessbezogene oder organisationsbezogene Messgrößen für WM vorschlagen (Kankanhalli, Tan 2004, S. 4-6). Sie kommen zu dem Ergebnis, dass Nutzerakzeptanz und Benutzerfreundlichkeit von WMS bisher zu wenig für die Messung herangezogen wurden. Weiterhin liegt der Fokus zu stark auf integrativen WMS (v.a. ECM) und zu wenig auf interaktiven (z.B. elektronische CoP). Zudem wurde eine Lücke zwischen Studien auf Mikroebene (Benutzer, System) und Makroebene (Organisation) identifiziert, die durch Berücksichtigung von Teamstrukturen und Gruppendynamik geschlossen werden könnte (ibid., S. 5).

### **2.5.1 Messansätze**

Messung des Intellektuellen Kapitals (IC, Intellectual Capital) ist ein bekannter top-down Ansatz mit einer ganzheitlichen Sicht auf immaterielle Güter einer Organisation, das IC oder Wissenskapital (Sveiby, Lloyd 1987). Konkrete Methoden, die hierzu zählen, sind der Intangible Asset Monitor (Sveiby 1997), der Skandia Navigator (Skyrme, Amidon 1997) und der Intellectual Capital Navigator (Stewart 1997), die alle Weiterentwicklungen der Balanced Scorecard (Kaplan, Norton 1996) darstellen. Es gibt auch eine Reihe von Forschungsberichten die beschreiben, wie die Balanced Scorecard auf konkrete WM-Instrumente, z.B. Semantic Content Management (Laugero, Globe 2002, S. 16ff), oder wie bestimmte Wissensziele, z.B. Ausbeutung des vorhandenen Wissens, Innovation und Weiterbildung angewendet werden kann (Arora 2002). Ein neuerer Ansatz, der auf IC abstellt, ist das WM Value Assessment Framework, das eine Reihe von Indikatoren vorschlägt, mit denen IC quantitativ gemessen werden kann (Conway 2003).

Bottom-up Ansätze zur Erfolgsmessung im WM benutzen IT-Systeme oder deren Benutzer als Startpunkt der Messung. Viele davon basieren auf dem bekannten Modell des Erfolgs von Informationssystemen von DeLone und McLean (DeLone, McLean 1992). Einige der Ansätze sind Erweiterungen (z.B. Li 1997; Pitt et al. 1995; Myers et al. 1998), andere sind Gegenvorschläge für Teile der Beziehungen zwischen den einzelnen Komponenten (z.B. Seddon 1997), wieder andere sind alternative Modelle, die einen vollständigen anderen Ansatz wählen (z.B. Ballantine et al. 1998). Unter den aktuelleren Erweiterungen die einen

klaren Bezug zu WM haben sind v.a. das organizational memory information system success model (Jennex, Olfman 2003) und das Modell für die Erfolgsmessung von WMS (Maier, Hädrich 2001; Maier 2004, S. 331ff) zu nennen.

Das Messen von Fertigkeiten einzelner Mitarbeiter ist seit längerem Forschungsgebiet des Personalwesens (HRM, Human Resource Management). Die Effektivität von Trainingsmaßnahmen kann zum Beispiel mit der Test-Retest Methode, der Pre-Post Performance Methode oder der experimentellen Kontrollgruppen-Methode gemessen werden (De Cenzo, Robbins 1994, S. 276f). Test-Retest heißt, dass Mitarbeiter einen Test vor der Schulungsmaßnahme machen müssen und einen Nachtest, nachdem die Maßnahme abgeschlossen ist. Die Methode zur Messung der Pre-Post Leistung ist sehr ähnlich, misst die Leistung aber nicht in einem gestellten Test, sondern in einer echten Arbeitssituation. In der experimentellen Kontrollgruppe werden zwei Gruppen mit ähnlichen Fertigkeiten und Voraussetzungen getestet, nachdem eine Gruppe an einer bestimmten Maßnahme teilgenommen hat und die andere nicht.

Ein weiterer Ansatz, der von Menschen als Startpunkt ausgeht, ist die soziale Netzwerkanalyse (Flap et al. 1998; siehe auch Maier 2004, S. 344f). Sie misst allerdings nicht die Fertigkeiten der Menschen, sondern die Veränderungen in den Kommunikationshäufigkeiten, Vertrauens- oder Beratungsbeziehungen, wodurch sich Rückschlüsse auf den Erfolg des WM-Instruments ziehen lassen.

Ein Ansatz, der zwar IT-lastig ist, jedoch trotzdem von Mitarbeitern als Startpunkt der Messung ausgeht, ist die Benutzerzufriedenheit mit WMS (Ong, Lai 2004). Ong und Lai haben ein System mit 26 Fragen entwickelt, das auf die Qualität der Inhalte, die Verständlichkeit, den Umgang mit Dokumenten, sowie Communities und Personalisierung eingeht.

Ein Messansatz der von den Prozessen als Startpunkt ausgeht, ist das Process Classification Framework (PCF) des American Productivity Center (APQC) (APQC 1998). Es hilft Organisationen die Leistungsfähigkeit ihrer Prozesse mit anderen Organisationen zu vergleichen (benchmarking), indem es eine detaillierte Taxonomie vorgibt, die aus Daten von annähernd 100 Firmen abgeleitet wurde (Kankanhalli, Tan 2004, S. 3). Ein weiterer Ansatz zur Analyse der Prozessleistung ist die WM Wertekette (Bots, de Bruijn 2002), die allerdings Wissensprozesse fokussiert, statt die Geschäftsprozesse zu analysieren und zu bewerten (Jennex, Olfman 2004).

Ein Ansatz, der die Lücke zwischen Top-Down und Bottom-Up Messansätzen schließen will (Smits, de Moor 2004) zielt sowohl auf konkrete WM-Instrumente (z.B. CoP) als auch auf Prozesse (nämlich Nonakas Wissenskonversionsprozesse Sozialisierung, Externalisierung, Kombination und Internalisierung. (vgl. Nonaka, Takeuchi 1995). Es basiert auf Arbeiten zum Intellektuellen Kapital und benutzt deren Methoden um eigene Indikatoren für jeden Prozess zu definieren und zu messen (z.B. direkte Kommunikationsbeziehungen als Indikator für Sozialisierung).

Intellectual Capital (IC) setzt sich nach Peppard & Rylander (2001) aus den drei Bestandteilen Humankapital (human capital), soziales Kapital (relationship capital) und Organisationskapital (organizational capital) zusammen (Paauwe 2004, S. 188). Humankapital besteht aus den Fähigkeiten und Fertigkeiten, sowie der geistigen Agilität der Mitarbeiter. Soziales Kapital besteht aus den wertvollen Beziehungen der Mitarbeiter mit Kunden, Lieferanten und anderen relevanten Parteien. Organisationskapital besteht aus den Prozessen, Routinen, Systemen, Marken, Patenten und Dokumenten in denen wertvolles Wissen für die Organisation enthalten ist.

Wiig dagegen unterscheidet in Humankapital und strukturelles Kapital, welches weiter unterteilt wird in Organisationskapital und Kundenkapital (Wiig 1997, S. 401). Eine ähnliche Klassifizierung findet sich auch bei Maier, der von einer ressourcen-orientierten Sicht ausgeht und immaterielle Ressourcen in personen-abhängige und personen-unabhängige Ressourcen aufteilt, wobei erstere aus implizitem und explizitem Wissen bestehen, während letztere in Organisationskapital und immaterielle Vermögenswerte unterteilt werden können (Maier 2004, S. 93).

Auch im Personalwesen ist die Erfolgsmessung ein wichtiges Thema und es verwundert nicht, dass auch dort die bestehenden Ansätze der Balanced Scorecard und des IC verwendet und erweitert werden. So setzen Walker und McDonald die von Becker et al. (2001) entwickelte HR scorecard erfolgreich bei der Firma Verizon ein (Walker, MacDonald 2001). Als operationalisierte Messwerte werden z.B. der HR ROI index, der Gewinn pro Mitarbeiter, Befragungsergebnisse über Mitarbeiterzufriedenheit, der Prozentsatz umgesetzter HR Programme, das Erreichen kritischer Fähigkeiten und das Halten von zentralen Wissensträgern definiert (S. 372). Paauwe entwickelt basierend auf den Erfahrungen mit der HR scorecard die 4logic HRM scorecard, welche die Bereiche Strategie (strategic logic), Beruf (professional logic), Gesellschaft (societal logic) und Lieferung (delivery logic) unterscheidet. Hervorzuheben ist außerdem die konsequente

Ausrichtung an den „Kunden des Personalwesens“: Mitarbeitern, Management und Aufsichtsrat. Interessant ist hier v.a. die Umsetzung der einzelnen Bereiche in konkrete Kenngrößen. So werden die Prozentsätze von neu eingestellten, bzw. freiwillig ausgeschiedenen Experten (high potentials) als Maß für die Beeinflussung der Unternehmenskultur herangezogen, die Anzahl innovativer Ideen für die Teamentwicklung und der prozentuale Anstieg der Fähigkeiten der Mitarbeiter als Kenngröße für Personalaktivitäten zu Verbesserung des Unternehmensklimas (siehe Paauwe 2004 Tabelle 9-7).

(Hannula et al. 2004) plädieren für ein Management, das an der Erfolgsmessung ausgerichtet ist. Die Erfolgsmessung sollte dabei sowohl Faktoren auf organisatorischer wie auch individueller Ebene berücksichtigen, da der Erfolg bei wissensintensiven Tätigkeiten in erster Linie von der Kompetenz und dem Einsatz Einzelner abhängt (ibid. S. 2). Sie schlagen weiterhin vor, ein bis zehn Bereiche für die Messung und 10 bis 20 konkrete Messgrößen zu definieren. Dabei sollen insbesondere Messgrößen zu den Kompetenzen der Mitarbeiter berücksichtigt werden. Konkretere Vorschläge werden aber nicht gemacht.

### **2.5.2 Integration der Ansätze**

Zusammenfassend lässt sich aus den vorgestellten Ansätzen ableiten, dass einige gemeinsame Startpunkte für die Erfolgsmessung existieren. Es gibt zwei Hauptströmungen für die Erfolgsmessung die sich auch im WM-Bereich wieder finden: Top-Down und Bottom-Up Ansätze. Beim Top-Down Vorgehen sind die Geschäfts- und / oder WM-Strategie Startpunkte, welche mit Intellectual Capital und Balanced Scorecard Methoden fokussiert werden. Beim Bottom-Up Ansatz sind Informationssysteme (IS), deren Inhalte, sowie die Mitarbeiter die wichtigsten Startpunkte. Für die Erfolgsmessung bei den IS nehmen die Weiterentwicklungen des DeLone/McLean IS success Modells den größten Raum ein. Mitarbeiter-bezogene Faktoren können mit sozialer Netzwerkanalyse, Modellen der Benutzerzufriedenheit und anderen Methoden des HRM erhoben werden. Die Messung der Inhalte zielt auf die Ermittlung der Wertigkeit der Wissensgüter (knowledge assets) ab, was allerdings aufwändiger ist als andere Ansätze und zudem fragliche Ergebnisse liefert.

Bei der Diskussion der beiden gegensätzlichen Ansätze drängt sich die Frage nach einem vermittelnden Konzept auf, was die Lücke zwischen Top-Down und Bottom-Up Ansätzen füllen kann. Zwei mögliche Kandidaten dafür sind WM-Instrumente, sowie Geschäfts- und Wissensprozesse (siehe Abbildung 2-8).

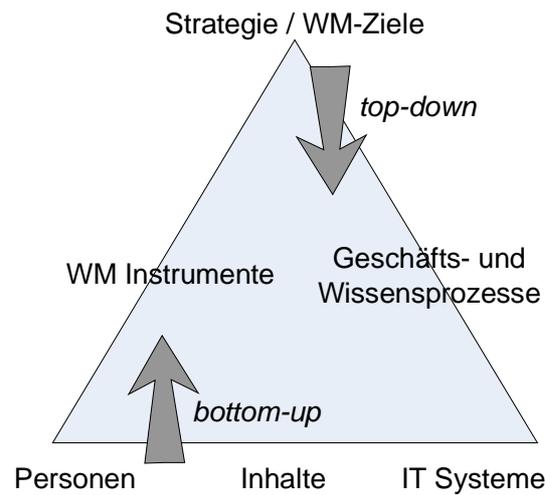


Abbildung 2-8: Startpunkte für die Erfolgsmessung im Wissensmanagement (eigene Darstellung)

Wie erläutert existieren bereits einige Versuche in diese Richtung. Es bleibt jedoch Raum für weitere Forschungen in diesem Gebiet. Im nächsten Abschnitt wird ein eigener Ansatz vorgestellt, der auf Wissensweitergabe als wichtigem Teil des WM abstellt und WM-Instrumente als Startpunkt für die Erfolgsmessung verwendet.

### 2.5.3 Erfolgsmessung der Wissensweitergabe

Für jede Phase des Wissensweitergabeprozesses existiert eine Reihe von Faktoren, die den Erfolg der Weitergabe begünstigen oder behindern können. Argote und Ingram sind der Meinung, dass Wissensweitergabe durch Änderungen des Wissens oder der Arbeitsleistung gemessen werden kann (Argote, Ingram 2000, S. 151). Allerdings kann Wissen selbst in der Regel nicht direkt gemessen werden und die Leistung der Mitarbeiter hängt nicht nur vom Wissen alleine, sondern auch von der Fähigkeit das Wissen anzuwenden, ihrer Motivation, der daraus resultierenden Einsatzstärke, sowie der Schwierigkeit der Aufgabe und dem situationsspezifischen Glück ab (Weinert, Waldmann 1988). Andererseits sind leistungsbasierte Messansätze besser geeignet den Erwerb impliziten Wissens zu erheben, als andere Ansätze, die versuchen Wissen direkt zu messen (Argote, Ingram 2000, S. 152). Argote und Ingram gehen weiterhin davon aus, dass es in Organisationen fünf Speicher für Wissen gibt, nämlich (a) Mitarbeiter, (b) Rollen und Organisationsstrukturen, (c) die eingespielten Abläufe und Arbeitspraktiken der Organisation, (d) die Organisationskultur, sowie (e) die physische Struktur des Arbeitsplatzes. Um den Wissenstransfer durch Änderungen des Wissens zu messen muss das Wissen in allen fünf Speichern gemessen werden (Argote, Ingram 2000, S. 152). Die vorliegende Arbeit beschränkt sich darauf, die Wissensweitergabe zwischen Individuen zu messen, auch wenn die Weitergabe nicht

direkt sondern mittelbar durch dokumentiertes Wissen erfolgt. Inkorporiertes bzw. eingebettetes Wissen in Organisationsstrukturen oder dem Arbeitsplatz wird nicht betrachtet.

(Cummings, Teng 2003, S. 41f) identifizieren in der Literatur vier Ansätze, wie erfolgreiche Wissensweitergabe als abhängige Variable operationalisiert wird:

1. als hohe Anzahl erfolgter Wissensweitergaben (Hakanson, Nobel 1998)
2. als Einhaltung von Zeit- und Budgetbeschränkungen bei gleichzeitiger Zufriedenstellung des Empfängers (Szulanski 1996)
3. als erfolgreiche „Wiedererschaffung“ (re-creation) des Wissens aus den übertragenen Informationen (Kim, Nelson 2000), also der Integration in die eigene Wissensbasis ohne Verfälschung
4. als Erlangen des Besitzes von, des Bekenntnisses zu und der Zufriedenheit mit dem übertragenen Wissen auf Seiten des Empfängers (Kostova 1999).

Die Ansätze drei und vier scheinen im Rahmen der vorliegenden Arbeit am besten geeignet, um den Erfolg der Wissensweitergabe zu messen, da sie im Einklang mit der Definition von Wissensweitergabe als Prozess der Interpretation weitergegebener Informationen steht. Die Anzahl an Weitergaben erscheint trotz der leichten Messbarkeit kein geeignetes Maß, da diese in keinem festen Verhältnis zur „Menge“ an übermitteltem Wissen steht. Die Einhaltung von Zeit und Budgetbeschränkungen wird zwar als wichtig für eine Organisation und deren Erfolg erachtet, steht jedoch in einem übergeordneten Kontext und sollte ergänzend zur erfolgreichen Wissensweitergabe verwendet werden, aber nicht zum Messen der Wissensweitergabe an sich.

Zusammenfassend kann Erfolg von Wissensweitergabe in dieser Arbeit also wie folgt definiert werden:

Erfolgreiche Wissensweitergabe wird gemessen durch die Ähnlichkeit und Vollständigkeit des neu gewonnenen Wissens beim Empfänger im Vergleich zu dem Wissen, das der Sender expliziert hat, sowie durch das Bekenntnis zu diesem Wissen. Dabei werden kürzer dauernde Weitergabeaktivitäten als erfolgreicher als länger dauernde bewertet, wenn beide zu dem gleichen Resultat führen.

Ähnlichkeit und Vollständigkeit des Wissens sind in der betrieblichen Praxis schwierig zu messen. Mit Testverfahren, die sowohl Fakten als auch die Anwendung des rekonstruierten Wissens abprüfen, kann man sich jedoch nach Meinung des Autors an beide Variablen annähern. Durch Anwendungsfragen wird auch zu einem gewissen Grad das Bekenntnis

überprüfbar, da nur Lösungsstrategien vorgeschlagen werden, an deren Erfolg man auch glaubt. Die Definition wurde in erster Linie im Hinblick auf die Simulation formuliert, wo die Messung der Variablen Ähnlichkeit, Vollständigkeit und Bekenntnis zum neuen Wissen wesentlich einfacher ist. Sie beinhaltet den Wiedererschaffungsaspekt von Kim und Nelson und die Besitzerlangung sowie das Bekenntnis zum neu erlangten Wissen von Kostova, lässt aber offen, ob der Empfänger das neue Wissen unmittelbar erfolgreich anwenden kann, was maßgeblich für dessen Zufriedenheit ist. Die bloße Anzahl an Transferaktivitäten (Ansatz eins) scheint nicht geeignet, den Erfolg zu messen, da weder die „Menge“ weitergegebenen Wissens pro Transferaktivität berücksichtigt wird, noch damit gesagt wird, dass der Empfänger überhaupt das explizierte Wissen wieder rekonstruieren kann. Ansatz zwei scheint ungeeignet, da er für jede einzelne Wissensweitergabeaktivität die Planung von Zeit und Kosten erfordern würde. Das erscheint auf der Mikroebene weder möglich noch sinnvoll.

Eine Wissensbilanz kann nicht nur die Summe des Wissens in allen Wissensbereichen über alle Mitarbeiter enthalten sondern sollte auch die Redundanz messen, also wie viele Mitarbeiter innerhalb eines Wissensgebiets das gleiche Wissen haben.

## **2.6 Zusammenfassung**

In diesem Kapitel wurden die Grundlagen des Wissensmanagement diskutiert und die der Arbeit zugrunde liegenden Auffassungen von Wissen, Wissensweitergabe und WM-Instrumenten dargelegt. Die Perspektive auf Wissen in Personen wurde als die favorisierte dargestellt, wobei die Sichten auf Wissen in Objekten und in sozialen Systemen als wichtige Ergänzung zu sehen sind, auch wenn sie nach Meinung des Autors für sich alleine genommen nicht tragfähig sind. Darauf aufbauend wird Wissensweitergabe als ein sozialer Prozess verstanden, der das Explizieren von Wissen in Personen, die Weitergabe von kodifiziertem Wissen in Dokumenten oder in Form von Sprache sowie dessen Internalisierung, Interpretation und Bewertung umfasst.

## **3 Erklärungsansätze für Wissensweitergabe und Arbeitsverhalten**

Im folgenden Kapitel werden die in Abschnitt 2.2.4 identifizierten Kontextbereiche für Wissensweitergabe genauer untersucht und in der sozial- und wirtschaftswissenschaftlichen Literatur, Psychologie und Pädagogik nach Konzepten gesucht, die Einflussfaktoren innerhalb der Kontextbereiche erklären. Dabei wird neben Grundlagenwerken v.a. Literatur aufgearbeitet, die die einzelnen Konzepte schon in den Kontext der wissensintensiven Tätigkeiten oder Wissensweitergabe einordnet. Die Auswahl der diskutierten Theorien und Konzepte erfolgt zum einen durch die Analyse empirischer Arbeiten zur Wissensweitergabe in Abschnitt 3.1 und deren theoretischem Unterbau und zum anderen durch Vergleich einiger zugänglicher Standardwerke und Lehrbücher in den angesprochenen Bereichen, wobei die häufiger diskutierten Theorien aufgegriffen werden.

### **3.1 Einflussfaktoren für Wissensweitergabe**

Wenn versucht wird, die Ergebnisse unterschiedlicher empirischer Studien zu harmonisieren und in Einklang zu bringen, muss beachtet werden, dass ein Teil des Kontexts verloren geht und in den meisten Fällen zumindest eine geringfügige Neuinterpretation der Ergebnisse vorgenommen wird. Trotz der vorgenommenen Vorauswahl der Studien, um einen möglichst homogenen Hintergrund zu erhalten, bleiben die Voraussetzungen, von denen in den Studien ausgegangen wird, i. d. R. leicht unterschiedlich.

Die Weitergabe von Wissen über den Austausch von Informationen und Rekonstruktion des Wissens wird in vielen Disziplinen erforscht (z.B. Psychologie, Pädagogik, Informationswissenschaft). Es werden vor allem Situationen beleuchtet, die den Wissensaustausch besonders schwierig machen, wie z.B. verschiedene Kulturkreise beim Austausch zwischen Geschäftseinheiten eines multinationalen Konzerns, verschiedene Unternehmenskulturen und Misstrauen beim Austausch in strategischen Allianzen, sowie bei Unternehmensübernahmen oder -zusammenschlüssen (Cummings, Teng 2003, S. 41). Dabei ist jedoch die Wissensweitergabe im einfachen Fall, innerhalb einer Firma im gleichen Gebäude bei gleicher Nationalität und Religion, gleichem Geschlecht und innerhalb derselben Abteilung schon komplex genug und kann derzeit nicht als verstanden

bezeichnet werden. In der vorliegenden Arbeit und in diesem Abschnitt wird daher der Fokus auf Studien gelegt, die intraorganisatorische Wissensweitergabe untersuchen.

In einer Reihe von Studien mit Studenten in IS Kursen haben Joshi, Sarker und Sarker untersucht, inwieweit die Menge des weitergegebenen Wissens von verschiedenen Faktoren wie Kommunikationsaktivität, Glaubwürdigkeit und Kompetenz der Quelle abhängt (Sarker et al. 2003; Joshi et al. 2004; Joshi et al. 2005). Während die erste Studie die Wissensweitergabe in verteilt arbeitenden Teams untersucht, beleuchten die Arbeiten aus 2004 und 2005 die Wissensweitergabe in physisch zusammen arbeitenden Teams. Die erzielten Ergebnisse sind nicht vollkommen konsistent. So wird zwar eine hohe Kommunikationsaktivität in allen drei Studien als wichtige Voraussetzung für die Wissensweitergabe identifiziert, Glaubwürdigkeit und Kompetenz der Quelle konnten aber nicht immer als Einflussfaktor bestätigt werden. Die Studien von 2003 und 2005 untersuchen weiterhin die Abhängigkeit von einer individualistischen oder kollektivistischen (Landes-) Kultur, was in beiden Studien bestätigt werden konnte. In 2005 wurde zudem der Einfluss von Gruppenzusammenhalt auf die Wissensweitergabe innerhalb der Gruppe untersucht, konnte aber nicht nachgewiesen werden.

(Ensign, Hébert 2004) untersuchen den Einfluss von Reputation auf die Wissensweitergabe von technischem Wissen bei Forschern in multi-nationalen und multi-divisionalen Konzernen in der Pharmazie. Die unabhängige Variable Reputation wird dabei durch das vergangene Verhalten und das erwartete zukünftige Verhalten definiert. Ersteres wird durch die Art, Dauer und Häufigkeit der vergangenen Interaktionen operationalisiert, während für das erwartete zukünftige Verhalten die Vorhersehbarkeit, die Reziprozität und existierende Verpflichtungen als Messgrößen dienen. Zudem wurden die Kontextvariablen organisatorische, geographische und technologische Distanz, der betreffende Forschungsprozess (Grundlagenforschung oder angewandte Forschung), die organisatorische Beziehung zwischen Quelle und Empfänger (horizontal oder vertikal), ob die betroffenen Abteilungen durch Firmenübernahmen zusammengekommen sind, die verwendeten Systeme und die Signifikanz des übertragenen Wissens berücksichtigt. Die Ergebnisse bestätigen, dass sowohl die Reputation des Empfängers als auch der Forschungsgruppe der er angehört einen Einfluss auf die Wissensweitergabe hat. Die Autoren betonen weiterhin, dass die Existenz von dauerhaften Austauschbeziehungen und sozialen Bindungen wichtige Voraussetzung für eine erfolgreiche Wissensweitergabe ist.

(Cummings, Teng 2003) analysieren die Wissensweitergabe von Forschungsergebnissen zwischen Organisationen oder Organisationseinheiten in 15 Branchen. Sie postulieren neun Faktoren, die innerhalb dieser Kontextbereiche die Wissensweitergabe positiv oder negativ beeinflussen. Auf Seiten der Wissensquelle, wo die Autoren den Wissenskontext (knowledge context) ansiedeln, sehen sie die Ausdrückbarkeit (articulability) als förderlich und das Maß der Eingebettetheit (embeddedness) als hinderlich an. Dem Beziehungskontext (relational context) zugeordnet identifizieren sie organisatorische, physische, normative und wissensbezogene Distanz als Hindernisse für die Wissensweitergabe. Auf Empfängerseite können eine Lernkultur (learning culture) und Dringlichkeit (priority) positiv wirken. Im Aktivitätskontext (activity context) wirken schließlich Transferaktivitäten förderlich für die Wissensweitergabe. In der durchgeführten Studie konnten aber nur die Faktoren Ausdrückbarkeit, Eingebettetheit, normative und wissensbezogene Distanz, sowie die Anzahl an Transferaktivitäten bestätigt werden.

(Szulanski 1996) untersucht Barrieren für den Transfer von Best Practices zwischen Organisationseinheiten einer Firma. Er berücksichtigt dabei die Faktoren Schwierigkeit des Erkennens relevanten Wissens, nicht bestätigte Güte des Wissens, fehlende Motivation, fehlendes Vertrauen in den Empfänger oder die Quelle, fehlende Aufnahmefähigkeit (absorptive capacity), mangelnde Dauerhaftigkeit der Wissensaufnahme (retentive capacity), unfruchtbarer organisatorischer Kontext und eine angespannte Beziehung zwischen Quelle und Empfänger. Die empirischen Ergebnisse weisen vor allem die Aufnahmefähigkeit des Empfängers, die Schwierigkeit relevantes Wissen zu identifizieren und die soziale Beziehung zwischen Quelle und Empfänger als potentielle Barrieren aus und bestätigen damit die These des Autors, dass wissensbezogene Faktoren wichtiger sind als motivationale (S. 36). Da die vorliegende Arbeit nicht die Weitergabe von Best Practices, sondern von technischem Faktenwissen untersucht, scheint die Identifikation von relevantem Wissen weitaus einfacher zu sein, so dass dieser Faktor als Barriere eine untergeordnete Rolle spielt.

Basierend auf qualitativen ethnographischen Studien untersuchen (Berends 2002; Berends et al. 2004) grundlegende Mechanismen zur Wissensweitergabe in zwei Forschungseinrichtungen. Sie unterscheiden die Transferprozesse nach den Dimensionen Weitergabe bestehenden Wissens vs. Entwicklung neuen Wissens, aktive oder passive Weitergabe (push vs. pull) und dem Ziel der Wissensweitergabe. Für letztere Dimension werden die Ausprägungen „eigenes Problem“ (Sender), „Problem des Ziels“ (Empfänger),

„gemeinsames Problem“ oder „kein bestimmtes Problem“ vorgeschlagen. Damit ergibt sich eine Matrix mit 24 möglichen Wissensweitergabemechanismen. Von diesen 24 konnten aber nur 16 Mechanismen in der Empirie bestätigt werden. Von den 16 konnten wiederum 8 Mechanismen häufiger beobachtet werden (12-54 von 227 Fällen) wogegen die übrigen 8 nur selten auftraten (1-8 Fälle).

*Tabelle 3-1: Wissensweitergabesituationen nach (Berends et al. 2004)*

<b>Bezeichnung</b>	<b>Wissenstyp</b>	<b>Push/Pull</b>	<b>Ziel</b>	<b>Häufigkeit</b>
-	Bestehendes Wissen	Push	Sender	54
<b>Thinking along</b>	Neues Wissen	Push	Empfänger	37
<b>Diffusion</b>	Bestehendes Wissen	Push	Ungerichtet	26
<b>Information Pooling</b>	Bestehendes Wissen	Push	Gemeinschaft	20
<b>Pushing</b>	Bestehendes Wissen	Push	Empfänger	18
<b>Information retrieval</b>	Bestehendes Wissen	Pull	Empfänger	16
<b>Collaborative problem solving</b>	Neues Wissens	Push	Gemeinschaft	14
<b>Self-suggestion</b>	Neues Wissen	Push	Sender	12

Der am häufigsten beobachtete Mechanismus wird von den Autoren nicht weiter erklärt. Es scheint sich aber nicht um Wissensweitergabe im engeren Sinne zu handeln, sondern vielmehr um die Anwendung von bestehendem Wissen auf das eigene Problem (siehe Tabelle 3-1). Thinking along beschreibt den Fall, dass der Sender eine Idee zu einem Problem des Empfängers hat und ihm diese mitteilt. Diffusion beschreibt die ungerichtete Weitergabe von Wissen in Form von Berichten und Publikationen einerseits oder in Form von Geschichten (vgl. Wiig 2003, S. 15f) über (Miss-)Erfolge bei sozial motivierten Gelegenheiten (z.B. gemeinsames Mittagessen).

Information Pooling beschreibt das gegenseitige Informieren über Projektfortschritte und Probleme z.B. bei Team-Besprechungen. Pushing beschreibt die Weitergabe von bestehendem Wissen zur Lösung eines Problems des Empfängers. Dieser Mechanismus ähnelt „thinking along“ mit dem Unterschied, dass in einem Fall das Wissen aktiv selbst generiert wird, im anderen Fall aus (meist externen) Quellen gelerntes oder schon vorhandenes Wissen weitergegeben wird. Der Anlass zur Weitergabe ist entweder das Lernen von neuem Wissen, die Kenntnisnahme des Problems des Empfängers oder das erstmalige Auftreten einer Gelegenheit zur Weitergabe. Information retrieval beschreibt die Weitergabe von bestehendem Wissen auf Nachfrage des Empfängers. Collaborative

problem solving beschreibt das gemeinschaftliche Lösen von Problemen, z.B. bei der unmittelbaren Zusammenarbeit zweier Personen oder bei der Diskussion eines Problems in einer Team-Besprechung. Dies wurde bei der Definition von Wissensweitergabe explizit ausgeschlossen und stellt daher keine Wissensweitergabe im Sinne dieser Arbeit dar. Self-suggestion schließlich beschreibt das Phänomen, dass manchmal schon das Reden über ein Problem mit einem Kollegen und der damit verbundene Zwang zur strukturierten Beschreibung ausreicht, um selbst neue Problemlösungsvorschläge zu generieren, ohne die aktive Mitarbeit des Kollegen in Anspruch zu nehmen. Inwiefern in diesem Fall von Wissensweitergabe gesprochen werden kann ist zweifelhaft, da der Sender oft schon auf einen Problemlösungsvorschlag kommt, bevor der Empfänger das Problem überhaupt vollständig verstanden hat, so dass auch die Problemlösung nicht nachvollzogen werden kann und das vom Sender neu generierte Wissen vom Empfänger nur zu einem sehr geringen Anteil rekonstruiert werden kann. Das Phänomen scheint eher im Lichte der Situated Cognition Theorie, sowie den Erkenntnissen der Kognitionspsychologie erklärt werden zu können, nach denen sich Wissen nicht aus dem Gedächtnis abrufen lässt, sondern rekonstruiert werden muss, wodurch sich die Wissensstrukturen bereits ändern können und dadurch scheinbar neue Erkenntnisse generiert werden.

(Wah et al. 2005) untersuchen die Wissensweitergabe in einem Weiterbildungsinstitut. Ihre theoretischen Vorüberlegungen basieren auf dem Social Capital Ansatz. Sie definieren zu jedem der drei Bausteine sozialen Kapitals einige Facetten, welche die Dimension ausmachen.

#### 1) Strukturelle Dimension sozialen Kapitals

- a) Organisatorische Fürsorge (organizational care)
- b) Achtung und Belohnung (recognition and reward)

#### 2) Relationale Dimension sozialen Kapitals

- a) Kompetenz (competence)
- b) Integrität (integrity)
- c) Offenheit (open-mindedness)

#### 3) Agency Dimension sozialen Kapitals

- a) Soziale Motive (pro-social motives)
- b) Vermeidung negativer Folgen des Nicht-Teilens (impression management)
- c) Geteilte Werte (shared values)

Zu diesen acht Facetten des sozialen Kapitals werden auch acht Voraussetzungen diskutiert, die für erfolgreiche Wissensweitergabe erfüllt sein müssen.

1. Es müssen Gelegenheiten zur Wissensweitergabe vorhanden sein, nämlich formelle Gelegenheiten wie Trainings, Arbeitsgruppen oder IT-Systeme, die Wissensteilung fördern, oder informelle wie persönliche Beziehungen und Communities of Interest
2. Es muss ein Kommunikationskanal vorhanden sein, bevorzugt durch physische Nähe.
3. Es müssen Erwartungen über einen Nutzen der Wissensweitergabe vorhanden sein.
4. Es müssen Erwartungen über Kosten das Wissen nicht weiterzugeben gegeben sein.
5. Der Kontext von Lernendem und Lehrendem muss kompatibel sein.
6. Intrinsische Motivation muss gegeben sein (z.B. Reziprozität).
7. Der Lernende und der Lehrende müssen sich leiden können (personal compatibility)
8. Opportunismus

Basierend auf diesen Vorüberlegungen entwickeln die Autoren sechs Hypothesen über Faktoren die positiv auf Wissensweitergabe wirken. Die Faktoren sind identisch mit den Facetten sozialen Kapitals ausgenommen 2b und 3c In einem Fragebogen werden diese Einflussfaktoren mit je 4-6 Fragen operationalisiert, die auf einer Skala von 1 (strongly disagree) bis 5 (strongly agree) gemessen werden. Hinzu kommen die Faktoren Kosten und Nutzen der Wissensweitergabe, sowie die Kosten des Hortens von Wissen. In der Studie wurden alle Hypothesen bestätigt. Allerdings ist hohe Kompetenz nur bei gleichzeitig gegebener Achtung und Belohnung für Wissensweitergabe ein positiver Einflussfaktor.

(Song et al. 2003) untersuchen Wissensweitergabe von technischem Wissen in der Produktentwicklung zwischen Geschäftseinheiten von 277 US-amerikanischen High-tech Firmen. Sie identifizieren basierend auf einer Analyse der Managementliteratur 17 potenzielle Einflussfaktoren, von denen 11 in Tiefeninterviews mit den Forschungsleitern von IBM, Philips, Microsoft, Motorola, Sony, Intel, and Merck bestätigt wurden. Die Faktoren sind physische Nähe, Informationstechnologie, Kontakt zu führenden Anwendern und Lieferanten, persönliches Engagement, formelle Belohnungen, das Budget der Geschäftseinheit, Langzeitorientierung, Redundanz in der Organisation, organisatorische Krisen, die Risikoeinstellung und Managementunterstützung für die Integration. Sie entwickeln für jeden Faktor eine Messskala mit 3-4 Messgrößen und kommen zu dem Ergebnis, dass Kontakt zu führenden Anwendern und Lieferanten, persönliches Engagement, Langzeitorientierung und organisatorische Krisen die Wissensweitergabe

positiv beeinflussen, wogegen Redundanz in der Organisation die Wissensweitergabe behindert.

(Cross et al. 2000) untersuchen die Voraussetzungen für erfolgreiche Wissensweitergabe und befragen dazu 40 Manager. Sie identifizieren dabei vier wichtige Faktoren: Meta-Wissen, also das Wissen, welche Person welches Wissen hat, Verfügbarkeit des Wissensträgers, den direkten Einbezug der Personen in die Problemlösung und eine gefestigte soziale Beziehung zu der Person, so dass eigene Wissenslücken zugegeben werden können, ohne Gefahr zu laufen zurückgewiesen oder verlacht zu werden. Bei der Identifikation von Experten wurden zwei verschiedene Gruppen von Mitarbeitern gefunden, die häufig gefragt wurden: diejenigen, die Experte in einem bestimmten technischen Gebiet sind (Spezialisten) und diejenigen, mit denen man zusammen ein bestimmtes Problem analysieren kann, wo noch kein spezielles Wissensgebiet bestimmt werden kann (Generalisten). Der identifizierte Wissensträger muss nicht nur zeitnah physisch, oder über ein Medium erreichbar sein, sondern der Fragende muss auch lernen, wie er effektiv mit dem speziellen Mitarbeiter kommuniziert. Dazu muss er dessen Angewohnheiten und Vorlieben kennen, um die Fragen entsprechend formulieren, das richtige Medium und die richtigen Modalitäten (Zeit, Ort, etc.) wählen zu können. Die Wissensträger müssen auch bereit sein, sich in die Problemlösung einbeziehen zu lassen, was nicht so sehr eine zeitliche Frage sondern vielmehr durch die Bereitschaft zum Verstehen und Durchdenken des Problems und die anschließende Weitergabe des eigenen Wissens bestimmt ist. Die Scheu vor dem Offenlegen eigener Wissenslücken kann sowohl von einer ungenügend festen sozialen Beziehung, als auch von der Persönlichkeit des Nachfragenden bestimmt sein.

In der folgenden Liste sind die Faktoren aus den beschriebenen Studien noch einmal explizit aufgeführt und zusammen mit den Faktoren aus den Literaturreviews, die im Anhang zusammengefasst sind, kategorisiert und den in Kapitel 2 erarbeiteten Kontextbereichen zugeordnet.

#### 1. Gelegenheitskontext

##### 1.1 Soziale Gelegenheiten

###### 1.1.1 Interaktionen zwischen Mitarbeitern (Handzic, Ladd/Ward)

###### 1.1.2 Co-Location, Co-Presence, Kommunikationshäufigkeit (Sarker/Joshi, Song)

##### 1.2 Organisatorisch verankerte Gelegenheiten

- 1.2.1 Organisierte Besprechungen (Handzic)
- 1.2.2 Personalbewegungen im Unternehmen (Job rotation) (B/K)<sup>17</sup>
- 1.2.3 Post-Projekt-Evaluationen (Song)
- 1.3 Gelegenheiten durch organisations-externes Lernen
  - 1.3.1 Interaktionen mit führenden Anwendern, Kunden und Lieferanten (Song, B/K)
  - 1.3.2 Trainings (B/K)
- 1.4 Technisch ermöglichte Gelegenheiten
  - 1.4.1 Zugang zu Kollegen / Wissensquellen (richtiges Medium) (Cross)
  - 1.4.2 IT-Infrastruktur, die Gelegenheiten zur Kommunikation und Zusammenarbeit bietet (Handzic)
- 2. Quellkontext
  - 2.1 Mentale Fähigkeiten zur Weitergabe
    - 2.1.1 Identifikation des relevanten Wissens (Szulanski)
    - 2.1.2 Kompetenz (Sarker/Joshi), Wissen (Ladd/Ward)
    - 2.1.3 Weitergabefähigkeit (didaktische Fähigkeit) (B/K)
  - 2.2 Einstellung zur Wissensweitergabe
    - 2.2.1 Individuelle Verpflichtung (commitment) (Song)
    - 2.2.2 Offenheit (Wah)
    - 2.2.3 Bereitschaft, an der Problemlösung teilzunehmen (Cross, B/K)
    - 2.2.4 Bereitschaft zur Wissensweitergabe (B/K, Wah)
  - 2.3 Motivation
    - 2.3.1 Motivation zur Wissensweitergabe (B/K)
    - 2.3.2 Gefühl, etwas zu leisten (Hendriks)
    - 2.3.3 Gefühl, verantwortlich zu sein (Hendriks)
    - 2.3.4 Einstellung zum Horten von Wissen (Wah)
- 3. Transferkontext
  - 3.1 Kommunikationshäufigkeit (Sarker/Joshi)

---

<sup>17</sup> B/K = Becker / Knudsen

3.2 Anzahl der Transferaktivitäten (Cummings/Teng)

3.3 Technologie (Boer, B/K)

3.4 Existenz und mediale Reichhaltigkeit eines Kommunikationskanals (B/K)

#### 4. Wissenskontext

4.1 Ausdrückbarkeit (Cummings/Teng), Impliztheit (B/K)

4.2 Eingebettetheit (Cummings/Teng)

4.3 Art des Wissens (Boer)

4.4 Bestätigungsstatus (B/K) (Verbindlichkeit)

4.5 Zweideutigkeit (B/K)

4.6 Verbreitungsgrad (B/K)

4.7 Ursprung des Wissens (B/K)

#### 5. Empfängerkontext

5.1 Einstellung zum Lernen

5.1.1 Individuelle Verpflichtung (commitment) (Song)

5.1.2 Offenheit (Wah)

5.2 Metawissen über Wissensquellen (Cross)

5.3 Mentale Fähigkeiten zum Lernen

5.3.1 Identifikation des relevanten Wissens (Szulanski)

5.3.2 Absorptive Capacity (Szulanski, B/K)

5.3.3 Depreciation, Retentive Capacity (Ladd/Ward)

#### 6. Nutzungskontext (Geschäftsprozesse)

6.1 Charakteristika der Aufgabe, die gelernt werden soll (Häufigkeit, Heterogenität, Unbestimmtheit) (B/K)

6.2 Aktivität (Boer)

#### 7. Beziehungskontext

7.1 Soziale Beziehung

7.1.1 Sichere Beziehung, um wichtige Fragen zu stellen (Cross, B/K)

7.1.2 Soziale Beziehung (Szulanski), Typ (B/K) und Art der Beziehung (B/K)

7.1.3 Stärke der Beziehung (B/K), Häufigkeit und Tiefe des Kontakts (Ladd/Ward)

7.1.4 Einbettung der Beziehung (B/K)

- 7.1.5 Informelle Beziehungen (B/K)
- 7.1.6 Relation Model (Boer)
- 7.2 Sozialer Status (Thomas-Hunt)
  - 7.2.1 Reputation von Quelle und Gruppe (Ensign/Hebert)
  - 7.2.2 Vertrauen (Handzic)
  - 7.2.3 Glaubwürdigkeit (Sarker/Joshi)
  - 7.2.4 Zuverlässigkeit (B/K)
- 7.3 Expertenstatus (Thomas-Hunt)
  - 7.3.1 Ansehen für Wissensweitergabe (Wah)
  - 7.3.2 Anerkennung der Wissensweitergabe als gute Arbeit (Hendriks)
- 7.4 Gemeinsamkeiten von Quelle und Empfänger
  - 7.4.1 Normendistanz (Cummings/Teng)
  - 7.4.2 Wissensdistanz (Cummings/Teng)
  - 7.4.3 Ähnlichkeit zwischen Quelle und Empfänger und deren Wissen (B/K, Ladd/Ward)
  - 7.4.4 Überlappung des Wissens von Quelle und Empfänger (B/K)
  - 7.4.5 Wissensredundanz (B/K)
- 8. Organisationskontext
  - 8.1 Organisationskultur
    - 8.1.1 Offene Atmosphäre (Handzic)
    - 8.1.2 Gemeinschaftsgefühl (Handzic), kollektive Identität (B/K)
    - 8.1.3 (Organisations-)Kultur (Sarker/Joshi, Boer)
    - 8.1.4 Aufgeschlossene Kultur für Wandel und Innovation (Ladd/Ward)
    - 8.1.5 Kultur mit Aufgabenorientierung (Ladd/Ward)
    - 8.1.6 Kultur der Herausforderung und des Wettbewerbs (Ladd/Ward)
    - 8.1.7 Vorherrschendes Relation Model (Boer)
  - 8.2 Organisationsstruktur
    - 8.2.1 Arbeitsteilung (Boer)
    - 8.2.2 Feedback-Mechanismen (Song)
    - 8.2.3 Formelle Organisation (Struktur, Autonomie, Kontrolle) (B/K)

#### 8.2.4 Teams (Song)

### 8.3 Strategie

#### 8.3.1 Langzeitorientierung (Song)

#### 8.3.2 Organisatorische Redundanz (überlappende Skills/Aufgaben) (Song)

#### 8.3.3 Wettbewerb zwischen Organisationseinheiten um Ressourcen oder Marktanteile (B/K) => Profitcenter

#### 8.3.4 Kongruenz zwischen individuellen und organisatorischen Zielen (Ladd/Ward)

### 8.4 Organisationskrise (Song) => externer Druck, starke Konkurrenz, schwacher Markt

### 8.5 Managementinstrumente

#### 8.5.1 Aufstiegschancen (Hendriks)

#### 8.5.2 Anreizsysteme (Wah, B/K)

#### 8.5.3 Organisatorische WM-Instrumente (B/K)

#### 8.5.4 Arbeitspraktiken, die Kommunikation und Imitation von Routinen fördern (B/K)

Es ergeben sich insgesamt 30 Konzepte, die durch 76 Faktoren detailliert werden. Diese sollen in den folgenden Abschnitten näher erörtert bzw. mit Hilfe geeigneter Theorien und empirischer Befunde erklärt werden.

## 3.2 Gelegenheits- und Nutzungskontext

Die in der Empirie bestätigten Faktoren wurden gesammelt und die dem Gelegenheitskontext zuordenbaren sind in Tabelle 3-2 zusammengefasst. Zur besseren Übersicht wurde versucht, die zehn identifizierten Faktoren weiter zu verdichten und zu klassifizieren.

Formelle Gelegenheiten werden durch organisatorische Maßnahmen ermöglicht und können von Führungskräften beeinflusst werden. Die letztendliche Umsetzung hängt aber von den Mitarbeitern ab, da Gelegenheiten auch wahrgenommen werden müssen, um zu erfolgreicher Wissensweitergabe zu führen. Informelle Gelegenheiten ergeben sich dagegen eher zufällig oder durch soziale Beziehungen zwischen Mitarbeitern, die nur begrenzt durch die Organisation beeinflusst werden können. Sind direkte Interaktionen schwierig, weil die Mitarbeiter physisch getrennt sind, so können technische Systeme bis zu einem gewissen Grad Ersatz dafür schaffen und den Zugang zu Wissensquellen ermöglichen. Aber selbst wenn ein direkter Zugang besteht, können technische Systeme

aus der Erfahrung des Autors die Hemmschwelle zum Ansprechen eines Kollegen senken. Gründe dafür können Bequemlichkeit aber auch die subjektiv geringere Störung und leichtere Ablehnmöglichkeit des Kollegen sein. Aufgrund der gesunkenen Hemmschwelle kann die häufigere Nachfrage aber auch lästig werden und somit den gegenteiligen Effekt hervorrufen. Schließlich kommt relevantes Wissen oft auch von außen in das Unternehmen, was in der vorliegenden Arbeit nicht weiter berücksichtigt wird, da es sich dabei um organisationsübergreifende Wissensweitergabe handelt. Es ist jedoch als Gelegenheit zur Wissensweitergabe zu berücksichtigen, wenn Mitarbeiter von außen neues Wissen erhalten haben und dieses dann intern weitergeben können.

*Tabelle 3-2: Einflussfaktoren im Gelegenheitskontext*

<b>Kategorie</b>	<b>Faktor</b>
Formelle, organisatorisch verankerte Gelegenheiten	Organisierte Besprechungen (Handzic et al. 2004)
	Personalbewegungen im Unternehmen, Job rotation (Becker, Knudsen 2003 )
	Post-Projekt-Evaluationen (Song et al. 2003)
	Interne Trainings (Becker, Knudsen 2003 )
Informelle, soziale Gelegenheiten	Interaktionen zwischen Mitarbeitern (Handzic et al. 2004; Ladd, Ward 2002)
	Co-Location, Co-Presence, Kommunikationshäufigkeit (Song et al. 2003; Joshi et al. 2005)
Technisch ermöglichte Gelegenheiten	Zugang zu Kollegen / Wissensquellen (richtiges Medium, Cross et al. 2000)
	IT-Infrastruktur, die Gelegenheiten zur Kommunikation und Zusammenarbeit bietet (Handzic et al. 2004)
Gelegenheiten zur Internalisierung organisations-externen Wissens	Interaktionen mit führenden Anwendern, Kunden und Lieferanten (Becker, Knudsen 2003 ; Song et al. 2003)
	Externe Trainings (Becker, Knudsen 2003 )

Gelegenheiten werden in der Literatur nur wenig diskutiert (z.B. in Hädrich, Maier 2004). Wesentlich häufiger findet sich das verwandte Konzept der Arbeitssituation, das im Folgenden daher anstatt der Gelegenheit genauer untersucht wird. Dabei muss beachtet werden, dass der Begriff Situation oft sehr weit gefasst wird und daher einer Operationalisierung nicht mehr zugänglich ist.

### 3.2.1 Arbeitssituation

Jost sieht in der Arbeitssituation eine wesentliche Determinante für das Arbeitsverhalten. Er fasst dabei den Begriff der Situation sehr weit und bezieht sowohl familiäre und private Dinge, als auch längerfristig gültige organisationsinterne Faktoren in die Situation mit ein, wenn sie sich auf das Verhalten des Mitarbeiters auswirken können (Jost 2000, S. 77f). Er unterscheidet zwischen der spezifischen und der generellen Arbeitssituation eines Mitarbeiters. Die generelle Arbeitssituation gilt für alle Mitarbeiter gleichermaßen, während die spezifische Arbeitssituation nur für den einzelnen Mitarbeiter gilt (ibid. S. 78f). Die spezifische Arbeitssituation kann in Ihrer Anforderung stark variieren, von Routinetätigkeiten bis hin zu hoch komplexen Aufgaben (Wiig 2003, S. 7). Die Situation kann als Anreiz, der Motive des Mitarbeiters aktiviert, oder als Verhaltensrestriktion aufgefasst werden (Jost 2000, S. 79). Anreize können extrinsisch oder intrinsisch motivieren und materiell oder immateriell sein (ibid. S. 81, siehe auch Abschnitt 3.3.2). Restriktionen können (arbeits-)rechtlicher (z.B. aus dem Arbeitsvertrag entstehend) oder sozialer Natur (z.B. Gruppennormen), sowie ressourcenbedingt sein (z.B. Projektbudget, ibid. S. 83ff). Situationen werden von Mitarbeitern unterschiedlich wahrgenommen (ibid. S. 87). Die Gründe hierfür sind individuelle Unterschiede der Mitarbeiter (ibid. S. 88, siehe auch Abschnitt 3.3.6).

Personen erkennen Situationen, entscheiden was zu tun ist, setzen die Aktion in die Tat um und beobachten das Ergebnis (Wiig 2003, S. 7). Situationen können dabei aus einer einzelnen Begebenheit bestehen, mehreren Episoden die im Zeitverlauf zusammen gehören, oder einer Bedingung, die sich dynamisch verändert bis sie eine Aktion verlangt (ibid.).

Entscheidend für das Erkennen von Gelegenheiten ist das Verständnis der sozialen Situation (vgl. Fincham, Rhodes 2003, S. 168). Situationen können als diskrete Entitäten beschrieben werden, die aus Zielen, Regeln, Verhaltenselementen, Konzepten, Umgebungsbedingungen und Rollen bestehen. Die Situationen können als Gelegenheiten verstanden werden, in denen die Person ein oder mehrere persönliche oder aufgabenbezogene Ziele erreichen kann (ibid.). Regeln können in allgemeingültige und situationsspezifische oder in interpretative, präskriptive und nicht-generalisierbare Regeln unterteilt werden (Fincham, Rhodes 2003, S. 169). *Situationsspezifische Regeln* verändern oder ergänzen allgemeingültige Regeln, so dass sich z.B. Gesprächsregeln signifikant unterscheiden können, je nachdem ob man als Verkäufer ein Kundengespräch führt (man

denke z.B. an die „der Kunde hat immer recht“ (Maxime), oder ob man Bewerber in einem Bewerbungsgespräch ist. *Interpretative Regeln* stellen Kriterien zur Verfügung, die zur Interpretation der sozialen Situation herangezogen werden können, so dass z.B. ein Ausruf im Fußballstadion als normal eingestuft wird, der im Büro als unangemessen oder sogar beleidigend empfunden würde. *Präskriptive Regeln* schreiben angemessene Reaktionen auf Ereignisse in sozialen Situationen vor. *Nicht-generalisierbare Regeln* können zur Erklärung von Verhalten (z.B. Rechtfertigung) in einer bestimmten Situation herangezogen, aber schwer auf andere Situationen übertragen werden. Sie tragen der Komplexität von sozialen Situationen Rechnung, die es nicht erlaubt Regeln aufzustellen, die in jeder Situation anwendbar sind (ibid.). Situationen besitzen auch ein sozio-kulturell gewachsenes Repertoire an möglichen Verhaltenselementen (Fincham, Rhodes 2003, S. 170). Obgleich prinzipiell alle Verhaltenselemente eines Menschen gezeigt werden könnten, werden nur die im Repertoire enthaltenen als angemessen empfunden. So wird es z.B. bei einer formellen Besprechung als unangemessen empfunden werden sich auf die Tischkante zu setzen, während es bei der anschließenden „After work party“ bei gleichen Räumlichkeiten und Personen nicht als störend empfunden werden könnte. Auch die kognitiven Konzepte, die ein Mensch besitzt können situationsspezifisch sein (ibid.). Vor dem Hintergrund der in Abschnitt 3.3.6 diskutierten Forschungsergebnissen scheint es, dass die entsprechenden Konzepte im Gedächtnis voraktiviert sind und damit eine gewisse Erwartungshaltung auftritt, dass die Konzepte in der Situation auch auftreten werden. Die Umgebungsbedingungen beeinflussen die Wahrnehmung einer Situation ebenfalls (Fincham, Rhodes 2003, S. 170). Trifft man z.B. einen Kollegen in der Kaffeeküche, dann fällt es leichter ein informelles Gespräch zu beginnen als in der als formeller wahrgenommenen Atmosphäre des Büros. Rollen sind schließlich Ausdruck der Arbeitsteilung in einer Organisation (z.B. Projektmanager, Entwickler), aber auch des Platzes, den eine Person in einer sozialen Beziehung einnimmt (z.B. Ehemann, Sportpartner) und mit Erwartungen bzgl. des Verhaltens einer Person in dieser Rolle verbunden (Fincham, Rhodes 2003, S. 171). Das Verhalten einer Person kann sich signifikant ändern, wenn die Person die Erwartungen anderer an die Rolle erfährt und diese Erwartungen internalisiert (siehe auch Abschnitt 3.7.2).

Es werden offene, geschlossene und definierte Situationen unterschieden (Fincham, Rhodes 2003, S. 169f). Geschlossene Situationen sind vollständig durch Regeln gesteuert, z.B. Rituale oder Zeremonien. Offene Situationen sind nur wenig durch Regeln gesteuert

und besitzen außerdem meist kein vordefiniertes Ziel, z.B. informelle Treffen mit Freunden. Definierte Situationen sind zwischen diesen beiden Extremen angesiedelt und besitzen zwar relativ genaue Regeln, aber auch eine gewisse Unschärfe in Bezug auf die Ziele der Beteiligten.

Konkrete Situationen, die zu Wissensweitergabe führen werden in der Literatur nur spärlich beschrieben (siehe oben). (Berends et al. 2004) unterscheiden in ihrer empirischen Studie Push- und Pull-Situationen. In Push-Situationen wird die Wissensweitergabe nicht von Personen initiiert, die Wissen aktiv nachfragen, sondern von Personen, die neues Wissen durch Lernen gewonnen haben und dies dann aktiv an Kollegen weitergeben. In den untersuchten Forschungseinrichtungen wurde sogar die überwiegende Anzahl an Wissensweitergabeaktivitäten durch Push-Situationen eingeleitet.

Für die organisatorisch verankerten Gelegenheiten nennt Drumm Coaching, Traineeprogramme, Stellenrotation und Hospitation (Drumm 2000, S. 401). Eine Reihe von Gelegenheiten, die zu Wissensweitergabe in Organisationen führen können, wurden von Argote et al. gesammelt. Sie sprechen zwar von Weitergabemechanismen, bei genauerer Analyse wird aber klar, dass diese bei einer Mikrobetrachtung als Gelegenheiten einzustufen sind. Sie identifizieren Mitarbeiterbewegungen in der Organisation (personnel movement), Training, Kommunikationsmöglichkeiten, Beobachtungsgenauigkeiten, Reverse Engineering von Produkten und Interaktionen mit Lieferanten und Kunden als solche Gelegenheiten in der Literatur (Argote et al. 2000, S. 3).

In Push-Situationen ist der vorherige Wissenserwerb durch die spätere Quelle des Wissensweitergabeprozesses entscheidend. Wissenserwerb erfolgt z.B. organisiert in Personalentwicklungsmaßnahmen. Als wissensorientierte Personalentwicklungsverfahren nennen Sonntag und Schaper kognitives Training, kooperative, arbeitsplatzbezogene Lernformen, computergestützte mediale Lernprogramme sowie komplexe Lehr-Lern-Arrangements in der Ausbildung (Sonntag, Schaper 2001, S. 248). Kognitive Trainingsverfahren dienen nicht zum Erlernen konkreter Arbeitsabläufe oder Fakten, sondern sollen die grundsätzlichen Denkleistungen wie mentale Simulation, Planen und Entscheiden schulen (ibid.). Als Kooperative Lernformen nennen Sonntag und Schaper im wesentlichen Coaching. Dabei sind drei Methoden relevant (Sonntag, Schaper 2001, S. 251). Beim kognitiven Modellieren dient der Coach als Modell, der die Lösung eines Problems demonstriert und dabei seine Wahrnehmungen und Überlegungen verbalisiert und damit für den Lernenden beobachtbar macht. Als „Anleiten und Zurücknehmen“ wird

die Beobachtung des Lernenden bei der Problemlösung bezeichnet, die durch Unterstützung in Form von Feedback, Hinweisen oder Erinnern an Teilprozesse ergänzt wird. Mit „Hilfestellung geben“ bezeichnen die Autoren eine weitgehend passive Begleitung des Lernenden, der nur im Falle von Überforderung unterstützt wird, damit er nicht die Motivation verliert. Unterstützt werden sollen diese Methoden durch Verbalisierung der Handlungen, bewusste Reflexion und selbsttätige Exploration des Problemraumes (Sonntag, Schaper 2001, S. 252).

Als technisch-medierte Gelegenheiten sind nicht nur Kommunikationsmedien sondern auch computergestützten Lernprogramme relevant. Dort sind v.a. Simulationsprogramme hervorzuheben, die weit verbreitete Hypertextsysteme, sowie Trainings- und Tutorensysteme unterstützen können (ibid. S. 253). In der Softwareentwicklung können z.B. Datenflüsse in komplexen Softwaresystemen simuliert werden, um das Zusammenspiel einzelnen Komponenten des Systems zu erlernen. Unter komplexen Lehr-/Lernarrangements verstehen Sonntag und Schaper die Konfrontation mit vollständigen, in ihrer Komplexität gestuften Arbeitsaufgaben (Sonntag, Schaper 2001, S. 255). Als Methoden können u.a. Übungsfirmen, Fallstudien und Planspiele eingesetzt werden. Im Bereich der Softwareentwicklung werden hier z.B. Wettbewerbe eingesetzt in denen selbst geschriebene Programmteile in einer genau spezifizierten Ablaufumgebung gegeneinander antreten. Dies ist allerdings eher als Lerngelegenheit einzustufen denn als Wissensweitergabebegelegenheit. Erfolg aber auch Misserfolg beim Lernen kann andererseits Anlass für Push-Gelegenheiten zur Wissensweitergabe sein.

Eine weitere formelle Gelegenheit, die hauptsächlich zur aktiven Nachfrage nach Wissen führen wird, ist das Beurteilungsgespräch. Es wird in Organisationen oft im jährlichen Turnus durchgeführt und dient u.a. der Rückmeldung über die wahrgenommene Leistung eines Mitarbeiters, der Festlegung von Qualifizierungszielen und -maßnahmen und dem Treffen verbindlicher Vereinbarungen (z.B. Gehaltserhöhung, Leistungsbonus Finke et al. 2001, S. 465). Dadurch kann der Mitarbeiter sein Selbstbild bzgl. Wissen und Leistung mit dem Fremdbild abgleichen. Er erhält weiterhin Richtungsvorgaben für neue Wissensziele und evtl. durch finanzielle Regelungen Rückmeldung zu seinen Erwartungen in der Vergangenheit, wie sich (Wissensweitergabe-)Aktionen auf sein zukünftiges Gehalt auswirken.

Pull-Gelegenheiten werden oft durch unerwartet auftretende Probleme bei der aktuellen Arbeitsaufgabe ausgelöst, z.B. wenn der gerade geschriebene Programmteil einen Fehler enthält, den der Mitarbeiter nicht selber findet oder beheben kann.

Als formelle Gelegenheiten ist in der Softwareentwicklung zusätzlich zu den oben beschriebenen die Projektphase der Dokumentation zu nennen (Walz et al. 1993, S. 63). Auch formelle Projektreviews sind solche Gelegenheiten (Henninger 1997). Eine Möglichkeit zur Verbindung von formellen und informellen Gelegenheiten in der Softwareentwicklung ist das Pair Programming, was im Rahmen vieler agiler Softwareentwicklungsmethoden vorgeschlagen wird und die Wissensweitergabe verbessert (Cockburn, Williams 2001, S. 7).

#### **3.2.2 Nutzung in Geschäftsprozessen**

Die Entscheidung des Autors, das Anwenden des weitergegebenen Wissens nicht mit in den Wissensweitergabeprozess aufzunehmen, sondern nur als Erweiterung im Gesamtmodell zu betrachten wird gestützt durch die Tatsache, dass nur ein Einflussfaktor diesem Kontextbereich zugeordnet werden konnte. So finden Becker und Knudsen (2003) die Charakteristika der Aufgabe, die erlernt werden soll (Häufigkeit, Heterogenität, Unbestimmtheit) als Einflussfaktor. Die Aktivitätentheorie (Kuutti 1996; Engeström 2000) beschreibt wie ein Akteur beim Lösen einer Aufgabe in einen Kontext eingebettet ist. Boer et al. beziehen dieses allgemeine Modell auf die Wissensweitergabe. Ein zentraler Punkt ist dabei die Einbettung der Wissensweitergabe in den Kontext der Aktivität. Der Akteur (Subjekt) ist über soziale Regeln und Normen in die Gruppe, Organisation und Gesellschaft eingebunden. Vermittelnde Artefakte wie Hard- und Software, Symbole und Modelle helfen bei der Ausführung der Aufgabe. Die Aufgabenverteilung bestimmt, wie die beteiligten Akteure ihren Teil zur Bewältigung der Aufgabe beitragen müssen (Boer et al. 2002b, S. 4).

Das durch Wissensweitergabe erworbene Wissen wird in den Geschäftsprozessen angewendet. Der Kerngeschäftsprozess innerhalb der Individualsoftwareentwicklung lässt sich anhand eines Vorgehensmodells beschreiben. Von diesen Vorgehensmodellen werden in der Literatur viele verschiedene diskutiert, die sich aber v.a. bei den Übergängen zwischen den einzelnen Phasen und vorgesehenen Iterationen, Zyklen und Rückschritten unterscheiden (Balzert 2001, S. 99ff). Die Frage, welche Phasen relevant sind, wird jedoch abgesehen von der Aufspaltung einzelner Phasen in detailliertere Subphasen (z.B. Entwurf

=> Grob- und Feinentwurf) relativ einheitlich beantwortet. So finden sich in vielen Quellen die Phasen (z.B. Balzert 2001, S. 99ff)

- |                            |                         |
|----------------------------|-------------------------|
| (1) Anforderungsdefinition | (2) Analyse             |
| (3) Entwurf                | (4) Implementierung     |
| (5) Test                   | (6) Betrieb und Wartung |

Etwas detaillierter ist die Liste der Softwareentwicklungsaktivitäten, die Edwards gesammelt hat (Edwards 2003, S. 14). Da sie trotzdem große Übereinstimmungen mit dem Phasenmodell von Balzert aufweist, wird sie hier nicht weiter diskutiert.

Bei der Anforderungsdefinition wird beim Kunden erhoben, welche Fähigkeiten die Software braucht. Dabei kann in funktionale und nicht-funktionale Anforderungen unterschieden werden. Nicht-funktionale Anforderungen sind z.B. Antwortzeitverhalten bei einer definierten Hardwareausstattung und vorgegebener Last, sowie Ausfallsicherheit und intuitive Bedienbarkeit. Funktionale Anforderungen sind anwendungsspezifisch auf die jeweilige Aufgabe ausgerichtet und sind z.B. Import eines spezifizierten Datenformats, Prüfen eines Datensatzes auf Einhaltung gewisser Bedingungen, oder Benachrichtigung des Benutzers über eingetretene Ereignisse.

In der Analysephase werden die Anforderungen weiter detailliert und priorisiert. Darauf aufbauend werden auf grobgranularer Ebene Architekturentscheidungen getroffen (Kirk, Tempero 2004, S. 4). Wenn der Kunde schon selbst eine detaillierte Anforderungsanalyse gemacht hat, so kann die Analysephase auch die Anforderungsdefinition ersetzen.

In der Entwurfsphase wird die detaillierte Architektur erarbeitet und ein Entwurfsdokument erstellt, das genaue Vorgaben für die Entwickler festhält (ibid.). Hier sollten auch Projektpläne erarbeitet werden in denen Aufgabenzuordnung, geschätzte Aufwände und Meilensteine vermerkt sind. Der Entwurf sollte vollständig abgeschlossen sein, bevor mit der Implementierung begonnen wird (ibid.). Je nach Kundenwunsch und Marktsituation wird entweder nach der Analyse oder nach der Entwurfsphase das bindende Angebot erstellt, was den Preis und die zu erbringende Leistung festlegt und damit die Grundlage des Vertrags wird.

In der Testphase können verschiedene Tätigkeiten unterschieden werden. Zu den manuellen Prüfverfahren zählen u.a. Inspektion, Review und Walkthrough (Balzert 2001, S. 302ff). Automatisierte und teilautomatisierte Verfahren werden als Tests bezeichnet.

Dort wird im Phasenmodell v.a. nach der Reichweite der Tests in Modul-, Integrations- und Systemtest unterschieden (ibid. S. 101).

Neuere Vorgehensmodelle sehen zwischen Test und Betrieb eine Einführungsphase vor, die Installation vor Ort, Tests durch die Endbenutzer, Schulung der Endbenutzer und Migration von Daten aus Vorgängersystemen beinhalten kann (siehe Noack, Schienmann 1999, S. 170ff für einen Überblick; Rational 1999, S. 6f für Details).

Betrieb und Wartung wird oft vom Kunden selbst übernommen. Ist das nicht der Fall, so sind im Normalfall signifikant andere Wissensgebiete für diese Phase nötig, als für die vorherigen Phasen. Es müssen Logdateien ausgewertet, Prozessorlast, Hauptspeicher- auslastung und freie Festplattenkapazität überwacht und das Backup konfiguriert werden. Zudem ist oft ein Helpdesk für die Endbenutzer zu betreiben. Im Fehlerfall ist allerdings wieder das Wissen der IT-Architekten und Programmierer erforderlich, die für Entwurf und Implementation der Applikation verantwortlich waren, damit die Ursache des Fehlers gefunden und behoben werden kann.

Für die Anwendung als direkt der Wissensweitergabe angeschlossener und damit finaler Prozessschritt der Betrachtung ist die Messung der erfolgreichen Anwendung des Wissens ein wichtiger Gesichtspunkt. Neben den üblichen Kriterien Zeit, Kosten, Qualität wird in der Softwareentwicklung oft die Produktivität der Entwickler (oft als Output-Leistung in Zeilen Programmcode (LoC – Lines of Code) pro Arbeitsstunde definiert), als Erfolgsvariable herangezogen (Bailey et al. 1982; Cain, McCrindle 2002; Hochstein et al. 2005).

Aus Sicht der Wissensweitergabe ist ein wichtiger Aspekt die Weitergabe von Wissen der Mitarbeiter einer Phase an Mitarbeiter, die für eine andere Phase zuständig sind. So fehlt z.B. den Mitarbeitern die am Helpdesk arbeiten das Wissen der Programmierer oder IT-Architekten, um eine Frage des Endbenutzers schnell beantworten zu können. Oder der Programmierer weiß nicht genug über die Gründe für eine Designentscheidung, um die Spezifikationen des Entwurfsdokuments im Sinne der IT-Architekten umsetzen zu können, oder dem IT-Architekt fehlt das Wissen über Implementierungsprobleme, die mit seinem Architekturvorschlag verbunden sind.

#### **3.2.3 Diskussion**

Die Gelegenheit zur Wissensweitergabe und die Anwendung des neu erworbenen Wissens stellen die Verbindungsstücke von den Geschäftsprozessen und Routinetätigkeiten in deren

Rahmen zum Wissensprozess der Wissensweitergabe dar. Bislang fehlen aber noch weitgehend die theoretischen Grundlagen dazu, wie diese Übergänge genau vonstatten gehen und was man zu deren Unterstützung tun kann (vgl. Hädrich, Maier 2004). Dies spiegelt sich auch in den Ausführungen dieses Kapitels wieder. Die aus Sicht des Autors wichtigste Erkenntnis hier ist zum ersten die Unterscheidung in Push- und Pull-Gelegenheiten, die von den wenigsten Arbeiten zur Wissensweitergabe bisher berücksichtigt wurde. Des Weiteren wurden formelle, organisatorisch verankerte und informelle, soziale Gelegenheiten zur Wissensweitergabe identifiziert und IT-Systeme als möglicherweise dabei unterstützenden Faktor diskutiert. Schließlich wurden die grundlegenden Tätigkeiten in den verschiedenen Phasen der Softwareentwicklung angesprochen und auf die Verschiedenartigkeit des benötigten Wissens für die einzelnen Phasen hingewiesen. Die daraus folgende Spezialisierung der Mitarbeiter auf eine oder wenige Phasen macht die Wissensweitergabe für die Fälle, in denen Wissen, das vorwiegend in einer Phase generiert wird auch in einer der anderen Phase benötigt wird, besonders interessant. Dort sind größere Schwierigkeiten zu erwarten, da das gemeinsame Wissen der Spezialisten aus verschiedenen Phasen wesentlich kleiner ist, als von Mitarbeitern, die auf dieselbe Phase spezialisiert sind. Eine detailliertere Diskussion der benötigten Wissensbereiche findet sich in Abschnitt 3.4.1.

### 3.3 Quellen- und Empfängerkontext

Die Quelle des Wissens in einem Wissensweitergabeprozess muss in erster Linie das „richtige“ Wissen haben, das auf Seite des Empfängers benötigt wird, oder zumindest gebraucht werden kann. Breites Wissen in einer Reihe von Wissensbereichen führt dabei eher zu erfolgreicher Wissensweitergabe in der Softwareentwicklung als jahrelange Erfahrung in dieser Industrie (Walz et al. 1993, S. 73f). In der folgenden Aufzählung werden die oben angeführten Faktoren zusammengefasst, bevor in den folgenden Abschnitten damit verbundene Konzepte und Theorien erörtert werden.

*Tabelle 3-3: Einflussfaktoren im Quellkontext*

Mentale Fähigkeiten zur Weitergabe	<ul style="list-style-type: none"><li>• Identifikation des relevanten Wissens (Szulanski)</li><li>• Kompetenz (Sarker/Joshi), Wissen (Ladd/Ward)</li><li>• Weitergabefähigkeit (didaktische Fähigkeit) (B/K)</li></ul>
------------------------------------	--

### 3 Erklärungsansätze für Wissensweitergabe und Arbeitsverhalten

Einstellung zur Wissensweitergabe	<ul style="list-style-type: none"> <li>• Individuelle Verpflichtung (commitment) (Song)</li> <li>• Offenheit (Wah)</li> <li>• Bereitschaft an der Problemlösung teilzunehmen (Cross, B/K)</li> <li>• Bereitschaft zur Wissensweitergabe (B/K, Wah)</li> <li>• Einstellung zum Horten von Wissen (Wah)</li> </ul>
Motivation	<ul style="list-style-type: none"> <li>• Motivation zur Wissensweitergabe (B/K)</li> <li>• Gefühl, etwas zu leisten (Hendriks)</li> <li>• Gefühl, verantwortlich zu sein (Hendriks)</li> </ul>

Tabelle 3-4: Einflussfaktoren im Empfängerkontext

Einstellung zum Lernen	<ul style="list-style-type: none"> <li>• Individuelle Verpflichtung (commitment) (Song)</li> <li>• Offenheit (Wah)</li> </ul>
Metawissen	<ul style="list-style-type: none"> <li>• Metawissen über Wissensquellen (Cross)</li> </ul>
Mentale Fähigkeiten zum Lernen	<ul style="list-style-type: none"> <li>• Identifikation des relevanten Wissens (Szulanski)</li> <li>• Absorptive Capacity (Szulanski, B/K)</li> <li>• Depreciation, Retentive Capacity (Ladd/Ward)</li> </ul>

#### 3.3.1 Entscheiden

In vielen Organisationen kann die tägliche Arbeit als Abfolge von einander ähnlichen, sich wiederholenden Entscheidungsaufgaben angesehen werden (Carley 1992, S. 24). Die zu lösenden Probleme sind dabei oft sehr ähnlich aber selten absolut identisch, da sich häufig einige Rahmenbedingungen und Parameter ändern. Carley führt als Beispiele hierfür die Flugkontrolle und den Börsenhandel an (S. 23f). Auch bei der Softwareerstellung müssen laufend kleinere Designentscheidungen getroffen werden, wenngleich ein Programmierer selten als klassischer Entscheidungsträger gesehen wird. Im Hinblick auf die Einflussfaktoren wird Entscheidung hier jedoch v.a. auf die Entscheidung Wissen nachzufragen, oder die Entscheidung Wissen weiterzugeben bezogen.

**Rational Choice Theory:** Die klassische Theorie rationalen Handelns geht von globaler Rationalität aus. Dies bedeutet, dass ein Entscheider uneingeschränkter Zugang zu allen Informationen hat, die für die Lösung seines Entscheidungsproblems relevant sind (Jost 2000, S. 312). Er kennt also alle Handlungsalternativen und deren Konsequenzen in Bezug auf seine Zielerreichung. Er kann weiterhin alle Informationen verarbeiten und die

Handlungsalternativen bewerten (homo oeconomicus). Wegen dieser Grundannahmen wurde das ökonomische Rationalitätsverständnis oft kritisiert (Heap 1992).

Die Bewertung von Alternativen im Modell der rationalen Entscheidung erfolgt oft mit Hilfe einer Nutzenfunktion, um auch bei konfligierenden Zielen eine Reihung der Alternativen vornehmen zu können. Der Nutzen wird dabei als Konsum von Gütern beschrieben. Die Nutzenfunktion gibt an, inwiefern Verzicht auf Konsum eines Gutes durch zusätzlichen Konsum eines anderen Gutes kompensiert werden kann. Unter einem Gut wird dabei alles verstanden, was die Bedürfnisse eines Mitarbeiters in irgendeiner Weise befriedigen kann und knapp ist (Jost 2000, S. 317). Die in der Mikroökonomie verwendete klassische Nutzenfunktion geht davon aus, dass der Nutzen aus den Erwartungswerten von Einkommen und Freizeit maximiert wird (Weise et al. 2002, S. 370):

$$(1) \max U_E = E(\text{Einkommen}) + E(\text{Freizeit}).$$

Das Einkommen setzt sich aus zwei Teilen zusammen, dem Gehalt der aktuellen Periode  $t$  (im Grundmodell als fix angenommen) und der Summe des erwarteten Gehalts in den Folgeperioden bis zum Planungshorizont  $n$ , das von der Leistung in der vorherigen Periode abhängt (streng genommen von der Leistung in allen vorherigen Perioden). Da zukünftige Auszahlungen für den Mitarbeiter bei den üblicherweise unterstellten positiven Zeitpräferenzen (Gegenwartsorientierung) weniger wert sind als gegenwärtige muss dieses Gehalt mit einem Zinssatz abdiskontiert werden. Für die Planungen werden i. d. R. verschiedene Szenarien berücksichtigt, zumindest ein optimistisches, ein pessimistisches und ein als realistisch eingeschätztes Szenario.

$$(2) E(\text{Einkommen}) = W_t + \sum_{j=t+1}^n \sum_{k=1}^m W_{j,k} (\text{Leistung}_{j-1,k}) * P_k(W_{j,k}) * (1 - i_G)^j \text{ }^{18}$$

Die Freizeit wird aus der Differenz der Gesamtzeit pro Tag und der Arbeitszeit berechnet. Zum Ermitteln des Erwartungswertes muss die erwartete Arbeitszeit mit der Wahrscheinlichkeit gewichtet werden, dass die entsprechende Arbeitszeit gearbeitet wird und mit dem internen Zinsfuß für Freizeit, der die Präferenzen des Nutzers widerspiegelt Freizeit heute gegen zukünftige Freizeit einzutauschen.

---

<sup>18</sup>  $i_G$  = interner Zinssatz für Geld,  $t$  = aktuelle Periode,  $n$  = Planungshorizont,  $j$  = Laufindex für Perioden,  $k$  = Laufindex für verschiedene Szenarien,  $m$  = Anzahl verschiedener Szenarien,  $P$  = Wahrscheinlichkeit,  $W$  = Gehalt (wage)

$$(3) E(\text{Freizeit}) = \sum_{j=t}^n \sum_{k=1}^m (24h - \text{Arbeitszeit}_{j,k}) * P(\text{Arbeitszeit}_{j,k}) * (1 - i_F)^j$$

Jost zeigt eine andere mögliche Zielfunktion für die Auswahl einer Stelle auf, die in Gleichungsform folgendermaßen lautet (Jost 2000, S. 147f).

$$(4) U = \text{Einkommen} - \text{Arbeitsbelastung} + \text{Prestige}$$

Die Arbeitsbelastung ist dabei durch die physikalischen Arbeitsbedingungen, die Arbeitszeit und die nötige Anstrengung zur Bewältigung der Arbeitsaufgabe definiert. Prestige ist durch häufige Kontakte mit Organisationsexternen, Privilegien, Ansehen der Abteilung (oder Firma) und Aufstiegsmöglichkeiten (also die Machtposition innerhalb der Organisationshierarchie) bestimmt. Jost gibt aber auch zu Bedenken, dass die individuellen Präferenzen stark differieren können, je feingranularer die Attribute werden, die den Grad der Zielerreichung angeben (ibid. S. 150). Insbesondere muss davon ausgegangen werden, dass nicht eine besonders leichte Arbeitsaufgabe angestrebt wird, sondern eine angemessen anspruchsvolle (siehe auch Abschnitt 3.3.5 zu Flow).

Für jedes einzelne Attribut kann eine Funktion angegeben werden, welche die Ausprägung der Zielvariablen mit einem individuell empfundenen Nutzen in Zusammenhang bringt. Dabei sind solche Nutzenfunktionen oft durch einen gewissen Schwellenwert gekennzeichnet unterhalb dessen kein Nutzen empfunden wird (ibid. S. 151). So muss das Einkommen beispielsweise hoch genug sein, um die üblichen Lebenshaltungskosten (Miete, Nahrung, etc.) zu decken, sonst wird die Stelle nicht als nutzenstiftend empfunden. Weiterhin sind die Nutzenfunktionen i. d. R. konkav (z.B. logarithmisch) um dem abnehmenden Grenznutzen zusätzlicher Einheiten des Gutes Rechnung zu tragen.

**Bounded Rationality:** Am Grundmodell des rationalen Entscheiders (homo oeconomicus) wurde oft Kritik geübt. Insbesondere zeigen sich bei Experimenten und empirischen Beobachtungen, dass drei Effekte gegen ein vollkommen rationales Verhalten von Entscheidungsträgern sprechen: der Framing-Effekt, die Tendenz Probleme zu vereinfachen und die Suche nach Begründungen für Entscheidungen (Rubinstein 2002, S. 14f). Der *Framing-Effekt* entsteht dadurch, dass Entscheider ihre Informationen über die Handlungsoptionen immer in einem Kontext eingebettet bekommen. Die Formulierung bei der Beschreibung von Handlungsoptionen, sowie die Einführung von irrelevanten Alternativen können sich auf die Auswahl auswirken, obwohl dies eine Verletzung der Konsistenzbedingung darstellt (siehe auch Jost 2000, S. 198). Die *Vereinfachung von Auswahlproblemen*, z.B. durch Elimination einer Reihe von Alternativen, oder die

willkürliche Berücksichtigung nur eines Teils der relevanten Bewertungskriterien verhindern ebenfalls vollkommen rationale Entscheidungen. Die *Suche nach Begründungen* stellt ein Problem dar, wenn inkonsistente Bewertungsfunktionen gewählt werden, die nur eine Teilordnung zwischen Handlungsoptionen herstellt und deswegen die Handlungsoption gewählt wird, die bei einem Vergleich der Teilordnungen am besten abschneidet.

Eine andere Erklärung stellt auf die *begrenzte kognitive Komplexität* des menschlichen Entscheiders ab, der gar nicht dazu in der Lage ist, alle Alternativen zu vergleichen und zu bewerten (Jost 2000, S. 183f). Insbesondere muss dabei das unvollständige (und evtl. veraltete) Wissen des Entscheiders, die Grenzen der sprachlichen Informationsvermittlung, der menschlichen Informationsaufnahme und -verarbeitung berücksichtigt werden. Zudem müssen zur Informationsbeschaffung und -bewertung Zeit und kognitive Kapazität aufgewendet werden, deren Kosten im Sinne von negativem Nutzen oder Opportunitätskosten ebenfalls berücksichtigt werden müssen (vgl. Jost 2000, S. 313, 318). Zu unterscheiden sind physische Kosten (z.B. Arbeitsanstrengung, kognitiver Aufwand), psychische Kosten (z.B. durch kognitive Dissonanzen) und soziale Kosten (z.B. beim Verstoß gegen ethische Normen).

Weiterhin kann das Einbeziehen von nicht ökonomischen Aspekten zu scheinbar irrationalen Verhalten führen. Berücksichtigt man dagegen soziale und moralische Ziele, so kann dieses Verhalten im Rahmen von Rationalitätskalkülen erklärt werden (Jost 2000, S. 314). Für Wissensweitergabe müssen also alle Güter berücksichtigt werden, die im Arbeitskontext relevant sind, z.B. soziales Ansehen, Macht, Entscheidungsspielraum, Karriere, Selbstverwirklichung und Wissen (ibid. S. 316f).

Der Framing-Effekt kann auch so interpretiert werden, dass rationales Maximierungsverhalten nur innerhalb eines "Frames" existiert, die Auswahl eines solchen Handlungsrahmens, der eine komplexe Situationsdefinition beinhaltet, dagegen durch Routinen gesteuert wird (Abraham 2001, S. 4). Dies führt zu so genannten *Framing-Modellen* (z.B. Tversky, Kahnemann 1986; Esser 1990).

Es können weiterhin zwei unterschiedliche Typen von Entscheidungsregeln unterschieden werden. Bei *kompensatorischen Entscheidungsregeln* werden zwei konfligierende Ziele genau gegeneinander abgewogen und eine Kombination gewählt, die den größtmöglichen Nutzen ergibt. Bei *nicht-kompensatorischen Entscheidungsregeln* wird hingegen nur die Erfüllung eines gewissen Grundniveaus hinsichtlich eines Ziels oder des Gesamtnutzens

festgelegt und die erste Alternative gewählt, die dieses Niveau erfüllt (Jost 2000, S. 328f). Dies wird in der englischsprachigen Literatur auch unter dem Stichwort *Satisficing* diskutiert (Mortensen 1988; McKenna 1985) und ist vor allem dann relevant, wenn im Rahmen von sequentiellen Suchprozessen über die Fortsetzung der Suche oder die Annahme eines Angebots entschieden werden muss (Abraham 2001, S. 3). Mitarbeiter in Organisationen arbeiten nach der Individuen-Theorie von March und Simon (siehe auch Abschnitt 3.7.1) eher nach einer satisfizierenden Strategie und wählen eine hinlänglich gute oder akzeptable Lösung anstatt der Besten (Weinert 1998, S. 558).

In der *Theorie industrieller Beziehungen* (industrial relation theory) werden vier unterschiedliche Kalküle unterschieden: instrumentell-rational (Zweckrationalität), wert-rational, emotional und traditionell (Paauwe 2004, S. 40). Die Zweckrationalität entspricht dem klassischen Ansatz des Utilitarismus und ist v.a. auf finanzielle Ziele ausgerichtet. Die Wertrationalität bringt ethische, religiöse, politische oder ästhetische Überlegungen mit in das Kalkül. Emotionale Komponenten können rationale Gründe verstärken, oder auch überlagern. Bei traditioneller Entscheidungsfindung werden keine neuen Überlegungen angestellt, sondern eine bewährte Strategie der Vergangenheit gewählt.

***Naturalistic Decision Making:*** Einen ganz anderen Ansatz verfolgen die Anhänger von Naturalistic Decision Making (NDM) Modellen. NDM geht davon aus, dass in Alltagssituationen keine rationalen Entscheidungsmodelle angewandt werden, sogar wenn die Entscheider mit dem Prinzip vertraut sind und es prinzipiell gut heißen (Norling et al. 2001, S. 217). Dies wurde auch empirisch bestätigt, wobei allerdings zu beachten ist, dass die untersuchten Personen mindestens Fortgeschrittene oder sogar Experten in ihrer Domäne waren und es sich um Domänen handelte, in denen akuter Zeitdruck vorherrscht (z.B. Krankenschwestern, Feuerwehrleute). Im Modell des wahrnehmungs-bestimmten Entscheidens von Klein erwartet der Entscheider als Nebenprodukt der Situationserkennung bestimmte Ereignisse, wogegen andere als unwahrscheinlich eingestuft werden (ibid.). Zudem wird seine Aufmerksamkeit auf bestimmte Aspekte der Situation gelenkt um eine genauere Einschätzung zu ermöglichen. Er versteht welche Ziele in der Situation erreichbar sind und welche Aktionen Erfolg versprechend sind (ibid. S. 218). Die Anzahl zu evaluierender Alternativen ist also von vornherein eingeschränkt, Klein postuliert sogar dass nur zwei Alternativen evaluiert werden (Klein 1999, S. 8). Der Entscheider selektiert die erste (oder die zuletzt erfolgreich eingesetzte) Alternative und evaluiert ihre

Erfolgswahrscheinlichkeit mittels mentaler Simulation. Ist die Erfolgswahrscheinlichkeit hoch, so wird die Alternative sofort umgesetzt. Andernfalls wird eine Alternative evaluiert.

***Situated Action und Situated Cognition:*** Ein weiterer Vorschlag zur Erklärung der menschlichen Entscheidungsfindung wird als Situated Action Theorie bezeichnet (Johnston 2001, S. 235) und kann als Bindeglied zu den Handlungstheorien des Lernens betrachtet werden (siehe Abschnitt 3.3.4). Dabei werden zwei wesentliche Änderungen gegenüber der Rational Choice Theorie eingeführt. Zum einen wird postuliert, dass Entscheidungen meist ausschließlich auf den unmittelbaren Wahrnehmungen der Situation beruhen und nicht auf ausführlichen Analyseprozessen, die das gesamte Weltbild des Entscheiders berücksichtigen. Situationen sind im Gegensatz zum Weltbild der Rational Choice Theorie Entscheider-zentrierte Sichten auf den gerade relevanten Ausschnitt der Welt (ibid.). Zum anderen wird der Umgebung des Entscheiders eine wichtigere Rolle zugeschrieben. Die Situation ändert sich in Folge der Aktionen des Entscheiders. Die Änderungen sind dabei nicht beliebig, sondern durch die Struktur der Umgebung festgelegt (ibid.).

Eng verwandt damit sind die Forschungen zu *Situated Cognition*. Law identifiziert vier Vertreter dieser Theorie, die jeweils unterschiedliche Aspekte betonen, nämlich Suchman, Greeno, Resnick und Clancey (Law 1998b). Suchman postuliert z.B., dass Pläne für das menschliche Verhalten nur insofern eine Rolle spielen, als sie vor der Handlung gemacht werden, beim Handeln selber jedoch ad-hoc auch andere Handlungen als geplant durchgeführt werden können (Law 1998b, S. 17). Weiterhin werden Pläne nach der Handlung zur Rechtfertigung der Handlung benutzt (ibid.). Neben einzelnen empirischen Bestätigungen dieser These finden sich aber auch mehrere gegensätzliche Belege (ibid. S.18ff). Clancey dagegen vertritt eine neuropsychologisch-basierte Interpretation von Situated Cognition. Der zentrale Aspekt ist dabei die fortlaufende Koordination zwischen Wahrnehmung, Handeln und Begriffsbildung, ohne dass dazu immer symbolische Repräsentationen im Gedächtnis notwendig sind (ibid.). Wissen wird als ständiger, situationsabhängiger Rekonstruktionsprozess in Form eines Regelkreises betrachtet.

### **3.3.2 Motive und Motivation**

Motivation ist der Schlüsselfaktor für das Leistungsverhalten und damit die Arbeitsleistung von Mitarbeitern. Eignung, Verfügbarkeit von Ressourcen und entsprechende Arbeitsbedingungen alleine reichen nicht aus (Drumm 2000, S. 435). Allerdings geht die Literatur heutzutage von einem positiven Menschenbild aus, in dem Arbeit nicht allein als negativer

Faktor verstanden wird, sondern auch aus dem Gefühl, etwas geschafft zu haben eine Befriedigung gezogen wird und es motivierend wirkt, wenn „der Schreibtisch freigearbeitet wurde“ (Fincham, Rhodes 2003, S. 282).

Motive sind Wertungsdispositionen, die für Individuen charakteristisch sind (Nerdinger 2001, S. 350). Situationen bieten Anreize, bestimmte Handlungen zu zeigen oder zu unterlassen. Anreize können Motive aktivieren, aktivierte Motive führen dann zu Verhalten (ibid.). Die Wechselwirkung von Anreiz und Motiven wird als Motivation bezeichnet. Mit anderen Worten ist Motivation in der Psychologie eine Sammelbezeichnung für vielerlei Prozesse und Effekte, deren gemeinsamer Kern darin besteht, dass ein Lebewesen sein Verhalten um der erwarteten Folgen willen auswählt und hinsichtlich Richtung und Energieaufwand steuert (Heckhausen 1989, S. 10).

Motivationstheorien können in Inhalts- und Prozesstheorien unterschieden werden. **Inhaltstheorien** der Motivation gehen von Taxonomien menschlicher Bedürfnisse aus und versuchen zu erklären, welche Anreize in welcher Form mit den Motiven wechselwirken (Nerdinger 2001, S. 352). Einer der bekanntesten Ansätze ist die Theorie von *Maslow*, die fünf Klassen von Motiven unterscheidet, nämlich (1) physiologische Bedürfnisse, z.B. Hunger, Schlaf, (2) Sicherheitsmotive, z.B. Schutz vor Schmerz und Tod, (3) soziale Bindungsmotive, z.B. Streben nach Liebe und sozialem Anschluss, (4) Selbstachtungsmotive, z.B. Geltung und Ansehen und (5) Selbstentfaltungsmotive, z.B. Realisierung der vorhandenen Begabungen (ibid.). Die Kritik an Maslows Modell trifft weniger die postulierten Klassen als den Anspruch der globalen Gültigkeit, die These der stufenweisen Aktivierung von Motiven und die Annahme, dass befriedigte Bedürfnisse nicht mehr motivieren, unbefriedigte aber immer (Weinert 1998, S. 145). *Alderfer* überarbeitet Maslows Theorie daher zur *ERG-Theorie*, die von nurmehr drei Klassen von Motiven ausgeht und keine stufenweise Aktivierung vorsieht. Die Klassen sind (1) Existenzbedürfnisse (E, existence), die sowohl physiologische Bedürfnisse als auch den Arbeitslohn und die Arbeitsbedingungen umfassen, (2) Beziehungsbedürfnisse (R, relatedness), die soziale Bedürfnisse nach Zugehörigkeit, aber auch Achtung und Wertschätzung umfassen und (3) Wachstumsbedürfnisse (G, growth), die das Streben nach Selbstverwirklichung und Produktivität beinhalten (Weinert 1998, S. 147f). *Alderfer* verzichtet nicht nur auf die stufenweise Aktivierung sondern erwähnt auch explizit die Möglichkeit, dass der umgekehrte Fall eintreten kann, nämlich niedrigere Bedürfnisse aktiviert werden, um die Frustration über die Blockierung höherer Ziele abzubauen (ibid.).

S. 148). *Herzberg* postuliert, dass es sog. Hygienefaktoren gibt, deren Abwesenheit demotivierend wirkt, deren Anwesenheit aber nicht automatisch zur Motivation führt, sondern eher zur Zufriedenheit mit der Situation (Weinert 1998, S. 149f). Soll ein Mitarbeiter motiviert werden, so müssen zusätzliche Faktoren gegeben sein, die Motivationsfaktoren genannt werden. Beispiele für Motivationsfaktoren sind Selbstverwirklichung und Autonomie (ibid.). Damit lässt sich z.B. erklären, warum finanzielle Anreize im WM-Umfeld meist nicht zu den gewünschten Ergebnissen führen, da sie als Hygienefaktor einzuordnen sind. Es gibt aber auch Kritik an Herzbergs *Zwei-Faktoren-Theorie*. Ein wesentlicher Punkt ist z.B. der mangelnde Einbezug von situationsspezifischen Variablen (Weinert 1998, S. 152). So ist anzunehmen, dass Mitarbeiter in Ländern der dritten Welt, wo Hunger und Krankheiten (z.B. AIDS) allgegenwärtig sind, andere Faktoren als Motivationsfaktoren einschätzen würden, als in Westeuropa, wo man wesentlich höhere Erwartungen hat.

*McClelland* stellt in seiner *Theorie der gelernten Bedürfnisse* auf einen anderen Punkt ab. Sie besagt, dass Bedürfnisse nicht angeboren, sondern kulturell erlernt sind (Weinert 1998, S. 153). Aufbauend darauf postuliert er die drei Schlüsselbedürfnisse (1) Leistungsbedürfnis, (2) Zugehörigkeitsbedürfnis und (3) Machtbedürfnis. Das Leistungsbedürfnis führt zum Streben nach ständiger Verbesserung der Leistung, das Zugehörigkeitsbedürfnis ist relevant für den Gruppenzusammenhalt, Zusammenarbeit und gegenseitige Unterstützung, während das Machtbedürfnis für Überzeugungsfähigkeit, Konkurrenzsituationen und allgemeine Bereitschaft zum Kämpfen relevant ist (ibid.). Abbildung 3-1 fasst die Inhaltstheorien der Motivation noch einmal zusammen.

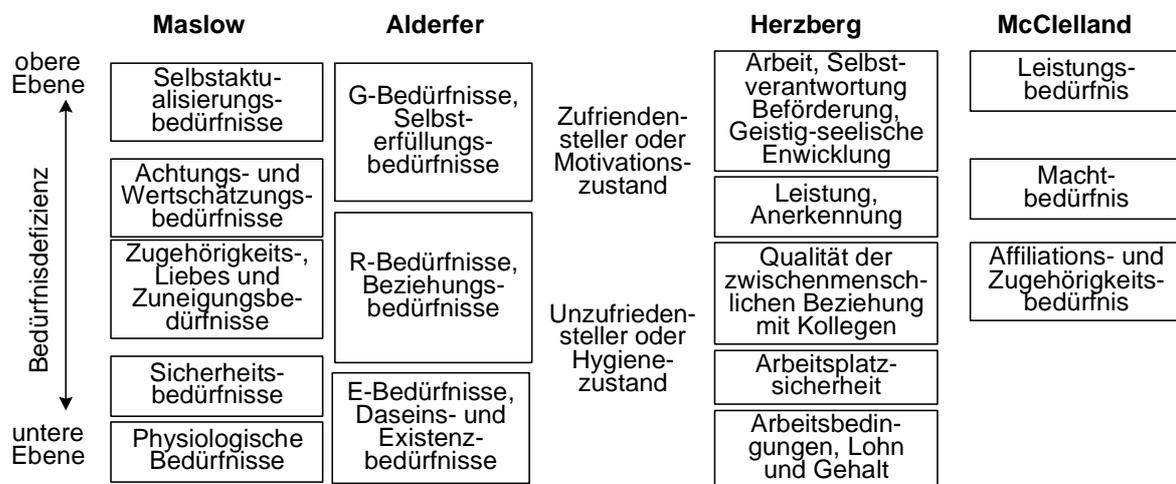


Abbildung 3-1: Inhaltstheorien der Motivation im Überblick (Weinert 1998, S. 157)

Für die Leistungsmotivation verwendet McClelland ein Erwartungs-Valenz-Modell ähnlich wie Vroom (siehe unten). Empirisch wurde McClellands Modell erfolgreich bei Führungskräften bestätigt, die hauptsächlich ein großes Machtbedürfnis haben, während Selbstständige v.a. ein hohes Leistungsbedürfnis haben, aber geringere Machtbedürfnisse, was sich negativ auf ihre Führungsqualitäten auswirkt (Weinert 1998, S. 154f). Inhaltstheorien sind aber kaum empirisch bestätigbar und nützen wenig bei der Erklärung individueller Unterschiede (Nerdinger 2001, S. 352).

**Prozesstheorien** der Motivation erklären Einflüsse der Motivation auf die erste (prädeziionale) und letzte (postaktionale) Phase der Handlung, in denen Handlungsalternativen ausgewählt und die Ergebnisse der Handlung bewertet werden (Nerdinger 2001, S. 252ff).

*Vrooms Expectancy Theorie*, basiert auf Atkinson und steht am Anfang der Entwicklung von Valenz-Instrumentalität-Erwartungs-Theorien (VIE-Theorien, Drumm 2000, S. 456ff). VIE-Theorien basieren darauf, dass Personen Erwartungen über Folgen ihrer Handlungen bilden (Nerdinger 2001, S. 354f). Diese wirken als Instrumentalität mit einer gewissen Stärke auf Motive, denen situations- und personenspezifisch ein gewisser Wert oder Nutzen, die Valenz zugeordnet wird (Weinert 1998, S. 158). Die Konzeptionalisierung ist letztlich ähnlich zur mikroökonomischen Nutzentheorie, bei der Erwartungswerte über Nutzenbeiträge einzelner Handlungsoptionen gebildet werden (siehe Abschnitt 3.3.1). Zwei Wahrscheinlichkeiten sind für die Bewertung der Anreize wichtig, die Wahrscheinlichkeit mit der eine gewisse Arbeitsanstrengung zu dem antizipierten Resultat führt (Erwartung) und die Wahrscheinlichkeit, mit der dieses Arbeitsergebnis zu dem erhofften Beitrag auf Bedürfnisebene führt (Instrumentalität). Der Mitarbeiter erwartet also z.B. dass er unter Aufwendung von zusätzlichen vier Stunden Arbeit nach Feierabend das Projekt termingerecht fertig stellen kann (Erwartung) und dafür eine Leistungsprämie bekommt (Instrumentalität). Weil sein Kontostand gerade wegen einer ungeplanten Ausgabe niedrig ist bewertet er diesen Leistungsbonus besonders hoch (Valenz). Nerdinger resümiert, dass diese Art der Betrachtung von Motivation sehr nützlich ist, es aber realistischer ist mit groben Eintrittswahrscheinlichkeiten für Handlungskonsequenzen zu rechnen (z.B. hoch, mittel, niedrig), als anzunehmen, die Mitarbeiter könnten diese auf wenige Prozent genau ermitteln (Nerdinger 2001, S. 356).

*Porter & Lawler* erweitern das Modell von Vroom um zusätzliche Komponenten, so dass ein Regelkreis von der Motivation über die Arbeitsleistung bis hin zur Zufriedenheit

entsteht (Weinert 1998, S. 161f). Abbildung 3-2 zeigt das Modell mit allen neun Variablen. Variable 1 (V1) entspricht der Valenz bei Vroom, während V2 der Instrumentalität entspricht. Die tatsächlich erhaltene Belohnung wirkt sich auf die Schätzungen in Zukunft auf. Das neue Zufriedenheitsniveau wiederum beeinflusst die subjektive Wertschätzung für eine Belohnung. Dabei wird im Allgemeinen von degressiven Zusammenhängen ausgegangen, also je höher die Zufriedenheit mit einem Bedürfnisbefriedigungsniveau, desto geringer die Motivation durch die Aussicht auf eine weitere Steigerung.

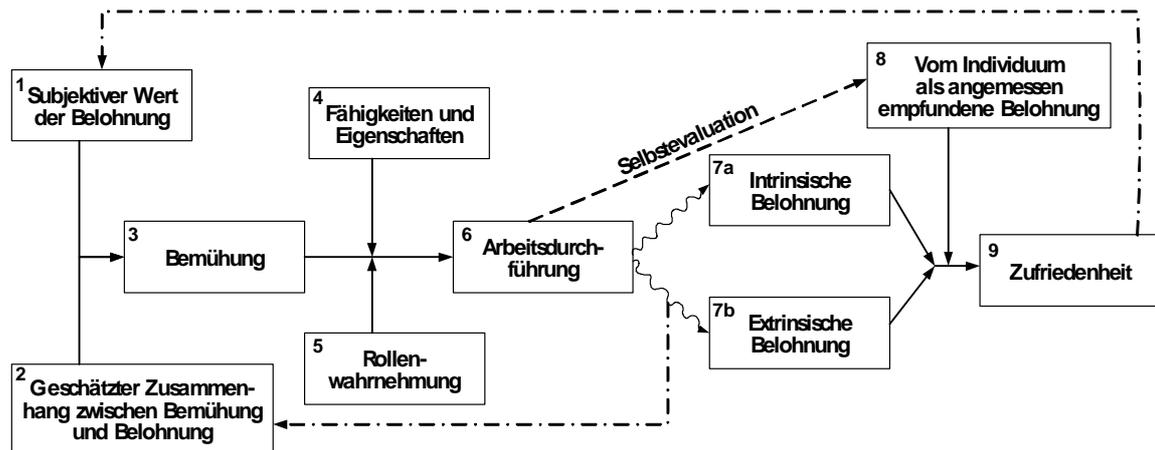


Abbildung 3-2: Motivationsmodell von Porter und Lawler nach (Weinert 1998, S. 163)

Eine weitere Kategorie von Prozesstheorien der Motivation sind die *Gleichgewichtstheorien*, die z.B. von Thibaut & Kelley, Homans, sowie Adams untersucht wurden (Drumm 2000, S. 455ff). Sie weisen große Ähnlichkeit zur Social Comparison Theorie auf (siehe Abschnitt 3.6.1) und basieren auf der Grundannahme, dass das eigene Verhalten, Leistung und die Belohnung dafür mit denen anderer Personen verglichen wird (Weinert 1998, S. 167). Dabei wird versucht ein psychologisches Gleichgewicht zu bewahren. Durch Diskrepanzen in den Kognitionen kommt es nach Festingers Theorie kognitiver Dissonanzen zu Spannungen, die als unangenehm empfunden und deshalb aufzulösen versucht werden (ibid.). Solche kognitiven Dissonanzen entstehen z.B. wenn das Verhältnis zwischen eigener Arbeitsanstrengung und der dafür erhaltenen Belohnung nicht als äquivalent zu dem einer Vergleichsperson empfunden wird. Vergleichspersonen werden dabei oft nach Geschlecht, Länge der Betriebszugehörigkeit, Organisationsebene und Ausbildung ausgewählt (Weinert 1998, S. 168). Das Bestreben nach Ausgleich der Spannung wird von Adams als Motivation interpretiert (ibid.).

Weitere Prozesstheorien der Motivation sind die SIR-Theorie von Hull und Lambert (Stimulus - intervenierende Prozesse - Response), die auf der Lerntheorie der klassischen Konditionierung (Stimulus-Response) aufbaut und als Neuerung dazu intervenierende

Prozesse in der Person vorsieht und die VIE-Theorie von Heckhausen, bei der drei verschiedene Erwartungen unterschieden werden, die Situationsvalenz (was passiert ohne mein Eingreifen), die Handlungsvalenz (Folgen eigener möglicher Handlungen) und die Ergebnisvalenz (Anreize, Drumm 2000, S. 454-64).

**Auswirkungen auf die Wissensweitergabe:** Heiss untersucht in einer Studie die Motive für die Teilnahme an Communities of Practice (Heiss 2004). Sie stellt dazu die in der Literatur genannten Motive den von ihr selbst ermittelten empirischen Ergebnissen gegenüber und kommt zu sieben Motiven, die sowohl in der Literatur als auch empirisch gefunden wurden. Dies sind Wissenserwerb & Lernen, Gerechtigkeit, Anschluss, Selbstdarstellung, Autonomie, Feedback und Sinnggebung (ibid. S. 166). Die nur empirisch ermittelten Motive sind Ähnlichkeitserleben und sozialer Vergleich, nur in der Literatur genannt wurden Anerkennung und Macht. Diese Fülle von Motiven kann nach Meinung des Autors aber noch weiter verdichtet werden.

Die **Relation Model Theory** (Fiske 1991; Fiske, Haslam 2005) kann in Bezug auf soziale Kontakte auch als Motivationstheorie aufgefasst werden und erklärt damit einige der von Heiss gefundenen Motive. Boer et al. haben die Relation Model Theory in einer empirischen (Boer, Berends 2003) und mehreren theoretischen Arbeiten (Boer et al. 2002a; Boer et al. 2002b) auf Wissensweitergabe übertragen und zeigen auf, wie sich widersprüchliche Ergebnisse früherer empirischer Arbeiten erklären lassen. Sie betonen die Situationsgebundenheit von Wissen und entsprechend auch der Wissensweitergabe in einem Kontext.

*Tabelle 3-5: Relation Model Theorie, Anreize und Organisationskultur nach Geißler und Fiske*

<b>Relation Model</b>	<b>Streben nach ...</b>	<b>Organisationskultur</b>
Communal Sharing (CS)	... Zugehörigkeit	Family culture
Authority Ranking (AR)	... Status, Macht	Law-and-Order
Equality Matching (EM)	... Wissen	Discourse
Market Pricing (MP)	... Geld	Market

Tabelle 3-5 listet die vier grundlegenden Modelle auf, nach denen soziale Beziehungen zwischen Menschen funktionieren (Fiske 1991) und ordnet sie den Organisationskulturen nach Geißler (1999) zu. Weiterhin wird zur Erklärung das zugrunde liegende Motiv mit angegeben (eigene Darstellung). Für Communal Sharing geht Fiske von einer Beziehung aus, die keine Gegenleistung erfordert, wie das z.B. in Familien der Fall ist. Man kann als Gegenleistung aber auch die Zugehörigkeit zur Gruppe auffassen. Im Authority Ranking

Modell werden soziale Beziehungen aufgebaut und aufrechterhalten um Macht und Status als Gegenleistung zu erhalten. Dies gilt bidirektional. Vorgesetzte helfen z.B. Untergebenen, um ihren Status zu festigen und Untergebene erledigen Arbeiten für Vorgesetzte um einen Teil des Ansehens des Vorgesetzten für sich selbst zu beanspruchen. Bei Equality Matching wird eine Leistung für den anderen erbracht, in der Hoffnung dieselbe Leistung in Zukunft zurück zu erhalten. Auf Wissensweitergabe angewendet wird also erhofft, dass man selber neues Wissen vom anderen erhält, wenn man ihm hilft. Im Market Pricing Modell schließlich wird monetäre Gegenleistung erwartet.

Das allgemeingültige Modell von Fiske wird von Boer et al. auf die Wissensweitergabe angewendet. Dabei entsteht das in Abbildung 3-3 gezeigte Modell, das neben dem für jede Interaktion neu wählbaren Relation Model fünf weitere bestimmende Faktoren aufzeigt. Dies sind das in der Gruppe, Organisation und der Gesellschaft vorherrschende Relation Model (implementation rules, entsprechend der Organisationskultur nach Geißler), der Typ des weitergegebenen Wissens, die Technologie zur Unterstützung der Wissensweitergabe, die Aktivität, in der das Wissen angewendet werden soll, so wie die Arbeitsteilung in der Gruppe oder Organisation.

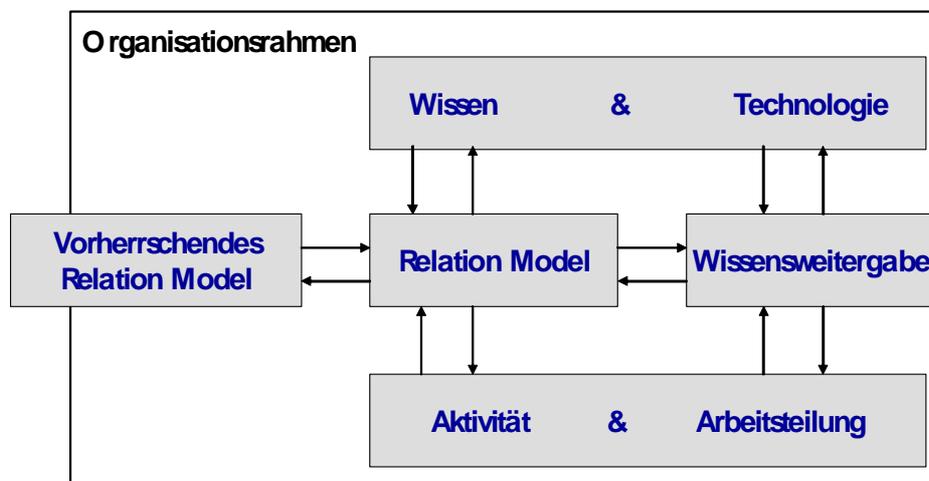


Abbildung 3-3: Konzeptuelles Modell der Einflussfaktoren für Wissensweitergabe nach (Boer et al. 2002b, S. 14)

Softwareentwickler wenden aus der Erfahrung des Autors heraus oft dieselben Modelle an, wie Boer und Berends das für die industrielle Forschung und Entwicklung berichten. Communal sharing wird aufgrund des gemeinsamen Interesses in Technologie angewendet (Boer, Berends 2003, S. 6). Weiterhin wird Equality matching angewendet, d.h. es wird Wissen an diejenigen weitergegeben, von denen man sich auch Wissen in anderen Bereichen als Gegenleistung verspricht (ibid. S. 8f). Schließlich wird Wissen

weitergegeben um als Experte anerkannt zu werden, also das Authority ranking Modell in der Expertise Variante (ARe) angewendet (ibid. S. 7).

Die von Heiss für Communities identifizierten Motive können relativ eindeutig den Beziehungsmodellen zugeordnet werden. Wissenserwerb/Lernen/Gerechtigkeit entsprechen dabei dem Modell Equality Matching, Anschluss und Ähnlichkeitserleben entsprechen dem Modell Communal Sharing. Selbstdarstellung, sozialer Vergleich und Anerkennung entsprechen dem Modell Authority Ranking in der Ausprägung Expertise (ARe), wogegen Macht als Motiv Authority Ranking in der Ausprägung formelle Autorität (ARf) darstellt (vgl. Boer, Berends 2003, S. 4f).

In einer empirisch unterlegten Analyse untersucht (Hendriks 1999) die Motive für Wissensweitergabe und den Einfluss von IT Werkzeugen auf diese Motive. Er stützt seine Argumentation vor allem auf die Theorien von Herzberg, berücksichtigt aber auch andere Theorien bei der Erarbeitung von einzelnen Motivationsfaktoren, die mit der 2-Faktoren-Theorie vereinbar sind (z.B. Maslow, Vroom, McClelland). Die untersuchten Motive sind das Gefühl, etwas geleistet zu haben, das Gefühl verantwortlich dafür zu sein, dass Wissen weitergegeben wird, die Anerkennung der Leistung, sowie Autonomie, Aufstiegsmöglichkeiten und herausfordernde Arbeitsaufgaben (ibid. S. 98).

Selbstbestimmung ist eine wichtige Determinante dafür, welche Art von Motivation erforderlich ist, damit Personen ein bestimmtes Verhalten zeigen. Nach der Selbstbestimmungstheorie von Deci und Ryan wechselt die Motivationsart mit abnehmender Selbstbestimmung von intrinsischer zu extrinsischer Motivation um schließlich bei allgemeiner Demotivation anzukommen (Guay et al. 2000, S. 176). Dass Selbstbestimmung auch für Wissensweitergabe relevant ist zeigen die Studien von Hendriks und Heiss (Hendriks 1999; Heiss 2004).

#### **3.3.3 Expertise**

Die Expertiseforschung untersucht, ob Experten Informationen anders verarbeiten und aufnehmen als Novizen und wie sich ihr Problemlöseverhalten unterscheidet. Mehrere Studien definieren Expertise als eine Kombination von Wissen und der Fähigkeit gute Resultate mit diesem Wissen zu erzeugen, meist innerhalb einer bestimmten Domäne (Ko, Dennis 2004, S. 3). Bredl stellt fest, dass es zwei verschiedene Schulen der Expertiseforschung gibt, die psychologisch geprägte und die soziologische (Bredl 2005, S. 12). „Der psychologische Ansatz auf der einen Seite beinhaltet ein kognitiv orientiertes, individualistisches Verständnis von Expertenwissen. Es definiert Expertise als eine

aufgabenspezifische Kompetenz im Problemlösen, welches eine Person befähigt, permanent eine herausragende kognitive Leistung zu bringen, wobei der Erfolg durch bestimmte Denkprozesse bestimmt wird. Der soziologische Ansatz andererseits qualifiziert eine Person nicht als Experten mit besonderen Kompetenzen oder Spezialwissen, sondern unter anderem durch die Fähigkeit, eine andere Person davon zu überzeugen, dass er im Besitz solcher bestimmter Kompetenzen ist“ (Bredl 2005, S. 12). Für die Wissensweitergabe in der Softwareentwicklung sind beide Definitionen relevant, da das psychologische Expertisemodell Erklärungsbeiträge zur Weitergabe im engeren Sinne liefert, während das soziologische Verständnis für die Anbahnung von Wissensweitergabe Erklärungen liefert. Im Folgenden wird auf die psychologische Perspektive abgestellt, während die soziologische Perspektive unter dem Stichwort Reputation in Abschnitt 3.6.4 diskutiert wird.

Die Expertiseforschung verfolgt meist einen kontrastiven Ansatz, bei dem Experten und Novizen verglichen werden (Bredl 2005, S. 13). Häufig wird nur zwischen zwei Gruppen unterschieden, deren Mitglieder oft über die Berufsjahre festgelegt werden. Es finden sich aber auch Arbeiten, die drei- (Bredl 2005), vier- (Patel, Groen 1991) und fünfstufige (Dreyfus, Dreyfus 1986) Entwicklungsprozesse differenzieren. Eine so feine Unterscheidung birgt aber auch die Gefahr, dass die Trennschärfe zwischen den einzelnen Stufen nicht gewährleistet ist (Bredl 2005, S. 13).

Das *Modell des Expertiseerwerbs* von Stuart und Hubert Dreyfus gilt als einflussreich und unterscheidet fünf Stufen, angefangen vom Novizen über den fortgeschrittenen Anfänger zum Kompetenten und danach zum gewandten Könnern, bis hin zum Experten (Dreyfus 2002, S. 2-7; vgl. auch Neuweg 1999 für die Übersetzungen der Begriffe). Das Modell differenziert die einzelnen Stufen nicht nur nach ihren Problemlösestrategien, sondern postuliert auch unterschiedliche Lernkonzepte, die für Personen auf den jeweiligen Stufen geeignet sind. Zu Beginn werden Novizen mit Regelwissen konfrontiert, das sie sich aneignen müssen und das auf eindeutig beobachtbare Aspekte der Situation abstellt (Dreyfus 2002, S. 2). Dreyfus illustriert dies am Beispiel des Schachspiels und des Autofahrens. Die Anwendung der Regeln führt aber zu unbefriedigenden Ergebnissen sobald die Situation geringfügig vom Standard abweicht (ibid.). Durch Erleben einiger realer Situationen werden für den fortgeschrittenen Anfänger zusätzliche Aspekte deutlich, die er durch Training zu erkennen lernt, die ihm aber vorher nicht zugänglich waren, weil sie schwer beschreibbar sind (implizites Wissen). Diese zusätzlichen Aspekte können in

die Instruktionen des Lehrers mit einbezogen werden und sollten in Form von Maximen die vorher gelernten Regeln ergänzen (ibid. S. 3). Das Lernen ist aber immer noch analytisch. Auf der Stufe der Kompetenten werden die zu beachtenden Aspekte immer zahlreicher, so dass ein ausschließlich analytisches Vorgehen die kognitiven Fähigkeiten übersteigt. Der Kompetente muss daher lernen, die relevanten Aspekte von den irrelevanten zu unterscheiden und dadurch die aktuelle Situation einer prototypischen zuzuordnen (ibid. S. 4). Dies kann nicht mehr durch Vorgeben von Regeln durch einen Lehrer erfolgen sondern muss vom Lerner selbst erarbeitet werden. Dreyfus betont die emotionale Komponente die in dieser Phase stark an Bedeutung gewinnt, da der Erfolg unsicher ist und somit Nervenkitzel vor und Glücksgefühle nach erfolgreichen Entscheidungen auftreten (ibid. S. 5). Diese Gefühle werden auch Teil der Situationserkennung und führen dazu, dass der geübte Könnner einen intuitiveren Zugang zu dem Problem hat. Zu dessen Lösung muss er aber immer noch bewusst entscheiden und dafür auf die Anwendung von Regeln und Maximen zurückgreifen (ibid. S. 6). Dies unterscheidet ihn vom Experten, der nicht nur die Situation intuitiv erkennt, sondern auch das, was getan werden muss. Er entscheidet nur noch darüber, wie es getan werden muss. Er ist dem geübten Könnner weiterhin dadurch überlegen, mehr Aspekte differenzieren zu können und daher die Aktion feiner darauf abzustimmen (ibid.). Dreyfus resümiert, dass das Gelernte nicht im Intellekt des Lernenden repräsentiert ist, sondern sich dem Lernenden als immer feingranularere Situationen darstellt und geht damit einen Schritt in Richtung Situated Cognition (siehe Abschnitt 3.3.1).

**Domäne:** Die Expertise besitzt ein Experte zunächst einmal nur in einer abgegrenzten Domäne (bei Dreyfus also Autofahren oder Schach). Solche Domänen können anhand ihrer Merkmale in wohl definierte und schlecht definierte Domänen unterschieden werden (Bredl 2005, S. 10). Wohl definierte Domänen zeichnen sich durch eindeutig identifizierbares Wissen, leicht messbare Expertise und allgemein anerkannte Lösungsstrategien aus (ibid.). Es scheint sich nach dem Verständnis des Autors weiterhin um eng abgrenzbare Problembereiche mit relativ unveränderlichem Wissen zu handeln. Bredl führt als Beispiele Schach, Physik und Programmieren an, wobei angemerkt werden muss, dass er auf die in der Expertiseforschung diskutierten Laborexperimente abzielt, die sich auf den stabilen Teil der physikalischen Erkenntnisse konzentrieren und überschaubare, klar definierte Programmieraufgaben betrachten.

Schlecht definierte Domänen sind dagegen dadurch gekennzeichnet, dass es keine Regeln oder Prinzipien gibt, die alle Fälle umspannen können, gleiche Merkmale in unterschiedlichen Kontexten unterschiedliches bedeuten können und multiple hierarchische Organisationsprinzipien von Merkmalskomplexen vorliegen (Bredl 2005, S. 11). Dies trifft laut Bredl auf die Unternehmensberatung zu, muss aber aus Sicht des Autors auch für die Softwareentwicklung angenommen werden, da die heutzutage zu entwickelnden Problemlösungen i. d. R. eine erhebliche Komplexität aufweisen und keine eindeutigen und allgemein anerkannten Lösungsstrategien dafür existieren.

Die Abgrenzung einer Domäne birgt auch die Schwierigkeit der Wahl des Granularitätsgrads. In der Softwareentwicklung könnte man z.B. Experten für Java-Programmierung, objektorientierte Programmierung, Programmierung allgemein oder Softwareentwicklung insgesamt definieren. Je umfangreicher die Domäne gewählt wird, desto eher wird sie als schlecht definiert gelten müssen.

**Wissen und Erfahrung:** Erfahrung, die durch rund zwei- bis vierstündige tägliche praktische Beschäftigung mit einer Domäne über 7-10 Jahre hinweg erworben wird, wird oft als Voraussetzung für Expertentum angesehen (Ericsson et al. 1993; siehe auch Bredl 2005, S. 15 der von 8-10 Jahren spricht). Mieg dagegen unterscheidet zwischen „expertise-by-experience“ und „expertise-by-knowledge“ (Mieg 2001; Bredl 2005, S. 15). Expertise durch Wissen ist besonders in den Bereichen relevant, wo Praxiserfahrung nur schwer zu erlangen ist, z.B. Archäologie oder Astronomie (Bredl 2005, S. 14). In der Softwareentwicklung ist nach Einschätzung des Autors beides relevant, Praxiserfahrung einerseits, aber auch ständige fachliche Weiterbildung um die technischen Fakten über neue Technologien zu erlernen. Da durch die schnelle technische Entwicklung selten Mitarbeiter mit Praxiserfahrung in neuen Technologien verfügbar sind, andererseits aber gerade der Einsatz dieser neuen Technologien oft zu einfacheren oder besseren Problemlösungen führt, sollten Experten durch ihre Erfahrung in verwandten (Sub-)Domänen in der Lage sein, ihr Wissen schnell auf die neue Domäne zu übertragen und auch dort effizient und effektiv zu arbeiten.

Die wesentlichen Merkmale von Expertise sind basierend auf der Literaturstudie von Bredl der Umfang des Wissens über die Domäne, sowie die Erfahrung in der Domäne. Experten wenden adäquatere Problemlösestrategien an als Novizen, besitzen ein umfangreiches Fachvokabular und ein ausgeprägtes Verständnis einzelner Begriffe, die Effizienz und

Schnelligkeit des Problemlösens oder die Flexibilität gegenüber neuen Problemstellungen sind größer (Bredl 2005, S. 14f).

(Creplet et al. 2001) sehen das Expertentum vor allem in der Fähigkeit begründet, neue Probleme zu verstehen und zu formalisieren. Experten benötigen vor allem Deutero-Lernen um dadurch neue Blickwinkel auf Probleme einnehmen zu können und so zu neuen Lösungsstrategien zu kommen (ibid. S. 1520, siehe Abschnitt 3.3.4).

Zu ähnlichen Ergebnissen kommen auch Studien, die Unterschiede von Novizen und Experten bei der Analyse komplexer Systeme untersuchen (Hmelo-Silver, Pfeffer 2004, S. 136). Jacobson fand heraus, dass Novizen einfache Ursache-Wirkungs-Zusammenhänge, zentrale Kontrolle und Vorhersagbarkeit bei ihren Erklärungen favorisierten, während Experten bei ihren Erläuterungen dezentrales Denken demonstrierten, oft mehrere Ursachen verbanden und von stochastischen Prozessen und Gleichgewichtszuständen ausgingen (Jacobson 2001).

Gobet und Wood fassen die Expertiseforschung in vier Grundprinzipien zusammen (Gobet, Wood 1999, S. 190). (1) Expertise benötigt den Aufbau einer großen Wissensbasis, die theoretische Konstrukte in Form von Chunks, Regeln und Schemata repräsentiert (siehe Abschnitt 3.3.5). (2) Es dauert lange Zeit und benötigt viel Erfahrung um ein Experte zu werden, wobei Simons 10-Jahres-Regel meist als die beste Schätzung akzeptiert wird. (3) Experten unterscheiden sich von Novizen nicht notwendigerweise in ihren grundsätzlichen Fähigkeiten zur Informationsverarbeitung oder ihrer Begabung, sondern hauptsächlich in der Zeit die sie sich aktiv kognitiv mit der Domäne auseinandergesetzt haben. (4) Expertise in einer Domäne lässt sich kaum in andere Domänen übertragen. Letzteres hängt jedoch aus Sicht des Autors stark von der Granularität der Domänendefinition ab, wie bereits oben erwähnt. Für umfangreiche Domänen trifft die Aussage sicher zu, je kleiner der Bereich gewählt wird, desto eher werden sich in verwandten Domänen Anknüpfungspunkte finden. Ein unmittelbares Expertentum in der neuen Domäne ist jedoch nicht vorstellbar.

Sonntag und Schaper nennen ähnliche und darüber hinaus noch weitere Merkmale von Experten (Sonntag, Schaper 2001, S. 244), nämlich (1) umfangreiches, und differenziertes bereichsspezifisches Wissen, (2) Nutzung komplexer Wissenseinheiten (Chunks) beim Verstehen der Problemsituation, (3) fallbasierte Organisation von Wissen, (4) eine tiefere Verarbeitung von Problemlöseanforderungen, (5) stärkere und problemadäquatere

Vernetzung von Wissensstrukturen, (6) weitergehende Prozeduralisierung von Wissen und (7) bessere Verfügbarkeit komplexer Informationsverarbeitungsstrategien.

**Auswirkungen auf die Wissensweitergabe:** Bei der Analyse komplexer Systeme konzentrieren sich Experten auf das Verhalten von aktiven und die Funktion von passiven Elementen, wogegen sich Novizen und Fortgeschrittene auf Strukturen konzentrieren (Hmelo-Silver, Pfeffer 2004, S. 136). Weiterhin tendieren Fortgeschrittene zur Beschreibung von konkreten Instanzen, während Experten auf Typebene bleiben und wichtige Eigenschaften des Typs hervorheben (ibid.).

Im Bereich der Softwareentwicklung stimmen mehrere Studien darin überein, dass Experten signifikant mehr Zeit für die Planung des Programms verwenden als Anfänger und besonders als Fortgeschrittene (Petre, Blackwell 1997, S. 111; Law 1998b, S. 25). Gleichzeitig können sie aber gegebene Probleme trotz längere Planungszeit schneller lösen und die Lösung ist oft effizienter (Law 1998b, S. 24f). Bei der Planung werden dabei auch visuelle Repräsentationen der Lösung in Gedanken erstellt und dienen als Basis für die Evaluation des Designs (Petre, Blackwell 1997, S. 119f). Bei diesen Aussagen muss jedoch beachtet werden, dass die Stufen Novize, Fortgeschrittener und Experte nicht direkt mit den Stufen 1, 3 und 5 von Dreyfus und Dreyfus übereinstimmen müssen.

Insgesamt kann festgehalten werden, dass sich Experten von Novizen in vielen Bereichen des Lernens, Erinnerens und der Anwendung von Wissen unterscheiden und daher auch von Schwierigkeiten bei der Wissensweitergabe zwischen Experten und Novizen ausgegangen werden muss. Die Anschlussfähigkeit der Wissensbasis bei der Wissensweitergabe von Experten zu Novizen ist basierend auf den dargestellten Analysen i. d. R. nicht gegeben und somit der Erfolg der Wissensweitergabe gefährdet.

#### **3.3.4 Lernen**

Lernen kann generell als Verhaltensänderung über einen gewissen Zeitraum hinweg aufgefasst werden (Schindler 2001, S. 40). Umgekehrt ist jedoch nicht jede Verhaltensänderung mit Lernen gleichzusetzen, da z.B. auch Ermüdung zu Verhaltensänderung führen kann (ibid.). Staehle weist besonders auf den reflexiven Charakter des Lernens hin und spricht von einer kritischen Auseinandersetzung mit dem eigenen Tun und Handeln (Staehle 1999, S. 913). Durch Lernprozesse wird Wissen aus der Umwelt akkumuliert und integriert.

**Behaviorismus:** Die bekanntesten und ältesten Lerntheorien werden als *behavioristische Lerntheorien* bezeichnet und können in Reiz-Reaktions-Lernen (Stimulus-Response, S-R) und Instrumentelles Lernen unterteilt werden (Edelmann 1996, S. 14; Schüppel 1997, S. 64f; Fincham, Rhodes 2003, S. 21ff). Die bekanntesten Vertreter sind

- Klassisches Konditionieren nach Pawlow und später Hull (auch Signallernen)
- Operantes Konditionieren nach Skinner
- Lernen durch Versuch und Irrtum nach Thorndike (Bekräftigungslernen, instrumentelle Konditionierung)

Sie gehen davon aus, dass durch einen Reiz ein bestimmtes Verhalten ausgelöst und durch entsprechende Belohnung oder Bestrafung des Verhaltens das zukünftige Verhalten angepasst wird. Zusammenfassend kann über Reiz-Reaktions-Versuche gesagt werden, dass die Ergebnisse aller, unter verschiedenen Bedingungen zustande gekommenen Vergleiche belegen, dass eine belohnte Verbindung immer beträchtlich verstärkt wird, bei Bestrafung dagegen nur eine geringe oder keine Schwächung eintritt (Angermeier 1994, S. 4). Verhalten, das kurz vor einem befriedigenden Zustand gezeigt wird, wird also mit höherer Wahrscheinlichkeit wiederholt (Angermeier 1994, S. 3). Die Verbindung zwischen Reiz und Reaktion wird dann gestärkt, wenn sie häufig und in kurzen Abständen auftritt (Gesetz der Übung). Ein Nachteil dieser Theorien ist, dass ausschließlich extern beobachtbare Fakten, wie Entzug von Reizquellen, Reizzuführung und Reaktion berücksichtigt werden. Interne Faktoren wie Motive und Einstellungen werden von den Untersuchungen ausgeschlossen (Stimulus-Response Modell, S-R). Daher wurden unterschiedliche Modelle entwickelt, die auch interne Variablen berücksichtigen (Stimulus-Organism-Response, S-O-R), z.B. von Hull. Das *instrumentelle Lernen* ist eine andere Weiterentwicklung des S-R Modells. Es basiert auf vier grundlegenden Bestandteilen, der Motivation, Situation, dem Verhalten und der Konsequenz (Edelmann 1996, S. 107). Basierend auf den Kombinationen aus Darbietung oder Entzug einer angenehmen oder unangenehmen Konsequenz, sowie keiner Konsequenz entstehen die vier Formen instrumentellen Lernens, positive Verstärkung, negative Verstärkung, Bestrafung und Löschung (Edelmann 1996, S. 113).

Es zeigte sich auch, dass Denkprozesse für die Erklärung des beobachteten Verhaltens mit herangezogen werden müssen, da zu bereits gelernten ähnliche Aufgaben nicht neu gelernt werden mussten, sondern sofort fehlerfrei gelöst wurden. Auch Beobachtung von Anderen kann zu schnellerem Lernerfolg und geringerer Anfangsfehlerrate führen. Lernen kann

dabei auch als Verringern von fehlerhaftem Verhalten oder Reduzieren von Ungewissheit über das richtige Verhalten interpretiert werden (Angermeier 1994, S. 10).

Tolman geht auch von Zwischenvariablen (intervening variables) aus, die er als Erwartungen formuliert. Es werden Hypothesen über Umweltreaktionen auf bestimmte Verhaltensweisen gebildet, die dann je nach tatsächlich beobachteter Umweltreaktion bestärkt oder weniger wahrscheinlich werden (ibid.). Auch Emotionen wie Frustration können zu einer Erhöhung der Intensität des gezeigten Verhaltens führt (Frustrationshypothese). Weitere Versuche zeigen, dass auch räumliche Strukturen gelernt werden (cognitive map) sowie Lernen durch Beobachten und Imitation stattfindet (ibid.).

Eine noch ungelöste Frage betrifft den zeitlichen Verlauf des Lernens. Die *Kontinuitätstheorie* besagt, dass Lernerfolg kontinuierlich ist und je nach Schwierigkeit der Aufgabe entweder leicht konvex von einem linearen Verlauf abweicht (schwierige Aufgabe, zunehmender Lernerfolg) oder leicht konkav gewölbt ist (leichte Aufgabe, abnehmender Lernerfolg). Eine konkurrierende Theorie postuliert, dass zunächst nur geringer Lernerfolg erzielt wird und nach einiger Zeit eine Einsicht in die Hintergründe entsteht, die dann im Weiteren zu einem großen Lernerfolg führt (exponentieller Kurvenverlauf). Die Empirie bestätigt beide Theorien in unterschiedlichen Umfeldern. Es scheint, dass bei größerer Erfahrung des Lernenden eher nicht-kontinuierliche Lernerfolge erzielt werden, wogegen bei geringerer Erfahrung ein kontinuierlicher Lernfortschritt erzielt wird (Angermeier 1994, S. 16). Für komplexe menschliche Aktivitäten scheinen behavioristische Theorien jedoch nur sehr bedingt geeignet, da Personen nicht direkt auf die Umgebung reagieren, sondern mit internen mentalen Modellen arbeiten (Fincham, Rhodes 2003, S. 32).

**Kognitivismus:** Besser geeignet zur Erklärung des Lernens von komplexen Aktivitäten sind *kognitive Theorien*, die Veränderungen im Verhalten auf Veränderungen in den mentalen Modellen der Personen zurückführen (Fincham, Rhodes 2003, S. 33). Ein Vertreter davon ist Gibson, der postuliert, dass zunächst die Reihe von möglichen Reizen in einer bestimmten Situation gelernt wird und anschließend eine Kategorisierung dieser Reize vorgenommen wird (ibid.). Danach werden die Zusammenhänge zwischen Reizen unterschiedlicher Sinne gelernt, so dass eine genauere Einschätzung der Situation vorgenommen werden kann (z.B. Auto fährt bergauf, Motor klingt ungewöhnlich => gewählter Gang ist zu hoch). Das *Lernen der Konstanz einiger Eigenschaften* hilft die Informationsverarbeitung zu erleichtern, da z.B. die Farben vieler Objekte als invariant

angesehen werden und dadurch wahrgenommene Farbveränderungen durch unterschiedliche Lichteinwirkung in den meisten Situationen vernachlässigt werden können (Fincham, Rhodes 2003, S. 33). Solche Konstanz kann dann auch für dynamische Prozesse gelernt werden, wie z.B. die Wirkungsweise von Maschinen, wodurch die Anzahl relevanter Reize für die Beurteilung einer Situation rapide sinkt und dadurch schneller Entscheidungen getroffen werden können (ibid. S. 34, siehe auch das Modell des Fertigkeitserwerbs nach Dreyfus und Dreyfus, Abschnitt 3.3.3).

*Kategorisierung* erfolgt durch Abstrahieren von Besonderheiten des Einzelfalls und Hervorheben gemeinsamer Eigenschaften (Edelmann 1996, S. 174). Dabei gibt es gewisse Akzeptanzgrenzen, die durch Konventionen oder subjektive Kriterien festlegen, welche Instanzen für eine Person zu einer Kategorie zugehörig sind und welche nicht mehr (ist z.B. der Wilseder Berg in der Lüneburger Heide mit 169 m Höhe als Berg zu bezeichnen oder nicht, ibid. S. 175).

Für Kategorien werden neben charakteristischen Eigenschaften oft auch prototypische Vertreter der Kategorie kodiert. Dies können sowohl typische reale Vertreter sein (z.B. Spatz für die Kategorie Vögel), als auch ideale Vertreter, die aus einer Kombination von Eigenschaften bestehen, die es in dieser Kombination real gar nicht gibt (Edelmann 1996, S. 180). Während Prototypen besonders bei Alltagsbegriffen eine wichtige Rolle spielen zeichnet Experten im professionellen Umfeld aus, dass sie auch präzise deren charakteristische Eigenschaften nennen können (Edelmann 1996, S. 182).

Drumm unterscheidet beim betrieblichen Lernen zwischen *Lernen im Lernfeld* und *Lernen im Tätigkeitsfeld* (Drumm 2000, S. 400), also Lernen, dass in eigens dafür eingerichteten Lernumgebungen mit gut passenden Beispielen und Lehrer oder Mentor stattfindet, oder Lernen während der täglichen Arbeit unmittelbar am praktischen Problem. Im Organisationsumfeld kann auch zwischen unternehmensinternen (z.B. Coaching) und -externen Lernmaßnahmen (z.B. externe Trainings) unterschieden werden (ibid. S. 400). Weiterhin sind aktive (z.B. Ausprobieren) und passive Lernprozesse (z.B. Lesen) zu unterscheiden. Letzteres wird unter dem Begriff verbales oder sprachliches Lernen in der Literatur diskutiert (Edelmann 1996, S. 201ff). Es kann auch nach dem Umfang und der Art des Lernens unterschieden werden (Schüppel 1997, S. 21ff). *Single loop learning* (Anpassungslernen) entspricht dem Lernen von Verhaltensmerkmalen. *Double loop learning* (Veränderungslernen) entspricht der Veränderung grundlegender Werte und Einstellungen. *Deutero learning* (Prozesslernen) entspricht dem Anpassen von

Lernstrategien. Weiterhin wird differenziert zwischen einer operationale Lernebene, die für die Steuerung von Handlungsrouinen wichtig ist und auf der durch Erfahrung und experimentelles Ausprobieren gelernt wird und einer konzeptionellen Lernebene, die durch Analyse komplexer Problemzusammenhänge Lösungsstrategien und Pläne generiert (Schüppel 1997, S. 70ff).

Bei sprachlichem Lernen bildet das *Regellernen nach Gagné* eine wichtige Grundlage (Edelmann 1996, S. 202ff). Dabei werden Regeln als universelle Basis menschlichen Wissens verstanden, die sowohl Attributwerte im Ontologiesinne (z.B. das Meer ist blau) als auch Regeln im engeren Sinne des Wortes umfassen (z.B. Zugvögel fliegen im Winter in den Süden). Regeln setzen Begriffe (Konzepte, Kategorien) zueinander in Beziehung und verknüpfen sie damit. Regeln können ebenso wie Kategorien hierarchisch sein (ibid. S. 204). Gagné sieht auch die Begriffsbildung (Kategorisierung), den Wissenserwerb (Regelbildung) und das Problemlösen (Anwendung der Regeln) als Hierarchie. Diese drei Stufen können auch als die ersten Stufen auf dem Weg zum Experten betrachtet werden (vgl. Abschnitt 3.3.3).

**Handlungstheorien** sind ein eigener Zweig der kognitiven Lerntheorien (Edelmann 1996, S. 8). Deren Theoriebasis werde unter dem Begriff Entscheiden in Abschnitt 3.3.1 diskutiert. *Handlungslernen* läuft in Anlehnung an das Modell der vollständigen Handlung phasenbezogen ab und muss sowohl Orientierungs-, Zielbildungs- und Planungsphasen, als auch praktische Ausführungs-, Kontroll- und Reflexionsphasen beinhalten (Sonntag, Schaper 2001, S. 245). Eine Form des Handlungslernens ist die *Imitation*, auch Modell-Lernen oder Beobachtungslernen genannt (Edelmann 1996, S. 282). Wird ein Verhalten beobachtet und als erfolgreich eingestuft, so wird es imitiert. Der Erfolg kann dabei sowohl aus der wahrgenommenen Umweltreaktion (z.B. ein erwünschtes Ergebnis tritt ein), als auch aus Attributen des Handelnden abgeleitet werden (z.B. Person erfüllt Vorbildfunktion). Zu beachten ist dabei, dass hier eine unterschiedliche Qualität von Lernen vorliegt. Grundsätzlich muss das Lernen von Verhalten (z.B. man gibt Wissen weiter, weil ein angesehener Kollege es auch tut) von Lernen von Sachverhalten unterschieden werden (z.B. eine relationale Datenbank kann unter bestimmten Bedingungen zum Speichern von XML-Dokumenten vorteilhaft sein) auch wenn es manchmal äquivalent sein kann (z.B. der Neustart eines Testsystems kann beim Debugging lange Fehlersuche ersparen).

Lernen kann auch entlang einer horizontalen (zeitlichen) Achse und einer vertikalen (hierarchischen) Achse betrachtet werden (Fincham, Rhodes 2003, S. 34f). Zeitlich gesehen führt Lernen zum Ausführen von Aktionen in der richtigen Reihenfolge und zum richtigen Zeitpunkt. Novizen sind z.B. oft damit überfordert eine Abfolge von Aktionen in einer begrenzten Zeit korrekt auszuführen, etwa beim Autofahren gleichzeitig Kuppeln, Schalten und Lenken. In Bezug auf die hierarchische Organisation führt Lernen zum Wechseln eines Verhaltensprogramms vom Executive Program zum Operating Program und zur Subroutine (ibid.). Während im Executive Program erst ein Modell entwickelt werden muss, welches Verhalten zum gewünschten Ergebnis führt wird im Operating Program festgelegt, welches Subroutinen am besten geeignet sind um die nötigen Teilschritte auszuführen. Subroutinen selbst wiederum sind so gut gelernt, dass sie erinnert und in Verhalten umgesetzt werden können, ohne Aufmerksamkeit zu beanspruchen (Fincham, Rhodes 2003, S. 35). Lernen kann innerhalb der einzelnen Stufen auch zur Verbesserung des Modells oder der Subroutinenauswahl führen.

**Aktivitätentheorie:** Dies steht auch im Einklang mit der Interpretation der Aktivitätentheorie als Lerntheorie (vgl. McAteer, Marsden 2004). Die Aktivitätentheorie ist weniger eine Theorie im Sinne eines klaren Ansatzes mit wohl definierten Aussagen, als ein Fokussieren der Interaktionen zwischen Akteuren und ihrer Umgebung, wohl wissend, dass die Aktivität soziokulturell konstruiert ist und ihr Sinn zum großen Teil im Kontext des Aktivitätensystems zu suchen ist (Thompson 2004, S. 584).

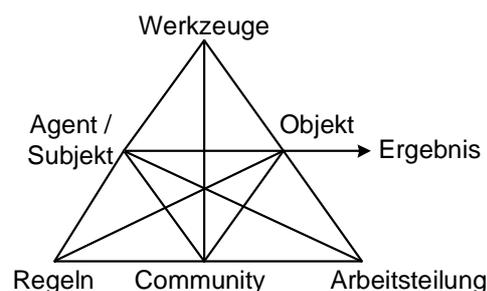


Abbildung 3-4: Elemente der Aktivitätentheorie nach (Thompson 2004, S. 585)

Abbildung 3-4 zeigt die Elemente der Aktivitätentheorie, die auf Engeström zurückgehen (Engeström 1987; Engeström 2000). Das zentrale Dreieck wird gebildet durch den Handelnden (Agent/Subjekt) der das Objekt der Aktivität transformiert, um ein Ergebnis zu erzielen und dabei durch die Aktivität in eine Gemeinschaft von Personen eingebettet ist (Community). Die Dreiecke außen verbinden jeweils zwei der zentralen Elemente über vermittelnde Elemente (mediating artifacts). Die Beziehung zwischen dem Subjekt und der

Gemeinschaft ist über Regeln festgelegt. Zur Erreichen des Ziels bedient sich das Subjekt verschiedener Werkzeuge, die sowohl technischer, als auch intellektueller Natur sein können (z.B. Methoden, Vorgehensweisen). Das Verhältnis der Gemeinschaft zum Ergebnis der Aktivität ist über die Arbeitsteilung geregelt (vgl. Thompson 2004, S. 585).

Aktivitäten bestehen aus einer Reihe von Aktionen, welche ihrerseits wiederum aus Operationen bestehen (Kuutti 1996, S. 30). Eine Aktivität kann situationsbedingt unterschiedliche Aktionen erfordern. Umgekehrt können Aktionen zu unterschiedlichen Aktivitäten gehören (ibid. S. 31). Vor dem Ausführen von Aktionen, werden typischerweise bewusst Pläne gemacht, die auf einem mentalen Modell der Aktionen aufbauen. Je besser das Modell, desto erfolgreicher die Aktion. Die Planungsphase wird Orientierung genannt (ibid.). Pläne werden aber in der Aktivitätentheorie nicht als genaue Beschreibungen des Ablaufs verstanden sondern eher im Sinne der Situated Cognition Theorie (siehe Abschnitt 3.3.1). Operationen sind wohl definierte, gewohnheitsmäßige Routinen. Durch Übung und Verbessern des Modells können Aktionen zu Operationen werden, so dass die Orientierungsphase überflüssig wird. Aber auch der umgekehrte Weg ist möglich, wenn sich die Ausführungsbedingungen ändern. Abbildung 3-5 zeigt, wie Aktivitäten zu Aktionen und Operationen werden können.

**Unstrukturierte Probleme, Exploration, Erzeugen von Wissen**

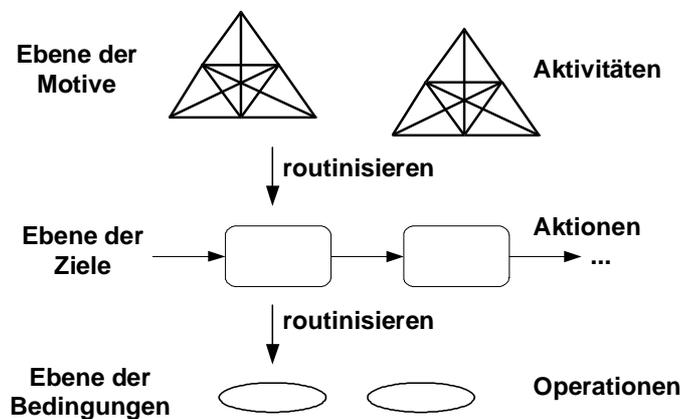


Abbildung 3-5: Zusammenhänge von Aktivitäten, Aktionen und Operationen in Anlehnung an (Hädrich, Maier 2004, S. 8)

Ein Lernmodell, das die Motive des Lernenden mit berücksichtigt, stellt Rich (2002) in seiner Dissertation vor (siehe Abbildung 3-6). Dort führen Wahrnehmungen aus der Umgebung zu dem Wunsch über bestimmtes Wissen zu verfügen. Wenn die Person dieses Wissen nicht besitzt, wird versucht, es sich über Lernen anzueignen. Wegen der begrenzten Kapazität des Gedächtnisses muss laut Rich auch veraltetes Wissen aktiv

entlernt werden, um aufnahmefähig für neues Wissen zu sein. Dieser Auffassung steht der Autor jedoch kritisch gegenüber, da aktives Vergessen kaum möglich scheint.

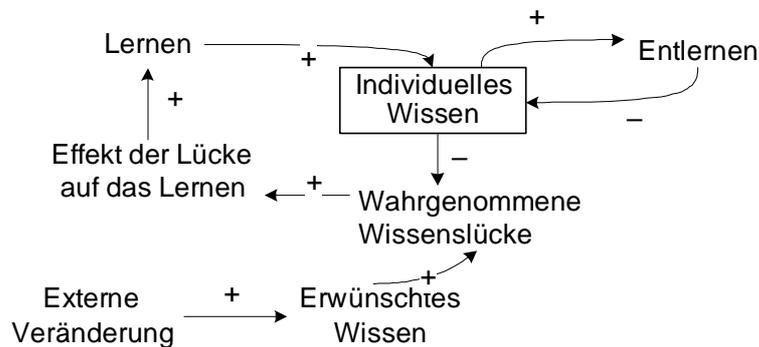


Abbildung 3-6: Zusammenhang zwischen Umgebung, Lernen und Wissen (Rich 2002, S. 5)

**Auswirkungen auf die Wissensweitergabe:** Lernen ist ein Teil des Wissensweitergabeprozesses, der für die Rekonstruktion des Wissens beim Empfänger sorgt. Definiert man Lernen als Verbessern der Effizienz oder Effektivität von Verhalten, so muss auch das Aneignen von Metawissen berücksichtigt werden. So steigt die Leistung von kleinen Gruppen signifikant, wenn durch gemeinsame Arbeitserfahrungen bekannt ist, wer über welches Wissen verfügt (Argote, Ingram 2000, S. 154).

Für die Wissensweitergabe ist besonders das informelle Lernen von Interesse. Es wird in Organisationen oft nicht als „richtiges Lernen“ akzeptiert, da es als Teil der normalen Aufgaben angesehen wird (Boud, Middleton 2003, S. 195). Die vielfach strapazierte Kaffeeküche als Ort informeller Wissensweitergabe ist nicht nur ein Mythos, sondern ebenso wie die Bedeutung eines gemeinsamen Mittagessens empirisch belegt (ibid. S. 196, siehe auch Lindvall, Rus 2003, S. 86). Entgegen der verbreiteten Ansicht, dass hauptsächlich junge neue Mitarbeiter von älteren Mitarbeitern mit langjähriger Erfahrung lernen, finden Boud und Middleton Belege dafür, dass oft junge, enthusiastische Mitarbeiter als Wissensquelle für die gesamte Gruppe dienen (ibid.). Trotz gelegentlicher Kontakte mit Organisationsexternen, die als Quelle für neues Wissen aus der Branche dienen sind die Hauptquellen für Wissen in den von Boud und Middleton untersuchten Unternehmen die Kollegen, v.a. die derselben Abteilung. Dies deckt sich auch mit der empirischen Untersuchung des Autors (siehe Abschnitt 6.1.4). Neue Mitarbeiter lernen in erster Linie von anderen Neulingen oder ihrem direkten Vorgesetzten (Boud, Middleton 2003, S. 197). Hierarchisch höhergestellte Kollegen sind Teil des Lernnetzwerks, werden aber meist nicht als erste kontaktiert, wenn Hilfe gesucht wird (ibid. S. 199).

### 3.3.5 Gedächtnis

Das Gedächtnis von Personen beruht auf Deutungen und Auslegungen ihrer Umgebung (Opwis, Plötzner 1996, S. 4f). Sie konstruieren durch Wahrnehmung, ihres Sprachverständnis und anderes Vorwissen, ihre Erwartungen und Ziele eine interne, subjektive Repräsentation einer Anwendungsdomäne (ibid.). Diese Repräsentation wird mentales Modell genannt.

**Gedächtnisbereiche:** Das Gedächtnis wird üblicherweise in Langzeit- und Kurzzeitgedächtnis aufgeteilt. Im Kurzzeitgedächtnis werden einige wenige ( $7 \pm 2$ ) Informations-Chunks (siehe unten) für ein paar Sekunden bewusst gehalten, die Informationen können aber durch Ablenkung auch leicht schneller vergessen werden (Fincham, Rhodes 2003, S. 34). Das Langzeitgedächtnis hat dagegen eine nahezu unbegrenzte Kapazität. Informationen kommen erst durch regelmäßige Wiederholungen ins Langzeitgedächtnis, werden dafür aber auch nicht so schnell wieder vergessen (Meyer, Sugiyama 2007, S. 20). Gobet und Wood gehen sogar davon aus, dass es überhaupt nicht vergessen wird, wenn es netzwerkartig mit anderem Wissen vernetzt ist, sondern nur durch neuere Erfahrungen überdeckt werden kann, so dass es später nicht mehr zugänglich ist, obwohl es prinzipiell noch im Gedächtnis verankert ist (Gobet, Wood 1999, S. 195).

Neben den beiden etablierten Gedächtnisbereichen Lang- und Kurzzeitgedächtnis gibt es auch Hinweise auf ein Langzeitarbeitgedächtnis, das 2-3 Tage weit reicht (Nuthall 2000, S. 96). Dieses ist in der Lage neue Informationen zu interpretieren, zu sortieren und in gewissem Umfang in das Vorwissen zu integrieren (ibid.). Mit fortschreitender Zeit passieren zwei Dinge. Zum einen werden durch die Vernetzung mit anderen Schemata Schlussfolgerungen gezogen und Ableitungen gemacht. Zum anderen werden Details vergessen und dafür abstraktere und generalisierte Versionen des Wissens erinnert (Nuthall 2000, S. 130).

**Chunking:** Das Konzept des Chunks bezeichnet ein Muster anderer Symbole und wird in vielen Studien als Erklärung dafür herangezogen, warum die Anzahl von Symbolgruppen im Kurzzeitgedächtnis relativ konstant ist, unabhängig davon, wie komplex die Symbolgruppen sind (Laird et al. 1986, S. 13). Auch bei Schach-Experten wurde nachgewiesen, dass sie komplexe Stellungen viel einfacher als Novizen reproduzieren können, was ebenfalls auf Chunking-Mechanismen zurückgeführt wurde (ibid.). Ein Beispiel aus der Alltagswelt ist das Lesen-Lernen bei Kindern. Zu Beginn müssen sie mühsam einzelne Buchstaben zusammensetzen, um ein Wort lesen zu können. Sind sie

geübter, so können sie ganze Wörter als zusammenhängende Einheit (Chunk) auf einen Blick erfassen und das dazu kodierte Sprachmuster abrufen, ohne sich über einzelne Buchstaben bewusst werden zu müssen. So werden auch falsch geschriebene Wörter in gewissen Grenzen problemlos erkannt. Mit zunehmender Übung können auch zusammengesetzte Wörter oder häufig benutzte Redewendungen als Einheit erkannt und effizient im Gedächtnis repräsentiert werden. Experten besitzen etwa 10.000 solcher Chunks über eine Domäne, wodurch die Chance groß ist, dass sie Wahrnehmungen innerhalb der Domäne in Zusammenhang mit diesen Chunks bringen können und darauf aufbauend erfolgreich handeln können (Gobet, Wood 1999, S. 194). Damit lässt sich erklären, warum Experten so schnell die wesentlichen Aspekte eines Problems erkennen und eine mögliche Lösung quasi direkt sehen (ibid.).

**Kodierung** bezeichnet die Aufnahme und Verankerung von Informationen im Gedächtnis. Als günstig für die Behaltensleistung hat sich eine duale Kodierung von Informationen in Sprache und Bildern erwiesen (Edelmann 1996, S. 6). Die Lernpsychologie spricht weiterhin von Assimilation, wenn neues Wissen im Vorwissen verankert wird, und vom Aufbau von Wissensnetzen, wenn Zusammenhänge zwischen Begriffen hergestellt werden (ibid.). Bruner unterscheidet drei Formen der Repräsentation von Wissen, nämlich die symbolische, bildhafte und handlungsmäßige Darstellung (Edelmann 1996, S. 221). Die symbolische Darstellung ist stark mit Sprache, also Wortklang oder Schriftzeichen verbunden. Die bildhafte Darstellung stellt auf Bilder und schematische Eindrücke ab. Die handlungsmäßige Darstellung entspricht episodischem Wissen und routinisierten Bewegungsabläufen (ibid. S. 233f).

In den letzten drei Jahrzehnten des 20. Jahrhunderts wurde viel Forschung betrieben, um die grundlegenden Gedächtnisprozesse des Menschen zu verstehen. Modelle wie Newells Unified Theory of Cognition (Newell 1990) oder Andersons ACT Modell (Anderson 1996), die das Gedächtnis als Speicher für *prozedurales* und *deklaratives Wissen* sehen, stießen auf breite Zustimmung und wurden vor allem durch Computerimplementierungen bekannt (siehe auch die Abschnitte zu SOAR und ACT-R in Kapitel 4.3). Deklaratives Wissen besteht aus Konzepten, die in Schemata vernetzt und hierarchisiert sind. Prozedurales Wissen kann aus deklarativem Wissen gewonnen werden und als Wenn-Dann-Regel dargestellt werden. Mentale Modelle von Sachverhalten bestehen aus prozeduralem und deklarativem Wissen (Sonntag, Schaper 2001, S. 243f). Prozedurales Wissen hat meist größere implizite Anteile als deklaratives (Nooteboom 2002, S. 27).

**Schemata** sind kognitive Strukturen, in denen allgemeines Wissen repräsentiert ist (Mandl et al. 1988, S. 125). Schemata, die Handlungsroutrinen abbilden, werden auch als Scripts bezeichnet (ibid.). Schemata weisen Leerstellen (slots) auf, die unterschiedliche Werte annehmen können (ibid.) und für das allgemeine Konzept, das im Schema repräsentiert wird, irrelevant sind oder deren konkrete Werte zu unterschiedlichen Subklassen des Konzepts führen. Neuere Arbeiten gehen davon aus, dass Schemata neben generischem Faktenwissen auch episodisches Wissen enthalten können (ibid.). Schemata werden bottom-up von Wahrnehmungen aktiviert (siehe auch Abschnitt 3.3.6), können dann aber ihrerseits wieder andere Schemata top-down aktivieren, was sich in bestimmten Erwartungen der Person niederschlägt (Mandl et al. 1988, S. 126). Für die Veränderung von Schemata können der Wissenszuwachs, die Feinabstimmung und die Umstrukturierung unterschieden werden (ibid.). Beim Wissenszuwachs werden bestehende Schemata an sich nicht verändert, sondern nur vorherige Leerstellen für eine bestimmte Instanz belegt oder ein weiterer Slot ergänzt (ibid. S. 127). In der Quelle nicht genannt ist der Wissenszuwachs, der zu ganz neuen Schemata führt, was jedoch ebenfalls möglich sein muss. Eine Umstrukturierung verändert ein Schema dagegen entweder durch Schemainduktion oder Mustervergleich. Schemainduktion erfolgt durch Erkennen von Zusammenhängen in Ereignisfolgen (ibid.), während Mustervergleich Konzepte eines Schemas auf ein anderes überträgt (z.B. Quadrat verhält sich zu Rechteck, wie Würfel zu Quader, ibid.). Die Neigung von Personen zur Umstrukturierung scheint jedoch gering. Versuche, die Schemata vor und nach Vermittlung von wissenschaftlichen Theorien vergleichen, zeigen, dass die Probanden oft bei ihrem vorherigen Alltagswissen bleiben, auch wenn dies in Widerspruch zu den neuen Erkenntnissen steht (Mandl et al. 1988, S. 128). Feinabstimmung betrifft kleinere Anpassungen eines Schemas, z.B. durch Präzisieren eines Defaultwertes oder Einschränken des Wertebereichs für einen Slot (ibid.). Ein **Produktionssystem** besteht aus den Produktionsregeln (prozedurales Wissen), dem Arbeitsspeicher mit deklarativem Wissen und dem Interpreter (Mandl et al. 1988, S. 136). Der Interpreter vergleicht die Bedingungen der Regeln mit den im Arbeitsspeicher vorliegenden Fakten, instantiiert Regeln, deren Bedingungen erfüllt sind, und führt sie aus. Die resultierenden Aktionen können den Inhalt des Arbeitsspeichers verändern, ergänzen oder löschen oder auch externe Handlungen anstoßen (ibid.). Beim Wissenserwerb werden zunächst neue Fakten durch Wahrnehmung erworben. Diese werden dann durch wiederholte Anwendung über Wissenskompilierung in spezifisches prozedurales Wissen

überführt und schließlich durch Generalisierung, Diskriminierung und Verstärkung verbessert (Mandl et al. 1988, S. 137).

**Mentale Modelle** sind interne Modelle der äußeren und inneren Realität (bzw. eines Ausschnitts davon) und stimmen mehr oder weniger gut mit dieser überein (Mandl et al. 1988, S. 146). Sie ermöglichen es Individuen, Schlussfolgerungen zu ziehen (Inferenz), Vorhersagen zu machen, Phänomene zu verstehen, Entscheidungen über Handlungen zu treffen und ihre Ausführung zu überwachen. Ein weiteres herausragendes Merkmal ist die Fähigkeit Ereignisse stellvertretend zu erfahren, also zu antizipieren oder deren Eintreten mental zu simulieren (ibid.). In der Softwareentwicklung ist diese Fähigkeit wichtig, da Kundenwünsche antizipiert und mögliches Benutzerverhalten bedacht werden muss, um daraus resultierenden Fehlerquellen vorzubeugen, v.a. da die dokumentierten Anforderungen aus verschiedenen Gründen selten den vollständigen Wünschen der Endbenutzer entsprechen.

Mentale Modelle sind umfassende Repräsentationen der Strukturen und Prozesse eines Realitätsausschnitts und integrieren sprachliches, bildhaftes und handlungsbezogenes Wissen (Edelmann 1996, S. 252). Auf ihnen operieren mentale Simulationen, die es z.B. erlauben, noch nicht implementierte Software im Vorfeld auf ihre Funktionalität zu testen oder die Auswirkungen von noch nicht realisierten Architekturen auf Eigenschaften der Software zu überprüfen (vgl. Petre, Blackwell 1997). Mentale Modelle sind i. d. R. bildhafte Vorstellungen, die bei Novizen meist aus oberflächlichen Eigenschaften des Realsystems (z.B. Form, Farbe), bei Experten hingegen hauptsächlich aus strukturellen und funktionalen Eigenschaften besteht (ibid.). Sie müssen v.a. im Alltag ihre Nützlichkeit beweisen, was dazu führt dass auch wissenschaftlich falsche Modelle von den Personen weiter genutzt werden, solange sie in der Praxis funktionieren (Mandl et al. 1988, S. 147).

Eine Spezialisierung der mentalen Simulation ist die *qualitative Simulation*, die für den Wissenserwerb beim Erschließen von Kausalmodellen benutzt wird (Mandl et al. 1988, S. 148). So ein Kausalmodell soll konsistent, korrespondent und robust sein, also widerspruchsfrei, dem Realsystem entsprechend und hinreichend generisch, um auch in ähnlichen Fällen angewendet werden zu können (ibid. S. 148f). Während des Lernprozesses werden oft Konsistenzverletzungen entdeckt und beseitigt, sowie implizite Annahmen, die dem Modell zugrunde liegen, expliziert und überprüft. Dadurch wird das Kausalmodell robuster (ibid. S. 149). Bei diesem Lernprozess sind Vergleiche das zentrale Mittel (Mandl et al. 1988, S. 151), wobei man die Vergleichsarten tatsächliche

Ähnlichkeit, Analogie, bloßer Anschein und abstrakte Abbildung unterscheidet, welche von Novizen und Experten unterschiedlich oft eingesetzt werden (ibid.).

**Prozesstheorien:** Dem gegenüber stehen Forschungen die belegen, dass Wissen nicht statisch speicher- und abrufbar ist, sondern es sich um einen dynamischen Prozess handelt, bei dem Wissen anhand von episodischen Erinnerungen rekonstruiert wird (Clancey 1997, S. 47ff; Cohen 1998, S. 30ff; Law 1998a, S. 38). Das Gedächtnis ist demnach kein passiver Informationsspeicher (Edelmann 1996, S. 6). Es besteht ein enger Zusammenhang zwischen der Aneignung des Wissens und dessen Abruf (ibid. S. 7). Insofern stellen Lernen und Gedächtnis keine voneinander getrennten Konzepte dar sondern legen nur unterschiedliche Schwerpunkte, beim Lernen auf das Aneignen, beim Gedächtnis auf das Speichern, Repräsentieren und Abrufen des Wissens (ibid.).

Schon 1932 wurde von Bartlett empirisch belegt, dass von einem Text der Inhalt nicht wortwörtlich erinnert wird, sondern nur der Kern des Textes, auffällige Details und die vom Leser empfundenen Emotionen in Bezug auf den Inhalt im Gedächtnis behalten werden (Edelmann 1996, S. 205f). Die Konsequenzen daraus sind nicht nur, dass Eindrücke im Nachhinein durch die erneute Interpretation bei der Rekonstruktion verfälscht werden können, was z.B. für Zeugenaussagen bei Verkehrsunfällen wichtig ist, sondern auch, dass sich die Wissensbasis durch den Erinnerungsprozess verbessern kann. Dies geht soweit, dass Clancey von einem ständigen Kreislauf von Wahrnehmung, Rekonstruktion und Interpretation der eigenen Wissensbasis vor dem Hintergrund der neuen Wahrnehmungen und dem Handeln spricht und Lernen als untrennbar mit Handeln verknüpft betrachtet (Clancey 1997, S. 3).

Die **Instanztheorie des Automatismus** (Logan 1988) besagt, dass Personen Wissen durch das Sammeln und Akkumulieren von Episoden oder Instanzen erwerben. Instanzen repräsentieren verarbeitete Episoden. Sie bestehen aus dem Ziel, das erreicht werden sollte, den Reizen, die in der Situation wahrgenommen wurden, deren Interpretation in Bezug auf das Ziel und der Verhaltensantwort auf die Reize (Logan 1988, S. 495). Dies steht in Einklang mit den Untersuchungen Abelsons, der drei Stufen des Schemaerwerbs von episodischen Repräsentationen über kategorische Verallgemeinerungen bis hin zu hypothetischen Repräsentationen führt, die komplexe Wenn-Dann-Regeln beinhalten (Mandl et al. 1988, S. 128f).

Wissen als Handlungsfähigkeit wird von einer umfangreichen Instanzbasis abgeleitet. Lernen muss daher die kognitiven Prozesse zur Transformation, Reduktion, Ausarbeitung

und Repräsentation von Informationen über diese Instanzen umfassen (Barreto, D'Eredita 2004, S. 4). Die Instanztheorie des Lernens stützt sich daher auf folgende Grundannahmen (ibid. S. 5). (1) Das Gedächtnis basiert auf Instanzen. Erfolgreiches Verhalten und die Variabilität des Erfolgs sind proportional zur Anzahl im Gedächtnis verankerter Episoden. (2) Bewusst wahrgenommene Episoden werden automatisch im Gedächtnis verarbeitet. (3) Werden entsprechende Hinweise wahrgenommen und als relevant eingestuft, so werden zugehörige Instanzen automatisch erinnert. (4) Nicht Wissen, sondern Daten und Informationen werden zwischen Personen ausgetauscht. (5) Wissen ist das Ergebnis von angeborenen kognitiven Lernprozessen, die unser Bild der Umgebung filtern und ein subjektives Bild erzeugen. Daher ist Wissen kontextabhängig und individuell.

Es gibt auch Hinweise darauf, dass die Gedächtnisrepräsentation von der Lernmethode abhängt (Nuthall 2000, S. 83), der häufige Umgang mit einem Typ von Lernmedium sich auf die Gedächtnisrepräsentation auswirkt (Issing 1988, S. 542) und umgekehrt Präferenzen für Lernmedien die Effizienz des Lernens beeinflussen (ibid.).

#### 3.3.6 Wahrnehmen

Die Wahrnehmung als Prozess der bewussten gedanklichen Verarbeitung einer Situation kann in drei Stufen unterteilt werden (Jost 2000, S. 101ff). Zuerst geschieht die *Informationsaufnahme und -selektion*. Neben den physiologischen Grenzen, die der menschlichen Wahrnehmung gesetzt sind, werden die von den Sinnesorganen aufgenommen Reize weiter gefiltert, um Informationsüberflutung zu verhindern. Diese Filterung (Aufmerksamkeit) wird dabei vom Wahrnehmungsziel, den Persönlichkeitseigenschaften des Wahrnehmenden und dem Wahrnehmungskontext mit seinen situativen Rahmenbedingungen gesteuert (ibid. S. 102). Im zweiten Schritt wird durch *Informationsverarbeitung* ein inneres Abbild aus den selektierten äußeren Reizen aufgebaut, wodurch ein sinnvoller Gesamtzusammenhang hergestellt wird. Nach dem Kontinuitäts- und dem Ähnlichkeitsprinzip wird versucht Reize in logische Reihen zu bringen oder nach gemeinsamen Charakteristika zu gruppieren (ibid. S. 103). Im dritten Schritt erfolgt die Beurteilung des inneren Abbilds anhand des Vorwissens. Da die Reize alleine oft keine schlüssigen Beurteilungen zulassen werden dazu Rückschlüsse gezogen und Vergleiche mit ähnlichen Situationen in der Vergangenheit angestellt (ibid. S. 104).

**Attribution:** Im Rahmen der Informationsverarbeitung wird auch das beobachtete Verhalten von anderen Personen beurteilt. Für die Erklärung dieses Phänomens liefert die *Attributionstheorie* einen Beitrag (Weinert 1998, S. 178f). Menschen versuchen

Ursachestrukturen in der für sie relevanten Umgebung zu identifizieren und zu verstehen (ibid.). Dabei kann beobachtetes Verhalten entweder internalen Faktoren (in der Person selbst begründet) oder externalen Faktoren (durch die Umgebung bedingt) zugeschrieben werden. Die Wahrscheinlichkeit der Zuschreibung (Attribution) von Ursachen wird laut Kelley durch die drei Konstrukte Konsens (Handeln andere Personen in der gleichen Situation genauso?), Distinktheit (Handelt dieselbe Person in dieser Situation genauso wie in anderen Situationen?) und Konsistenz (Handelt dieselbe Person über den Zeitverlauf hinweg immer genauso?) bestimmt (Weinert 1998, S. 179f). Sind alle drei Konstrukte hoch, so neigen Personen dazu das Verhalten anderer auf externe Faktoren zurückzuführen, sind Konsens und Distinktheit niedrig, so wird das Verhalten eher internalen Faktoren zugeschrieben (ibid. S. 180).

Wilkening hat die Untersuchungen zur Rolle von Wissen in der Wahrnehmung aufgearbeitet und zwei widerstreitende Strömungen identifiziert (Wilkening 1988). Die eine Strömung geht von einer direkten oder bottom-up Wahrnehmung aus und postuliert, dass die von der Umgebung ausgesandten Reize genügend Informationsgehalt (im Sinne von Shannon & Weaver) haben, um ein kognitives Konzept zu aktivieren. Die andere Strömung geht von einer indirekten oder top-down Wahrnehmung aus, in der empfangene Reize aus der Umgebung immer vor dem Hintergrund des Vorwissens interpretiert werden, was dazu führt, dass auch Dinge wahrgenommen werden können, die gar nicht vorhanden sind, sondern durch unsere Erwartungen, diese Dinge wahrzunehmen induziert sind (z.B. bei Halluzinationen, ibid. S. 223). Für beide Strömungen gibt es empirische Evidenz sowohl in der bildhaften, als auch der sprachlichen Wahrnehmung.

Insgesamt betrachtet spricht viel dafür, dass Sprachwahrnehmung sowohl top-down als auch bottom-up stattfindet, weil man ohne die bottom-up Komponente in Gesprächen oder beim Lesen nie neue Wörter lernen würde und man umgekehrt viel öfter das hören würde, was man erwartet, unabhängig davon ob es tatsächlich gesagt wurde (Wilkening 1988, S. 216). Insbesondere die Forschungen im Bereich Computer Vision haben geholfen, das Zusammenspiel der beiden Bereiche besser zu verstehen. Bei dem Versuch die menschlichen Wahrnehmungen im Computer zu simulieren stellte sich heraus, dass dies ohne Vorwissen nicht möglich ist, weil das resultierende Programm nicht ausreichend verallgemeinern kann und kaum fehlertolerant ist. Die bottom-up Wahrnehmung bildet aber die Basis (ibid. S. 221).

**Aufmerksamkeit:** Eine weitere bedeutungsvolle Komponente der Wahrnehmung ist die selektive Aufmerksamkeit. Sie besagt, dass Wahrnehmung insofern durch Vorwissen gesteuert ist, dass bekannte Muster (in jeglicher Form) schneller wahrgenommen werden und im Prozess der Wahrnehmung den Filter zwischen der unbewussten und der bewussten Wahrnehmung passieren während andere Informationen ausgefiltert werden (Cowan 1988, S. 163; Mandl et al. 1988, S. 130f). Einen empirischen Beleg dafür liefert die Forschung zum Perceptual-Defense-Effekt, die einen Zusammenhang zwischen negativ belegten Reizen und der Wahrnehmungsgeschwindigkeit festgestellt haben (Voß 2004, S. 7). Dies gilt allerdings nur so lange, wie der Reiz nicht mit unmittelbarer Lebensgefahr oder großer Angst verbunden ist. Dann zeigt sich wiederum eine schnellere Wahrnehmung, so dass sich ein umgekehrt U-förmiger Verlauf der Wahrnehmungskurve ergibt (ibid.). Kritiker halten dem entgegen, dass die negativen Reize aufgrund von top-down Verarbeitung schlechter erkannt werden, weil sie unerwartet kommen (z.B. bei Tabuwörtern, Voß 2004, S. 11). Bei diesen Versuchen wurde auch deutlich, dass die Wahrnehmung ein komplexer Prozess ist, der nur schwer analytisch von Gedächtnis und Informationsverarbeitung der erkannten Reize, oder auch der Reaktion darauf zu trennen ist (Voß 2004, S. 10).

Die Aufmerksamkeit weist auch einen zirkulären Zusammenhang mit (implizitem) Lernen auf (Jiang, Chun 2001). So kann z.B. beim Lesen von Texten ein Satz oder Wort erst dann bedeutsam werden, wenn man ihn zum wiederholten Mal in verschiedenen Aufsätzen unterschiedlicher Autoren liest, während man ihn/es beim ersten Lesen zwar wahrgenommen, aber keine Bedeutung zugemessen und daher halb bewusst gefiltert hat. Andererseits beeinflusst die Aufmerksamkeit auch, was im Gedächtnis bleibt, also das Lernen, so dass eine Art Regelkreis entsteht (ibid. S. 1106). Die durchgeführten Experimente weisen darauf hin, dass die Aufmerksamkeit nicht nur von bewussten kognitiven Mechanismen gesteuert ist, sondern auch von unbewussten, und auch nicht bewusst wahrgenommene Reize teilweise im Gedächtnis verankert bleiben (ibid. S. 1122).

**Reizverarbeitung:** Visuelle Reize werden im Wahrnehmungsprozess zunächst mit allen erkannten Merkmalen in einer parallelen Verarbeitung wahrgenommen. Die tiefere Verarbeitung wie z.B. das Zusammensetzen der Merkmale zu Objekten oder die semantische Analyse erfolgt dann jedoch seriell (Voß 2004, S. 13). Um auf wichtige Reize adäquat reagieren zu können, müssen sie aus der großen Zahl gleichzeitig wahrgenommener Reize herausgefiltert und bevorzugt in die tiefere Verarbeitung übergeben werden. Diese Selektion kann schon in der unbewussten Phase der

Wahrnehmung stattfinden (early selection) oder erst in der bewussten Phase der kognitiven Verarbeitung (late selection, *ibid.*). Auch hier sind sowohl bottom-up als auch top-down Verarbeitung relevant. Ein Beispiel, in dem die top-down Verarbeitung eine wichtige Rolle spielt, ist die sog. guided search, also eine bewusste Suche nach einem Reiz mit einem bestimmten Merkmal, z.B. ein Wort in einem Text (*ibid.*).

Weitere Experimente belegen, dass es eine automatische Verarbeitung von Reizen gibt, auch wenn diese durch bewusste Wahrnehmungssteuerung zu verhindern versucht wird. Wenn z.B. Wörter in unterschiedlichen Farben präsentiert werden und nur deren Farbe zu benennen ist, so werden doch auch die Wortinhalte interpretiert und führen dazu, dass bei einem Widerspruch (z.B. das Wort rot ist in blauer Farbe geschrieben) die Reaktionszeit signifikant steigt (Voß 2004, S. 16).

Was die Theorie der Filterung von Reizen nicht befriedigend erklären kann ist das schnelle Erkennen von Veränderungen in ausgeblendeten Reizen (Cowan 1988, S. 177). Die Gewöhnungshypothese als Variante der Theorie einer Filterung in der Mitte zwischen früher und später Selektion schließt diese Erklärungslücke indem sie postuliert, dass eine späte Filterung nur bei relativ gleichmäßigen Reizen in dem zu ignorierendem Wahrnehmungskanal möglich ist. Andernfalls wäre zu viel kognitiver Aufwand nötig, um den unerwünschten Kanal zu blockieren, so dass keine Aufmerksamkeit für den erwünschten Kanal übrig bleiben würde (Cowan 1988, S. 178). Veränderungen in den nicht bewusst wahrgenommenen Reizen ziehen jedoch sehr schnell die Aufmerksamkeit auf sich, was auch evolutionäre Ursachen haben könnte, da z.B. das rechtzeitige Erkennen von Gefahren davon abhängen kann.

#### **3.3.7 Emotion**

Emotionen wurden bislang in der Betriebswirtschaftslehre nur sehr wenig berücksichtigt, wenn es um die Erklärung des Arbeitsverhaltens von Mitarbeitern geht (Sturdy 2003, S. 82). In der Psychologie sind solche Forschungen schon länger unternommen worden und können als zusätzliche Erklärungskomponente herangezogen werden. Die Bedeutung von Emotionen für die Wissensweitergabe wird auch in der Literatur von einigen Autoren herausgestellt (z.B. Eppler, Reinhardt 2004, S. 2; Stieler-Lorenz, Paarmann 2004, S. 179).

Das Konzept der Emotionen ist umstritten, da es sehr unterschiedliche Definitionen gibt und einzelne Gefühle in unterschiedlichen Kontexten unterschiedlich gebraucht werden (Sloman 2001, S. 4). In einer ersten Annäherung können Emotionen als relativ kurzfristige positive oder negative Zustände einer Person angesehen werden, die aus neurologischen

und kognitiven Elementen bestehen und nicht vollständig kontrolliert werden können (Lawler, Thye 1999, S. 219).

Zwei grundlegende Fragen im Zusammenhang mit Emotionen sind, ob sie anhand einer kleinen Anzahl von Dimensionen beschrieben werden können, und ob es Kategorien von Emotionen gibt, die sich trennscharf abgrenzen lassen (ibid. S. 220). Russels *Circumplex Modell des Affekts* ist eine bekannte Antwort auf die erste Frage und gut empirisch bestätigt. Es ordnet Emotionen auf den Dimensionen emotionale Valenz (von Behagen bis Unbehagen) und Erregung an (niedrig bis hoch). Andererseits gibt es auch Kritiker, die entgegenhalten, dass in diesem Modell sehr unterschiedliche Emotionen (z.B. Angst und Wut) sehr dicht zusammen liegen und dass Emotionen nicht kontinuierliche, sondern diskrete Variablen sind (ibid.). Bzgl. der Kategorisierung gibt es eine Reihe von Ansätzen. Kemper fand Belege für die Unterscheidung zwischen grundlegenden Emotionen, die neurologisch (hormonell) „verdrahtet“ sind, während andere Emotionen sozial konstruiert sind (Lawler, Thye 1999, S. 221). Ansätze zur Konzeptualisierung von Emotionen lassen sich in sechs Perspektiven einteilen, die in Abbildung 3-7 gemäß ihrer Bedeutung für soziale Austauschbeziehungen dargestellt sind (vgl. Abschnitt 0).

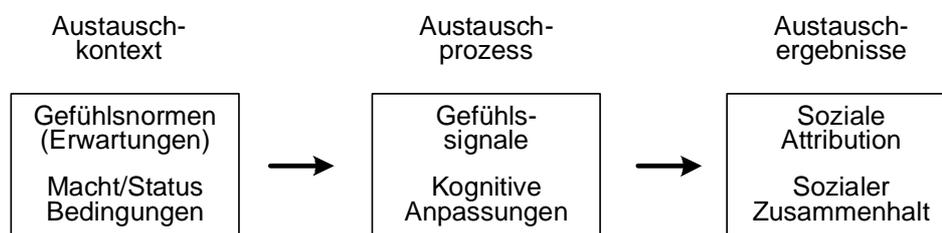


Abbildung 3-7: Emotionen in sozialen Austauschbeziehungen nach (Lawler, Thye 1999, S. 223)

Im Verständnis des *kulturell-normativen Ansatzes* wird das öffentliche Zeigen von Emotionen von sozialen Normen begrenzt (Lawler, Thye 1999, S. 224). Wenn die tatsächlich empfundenen Emotionen von den als angemessen empfundenen Emotionen abweichen, kann die Person versuchen die Gefühle zu unterdrücken (Emotions-Management, ibid.). So ist z.B. im Geschäftsumfeld üblicherweise emotionale Neutralität gefordert (ibid.). Für die Wissensweitergabe können Emotionen aber auch Belohnung sein, indem z.B. Dankbarkeit oder Hochachtung als Reaktion auf das erhaltene Wissen ausgedrückt wird (vgl. Lawler, Thye 1999, S. 225).

Die *strukturellen Beziehungsansätze* leiten bestimmte Emotionen von der sozialen Beziehung zwischen zwei Akteuren ab. Kemper postuliert z.B. dass ein Zugewinn an Macht oder Ansehen zu positiven Emotionen führt, während deren Verlust zu negativen

Emotionen führt (Lawler, Thye 1999, S. 225). Ein relativer Machtgewinn verursacht dementsprechend Sicherheitsgefühle, ein relativer Machtverlust führt zu Angst, was auch für noch nicht realisierte aber bereits antizipierte Machtveränderungen zutrifft (ibid. S. 226). Ein Gewinn an Ansehen führt zu Zufriedenheit oder Glück, während ein Verlust zu Scham, Wut oder Depression führen kann.

*Sensorisch-informatorische Ansätze* betrachten Emotionen als Informationsquelle (Signal) sowohl für die Person selber, die aus emotionalen Reaktionen Rückschlüsse über ihre Einstellung gegenüber dem Interaktionspartner ziehen kann, als auch für die Partner, die dadurch über interne Zustände der Person informiert werden (Lawler, Thye 1999, S. 228).

*Kognitive Ansätze* betrachten vor allem die Auswirkungen von Emotionen auf das Gedächtnis (vgl. Abschnitt 3.3.5). Es gilt allgemein als bewiesen, dass positive Emotionen auch positiven Einfluss auf die Erinnerungsleistung haben und Konzepte, die mit positiven Emotionen verknüpft sind, besonders leicht erinnert werden (Lawler, Thye 1999, S. 230f). Darüber hinaus werden Dinge leichter erinnert, die mit Emotionen verknüpft sind, die den aktuell empfundenen ähnlich sind (Schwarz 2000, S. 433). Andererseits können Emotionen auch die Leistung des Kurzzeitgedächtnisses negativ beeinflussen, da kognitive Leistung für die Verarbeitung der Emotionen benötigt wird und damit für andere Aufgaben nicht mehr zur Verfügung steht (Lawler, Thye 1999, S. 231). Auch Entscheidungen werden von Emotionen beeinflusst, was eine wichtige Grundlage für die Theorie der beschränkten Rationalität darstellt (ibid., siehe auch Abschnitt 3.3.1).

*Attributionsansätze* stellen darauf ab, dass den unmittelbaren Emotionen, die durch das Ergebnis von sozialen Interaktionen hervorgerufen werden, auch mittelbare Emotionen folgen, je nach dem, wer für das Ergebnis verantwortlich gemacht wird (man selbst oder die anderen). Als Quelle der Emotionen können dabei nicht nur Personen sondern auch soziale Konstrukte (Gruppe, Beziehung) oder eine Situation herangezogen werden (Lawler, Thye 1999, S. 235).

*Ansätze sozialer Formation* sehen in Emotionen primär die Grundlage für Zusammenhalt im sozialen Miteinander. Dauerhafte Beziehungen entstehen demnach über gemeinsam empfundene Gefühle (Lawler, Thye 1999, S. 236), z.B. die geteilte Freude über ein erfolgreich abgeschlossenes Projekt aber auch gemeinsam durchstandene Verzweiflung bei der Suche nach einem Programmfehler.

Allgemein können eine Reihe von Emotionen für das Verhalten und Arbeitseinsatz von Mitarbeitern in Organisationen relevant sein, z.B. Spaß, Hoffnung, Erleichterung, Stolz,

Dankbarkeit, Langeweile, Angst oder Enttäuschung (Hanne, Neu 2004, S. 8). Auch Emotionen aus dem sozialen Umfeld können für die Wissensweitergabe relevant sein, z.B. Empathie, Sympathie, Wut, Neid oder Geringschätzung (ibid.).

Die Auswahl der Aspekte einer Situation, die ausgewertet werden und zu Emotionen führen waren der Fokus vieler Studien (Staller, Petta 2001, Absatz 2.4). Mittlerweile herrscht weitgehende Übereinkunft, dass Veränderungen in der Umgebung die Aufmerksamkeit auf sich ziehen, den wahrgenommenen Stimuli ein Wert zugeordnet, deren Relevanz für eigene Ziele bewertet und der oder die Verantwortliche identifiziert wird, sowie die möglichen Aktionen zum Umgang mit dem Ereignis eingeschätzt und diese auf ihre Beziehung zu sozialen Normen evaluiert werden (ibid.). Die Bewertung der Situation führt zu Emotionen, die ihrerseits durch Aktionstendenzen, sog. Impulse, zu Aktionen führen können (Staller, Petta 2001, Absatz 2.8). Diese Impulse können teilweise unterdrückt werden, führen aber oft zur Unterbrechung der aktuellen Aktion und zu unkontrolliertem Verhalten (ibid.).

Ein weiterer Aspekt an Emotionen ist ihre Intensität. Die Intensität ist ein mehrdimensionales Konzept, was mindestens die Dauer einer Emotion, die Stärke des Impulses und die Verzögerung bis zum Ausbruch der Emotion umfasst (Staller, Petta 2001, Absatz 2.12). Die Ursachen für die Intensität einer Emotion sind genauso vielfältig wie für Emotionen selbst.

Emotionale Faktoren können auch bei der Beurteilung von vergangenen Wissensweitergabeaktivitäten eine Rolle spielen. Insbesondere beim Vergleich von realisierten Ergebnissen mit den vorher antizipierten Ergebnissen (Erwartungen) sind Enttäuschung und Bedauern relevante Emotionen (Jost 2000, S. 201f). Der *Disappointment-Effekt* tritt auf, wenn ein positives Ergebnis negativ beurteilt wird, weil ein noch besseres Ergebnis erwartet wurde. Wäre hingegen ein schlechteres Ergebnis erwartet worden, dann würde das gleiche Ergebnis positiv beurteilt. Der *Regret-Effekt* tritt auf, wenn eine Maßnahme nicht mit vollem Einsatz (z.B. Arbeitseinsatz, finanzieller Einsatz) durchgeführt wurde und das Ergebnis von anderen negativ beurteilt wird. Der Entscheider wünscht sich dann, sich für mehr Einsatz entschieden zu haben, und empfindet Bedauern wegen der entgangenen Chance.

Unter dem Stichwort *Flow* wird ein emotionaler Zustand diskutiert, bei dem durch Ausüben einer herausfordernden Tätigkeit ein Glückszustand erreicht wird

(Csikszentmihalyi 2004). Folgende Vorbedingungen und Folgen sind typisch für ein Flow-Erlebnis (siehe Tabelle 3-6):

Tabelle 3-6: Vorbedingungen und Folgen von Flow-Erlebnissen

Vorbedingungen	Folgen
Klare Ziele	Verändertes Zeitgefühl
Aufgabe entspricht den Fähigkeiten	Steigende Konzentration
Zeitnahe Rückmeldung	Gegenwartsorientierung
Beherrschen der Situation	Aussetzen des Ich-Bewusstseins

Für Wissensarbeiter allgemein und Softwareentwickler im Speziellen kann argumentiert werden, dass die Vorbedingungen meist erfüllt sind. Ohne klare Ziele wird kein Entwickler in der Lage sein ein Programm zu entwickeln. Die zeitnahe Rückmeldung ist durch Testen des Programms und entsprechende Compiler-Meldungen gegeben. Der Entwickler beherrscht die Situation, da er innerhalb seiner Aufgabe weitgehende Entscheidungsfreiheit hat. Nur bei den Fähigkeiten, die der Aufgabe entsprechen sollen, kann nicht automatisch davon ausgegangen werden, dass eine Entsprechung vorhanden ist, so dass nicht in allen Fällen die Voraussetzungen für Flow-Erlebnisse erfüllt sind. Die Entsprechung von Fähigkeiten und Aufgabe bei zugleich herausfordernder Tätigkeit ist aber ein zentraler Aspekt.

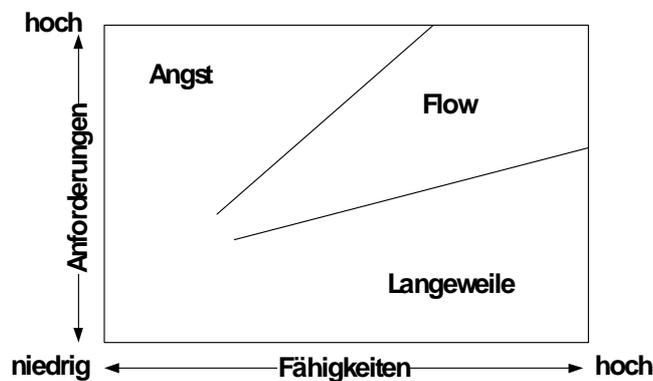


Abbildung 3-8: Zusammenhang zwischen Anforderungen, Fähigkeiten und Flow (adaptiert von Csikszentmihalyi 2004, S. 93)

Abbildung 3-8 illustriert, dass bei Unterforderung Langeweile entsteht, bei Überforderung Angst vorherrscht und nur in einem engen Bereich dazwischen Flow-Erlebnisse zustande kommen können. Daraus lässt sich schlussfolgern, dass Flow-Erlebnisse vermutlich nicht für die Wissensweitergabe direkt eine entscheidende Rolle spielen, über den Einfluss auf das Arbeitsverhalten und die Abhängigkeit von den Fähigkeiten des Mitarbeiters aber doch recht eng in Beziehung dazu stehen. Berücksichtigt man weiterhin, dass positive

Emotionen zu verbesserten Erinnerungsleistungen der damit verbundenen Sachverhalte führen und Erfolgserlebnisse auch zu Push-Gelegenheiten führen können, da über positive Erlebnisse gern berichtet wird, so scheint es dass die Wissensweitergabe indirekt sogar über mehrere Wege betroffen ist.

#### **3.3.8 Diskussion**

Bei der Diskussion der Konzepte Gedächtnis, Lernen, Wahrnehmung, Expertise, Entscheidung, Motivation und Emotion wird deutlich, dass eine enge Verbindung zwischen ihnen besteht. Insbesondere Gedächtnis, Lernen und Wahrnehmung einerseits und Entscheidung, Motivation und Emotion andererseits sind nur analytisch zu trennen, in der Praxis aber eng verbunden. Die Expertise des Individuums muss hingegen zur Unterscheidung genutzt werden, da Novizen und Experten in vielen Bereichen zu unterschiedlich sind, als dass sie einheitlich behandelt werden könnten.

Weiterhin fällt auf, dass in mehreren Bereichen die Forschung von einem statischen Verständnis des Konzepts zu einem dynamischen Prozessverständnis vorangeschritten ist. Das gilt für Gedächtnis und Wahrnehmung wie für Motivation und Emotion. Zudem sind dafür neben den Prozesstheorien, die temporale Abläufe und Zusammenhänge aufzeigen, die Inhaltstheorien gleichermaßen wichtig. Es ist nicht nur relevant, wie der Ablauf der Motivation eines Mitarbeiters ist, sondern auch welche Anreize ihn motivieren. Da dies nicht pauschal und allgemeingültig definiert werden kann, wurden die Inhaltstheorien unpopulär. Für eng abgegrenzte Domänen wie die Softwareentwicklung und speziell für die Wissensweitergabe lassen sich inhaltliche Aussagen machen und diese sind auch nötig, um ein Domänenmodell zu schaffen. Die inhaltlichen Aspekte des Gedächtnisses werden in Abschnitt 3.4 genauer behandelt. Für die Motivation der Wissensweitergabe scheinen die auf der Relation Model Theorie aufbauenden Anreize Wissen, Geld, Macht, Expertisestatus und soziale Zugehörigkeit die relevantesten zu sein, da sie sich in jede Inhaltstheorie einfügen und auch empirisch gut bestätigt sind. Bei den Emotionen ist unklar, welche für die Wissensweitergabe relevant sind. Potentiell gibt es viele Kandidaten. Dankbarkeit und Hochachtung, Enttäuschung und Bedauern, Angst und Zufriedenheit sowie viele weitere wurden in diesem Zusammenhang in der Literatur genannt. Diese können auch als direkte Motivatoren verstanden werden, entweder durch die Hoffnung auf positive Gefühle oder das Vermeiden von negativen. Das Flow-Konzept erscheint durch den direkten Bezug zur Arbeitssituation am besten geeignet.

Bei den Entscheidungstheorien wird am deutlichsten, dass mehrere Ansätze parallel für verschiedene Situationen und Expertisegrade nötig sind. In Routinesituationen (z.B. Programmieren einer Schleife) kann nicht von einer (begrenzt) rationalen Planung ausgegangen sondern muss ein direktes Erkennen der geeigneten Aktion unterstellt werden. In unbekanntem Situationen, v.a. mit großer Tragweite wie z.B. Entscheidung über die Softwarearchitektur in einem größeren Projekt, spricht jedoch trotz aller Kritik vieles für eine begrenzt rationale Entscheidung mit Enumerieren einer Reihe von Alternativen und genauem Abwägen von möglichen Vor- und Nachteilen. Zwischen diesen beiden Extremen stehen die gemäßigten Varianten der Situated Action Theorien und Framing-Modelle. In beiden Theorien wird davon ausgegangen, dass durch die Wahrnehmungen ein gewisser Rahmen abgesteckt wird, der von vornherein nur eine begrenzte Auswahl an Alternativen zulässt. Aus diesen Alternativen werden dann ein Favorit und eine Alternative nach den damit gemachten Erfahrungen gewählt. Beide werden dann mittels mentaler Simulation evaluiert. Wird ein erfolgreicher Einsatz der favorisierten Alternative antizipiert und bietet die Alternative keine gravierenden Vorteile, so wird der Favorit ausgewählt. So ein Vorgehen ist für gelegentlich auftretende Situationen mit mittlerer Wichtigkeit der Entscheidung und ausreichendem Wissen zur Lösung der Aufgabe plausibel. Fehlt jedoch das benötigte Wissen, so wird vermehrt planerisch entschieden, auch wenn der Plan anschließend nicht vollständig umgesetzt wird, weil neue Wahrnehmungen und Erkenntnisse den Plan unvorteilhaft erscheinen lassen (vgl. Law 1998a).

#### **3.4 Inhaltskontext**

Im Inhaltskontext sind die Art und verschiedenen Merkmale des weiterzugebenden Wissens relevant. Boer et al. identifizieren die Art des Wissens als relevanten Einflussfaktor Cummings & Teng sowie Becker & Knudsen die Merkmale Ausdrückbarkeit bzw. Impliztheit, Eingebettetheit, Bestätigungsstatus, Zweideutigkeit, Verbreitungsgrad und Ursprung des Wissens.

- Art des Wissens (Boer)
- Ausdrückbarkeit (Cummings/Teng), Impliztheit (B/K)
- Eingebettetheit (Cummings/Teng)
- Bestätigungsstatus (B/K) (Verbindlichkeit)
- Zweideutigkeit (B/K)
- Verbreitungsgrad (B/K)

- Ursprung des Wissens (B/K)

Die Art des Wissens wird im folgenden Abschnitt v.a. aus Sicht der Anwendungsdomäne diskutiert, während bei den Merkmalen im übernächsten Abschnitt einige domänenübergreifende Erklärungsansätze vorgestellt werden.

#### 3.4.1 Art des Wissens

Bevor man systematisch den Einfluss der Charakteristika von Wissen untersuchen kann, welche auch schon in Abschnitt 2.1.2 genannt wurden, ist es nötig zu identifizieren, welches Wissen im Softwareentwicklungsumfeld relevant ist. (Lethbridge 1999) untersucht die im Studium vermittelten Wissensbereiche und deren Relevanz für die Praxis. Befragt wurden 200 Personen die in der Softwareentwicklung in den Phasen Analyse, Entwurf, Implementierung, Test, Installation/Kundensupport und Wartung arbeiten. Die Befragten hatten zu etwa je einem Viertel weniger als 6, 6-12, 12-18 oder mehr als 18 Jahre Berufserfahrung. Damit ist die Mehrheit der Befragten nach der 10-Jahres Regel als Experte einzustufen (siehe Abschnitt 3.3.3). Abgefragt wurden 75 Wissensbereiche, die der Informatik oder sehr technischen und mathematischen Bereichen wie Differentialgleichungen und Mikroprozessorarchitektur entstammen aber auch Wissensbereiche der Wirtschaftsinformatik bis hin zu Marketing und Philosophie umfassen. Die Wissensbereiche, die von den Befragten als wichtig eingestuft wurden (auf einer Lickert-Skala von 0 bis 5 mindestens eine 3,0 im Durchschnitt) sind

- Konkrete Programmiersprachen
- Datenstrukturen
- Softwareentwurf und Entwurfsmuster
- Softwarearchitekturen
- Anforderungsanalyse
- Human Computer Interaction/  
Benutzerschnittstellen
- Objektorientierte Konzepte und  
Technologien
- Ethik und Professionalität
- Analyse- und Entwurfsmethoden
- Präsentationen vor Publikum halten
- Projektmanagement
- Testen
- Entwicklung von Algorithmen
- Technische Dokumentation
- Betriebssysteme
- Datenbanken
- Führungseigenschaften
- Konfigurations- und Release  
Management
- und Datenübertragung und Netzwerke

Von diesen 19 Wissensbereichen sind v.a. „Ethik und Professionalität“ und „Führungseigenschaften“, aber auch „Präsentationen halten“ und „Technische Dokumentation“, zu einem gewissen Teil auch „Projektmanagement“ als so genannte Softskills einzustufen, die sich schwer durch Wissensweitergabe vermitteln lassen, da sie zu einem hohen Anteil aus impliziten, schwer explizierbaren Wissen bestehen. Da die Umfrage in den USA stattfand wurde die abgefragte „Fremdsprache außer Englisch“ als wenig relevant eingestuft. Im deutschsprachigen Raum weisen jedoch einige Autoren darauf hin, dass Englischkenntnisse einen wichtigen Wissensbereich darstellen, da im IT-Bereich Englisch die dominierende Sprache ist (Balzert 2001, S. 141). Aber auch Sprachkenntnisse erscheinen für die Wissensweitergabe als wenig relevant aufgrund des starken Übungsanteils, der für den Ausbau der Kenntnisse benötigt wird, wogegen Vokabelwissen in Wörterbüchern in Buchform (z.B. Merriam Webster) oder online (z.B. dict.leo.org) leicht zugänglich ist.

In der eigenen empirischen Studie wurde die Relevanz von Branchen- und Fremdsprachenkenntnissen geringer eingeschätzt (Mittelwert 3,53 und 3,95<sup>19</sup>) als die von Programmierkenntnissen und Wissen über den Softwareentwicklungsprozess (Mittelwert jeweils 4,32).

Andere Studien größerer Firmen zeigen eine ähnliche Struktur, wenngleich sich höhere Detaillierung und deutlich mehr Skills ergeben. So wird in (Hiermann, Höfferer 2005) berichtet, dass ein IT-Unternehmen mit 600 Mitarbeitern und rund 200 Projekten pro Jahr einen Skillbaum mit 2500 Einträgen benutzt. Die Tiefe eines Astes ist maximal 4 Stufen und die oberste Ebene besteht aus den Kategorien Programmiersprachen, Datenbanken, Betriebssysteme, Transaktionsmonitore, Werkzeuge, Zertifikate, Software, Methoden, Fremdsprachen und technisches Wissen (ibid. S. 6).

Zusammenfassend lässt sich sagen, dass in der Softwareentwicklung eine Reihe von Wissensarten relevant sind. Neben direkt für die Programmierung erforderlichem Wissen (z.B. Programmiersprachen, Datenbanken, Netzwerke und Betriebssysteme) ist Konzept- und Methodenwissen (v.a. in enger Verbindung mit den Phasen eines Softwareentwicklungsvorgehensmodells, also z.B. Anforderungsanalyse oder Testen), Kenntnisse über Werkzeuge zur Unterstützung der Softwareentwicklung (z.B. integrierte Entwicklungsumgebungen), über Anwendungssysteme, zu denen eine Integration

---

<sup>19</sup> auf einer Lickert-Skala von 1=unwichtig bis 5=sehr wichtig, n=29, siehe auch Abschnitt 6.1.4

vorgenommen werden soll (z.B. ERP), sowie eher kundenorientierte Kenntnisse wie Fremdsprachen und Kenntnisse der Anwendungsdomäne wichtig. Abbildung 3-9 zeigt die ersten drei Stufen eines beispielhaften Skillbaums. Auf der dritten Stufe sind die Einträge nur exemplarisch und nicht vollständig. Für das Erheben von Skills in Unternehmen müssten einige Bereiche in einer vierten Stufe weiter verfeinert werden, auf der dann z.B. konkrete Programmiersprachen wie Java, PHP und Fortran auftauchen würden.

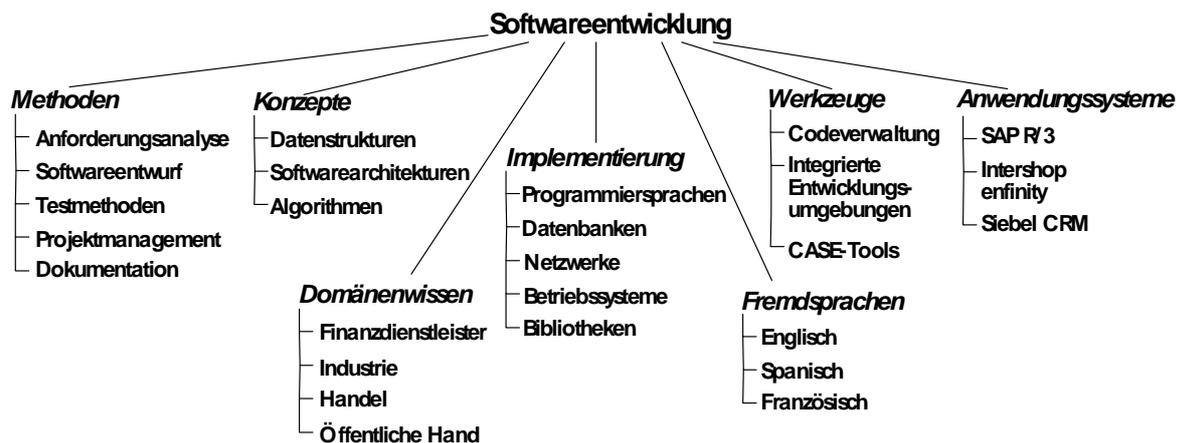


Abbildung 3-9: Beispiel für einen Skillbaum in der Softwareentwicklung (eigene Darstellung)

Eine ähnliche hierarchische Struktur mit mehr oder weniger Kategorien und Hierarchieebenen ist in vielen Firmen zu finden. In der Literatur ist bisher noch nicht diskutiert worden, ob und wenn ja in welcher Form eine Aggregation der Skillwerte auf unterer Ebene zu Kennzahlen auf höherer Ebene erfolgen kann. Eine Arbeit in der sich zumindest ein Praxisvorschlag findet ist die Dissertation von Szierpinski (Szierpinski 2007 (in Vorbereitung)). Nachdem das Erheben der Skills keine Messung sondern eine subjektive Einschätzung ist, bei der meist eine Selbsteinschätzung des Mitarbeiters mit einer Genehmigung und Absprache eines Vorgesetzten gekoppelt ist, wird durch eine Aggregation eine weitere Fehlerquelle eingeführt. Zudem dürfen die nominal skalierten Skillwerte streng genommen nicht addiert werden, da ein Wert von 4 zwar besser aber nicht zwangsläufig doppelt so gut wie ein Wert von 2 ist. Es ist auch fraglich, ob ein Mitarbeiter mit Wert 4 in Java-Programmierung genauso gut, besser oder schlechter Programmiersprachen beherrscht, als ein Mitarbeiter mit den Werten 2 in C++, Visual Basic und PHP. Man kann argumentieren, dass die Java Kenntnisse leicht auf andere Programmiersprachen übertragbar sind und deshalb der hohe Wert mehr zählt als Kenntnisse mehrerer Sprachen. Andererseits könnte man auch argumentieren, dass die Kenntnisse mehrerer Sprachen dazu beitragen, grundlegende Programmierkonzepte zu verstehen und damit ein besseres Verständnis von Programmiersprachen im Allgemeinen

anzunehmen ist. Auch die generelle Ausrichtung des Unternehmens auf Spezialisten oder Generalisten kann die Entscheidung für einen bestimmten Aggregationsmechanismus beeinflussen. Auch mathematische Formeln der Form

$$(5) \text{ Skill } A = (\text{Max}(\text{Skill } A_1-A_n) + \text{Mean}(\text{Skill } A_1-A_n \setminus A_{\text{max}}))/2$$

wären vorstellbar<sup>20</sup>.

Eine weitere Unterscheidung, die in der Literatur bislang in diesem Zusammenhang nicht diskutiert wird, ist zwischen projekt-spezifischen Wissen und allgemeinem Wissen (Peinl 2006, S. 396). Projekt-spezifisches Wissen wird zumeist direkt im Projekt generiert (z.B. Architekturvorschlag, verwendete Standards, Integrationslösungen) und kann nicht direkt auf andere Projekte übertragen werden. Nur durch Verallgemeinerung und Erfahrung hat dieses Wissen Auswirkungen auf das allgemeine Wissen. Ähnlich, aber etwas besser wieder verwendbar, ist kundenspezifisches Wissen, das zu Domänenwissen abstrahiert werden kann und in engem Zusammenhang mit projekt-spezifischem Wissen steht. Die beiden Typen von Wissen können unterschiedliche Transfermechanismen bedingen. So könnten z.B. Entwurfsentscheidungen (projekt-spezifisches Wissen) nicht ernst genug genommen werden, wenn sie nur mündlich weitergegeben werden (Walz et al. 1993, S. 72).

Das nicht projekt-spezifische Wissen wird von Robillard als deklaratives Wissen bezeichnet und kann weiter in Themenwissen (z.B. Entwurfsmuster, Namen und Funktionen von Klassen einer Bibliothek) und episodisches Wissen unterteilt werden (z.B. welcher Algorithmus in ähnlichen Situationen gut funktioniert hat, Robillard 1999, S. 87f).

### 3.4.2 Eigenschaften des Wissens

Die empirisch identifizierten Eigenschaften finden sich in den theoretischen Abhandlungen zu Wissen nur spärlich wieder. Einige diskutierte Eigenschaften kommen dem aber recht nahe und werden im Folgenden vorgestellt.

**Informationshärte:** Eine relevante Eigenschaft des Wissens für die Wissensweitergabe ist dessen „Härte“, also die Verlässlichkeit der Informationen (Watson 2003, S. 34). Watson ordnet exemplarische Informationen auf einer Skala von 1 bis 10 analog der Härteskala für Mineralien an (siehe Tabelle 3-7). Von den dort aufgeführten Informationen sind für die

---

<sup>20</sup> Die Formel bedeutet, der aggregierte Skill A wird berechnet aus dem Durchschnitt (dividiert durch zwei) der Summe aus dem Maximalwert der Subskills und dem Durchschnitt der Subskills ohne diesen Maximalwert

### 3 Erklärungsansätze für Wissensweitergabe und Arbeitsverhalten

---

Wissensweitergabe in dem Sinne, wie sie in dieser Arbeit behandelt wird, nur wenige relevant, da einige davon einen starken Daten- oder Informationscharakter haben und nicht kodifiziertes Wissen darstellen.

*Tabelle 3-7: Informationshärte nach (Watson 2003, S. 35)*

<b>Mineral</b>	<b>Härte</b>	<b>Information</b>
Talk	1	Gerüchte aus unbekanntem Quellen: Kaffeeküchentratsch, Hörensagen
Gips	2	Bekannte Quelle, Novize: Meinungen, Gefühle, Ideen
Calcit	3	Bekannte Quelle, Experte: Vorhersagen, Schätzungen
Fluorit	4	Nicht beeidetes Zeugnis: Erklärungen, Begründungen, Auswertungen, Interpretationen
Apatit	5	Eidesstattliches Zeugnis: Erklärungen, Begründungen, Auswertungen, Interpretationen
Orthoklas	6	Budgets, formelle Pläne
Quarz	7	Nachrichten, nichtfinanzielle Daten, Industriestatistiken, Fragebogendaten
Topaz	8	Nicht überprüfte Finanzdaten, Behördenstatistiken
Korund	9	Überprüfte Finanzdaten, Behördenstatistiken
Diamant	10	Börsen und Rohstoffmarktdaten

Für die Wissensweitergabe in der Softwareentwicklung kommen die Härtegrade eins bis sieben in Betracht. Eidesstattliche Zeugnisse sind z.B. verpflichtende Angebotsunterlagen oder auch Beschlüsse von Gruppenbesprechungen. Nicht beeidete Zeugnisse sind dagegen z.B. Ausarbeitungen einzelner Mitarbeiter zu einem Thema oder Inhalte im Intranet, die ohne Genehmigungsworkflow publiziert wurden.

**Organisatorische Legitimation:** In eine ähnliche Richtung wie die Informationshärte geht die organisatorische Legitimation, die angibt wie stark sich eine Organisation verpflichtet, kodifiziertes Wissen als korrekt anzuerkennen und die daraus folgenden Handlungen durchzuführen (vgl. Maier et al. 2005, S. 42f). In der Organisation selbst erzeugtes Wissen startet dabei oft in Form von Ideen einzelner Mitarbeiter, die noch keinerlei Verpflichtung auf Umsetzung seitens der Organisation besitzen. Wenn der Mitarbeiter sie aber für wichtig hält, so kann er sie als Teil des persönlichen Erfahrungsmanagements dokumentieren (z.B. in einem Weblog) und mit Kollegen diskutieren und verfeinern. Wurde die Idee dann innerhalb eines Projekts angewendet und hat funktioniert, so sollte das Projektteam die gemachten Erfahrungen in Form von Lessons learned dokumentieren (vgl. Maier et al. 2005, S. 46). In diesem Stadium besitzt das Wissen bereits ein geringes

Commitment, da einige Mitarbeiter gemeinsam die Anwendbarkeit bezeugen. Wird das Wissen dann in weiteren Projekten angewendet und verfeinert und ist auch dort erfolgreich, so kann es von der Organisationsleitung, oder zumindest hochrangigen Mitarbeitern zur Good oder Best Practice erklärt werden. Dies bedeutet eine Empfehlung, in allen ähnlichen Projekten oder Situationen das Wissen anzuwenden. Die stärkste Form der Verpflichtung besteht in der Anwendung des Wissens um die betreffenden Geschäftsprozesse der Organisation so zu ändern, dass das Wissen dort verpflichtend zur Anwendung kommt (vgl. Maier et al. 2005, S. 45). Bei diesem Knowledge Process Reengineering sollte außerdem darauf geachtet werden, dass Wissensprozesse, die sich an die veränderten Geschäftsprozesse anschließen, berücksichtigt und unterstützt werden.

In der Softwareentwicklung könnte so ein Lebenszyklus von der Idee bis zum veränderten Geschäftsprozess z.B. bei der Angebotserstellung auftreten. Zu Beginn hat ein Mitarbeiter eine Idee für eine neue Architektur, die für den Kunden einige Vorteile gegenüber den in der Vergangenheit in der Organisation oft genutzten Architekturen bietet. Er dokumentiert sie und diskutiert sie mit dem Projektteam. Evtl. wird vor der Entscheidung, das verbindliche Angebot an den Kunden zu schicken, ein Prototyp gebaut, der die Architektur testet. Mit dem Anbieten der Software mit der neuen Architektur besteht bereits eine gewisse Verpflichtung seitens der Organisation. Wird das Kundenprojekt mit der neuen Architektur erfolgreich abgeschlossen, so dokumentieren die Teammitglieder die Erfahrungen damit in einem Lessons learned Dokument. Bewährt sie sich auch in anderen Projekten so kann sie zur Best Practice erklärt werden und schließlich die neue Standardarchitektur der Organisation werden. Zusammen mit dem Commitment ist auch zu erwarten, dass der Verbreitungsgrad des Wissens steigt und mit zunehmender Anwendung eine Form der Routinisierung, also des Lernens, einsetzt (siehe auch Abschnitt 3.3.4 zur Aktivitätstheorie).

**Reifungsprozess des Wissens:** In der Arbeit (Maier, Schmidt 2007) werden die beiden Aspekte Informationshärte und organisatorische Legitimation aufgegriffen und in einem Prozess der Wissensreifung verknüpft (siehe Abbildung 3-10). Der Prozess geht auch insofern über die oben diskutierten Ansätze hinaus, als er die didaktische Aufbereitung und formelle Schulung von Mitarbeitern als klassische Themen des E-Learning mit einbezieht und als finale Phasen des Wissensreifungsprozesses betrachtet. Der Prozess beginnt mit dem Entstehen neuer Ideen, die dann in Communities diskutiert und verfeinert werden, wodurch auch eine gemeinsame Terminologie entsteht, die für die Wissensweitergabe eine

### 3 Erklärungsansätze für Wissensweitergabe und Arbeitsverhalten

wichtige Voraussetzung darstellt. In Phase drei wird das Wissen verstärkt dokumentiert und durch die Zielorientierung der Dokumente auch eine Formalisierung herbeigeführt.

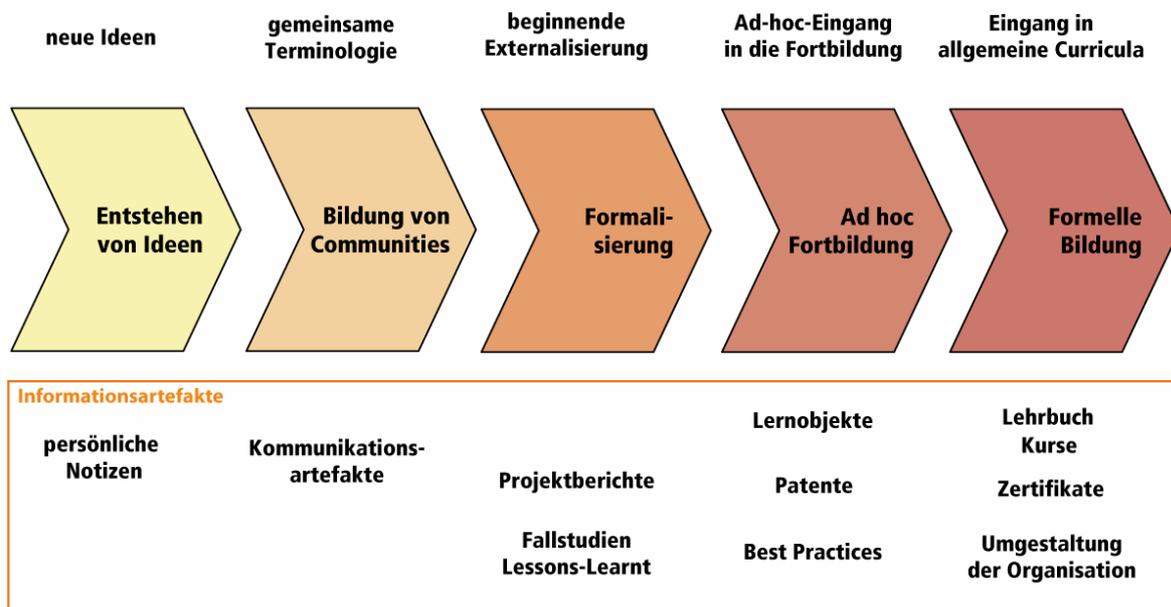


Abbildung 3-10: Wissensreifungsprozess nach (Maier, Schmidt 2007)

Ist das Wissen weit genug gereift, so sollte es in Phase vier auch in Weiterbildungsmaßnahmen berücksichtigt werden. Zunächst nur sporadisch und ad-hoc, gegen Ende des Reifungsprozesses schon systematisch bis das zu Beginn neue Wissen fester Bestandteil des Curriculums wird. Im Laufe des Prozesses entstehen auch Informationsartefakte, angefangen von persönlichen Notizen, über Weblog Einträge und Projektberichten bis hin zu Lernobjekten und Kapiteln in Schulungsunterlagen.

**Information property rights:** Bei der Wissensweitergabe gibt die Wissensquelle auch einen Teil ihrer Kontrolle über das Wissen ab, insbesondere wenn es sich nicht um persönliche Weitergabe zwischen Personen handelt sondern um elektronisch medierte Weitergabe, bei der das Wissen in kodifizierter Form der gesamten Organisation zur Verfügung steht (ähnlich auch bei Homburg, Meijer 2001, S. 1). Die Theorie der Informationseigentumsrechte (information property rights) liefert dazu einen Erklärungsbeitrag. Sie postuliert, dass die aus Organisationssicht günstigere zentrale Kontrolle über Informationsressourcen aus Sicht der Wissensweitergabe ungünstiger sein kann, da die Mitarbeiter zusätzliche Anreize für die Weitergabe ihres Wissens benötigen, wenn sie die Eigentumsrechte vollständig aufgeben müssen (ibid. S. 4). Dabei sind die Eigentumsrechte nicht im rechtlichen Sinne zu verstehen, sondern im faktischen Sinne der Kontrolle über das Wissen (ibid. S. 3). Unterschieden werden dabei die Rechte zur

Nutzung (usus), zur Rekonstruktion (abusus) und Ausbeutung des Wissens (usus fructus, ibid. S. 3). Die Eigentumsrechte wiegen umso schwerer, je mehr Investition in das Wissen gemacht wurde (z.B. in Form von Training) und je exklusiver das Wissen des Mitarbeiters ist (ibid.). Maier und Schmidt schlagen als Maßnahme gegen den Kontrollverlust vor, die Autorenschaft stärker sichtbar zu machen (Maier, Schmidt 2007, S. 8).

**Auswirkungen auf die Wissensweitergabe:** Für die Wissensweitergabe sind der Härtegrad und die organisatorische Legitimation Faktoren, die Vertrauen in die Glaubwürdigkeit der Quelle erhöhen können. Auch wenn die Person, die das Wissen weitergibt kein großes Vertrauen beim Empfänger besitzt, so kann doch der Hinweis darauf, dass es sich bei dem Wissen um im Projektteam einheitlich verabschiedete Erfahrungen handelt das Vertrauen in das Wissen an sich stärken. Der Reifungsprozess ist aus Sicht der Wissensweitergabe insbesondere dort relevant, wo Übergänge zwischen den Phasen stattfinden, weil insbesondere dafür Wissensweitergabeaktivitäten nötig sind. Es ist zu erwarten, dass in den verschiedenen Phasen auch verschiedene Weitergabemechanismen relevant sind und auch unterschiedliche Anreize vorhanden sein müssen. Die Eigentumsrechte an persönlichem Wissen hängen ebenfalls eng mit Vertrauen zusammen (siehe Abschnitt 3.6.5). Vertraut die Wissensquelle darauf, dass der Empfänger das Wissen nicht gegen sie verwendet, z.B. in dem er eine Idee als seine eigene ausgibt, so ist sie leichter bereit das Wissen weiterzugeben. Da bei anonymen elektronischen Speichern dieses Vertrauen fehlt, muss über entsprechende Anreize eine Kompensation für die Aufgabe des Eigentumsrechts geschaffen werden. Bei dezentral kontrollierten Informationsspeichern (z.B. Gruppennetzlaufwerken) ist der subjektive Kontroll- und damit Eigentumsverlust wesentlich geringer, so dass schon bei geringen Anreizen die Bereitschaft zur Wissensweitergabe vorhanden ist.

#### **3.4.3 Diskussion**

Für die Wissensweitergabe spielen die Art und die Eigenschaften des weitergegebenen Wissens eine wichtige Rolle. Nicht jede Art von Wissen lässt sich gleich gut und mit den gleichen Mechanismen weitergeben. Domänenwissen hat oft episodischen Charakter und kann in Form von Geschichten (Stories) über abgeschlossenen Projekte in der Domäne übermittelt werden, wogegen Methoden und Konzepte des Softwareengineering sowie Programmierkenntnisse aus deklarativem und prozeduralen Wissen bestehen, was zwar auch expliziert aber nur durch zusätzliche praktische Übung tief greifend erlernt werden kann. Werkzeuge und Anwendungssysteme werden ebenfalls durch eigene Erfahrung am

besten erlernt, wobei der Wissensweitergabe außerhalb von expliziten Schulungsmaßnahmen (z.B. interne Trainings) v.a. die Rolle der Effizienzsteigerung und Problemlösung zukommt. Kleine Hinweise, die zwischen Mitarbeitern ausgetauscht werden und oft durch Demonstration bei gleichzeitigen Erklärungen weitergegeben werden, helfen umständlichere Abläufe zu vereinfachen (z.B. durch Erklären von Tastaturkürzeln oder Symbolen, die tief in Untermenüs versteckte Befehle direkt aktivieren). Auch Artefakte wie Makros, die ebenfalls als eine Form kodifizierten Wissens angesehen werden können, spielen hier eine Rolle.

Härtegrad und Commitment wirken im Gegensatz dazu eher subtil auf Seite des Empfängers. Je nach Härtegrad und Commitment variiert die Glaubwürdigkeit und damit die Bereitschaft des Empfängers das weitergegebene Wissen in der Evaluationsphase als richtig einzuschätzen und seine Handlungen darauf zu gründen. Die Information property rights Theorie ist hingegen hauptsächlich für die Quelle relevant. Ist mit der Weitergabe eine weit reichende Aufgabe von Eigentumsrechten verbunden, so müssen zusätzliche Anreize vorhanden sein, um die Weitergabe zu ermöglichen.

### **3.5 Übermittlungskontext**

Hat die Quelle sich für die Weitergabe entschieden und das Wissen expliziert, so muss das explizierte Wissen an den Empfänger übermittelt werden. Dazu kommt in erster Linie eine mündliche Übermittlung in Frage, aber auch schriftliche papierbasierte oder elektronische Übermittlung, sowie (unterstützende) Gesten oder Diagramme als Übermittlungsform sind zu berücksichtigen. In der Empirie wurden dementsprechend auch kommunikations- und technologiebezogene Einflussfaktoren identifiziert.

- Kommunikationshäufigkeit (Sarker/Joshi)
- Anzahl der Transferaktivitäten (Cummings/Teng)
- Technologie (Boer, B/K)
- Existenz und mediale Reichhaltigkeit eines Kommunikationskanals (B/K)

#### **3.5.1 Kommunikation**

Mehrere Autoren stimmen darin überein, dass Kommunikation „den wechselseitigen Austausch von Daten, Nachrichten oder Informationen zwischen zwei Personen, einer Person und einer Maschine (Mensch-Maschine-Kommunikation) oder zwischen

Maschinen (Computer-Kommunikation)“ darstellt (Lehner et al. 1995, S. 142; ähnlich auch bei Reichwald 1993, S. 2174; Krcmar 2000, S. 165).

Aus betriebswirtschaftlicher Sicht wird Kommunikation als ein Prozess angesehen, der die betriebliche Koordination einer arbeitsteiligen Unternehmung ermöglicht. „Unter Kommunikation soll ein Prozess verstanden werden, bei dem Informationen zum Zwecke der aufgabenbezogenen Verständigung ausgetauscht werden.“ (Reichwald 1993, S. 450). Kienle definiert Kommunikation als „durch Zeichen vermittelte Interaktion, wobei auf gemeinsamen Kontext Bezug genommen wird“ (Kienle 2003, S. 20) und versteht unter Interaktion menschliche Handlungen, die wechselseitig aufeinander bezogen und intentional sind (ibid. S. 17).

Wenn über Kommunikation gesprochen wird, so ist im Bereich der Wirtschaftsinformatik oft die *Kommunikationstheorie* (Shannon, Weaver 1969) das zuerst genannte Konstrukt. Für die Analyse der Kommunikation von expliziertem Wissen ist sie aber ungeeignet, da sie von der festen Kopplung technischer Systeme ausgeht, die physikalisch miteinander verbunden sind (Herrmann, Kienle 2004, S. 51). Demgegenüber werden bei der menschlichen Kommunikation Ausdrücke hervorgebracht, die von außen wahrnehmbar sind und so dem Empfänger ermöglichen, das Gemeinte zu rekonstruieren (ibid. S. 52). Ausdrücke können dabei z.B. aus Textelementen, durch Schall übertragenen Worten, Gestik und Mimik bestehen (ibid.). Diese Auffassung folgt weitgehend dem „pragmatisches Kommunikationsmodell“ (Watzlawick et al. 2000), das den Beziehungsaspekt der Kommunikation hervorhebt, welcher das zwischenmenschliche Verhältnis wiedergibt. Ein anderes Modell, das auf der Kommunikationstheorie von Shannon und Weaver einerseits und der Semiotik andererseits aufbaut (siehe Abschnitt 2.1.1), stammt von Reichwaldt (siehe Abbildung 3-11).

Voraussetzung für eine erfolgreiche Kommunikation bei Reichwald ist, dass „die Kommunikationspartner über einen gemeinsamen Zeichenvorrat verfügen, die gleichen Verbindungsregeln kennen [...] und einheitliche Begriffszuordnung vornehmen“ (Reichwald 1993, S. 2176). Darüber hinaus kann man bei der Kommunikation den Inhalts- und den Beziehungsaspekt unterscheiden. Der Beziehungsaspekt bildet die Basis für die Interpretation einer Nachricht und ist oft in nonverbalen Zeichen enthalten. „Ob eine bestimmte Aussage ernst zu nehmen oder aber ironisch gemeint ist, kann häufig erst entschieden werden, sofern der Empfänger die parallelen nonverbalen Zeichen korrekt decodiert“ (Seiwert 1992, S. 1112). Daher kann der Inhaltsaspekt vergleichsweise einfach

in digitaler Form, der Beziehungsaspekt aber oft nur in analoger Form übertragen werden, wodurch der Mediatisierung von Kommunikationsprozessen wesentliche Grenzen gesetzt sind (Reichwald 1993, S. 2178).

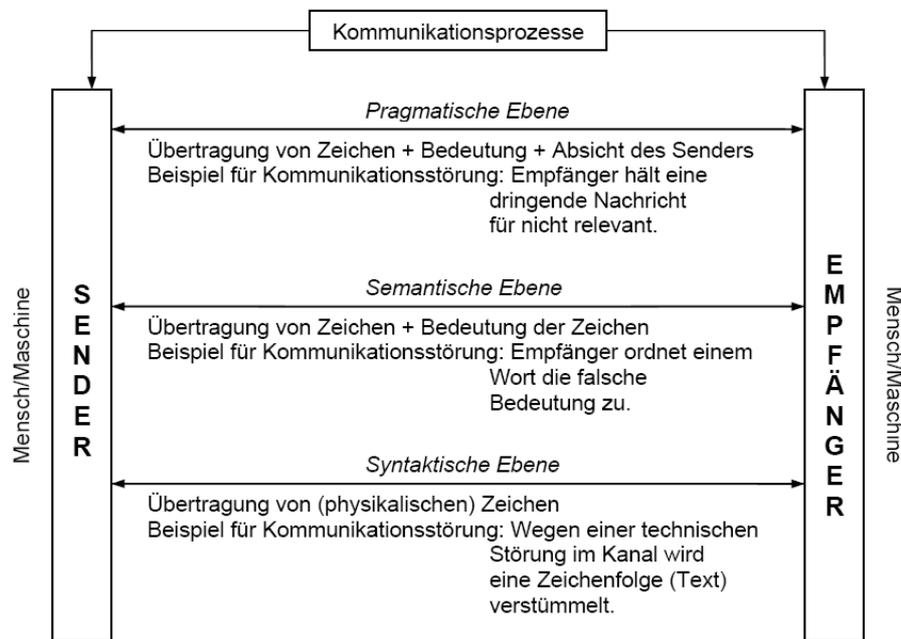


Abbildung 3-11: Kommunikationsmodell nach (Reichwald 1993, S. 2176)

Eine Erweiterung dieses Modells stellt das *Vier-Seiten-Modell* dar. Dort wird der Nachricht neben dem Sachinhalt und der Beziehung des Senders zum Empfänger auch eine Selbstoffenbarung (Mitteilung des Senders über sich selbst enthält) und ein Appell zugeschrieben, durch den der Sender den Empfänger zu einer Handlung veranlassen will (Edelmann 1996, S. 10)

Neben diesen Modellen, die den Fokus auf die Nachricht und die Beziehung zwischen den Kommunikationspartnern legen, gibt es auch Modelle, welche die Prozesse auf Sender und Empfängerseite genauer untersuchen. Abbildung 3-12 zeigt so ein Prozessmodell der Kommunikation, das die einzelnen Aktivitäten und deren Reihenfolge darstellt.

Nachdem der Sender A eine Vorstellung davon entwickelt hat, was er dem Empfänger B mitteilen möchte, erstellt er zuerst ein Kommunikationskonzept, das die Mittel beinhaltet wie er die Idee kommunizieren will. Dann wird der Ausdruck erzeugt, der vom Empfänger B wahrgenommen werden kann und der einen Eindruck bei B erzeugt. Daraus entwickelt B eine Idee. Der innere Kontext spielt dabei in nahezu allen Phasen eine wichtige Rolle (Herrmann, Kienle 2004, S. 57).

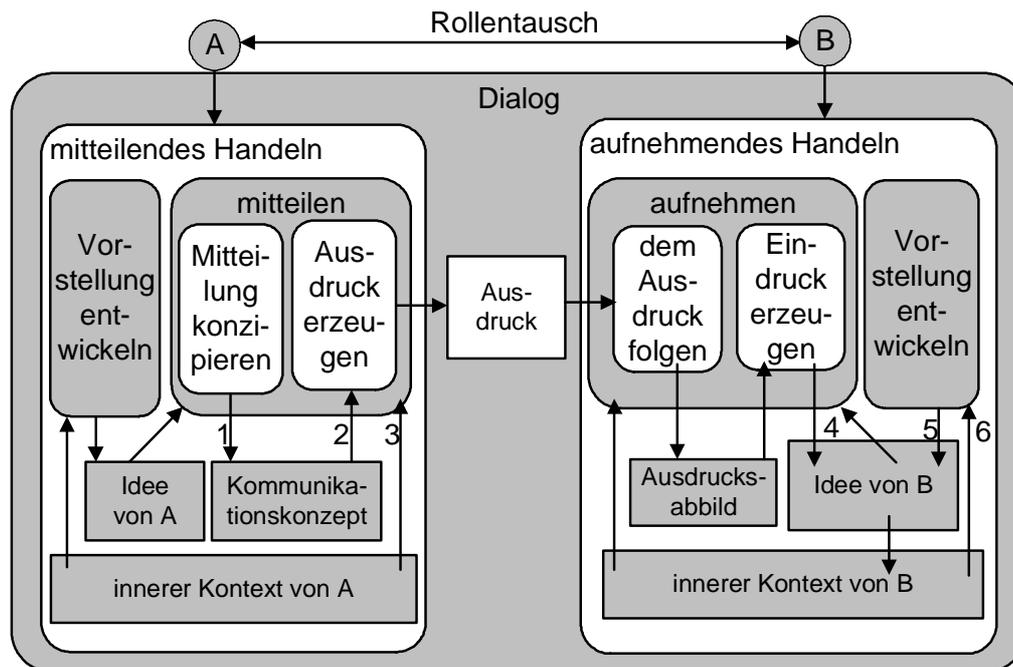


Abbildung 3-12: Kommunikationsmodell nach (Herrmann, Kienle 2004, S. 57)

Es sei hier explizit darauf hingewiesen, dass bei der persönlichen Kommunikation oft Feedbackschleifen auftreten werden (vgl. Herrmann, Kienle 2004, S. 59; siehe auch das ähnliche Modell in Gebert 1992, S. 1111, welches die Antwort explizit darstellt). Das explizierte Wissen fließt also nicht nur von der Quelle zum Empfänger, sondern der Empfänger gibt immer wieder Rückmeldung, um der Quelle Gelegenheit zu geben, die Vollständigkeit und Richtigkeit der Rekonstruktion des Wissens auf Empfängerseite zu kontrollieren, oder vereinfacht ausgedrückt, um festzustellen, ob der Empfänger verstanden hat und den Ausführungen folgen konnte (in Abbildung 3-12 als Rollentausch bezeichnet). Diese Feedbackschleifen sind im Wissensweitergabemodell aus Übersichtlichkeitsgründen nicht berücksichtigt, beeinflussen den Erfolg der Wissensweitergabe aber sehr wohl. Der Kommunikationsprozess ist also bei persönlicher und insbesondere bei synchroner 1-1 Kommunikation interaktiv und erhöht damit die Chance auf erfolgreiche Weitergabe.

Einen weiteren Ansatzpunkt für die Erklärung von Kommunikation bietet die *Sprechakttheorie*, welche von Austin entwickelt (Austin 1962) und von Searle verfeinert wurde (Searle 1969; Searle 1992). Sie setzt am pragmatischen Aspekt der Kommunikation an und betrachtet das Sprechen als Handlung. Sprechen ist demnach eine komplexe Form regelgeleiteten Verhaltens (Searle 1992, S. 24) und Kommunikation wird als intentionale Handlung von Kommunikationspartnern in Situationen einer gemeinsamen Aktivität gesehen. Ein Sprechakt besteht aus einem lokutionärem, illokutionärem und perlokutionärem Teil. Der lokutionäre Teil stellt den Satzinhalt dar, der illokutionäre die

Absicht, die mit der Äußerung (utterance) des Sprechaktes verbunden ist und der perlokutionäre Teil besteht aus der hervorgerufenen Reaktion beim Empfänger (Labrou, Finin 1997, S. 5). Sprechakte stellen die Basis für die menschliche Kommunikation dar und können kurze Ausrufe („Achtung!“) oder auch lange Sätze sein. Ähnlich wie Wissen nach Polanyi einen impliziten Anteil hat, kann auch bei der Kommunikation „der Sprecher mehr meinen als er sagt“ (Searle 1992, S. 32). Im Gegensatz zu Polanyi ist Searle jedoch davon überzeugt, dass es für einen Sprecher trotzdem prinzipiell möglich ist genau das zu sagen, was er meint (ibid.). Weiterhin klassifiziert Searle Sprechakte nach dem illokutionären Zweck, dem psychischen Zustand und der Entsprechungsrichtung (direction of fit) in fünf Großgruppen (Hindelang 1983, S. 45ff). Mit Repräsentativa legt der Sprecher sich darauf fest, eine wahre Aussage über einen Sachverhalt zu machen (z.B. behaupten, mitteilen, berichten). Direktiva sollen die Adressaten des Sprechaktes dazu bewegen etwas zu tun (z.B. bitten, befehlen, vorschlagen). Kommissiva dienen dazu, dass der Sprecher sich selbst zu einer zukünftigen Handlung verpflichtet (z.B. versprechen, vereinbaren, aber auch drohen). Mit Expressiva soll die mentale Einstellung des Sprechers zu einem Sachverhalt zum Ausdruck gebracht werden (z.B. danken, gratulieren, sich entschuldigen). Deklarationen schließlich schaffen durch die Äußerung neue Tatsachen (z.B. ernennen, freisprechen, kapitulieren).

**Auswirkungen auf die Wissensweitergabe:** Mengis und Eppler berücksichtigen bei den von ihnen entwickelten Dimensionen von Wissensdialogen ebenfalls den lokutionären Teil, den sie als Nachricht bezeichnen und den illokutionären Teil, der als Gesprächsabsicht bezeichnet wird (Eppler, Mengis 2004, S. 93ff). Sie argumentieren, dass für eine erfolgreiche Kommunikation die Absicht von allen Gesprächsteilnehmern geteilt werden muss und schlussendlich immer das Fördern von Lernen und Verständnis das Ziel ist. Die Absicht muss offen genug für unvorhergesehene Ergebnisse sein. Die Nachricht sollte qualitativ hochwertig, zielgruppenadäquat, verdichtet, in gemeinsamen Erfahrungen verankert und kompromissfähig sein. Darüber hinaus lenken Eppler und Mengis das Augenmerk auf die mit der Kommunikation verbundenen Prozesse, die Gruppendynamik und die mentalen Modelle der Gesprächsteilnehmer.

Noch stärker in Richtung soziale Kommunikation geht die *universelle Theorie der Höflichkeit* (Brown, Levinson 1987). Sie fasst Kommunikation als potentiell Image-Gefährdende Situation auf, in der die Kommunikationspartner höflich sein müssen, damit jeder sein „Gesicht wahren“ kann, also das öffentliche Selbstbild, das jeder kompetente

Erwachsene in der Gesellschaft gerne für sich in Anspruch nimmt (Carlo, Yoo 2003, S. 3). Höflichkeit wird als wichtig für Geschäftskommunikation angesehen, weil dabei drei prototypische Situationen auftreten können, die zu einem Gesichtsverlust des Kommunikationspartner führen können (Carlo, Yoo 2003, S. 4).

- Informationssuche könnte auf Ignoranz stoßen oder dazu führen, dass man sich als in der Schuld des Helfenden stehend fühlt.
- Informationsnachfrage kann neugierig wirken, aber auch anmaßend, wenn man jemanden dazu auffordert, etwas zu tun.
- Die genutzte Ausdrucksweise kann dazu führen, dass die soziale Distanz und relative Macht zwischen den Kommunikationspartnern betont oder übergangen wird.

Bei Wissensweitergabe ist dies besonders relevant, da alle drei Punkte zutreffen können. Zuzugeben, selbst nicht über genügend Wissen zu verfügen, kann als Zumutung empfunden werden und findet nur statt, wenn genügend Vertrauen besteht, dass dieses Zugeständnis nicht gegen die eigenen Interessen eingesetzt wird. Auch in Push-Situationen kann das Weitergeben von Wissen als Zumutung auf Empfängerseite empfunden werden, weil der Empfänger implizit dazu aufgefordert wird sein Weltbild zu ändern (zumindest in geringem Umfang) und zusätzlich eine implizite Aufforderung zur Anerkennung des Wissens der Quelle empfunden werden kann. Auch die Wortwahl und Betonung bei der Wissensweitergabe kann zu Emotionen beim Empfänger führen und dadurch eine erfolgreiche Rekonstruktion des Wissens verhindern, bzw. in der Evaluationsphase zur Zurückweisung führen.

Die räumliche Nähe ist für Kommunikation aus zwei Gründen wichtig. Zum einen ermöglicht räumliche Nähe direkte Kommunikation, während räumliche Distanz meist technologievermittelte Kommunikation erzwingt. Da bei direkter Kommunikation die Verständigungsmöglichkeiten größer sind, folgern Canals et al., dass auch die Wissensweitergabe durch räumliche Nähe erleichtert wird (Canals et al. 2004, S. 12). Zum anderen bedeutet größere Nähe oft auch ähnlichere Verhaltensweisen und Sprache, was für die Wissensweitergabe ebenfalls förderlich ist (ibid.).

Was Canals et al. für organisationsübergreifende Beziehungen festgestellt haben, kann in abgeschwächter Weise auch auf Beziehungen zwischen Individuen innerhalb eines Firmengebäudes übertragen werden. Je näher die Arbeitsplätze einander sind, desto mehr soziale Kontakte entstehen und desto höher die Wahrscheinlichkeit für Wissensweitergabegelegenheiten.

#### 3.5.2 Technologievermittelte Kommunikation

Sobald kein direkter Face-to-Face Austausch zwischen Quelle und Empfänger möglich ist, erschweren sich die Aufgaben des Mitteilens und Verstehens, da bei (computer-) vermittelter Kommunikation nur reduzierte Ausdrucksmöglichkeiten zur Verfügung stehen (Herrmann, Kienle 2004, S. 60). Insbesondere die asynchrone Kommunikation ist hierbei problematisch, da keine schnelle Rückmeldung vom Empfänger kommt (ibid.). Als theoretische Grundlagen können hier die Media Richness, Media Synchronicity und Social Presence Theory herangezogen werden.

Das Konzept des Medienreichtums (*media richness*) geht auf Daft und Lengel zurück (1986). Die mit Medienreichtum verbundene psychologische Interaktion nimmt von einem direkten physischen Treffen (face-to-face) bis hin zur klassischen Post stetig ab. In absteigender Reihenfolge ordnen sich zwischen diesen beiden Extremen Videokonferenzen, Telefonieren, Videobroadcasts und Email ein (Carmel 1999, S. 49). Die zentrale Aussage der Media Richness Theory ist, dass die Auswahl des Mediums an die Anforderungen der Arbeitsaufgabe angepasst werden sollte. Insbesondere sollten reichhaltigere Medien für mehrdeutige Aufgaben verwendet werden und weniger reichhaltige für eindeutige Aufgaben (zur Erhöhung der Effektivität, Dennis, Kinney 1998, S. 257). Die Richtigkeit dieser Aussage wurde aber angezweifelt und konnte in mehreren empirischen Studien (z.B. für Videokonferenzen im Gegensatz zu textbasiertem, computermediertem Austausch) nicht bestätigt werden (Dennis, Kinney 1998, S. 269f). Die Kritiker haben argumentiert, dass die Wahrnehmung von Medieneigenschaften sozial konstruiert und daher von Normen, Erfahrungen der Nutzer, einer kritischen Masse von Nutzern und dem Organisationskontext abhängt (Carlo, Yoo 2003, S. 2f).

Davon motiviert entwickelten Dennis und Valacich die *Media Synchronicity Theory* (Dennis, Valacich 1999). Diese besagt, dass Arbeitsziele von zwei Prozessen bestimmt werden, der Übermittlung von Inhalten (conveyance) und der Einigung (convergence) (Watson et al. 2005, S. 6). Medien haben eine Reihe von Eigenschaften, die für die beiden Prozesse förderlich oder hinderlich sein können. Insbesondere identifizieren Dennis und Valacich die Eigenschaften Schnelligkeit der Rückmeldung, Symbolreichtum, Parallelität, Überprüfbarkeit und Reproduzierbarkeit. Die Geschwindigkeit der Rückmeldung misst die Zeit in der das Medium dem Empfänger einer Nachricht erlaubt, dem Sender eine Rückmeldung zu geben. Symbolreichtum gibt die Anzahl an Möglichkeiten an, eine Information darzustellen, z.B. nur textuell, oder auch grafisch. Parallelität gibt die Anzahl

gleichzeitiger Gespräche mit unterschiedlichen Empfängern an. Überprüfbarkeit ist der Grad in dem das Medium zulässt die Nachricht vor dem Übermitteln zu überprüfen und einzustudieren. Reproduzierbarkeit ist das Ausmaß in dem das Medium zulässt, ältere Nachrichten noch einmal anzusehen oder zu hören. In Tabelle 3-8 werden gängige Medien nach diesen Kriterien charakterisiert.

Tabelle 3-8: Charakterisierung verschiedener Medien nach (Dennis, Valacich 1999, S. 3)

	Rückmeldung	Symbolreichtum	Parallelität	Überprüfbarkeit	Reproduzierbarkeit
Face-to-face	hoch	niedrig-hoch	niedrig	niedrig	niedrig
Videokonferenz	mittel-hoch	niedrig-hoch	niedrig	niedrig	niedrig
Telefon	mittel	niedrig	niedrig	niedrig	niedrig
Schriftliche Post	niedrig	niedrig-mittel	hoch	hoch	hoch
Anrufbeantworter	niedrig	niedrig	niedrig	niedrig-mittel	hoch
E-Mail	niedrig-mittel	niedrig-hoch	mittel	hoch	hoch
Online chat	mittel	niedrig-mittel	mittel	niedrig-mittel	niedrig-mittel
Asynchrone Groupware	niedrig	niedrig-hoch	hoch	hoch	hoch
Synchrone Groupware	niedrig-mittel	niedrig-hoch	hoch	mittel-hoch	hoch

Der Schlüssel zur erfolgreichen Kommunikation ist die Auswahl des richtigen Mediums anhand der Eigenschaften, die für die Aufgabe benötigt werden (Watson et al. 2005, S. 6). Es muss jedoch angemerkt werden, dass einige der Einstufungen recht willkürlich erscheinen. So ist beispielsweise nicht nachvollziehbar, warum analoge Nachrichten (schriftliche Post) mehr Parallelität erlauben als digitale (E-Mail). Eine wesentliche Erkenntnis von Dennis und Valacich ist, dass unterschiedliche Medien für Conveyance und Convergence Prozesse vorteilhaft sind, in der Praxis aber beide Prozesse oft gemeinsam auftreten und somit zwischen verschiedenen Medien gewechselt werden muss (ibid. S. 7f).

*Social Presence* kann als der Grad aufgefasst werden, in dem der Gesprächspartner bei medienvermittelter Kommunikation als reale Person wahrgenommen wird (Guanawardena 1995, S. 151). Als zugrunde liegende Konzepte dienen dabei die Vertrautheit, die sich in Form von wahrgenommener physischer Distanz, Augenkontakt, Lächeln und persönlichen Gesprächsthemen ausdrückt, und die Direktheit, die z.B. durch formelle Kleidung,

Gesichtsdruck und Abstand in direkten Gesprächen zum Ausdruck kommen kann (ibid.). Andere Autoren sehen soziale Präsenz eher auf den Dimensionen persönlich-unpersönlich, sensibel-unsensibel, warm-kalt und sozial-unsozial (Short 1976). Diese Konzeptualisierung wird von Lombard und Ditton als soziale Reichhaltigkeit des Mediums eingestuft (Lombard, Ditton 1997). Weitere Konzeptualisierungen sehen soziale Präsenz als realistische Vermittlung, Versetzen des Nutzers in die virtuelle Welt oder umgekehrt, als Eintauchen, die Vermittlung eines sozialen Akteurs innerhalb des Mediums oder des Mediums selbst als sozialem Akteur. Das Konzept wird also recht unterschiedlich definiert, wobei es trotz allem ein ähnliches Grundverständnis zu geben scheint. Eine neuere Studie zu Social Presence misst das Konstrukt über den Umfang, in dem sozialer Kontext transportiert wird, die Interaktivität des Mediums und die empfundene Privatheit der Kommunikation (Tu 2002).

Neuere Untersuchungen zur Kommunikation bei wissensintensiver Arbeit bzw. Softwareentwicklung stammen von (Wong, Dalmadge 2004) und (Mäki et al. 2004). Wong und Dalmadge erweitern die Media Richness Theory selber zu einem Framework, das die Wissensintensität und Komplexität der Prozesse berücksichtigt und Medien anhand der Kriterien Reichhaltigkeit, Interaktivität und Genauigkeit klassifiziert (siehe Abbildung 3-13). Die Autoren berücksichtigen dabei allerdings die Media Synchronicity Theory nicht und können auf keine eigenen empirischen Ergebnisse zur Validierung verweisen.

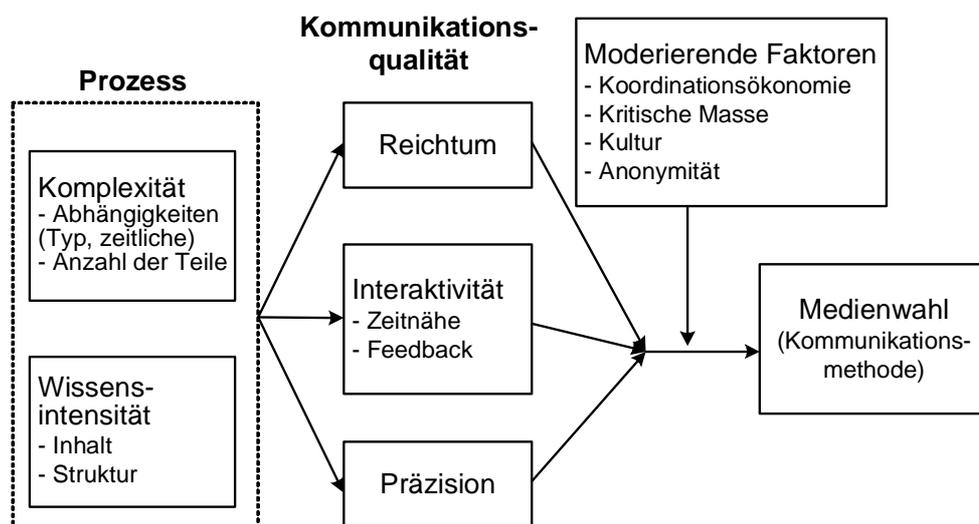


Abbildung 3-13: Medienauswahl in wissensintensiven Prozessen (Wong, Dalmadge 2004, S. 3)

Die Verwendung von Kommunikationsmedien bei der Softwareentwicklung in einem international verteilt arbeitenden Softwareunternehmen ist das Untersuchungsobjekt in

(Mäki et al. 2004). In Tabelle 3-9 und Tabelle 3-10 sind die Ergebnisse der Studie zusammengefasst.

Tabelle 3-9: Gründe für die Nutzung von Medien (Mäki et al. 2004, S. 6)

<b>Kommunikationsmedium</b>	<b>Zweck / Grund der Benutzung</b>
Telefon	<ul style="list-style-type: none"> <li>- Kurze, dringende Fragen oder Angelegenheiten</li> <li>- Komplizierte Angelegenheiten</li> <li>- Wenn interaktive Kommunikation oder Diskussion nötig ist (also E-Mail nicht adäquat ist)</li> </ul>
E-Mail	<ul style="list-style-type: none"> <li>- Angelegenheiten nicht dringend</li> <li>- Mehrere Leute sollen einbezogen werden</li> <li>- Dateianhänge müssen verschickt werden</li> </ul>
Telefonkonferenz/ Netmeeting	<ul style="list-style-type: none"> <li>- Regelmäßige oder vorangekündigte interne Besprechungen (über mehrere Standorte hinweg)</li> </ul>

Zwei Hauptprobleme wurden in der Studie identifiziert (ibid. S. 8). Zum einen waren die Zuständigkeiten unklar, sowohl bei den verwendeten IT-Systemen (Lotus Notes und Intranet-Webseiten) als auch bei den Personen, was zu erschwertem Zugang zu Informationen führt. Zum anderen waren die Experten schlecht erreichbar, weil sie mit Arbeitsaufgaben und Anfragen überlastet waren.

Tabelle 3-10: Vor- und Nachteile von Kommunikationsmedien (Mäki et al. 2004, S. 7)

<b>Kommunikationsmedium</b>	<b>Vorteile</b>	<b>Nachteile</b>
Telefon	<ul style="list-style-type: none"> <li>- Schnelle, unmittelbare Antwort/Rückmeldung</li> </ul>	<ul style="list-style-type: none"> <li>- Keine Aufzeichnungen für spätere Benutzung</li> <li>- Zeitdifferenz zwischen den Standorten</li> <li>- Kein Augenkontakt, keine physische Präsenz</li> <li>- Abmachungen sind nicht immer offensichtlich und werden unterschiedlich verstanden</li> </ul>
E-Mail	<ul style="list-style-type: none"> <li>- Archivierung, schriftlicher Beleg, zur späteren Einsicht</li> <li>- Größere zeitliche Unabhängigkeit und Vorbereitung der Antwort</li> </ul>	<ul style="list-style-type: none"> <li>- Langsame oder gar keine Antworten (auch wegen Zeitdifferenz)</li> <li>- Überflutung mit zu vielen E-Mails</li> <li>- Empfänger der E-Mails manchmal nicht richtig gewählt</li> </ul>

### 3 Erklärungsansätze für Wissensweitergabe und Arbeitsverhalten

<b>Kommunikationsmedium</b>	<b>Vorteile</b>	<b>Nachteile</b>
E-Mail	<ul style="list-style-type: none"><li>- Erreichbarkeit mehrerer Personen</li><li>- Besseres Verständnis der klaren schriftlichen Kommunikation</li><li>- Unterbricht die Arbeit nicht</li></ul>	<ul style="list-style-type: none"><li>- Zeitraubend zu erstellen</li><li>- Nicht für die Diskussion komplexer Angelegenheiten geeignet</li></ul>
Telefonkonferenz/ Netmeeting	<ul style="list-style-type: none"><li>- Keine Reise nötig</li><li>- Schnell und flexibel</li><li>- Möglichkeit zum Weitergeben von Daten, Bildern und Graphen</li></ul>	<ul style="list-style-type: none"><li>- Schwierig zum Diskutieren komplexer Angelegenheiten</li><li>- Wegen Sprachqualität oft schwierigen Gespräch zu folgen</li><li>- Kein Blickkontakt</li></ul>

Die Nachteile von mediengestützter Kommunikation ohne direkten persönlichen Kontakt gegenüber Face-to-Face Kommunikation sind schwer zu fassen, da sie sich von Fall zu Fall unterscheiden. Kennen sich die beiden Gesprächspartner gut, so kann ein telefonisches Gespräch genauso hilfreich sein wie ein persönliches. Kennen sich die Gesprächspartner nicht, so fehlt der Blickkontakt zur besseren Verständigung in jedem Fall.

#### 3.5.3 Diskussion

Die dargestellten Theorien und Ansätze bewegen sich z.T. auf unterschiedlichen Ebenen der Kommunikation, so dass die Beziehung zueinander nicht unmittelbar klar wird. Fincham stellt eine Ordnung her, indem er Wörter als kleinste Einheit der Kommunikation sieht und die anderen Konstrukte daraus aufbaut (Fincham, Rhodes 2003, S. 177f). Aus Wörtern entstehen Sätze und Sätze formen laut Fincham soziale Akte (z.B. verlangen, behaupten, beschweren), die sich weitgehend mit den oben dargestellten Sprechakten decken. Die Sprechakte bilden Episoden, die eine interne Homogenität aufweisen (z.B. das Vorstellen einer Person oder eines Themas) und mehrere Episoden zusammen bilden eine Begegnung (z.B. ein Interview, eine Verhandlung, eine Besprechung). Diese hierarchische Bildung von Begegnungen aus Wörtern bezeichnet Harré als Verkettung (Fincham, Rhodes 2003, S. 178). Solche Begegnungen können in offene, geschlossene und definierte eingeteilt werden (siehe auch Abschnitt 3.2.1), wobei offene Begegnungen kein bestimmtes Ziel haben (z.B. ein Abend mit Freunden, ein Treffen auf dem Gang), geschlossene Begegnungen streng formalisiert sind (z.B. Zeremonien, Rituale) und definierte Begegnungen zwar ein Ziel und gewisse Regeln haben, die jedoch nicht alle Episoden und Sprechakte vorschreiben sondern große Freiheiten lassen (Fincham, Rhodes

2003, S. 178f). Solche definierten Begegnungen haben trotzdem oft eine gemeinsame Grundstruktur der Form Begrüßung, Beziehungsaufbau, zentrale Aufgabe, soziale Beziehung wiederherstellen und Verabschiedung (ibid. S. 182). Wissensweitergabe ist also eine definierte Begegnung anzusehen.

Der Übermittlungsschritt ist aber auch der Hauptansatzpunkt für IT-Unterstützung der Wissensweitergabe. In der Softwareentwicklung werden neben den auch in anderen Branchen genutzten Email, Dokumenten- und Contentmanagement-, sowie Groupware- und E-Learningsystemen auch Codeverwaltungssysteme (z.B. CVS, Subversion, Visual Source Safe) zum Austausch von kodifiziertem Wissen in Form von inline Dokumentation genutzt (vgl. Teubner, Nietsch 2000; Maier et al. 2005; Lindvall, Rus 2003). Auch Fehlerberichte und deren Lösungen, die in Bugtrackingsystemen abgelegt sind, können eine wertvolle Wissensquelle darstellen (Lindvall, Rus 2003, S. 81).

### **3.6 Beziehungskontext**

Im Beziehungskontext finden sich alle Einflussfaktoren, die mit der sozialen Einbettung von Quelle und Empfänger in Verbindung stehen. Die Anzahl an Faktoren, die in vielen Studien identifiziert wurden zeigt die Bedeutung dieses Kontextbereichs (siehe Tabelle 3-11).

Bemerkenswert sind von den Faktoren v.a diejenigen in der Kategorie Gemeinsamkeiten von Quelle und Empfänger, da sie zwar als Folge des sozialen Netzes interpretiert werden können (z.B. gleiche Professoren im Studium gehört, gleiche Vorbilder), aber auch positiv auf die Wissensweitergabe wirken können, wenn sich die beiden Personen überhaupt nicht kennen und keine gemeinsamen Bekannten (im engeren Sinne des Wortes) haben. So können Überlappungen im Wissen aus Lesen der gleichen Bücher resultieren, oder auch daraus, dass Projekterfahrungen mit ähnlichen Unternehmen gemacht wurden (z.B. mit zwei Telekommunikationskonzernen in verschiedenen Ländern). Die Normendistanz kann dadurch gering sein, dass beide in ähnlichen Kulturkreisen aufgewachsen sind, die nicht einmal in den gleichen Ländern sein müssen. Auch eine gemeinsame Religion kann für eine geringe Normendistanz sorgen.

Tabelle 3-11: Einflussfaktoren im Beziehungskontext und deren Kategorisierung

Kategorie	Einflussfaktor
Soziale Beziehung	Sichere Beziehung um wichtige Fragen zu stellen (Cross, B/K)
	Soziale Beziehung (Szulanski), Typ (B/K) und Art der Beziehung (B/K)
	Stärke der Beziehung (B/K), Häufigkeit und Tiefe des Kontakts (Ladd/Ward)
	Einbettung der Beziehung (B/K)
	Informelle Beziehungen (B/K)
	Relation Model (Boer)
Sozialer Status (Thomas-Hunt)	Reputation von Quelle und Gruppe (Ensign/Hebert)
	Vertrauen (Handzic)
	Glaubwürdigkeit (Sarker/Joshi)
	Als zuverlässig wahrgenommen (B/K)
Expertenstatus (Thomas-Hunt)	Ansehen für Wissensweitergabe (Wah)
	Anerkennung der Wissensweitergabe als gute Arbeit (Hendriks)
Gemeinsamkeiten von Quelle und Empfänger	Normendistanz (Cummings/Teng)
	Wissensdistanz (Cummings/Teng)
	Ähnlichkeit zwischen Quelle und Empfänger und deren Wissen (B/K, Ladd/Ward)
	Überlappung des Wissens von Quelle und Empfänger (B/K), Wissensredundanz (B/K)

Im Folgenden werden unter dem Begriff soziale Netzwerke die Forschungsergebnisse über soziale Beziehungen allgemein diskutiert. Anschließend werden einzelne Faktoren genauer betrachtet, zu denen Forschungsergebnisse gefunden wurden. Aspekte des Sozial- und Expertenstatus werden unter dem Begriff Reputation erforscht, zu den Faktoren Vertrauen und Normendistanz konnte ebenfalls Literatur identifiziert werden. Macht ist schließlich ein Konstrukt, das sowohl die Art der sozialen Beziehung beeinflusst als auch Auswirkungen auf den sozialen Status hat.

### 3.6.1 Soziale Netzwerke

Die soziale Netzwerkanalyse erforscht, wie vernetzte Strukturen von Verwandten, Bekannten, Freunden und Kollegen, etc. zustande kommen, welche Bedingungen die Größe und Dichte der Netze beeinflussen und welche Rückschlüsse die Kenntnis eines Ego-Zentrierten Netzes auf das Individuum erlaubt.

**Begriffe:** Die elementare Betrachtungseinheit ist dabei die *Dyade*. So ein minimales soziales Netz besteht aus zwei Personen und der Beziehung zwischen ihnen. Die soziale Netzwerkanalyse spricht von den beiden Personen als *Ego* und *Alter*. Ein Netzwerk wird meist als *Graph* dargestellt, wobei die Knoten die Personen und die Kanten die Beziehungen repräsentieren. Die Distanz innerhalb eines sozialen Netzwerks bestimmt sich über die Anzahl der Kanten, die zwischen zwei Knoten liegen. Personen, die eine direkte Beziehung zueinander haben besitzen also eine Distanz von eins, der Freund eines Freundes (friend-of-a-friend, FOAF) eine Distanz von zwei. Beziehungen können einseitig oder reziprok sein sowie einfach oder multiplex. *Reziprok* bedeutet, dass eine Beziehung, die von Ego aufgebaut wird, auch von Alter erwidert wird. *Multiplex* bedeutet, dass mehrere Beziehungen zwischen Ego und Alter bestehen, z.B. eine berufliche als Kollegen und eine private Freundschaftsbeziehung. Die *Dichte* eines Netzwerks gibt an, wie viele Beziehungen zwischen den Akteuren bestehen im Verhältnis zu der maximal möglichen Anzahl an Beziehungen. Sie dient meist nur als Kontrollvariable. Die *Zentralität* eines Netzwerks bestimmt, ob einige wenige Akteure mit vielen Beziehungen den Mittelpunkt bilden, um den sich viele Akteure mit weniger Beziehungen gruppieren, oder ob die Beziehungen relativ gleichmäßig verteilt sind. Die Reichweite eines Netzwerkes schließlich gibt an, wie viel Prozent der Mitglieder im Durchschnitt erreicht werden, wenn eine Nachricht von einem beliebigen Individuum aus an alle ihre Freunde und deren Freunde weitergegeben wird (zu all diesen Begriffen vgl. Kilduff, Tsai 2003, S. 13ff).

**Balance Theorie:** Die Balance Theorie wurde von Heider (1958) als Teil der Theorie kognitiver Konsistenz entwickelt. Sie besagt, dass Personen sich reziproke und transitive Beziehungen wünschen. *Transitiv* bedeutet, dass wenn A mit B und C befreundet ist, auch B und C befreundet sein sollen. Entspricht ihre Wahrnehmung dem nicht, werden also unausgewogene (unbalanced) Beziehungen wahrgenommen, so wird entweder versucht, die Beziehungen doch noch ausgewogen zu gestalten, oder die Beziehung wird abgebrochen, um kognitive Dissonanzen zu vermeiden (Kilduff, Tsai 2003, S. 42ff). Eine weitere Alternative besteht darin, dass Ego sich einredet, die Beziehung wäre ausgewogen. Letzteres tritt jedoch in der Regel erst bei größeren Distanzen im sozialen Netzwerk auf (Kilduff, Tsai 2003, S. 72f). Eine Implikation der Balance Theorie ist, dass Menschen dazu tendieren, sich in Cliques zu organisieren. Eine Clique ist eine Teilgruppe innerhalb einer größeren Gruppe bei der die durchschnittliche Freundschaft (oder eine andere Beziehung) für alle Mitglieder der Teilgruppe untereinander größer ist, als die mit anderen Mitgliedern

der Gruppe. In einer Clique müssen alle Mitglieder direkt miteinander verbunden sein. Erst ab mindestens drei Personen wird von einer Clique gesprochen. Da diese Definition zu eng gefasst ist, um in der Realität oft aufzutreten, wurden Einschränkungen getroffen, die das Konzept besser anwendbar machen. *N-Cliquen* sind Cliquen bei der die Mitglieder eine Distanz  $N > 1$  aufweisen dürfen, wobei vor allem  $N=2$  interessant ist. Der *k-plex Ansatz* zur Identifikation von Cliquen schränkt die strikte Definition ein, indem Mitglieder einer Clique nicht mit allen anderen Mitgliedern verbunden sein müssen sondern nur mit  $n - k$  Mitgliedern, wobei  $n$  die Größe der Clique angibt (Kilduff, Tsai 2003, S. 46). Verschiedene Cliquen können auch Überlappungsbereiche aufweisen. Auf Personen, die Mitglieder in einer Clique sind, lastet ein gewisser sozialer Druck. Sie können nicht ohne weiteres die Beziehung zu einem Cliquenmitglied lösen, da die anderen Mitglieder darauf bedacht sind transitive Beziehungen herzustellen, also ihre Beziehungen ausgewogen zu gestalten. Es wird somit Druck aufgebaut, die Beziehung weiter aufrecht zu erhalten, oder es kann im Extremfall auch zum Ausschluss aus der gesamten Clique kommen. Beziehungen zwischen zwei Personen, die derselben Clique angehören werden nach dem deutschen Soziologen Georg Simmel auch Simmelsche Beziehungen genannt.

***Social Comparison Theory:*** Die Theorie sozialer Vergleiche wurde von Festinger (1954) entwickelt und postuliert, dass Menschen über sich selbst lernen, indem sie sich mit anderen, ähnlichen Menschen vergleichen. Diese Vergleiche wirken sich auch auf die Einstellungen und Meinungen der Menschen aus, insbesondere wenn keine objektiven Kriterien zur Meinungsbildung vorhanden sind (Kilduff, Tsai 2003, S. 49ff). Für die vorliegende Arbeit ist von besonderem Interesse, dass dies auch auf Wissen zutrifft. Man sucht sich Freunde mit ähnlichem Wissen, um sein eigenes Wissen genauer einschätzen zu können. Die Ähnlichkeit zu anderen Personen kann dabei auf Basis mehrerer Merkmale ermittelt werden, so dass häufig mehrere Gruppen mit jeweils mindestens einer ähnlichen Merkmalsausprägung ausfindig gemacht werden und Mitgliedschaft in allen diesen Gruppen besteht. Je diverser diese Gruppen sind, desto loser die Bindung der Person zu einer der Gruppen. Ein weiterer Faktor, der über die Stärke der Bindung bestimmt, ist die Anzahl der Gruppenmitglieder. Existieren in einer großen Organisation z.B. nur wenige Personen einer bestimmten Rasse, eines Geschlechts oder einer Religion, so ist die Bindung innerhalb dieser Gruppe stärker, als in Organisationen, wo die Anzahl der Gruppenmitglieder groß ist (ibid.).

Im Gegensatz zu diesem Bedürfnis, sich mit Menschen zu umgeben, die ähnliche Charakterzüge oder Merkmale haben wie man selbst, ist es für die Wissensweitergabe wichtig, dass Kontakte zu Mitgliedern anderer Gruppen bestehen, da gerade von denen wertvolle Anregungen und Neuigkeiten zu erfahren sind. Hansen stellt bei seinen Untersuchungen fest, dass über so genannte *weak ties*, also Beziehungen, die nur schwach ausgeprägt sind, was u.a. in seltener Kommunikation abzulesen ist, wenig komplexes Wissen besonders gut weitergegeben wird, wogegen die Weitergabe komplexen Wissens auf starke Beziehungen (*strong ties*) zwischen dem Lernenden und dem Lehrenden angewiesen ist (Hansen 1999). Man kann also schlussfolgern, dass für einen Generalisten das Pflegen vieler schwacher Beziehungen vorteilhaft ist, um breites Wissen in vielen Bereichen zu erlangen, wogegen Spezialisten auf die Existenz von starken Bindungen zu anderen Experten im selben Bereich angewiesen sind, um ihr Wissen weiter zu vertiefen. Mit *weak ties*, die naturgemäß in Organisationen hauptsächlich zwischen Mitarbeitern verschiedener Abteilungen oder Standorten auftreten, ist aber noch mehr verbunden. Durch die Beziehung erhalten beide Personen die Rolle eines Bindeglieds (*boundary spanner*, wörtlich: Grenzenüberspanner), das die strukturelle Lücke (*structural hole*) im Beziehungsnetzwerk schließt. Dies leistet einen großen Beitrag zum sozialen Kapital der beiden Personen. Das soziale Kapital kann grob als die potentiellen Ressourcen betrachtet werden, die in den sozialen Beziehungen einer Person inhärent sind (Kilduff, Tsai 2003, S. 28; vgl. auch Lin 2001). Die Vorteile, die aus dem Füllen einer strukturellen Lücke im Beziehungsnetzwerk resultieren, können aber nur immer für kurze Zeit genutzt werden (Kilduff, Tsai 2003, S. 57).

Diese Überlegungen können auf ein noch allgemeineres Konzept zurückgeführt werden, nämlich das *Linsenmodell von Brunswik* (1952). Es geht davon aus, dass für Urteilsprozesse jeglicher Art (Prognosen und Evaluationen) eine Menge von Referenzpunkten herangezogen wird (Jost 2000, S. 186f). Das bedeutet, dass Urteile im Verhältnis zu diesen Referenzpunkten und nicht absolut erfolgen.

***Cognitive Network Theory:*** Ein Versuch die Ansätze der Psychologie mit denen der Soziologie zu vereinen wird von Kilduff und Tsai kognitive Netzwerktheorie genannt. Sie erweitert den Einfluss von Netzwerken auf Meinungen und Verhalten um den Aspekt, dass oft nicht die tatsächlichen Verhältnisse den Ausschlag geben, sondern die wahrgenommenen (Kilduff, Tsai 2003, S. 70ff). Das soziale Kapital einer Person enthält neben der direkten Komponente, dass über Beziehungen bestimmte Dinge erreicht werden

können (z.B. Beförderung, neuer Job) auch eine indirekte, nämlich dass einem Vorteile zuteil werden, weil die darüber entscheidenden Personen glauben, dass man einen einflussreichen Fürsprecher hat, den man nicht verärgern will, bzw. der im Umkehrschluss über die Transitivität von Beziehungen zum eigenen Freund wird. Des Weiteren werden auch die eigenen Wahrnehmungen von den sozialen Netzwerken in die man eingebunden ist beeinflusst. Während sozial Gleichgestellte dazu tendieren unterschiedliche Wahrnehmungen einander anzugleichen und einen Kompromiss zu finden, wurde bei Personen mit einseitigen Beziehungen eine Tendenz von Alter zur Ablehnung der Wahrnehmung von Ego festgestellt, wohingegen Ego sich der Meinung von Alter annähert. Ist Alter ein Vorgesetzter von Ego und beide kommen nicht gut miteinander aus, so nähern sich ihre Wahrnehmungen entweder gar nicht an, oder Ego wird durch sozialen Druck quasi gezwungen die Meinung von Alter zu übernehmen (Kilduff, Tsai 2003, S. 76f). Ein weiterer Aspekt der Theorie beschäftigt sich mit der Wahrnehmung von sozialen Netzwerken durch die Mitglieder der Netzwerke, die sich signifikant von den tatsächlichen Netzwerken unterscheiden können. Diese Diskrepanzen haben insbesondere bei Entscheidungsträgern Auswirkungen, da die Untergebenen getroffene Entscheidungen ganz anders aufnehmen können als das vorhergesehen wurde, z.B. weil befreundete Personen nicht mehr zusammen arbeiten dürfen, oder Kollegen, die sich nicht leiden können in Zukunft eng zusammenarbeiten müssen.

***Social Exchange Theory:*** Die Theorie des sozialen Austauschs geht auf Emerson (1976) zurück und geht von der Grundannahme aus, dass zwischenmenschliches Verhalten ein Austausch von Aktivitäten darstellt, die mit Belohnungen und Kosten verbunden sind (Zafirovski 2005, S. 2f). Aktivität ist dabei im weiteren Sinne zu verstehen, so dass auch Kommunikation eingeschlossen ist (siehe Abschnitt 3.5.1 zur Sprechakttheorie). Austausch ist in diesem Zusammenhang eine soziale Interaktion, die auf reziproken Stimuli basiert (ibid. S. 3). Wenn die Reziprozität nicht gewährleistet ist, so funktioniert die Austauschbeziehung langfristig gesehen nicht mehr. Individuen etablieren und pflegen soziale Beziehungen auf der Basis von Erwartungen, dass diese zum gegenseitigen Vorteil sein werden, und nicht aufgrund von normativen Verpflichtungen. Diese Vorteile können intrinsische oder auch extrinsische Motive befriedigen (ibid.). Eigeninteresse und gegenseitige Abhängigkeiten sind zwei der Kernelemente der Theorie (Lawler, Thye 1999, S. 217). Der Austausch dient dem Erreichen von Zielen, die ein Individuum alleine nicht erreichen könnte (ibid.).

Obwohl einige Ähnlichkeiten zu ökonomischen Austauschbeziehungen vorhanden sind, existieren auch Unterschiede. Soziale Austauschbeziehungen basieren mehr auf intrinsischen Belohnungen und persönlichem Vertrauen als auf dem Austausch von Gütern mit extrinsischen Belohnungen und unpersönlichen Markt- und Rechtsgrundlagen (Zafirovski 2005, S. 4). Sie sind meist langfristige Beziehungen gegenüber den in der Ökonomie verbreiteten einmaligen Transaktionen und somit einer langfristigen Kundenbindung ähnlich.

Für die intrinsischen Belohnungen von Austausch spielen Emotionen eine wichtige Rolle (Lawler, Thye 1999, S. 218). Sie können in Form von Erleichterung, Zufriedenheit, Aufregung, Stolz auf sich selbst oder Dankbarkeit gegenüber dem Interaktionspartner aus einem Austausch resultieren (ibid.). Aber auch negative Gefühle sind als Ergebnis denkbar. Diese Gefühle beeinflussen die zukünftigen Interaktionen mit dem Austauschpartner (ibid. S. 222, siehe auch Abschnitt 3.3.7).

**Auswirkungen auf die Wissensweitergabe:** Studien haben belegt, dass Charakteristika des sozialen Netzwerks die Wissensweitergabe beeinflussen. So wurden z.B. die Vorhersagen der structural hole Theorie von Burt (1992) bestätigt, dass fehlende Redundanz in ihrem sozialen Netzwerk in Organisationen stark mit deren Fähigkeiten zusammenhängt, Wissen und neue Fähigkeiten zu erwerben (McEvily, Zaheer 1999). Redundante soziale Bindungen behindern den Zugang zu neuem Wissen. Die Art der sozialen Beziehung ist ebenfalls entscheidend für die Wissensweitergabe. Schwache Beziehungen, also unregelmäßige und entfernte Beziehungen, sind für die Weitergabe eindeutigen, kodifizierten Wissens gut geeignet, während starke Beziehungen für die Weitergabe impliziten Wissens besser geeignet sind (Hansen 1999). Auch in Studentengruppen, die an einer Wirtschaftssimulation teilnahmen war die Wissensweitergabe besser, wenn stärkere soziale Kontakte zwischen den Studenten vorlagen (Argote, Ingram 2000, S. 162).

#### **3.6.2 Normen**

Konformität mit einer Norm bedeutet die generelle Erwartung in das Verhalten zu erfüllen, wobei Erwartung dabei nicht im kognitiven sondern im normativen Sinne gemeint ist (Habermas 1981, S. 85). Dieses normative Aktionsmodell liegt hinter der Theorie der Rolle, die in der Soziologie weit verbreitet ist, und geht auf Emile Durkheim zurück (Elster 1989, S. 99). Ein auf Normen basiertes Verhaltensbild muss nicht zwangsläufig im Widerspruch zum Bild des rationalen Entscheiders stehen, wenn Normen als emotionale

und verhaltensbasierte Neigungen aufgefasst werden, die als Motive fungieren (Elster 1989, S. 102).

Echte soziale Normen müssen von Verhaltensregeln unterschieden werden (Verhagen 2000, S. 15). Regeln werden von Autoritätsstrukturen aufgestellt und basieren auf gegenseitigen Vereinbarungen. Soziale Normen im engeren Sinne bestehen dagegen aus Konventionen, die eher implizit vorliegen (ibid.). Weiterhin können noch moralische Normen und vernunftbasierte Normen unterschieden werden. Moralische Normen werden wegen des eigenen Gewissens eingehalten, vernunftbasierte Normen weil es rational ist, während soziale Normen eingehalten werden, weil es von der sozialen Umgebung erwartet wird, und Regeln werden befolgt, weil man sich dazu bereit erklärt oder verpflichtet hat (ibid.).

Das von Verhagen als Regel bezeichnete Konzept nennt Elster juristische Norm und führt weiterhin die Unterscheidung von sozialen Normen und privaten Normen ein, wobei letztere von Personen dazu benutzt werden, um Willensschwäche auszugleichen (Elster 1989, S. 100). Im Gegensatz zu sozialen Normen werden private Normen aber nicht mit anderen geteilt. Private Normen sind außerdem von habitualisiertem Verhalten und Gewohnheiten abzugrenzen, deren Nichtbeachtung nicht zu Schuldgefühlen führt (ibid.).

Normen sind oft sehr einfach aufgebaut, z.B. „Mache X!“, „Mache X nicht!“, „Wenn andere Y tun, dann mache X!“, oder „Um Y zu erreichen musst Du X tun!“ (Staller, Petta 2001).

Elster nennt einige Beispiele für soziale Normen, die ein besseres Bild davon vermitteln, was darunter verstanden werden kann (Elster 1989, S. 101, Absatz 1.3).

- Konsumnormen, z.B. Tischmanieren und Dresscodes
- Regeln gegen widernatürliches Verhalten, z.B. Abscheu vor Inzest und Kannibalismus
- Normen für den Einsatz von Geld, z.B. Bestechung ist auch im legalen Bereich negativ zu bewerten
- Normen der Reziprozität, z.B. wird einem ein Gefallen erwiesen, so sollte man in absehbarer Zukunft der entsprechenden Person ebenfalls einen Gefallen tun
- Vergeltungsnormen, z.B. es ist in gewissen Umfang erlaubt Rache zu nehmen
- Arbeitsnormen, z.B. dass man für den Lebensunterhalt arbeiten soll, oder dass im Arbeitskontext nicht zu wenig, aber auch nicht zu viel Einsatz gezeigt wird

- Kooperationsnormen, z.B. wenn die meisten anderen kooperieren, dann tu es auch
- Verteilungsnormen, z.B. was als gerechtes Einkommen empfunden wird

Da verschiedene als gültig anerkannte Normen sich aber auch widersprechen können wird oft diejenige angewendet, die der Person die meisten Vorteile verspricht (Elster 1989, S. 102f). Dies kann i. d. R. allerdings nicht beliebig erfolgen, da jeder ein Selbstbild hat, das er vor sich selbst und vor anderen wahren will (Elster 1989, S. 104). Somit verbietet es sich immer nur die Norm anzuwenden, die eigene Vorteile bringt, sondern erfordert eine gewisse Konsistenz in der Wahl.

Die Sanktionierung von Normen muss dabei nicht unbedingt von außen passieren, sondern kann auch intern in Form von positiven oder negativen Emotionen erfolgen, z.B. Gewissensbissen oder Schuldgefühlen (Elster 1989, S. 105). Emotionen treten auch als Folge von externen Sanktionen auf (z.B. Scham als Folge von Zurechtweisung) und wirken in zukünftigen Situationen auch dann noch nach, wenn keine externen Sanktionen zu erwarten sind (Staller, Petta 2001, Absatz 1.4). Dies ist wichtig für die Effektivität von Normen, da es zwar Metanormen gibt, die verlangen eine Person für die Nichteinhaltung einer Norm in irgendeiner Form zu bestrafen, ja sogar dafür, Personen die dies nicht tun negativ zu sanktionieren, aber dies trotzdem oft aus mangelnder Courage unterlassen wird (Elster 1989, S. 105). So wird z.B. die Norm, dass Männer Frauen in der Öffentlichkeit nicht belästigen sollen, allgemein geteilt. Trotzdem sind nur wenige bereit einzugreifen, wenn sie entsprechende Normverletzungen beobachten, auch auf die Gefahr hin, dass ihnen dies später vorgeworfen wird. Emotionen wirken aber nicht nur als Sanktionsmechanismus für Normen sondern umgekehrt wirken Normen auch regulierend auf Emotionen (Staller, Petta 2001, Absatz 1.7f). So gibt es Normen die vorschreiben, welche Gefühle in welchen Situationen angemessen sind, und welche Gefühle gezeigt werden dürfen (z.B. „Ein Indianer kennt keinen Schmerz!“, was soviel heißt wie „Ein Mann weint nicht!“). Stimmen tatsächliche Gefühle und durch Normen vorgeschriebene nicht überein, so führt das zu kognitiven Dissonanzen (z.B. „Ich sollte eigentlich glücklich sein, aber ...“).

Soziale Normen bilden sich nicht nur auf Gesellschafts- oder Unternehmensebene sondern auch auf Gruppenebene (siehe Abschnitt 3.7.2). Es gibt vier Normen, die in Arbeitsgruppen fast immer von allen Mitgliedern geteilt werden (Fincham, Rhodes 2003, S. 193). *Fairness* bedeutet, dass alle Mitglieder ihren Fähigkeiten entsprechend an der Zielerreichung mitarbeiten. *Reziprozität* bedeutet, dass Mitglieder empfangene Hilfeleistungen erwidern.

*Vernünftig* bedeutet, dass alle Gruppenmitglieder sich gemäß gesellschaftlichen Normen verhalten, also z.B. nicht betrügen oder bestechlich sind. *Rollengerecht* bedeutet, dass alle Mitglieder die Erwartungen an die ihnen zugewiesene Rolle erfüllen. Normen sind dabei i. d. R. mit einer gewissen Toleranzgrenze verbunden, innerhalb der bei Verstoß keine Sanktionen mit dem Verhalten verbunden sind. Dieser Spielraum ist auch als *Akzeptanzzone* bekannt (ibid.).

Gruppennormen sind demnach Verhaltensstandards, die auf weit verbreiteten Erwartungen an das Verhalten von Gruppenmitgliedern basieren (Fehr, Fischbacher 2004, S. 185). Innerhalb von Organisationen sind z.B. Normen relevant, die auf den Arbeitseinsatz abzielen. Ein Beispiel ist das Free Rider Problem, bei dem einzelne Gruppenmitglieder davon profitieren, dass ihr mangelnder Arbeitseinsatz durch andere abgedeckt wird (ibid.). Die Belohnung der Gruppe für das Erreichen des Ziels kommt dabei trotzdem allen Gruppenmitgliedern zugute und stellt demnach ein öffentliches Gut für die Gruppenmitglieder dar (ibid.). Aber auch die Einhaltung von Normen selbst kann für die Gesellschaft als öffentliches Gut verstanden werden, da alle Mitglieder von der Einhaltung profitieren (Castelfranchi et al. 1998, Absatz 1.3). Empirische Untersuchungen zeigen (Fehr, Fischbacher 2004, S. 186), dass die Hälfte der Probanden ihren eigenen Arbeitseinsatz annähernd proportional zur wahrgenommenen Anstrengung der Gruppe steigern (bedingte Kooperation). Weitere 30% der Probanden zeigen unabhängig von der Gruppenanstrengung kaum Arbeitseinsatz (Free Rider). 14% der Probanden zeigten bis zu einem Schwellenwert bedingte Kooperation, bei weiter steigender Gruppenanstrengung schränkten sie jedoch ihren eigenen Arbeitseinsatz systematisch ein, so dass ein hügelartiger Kurvenverlauf entstand.

Weitere Experimente mit Dritten zeigen, dass mangelnde Kooperation in 50% der Fällen negativ sanktioniert wird, obwohl die Beobachter weder von der Defektion betroffen waren noch Vorteile aus der Sanktionierung ziehen konnten sondern im Gegenteil sogar Kosten (z.B. unangenehme Situation, Verlust der Beziehung) damit verbunden sein konnten (Fehr, Fischbacher 2004, S. 186). Dabei fällt die Sanktion umso stärker aus, je weiter der Arbeitseinsatz vom Gruppendurchschnitt nach unten abweicht (ibid. S. 188).

Blieben Normverletzungen ohne externe Sanktionen, so sank in einem weiteren Experiment die Kooperationsrate im Zeitverlauf von anfänglich rund 50% auf unter 20%. Das Wissen um eine mögliche Sanktion hob die Kooperationsrate sprunghaft auf 65% an

(Fehr, Fischbacher 2004, S. 187). Die Erfahrung, dass tatsächlich Sanktionen erfolgen steigerte die Rate im Zeitverlauf rasch auf über 90%.

Normen wirken nicht nur begrenzend auf das Verhalten von Personen, sondern fordern auch dazu auf, aktiv ihre Einhaltung zu bewirken (Castelfranchi et al. 1998, Absatz 1.3). Die volle Akzeptanz einer Norm ist erst dann erreicht, wenn man sich sowohl selber an sie hält als auch eine Verletzung durch andere nicht toleriert (ibid.).

**Auswirkungen auf die Wissensweitergabe:** Auch für die Wissensweitergabe existieren in vielen Organisationen Normen. Leitsätze wie „angesehene Experten teilen ihr Wissen gerne“ oder „das Wissen unserer Mitarbeiter ist das höchste Gut des Unternehmens und wird unter Mitarbeitern mit Freude weitergegeben“ stehen für den Versuch des Managements eine Organisationskultur zu etablieren, die Wissensweitergabe positiv sanktioniert (siehe auch Abschnitt 3.7). Normen können der Wissensweitergabe aber auch im Wege stehen, wenn z.B. das ungeschriebene Gesetz gilt, dass man andere Kollegen nicht mit Fragen belästigt. Dadurch werden Pull-Gelegenheiten für Wissensweitergabe verhindert und das gesamte soziale Beziehungsnetzwerk am Arbeitsplatz geschwächt.

#### **3.6.3 Macht**

Macht kann als das Ausmaß definiert werden, mit dem Handlungen anderer über Kontrolle beeinflusst werden können (Gray 2001, S. 371). Im Panopticon von Foucault wird Macht über ständige Beobachtungsmöglichkeiten und damit Kontrolle dargestellt (Fincham, Rhodes 2003, S. 263f). Diese wird in heutigen Organisationen über das Berichtswesen realisiert. Aber auch über langjährige Erfahrung und gute Ausbildung, wie sie Experten auszeichnet, kann Macht entstehen (Fincham, Rhodes 2003, S. 264). Experten werden um ihre Einschätzungen über Dinge und Personen gebeten und haben dadurch eine Art normative Macht. Überspitzt dargestellt wird ihre Meinung als Tatsache oder allgemeingültiges Gesetz angesehen (ibid.). Ein Bereich, in dem dieses Phänomen ebenfalls deutlich wird ist das Erschaffen von neuen Fachausdrücken, die dann zu Modewörtern werden und damit großen Einfluss ausüben (ibid.). Es ist damit nicht das Wissen selbst, sondern die Kontrolle des Wissens das die Quelle für Macht darstellt (Reiff 1974, S. 451).

Foucault erklärt aber nicht, was die zentrale Motivation für Machterwerb und -ausübung ist, da Macht bei ihm nicht auf eine identifizierbare Person beschränkt ist, sondern als ein Netzwerk von sozialen Beziehungen und Mechanismen betrachtet wird (Fincham, Rhodes 2003, S. 265). Macht als Motivation ist interessant, weil mit mehr Macht oft auch mehr Respekt, mehr Privilegien und höhere Entlohnung einhergehen (Weinert 1998, S. 596).

Macht kann auch durch soziale Austauschbeziehungen entstehen (siehe Social Exchange Theory im Abschnitt 3.6.1). Wenn diese Austausche einseitig sind, entstehen Abhängigkeitsverhältnisse, die als Macht interpretiert werden können (Zafirovski 2005, S. 7). Zwei oder mehr solcher Abhängigkeitsbeziehungen bilden ein Machtnetzwerk, welches zur Geschlossenheit neigt (siehe Abschnitt 3.6.1). Ein wichtiger Prozess innerhalb solch eines Netzwerks ist die Legitimation oder Transformation der Macht in Autorität (ibid.). In der sozialwissenschaftlichen Literatur wird Macht z.T. als Zugang zu Ressourcen und deren Austausch konzeptualisiert (Zafirovski 2005, S. 6). Macht kann aber auch als strukturelles Potential betrachtet werden, das aus den Mustern und der Qualität von Austauschmöglichkeiten entsteht (Lawler, Thye 1999, S. 226).

Ein weiterer Aspekt von Machtbeziehungen ist, dass es i. d. R. auch Widerstände gegen die Ausübung von Macht gibt (Fincham, Rhodes 2003, S. 277). So bilden sich oft Gruppennormen, die den Arbeitseinsatz auf das Minimum dessen beschränken, was durch Kontrolle der Mächtigen gerade noch toleriert wird (ibid.). Dies wurde v.a. für manuelle Tätigkeiten untersucht, gilt aber auch für Fachleute und Experten (ibid.). So wurde festgestellt, dass Programmierer nach kontinuierlicher Beschneidung ihrer Autonomie und Selbstkontrolle durch steigende Reglementierung mit Vorgaben vom Management zu kontraproduktiven Gegenmaßnahmen griffen. Sie sprachen den vom Management befragten Analysten den Sachverstand ab und hielten benötigte Informationen zurück, um eine Gegenmachtposition aufzubauen (ibid.). Zudem wurden vermehrt für Laien unverständliche Fachbegriffe in Diskussionen verwendet, um die Unabkömlichkeit der Programmierer zu unterstreichen. Ein gewisser Grad an Autonomie, also Abwesenheit von Kontrolle durch Mächtige, ist demnach eine Grundvoraussetzung für das Vertrauen, dass weitergegebenes Wissen nicht zum eigenen Schaden eingesetzt wird, und ermöglicht somit erst Wissensweitergabe.

Neue Arbeitssysteme wie autonome Arbeitsgruppen können klassische Kontrollsysteme meist nicht vollständig ersetzen sondern existieren parallel dazu (Fincham, Rhodes 2003, S. 279). Kontrolle durch Peer Review kann z.B. unerwünschte Effekte erzeugen. Diese Form der Kontrolle ist zwar schwerer zu hintergehen als Kontrolle durch den Vorgesetzten, allerdings wird auch das Vertrauen in die Gruppenmitglieder und damit eine wichtige Basis für die erfolgreiche Zusammenarbeit zerstört (ibid.). Weiterhin bilden sich oft Gruppennormen aus, die das System unterminieren, indem negative Bewertungen von Gruppenmitgliedern durch die Gruppe sanktioniert werden (ibid.).

Eine weitere Möglichkeit Widerstand gegen vom Management ausgeübte Macht zu leisten besteht in der von Collinson „resistance through persistence“ genannten Strategie (Fincham, Rhodes 2003, S. 283). Dabei versuchen Mitarbeiter mehr Wissen über die internen Abläufe der Organisation zu erlangen und durch geschickten Einsatz dieses Wissens stärker in die Entscheidungsprozesse eingebunden zu werden (ibid.). Durch Sammeln von Informationen, Beobachten von Arbeitspraktiken und Hinterfragen von Entscheidungen gewinnen sie an Einfluss (ibid.). In dieser Form besteht also ein direkter Zusammenhang zwischen Macht und Wissen.

**Auswirkungen auf Wissensweitergabe:** Wissensträger sind (opportunistisches Verhalten unterstellt) zur Weitergabe ihres Wissens eher bereit, wenn es ihren eigenen Interessen nützlich erscheint (Schindler 2001, S. 81). Macht ist dabei ein wichtiger Einflussfaktor. Der bekannte Ausdruck „Wissen ist Macht“ wird dem britischen Philosophen Sir Francis Bacon (1561-1626) zugesprochen und spielt darauf an, dass Wissen als Machtinstrument eingesetzt werden kann, wenn Macht pragmatisch als Einfluss auf die Handlungen anderer definiert wird (ibid.).

Der Einsatz von Wissen als Machtinstrument wird von mehreren Autoren diskutiert, deren Ergebnisse Schindler folgendermaßen zusammenfasst: in solchen „politisierten“ Organisationen können Prozesse des Wissensmanagement gefördert oder behindert werden, je nach dem strategischen Kalkül der Wissensträger (Schindler 2001, S. 81).

Die Gleichung „Wissen = Macht“ kann auch mit dem Besitztumseffekt in Verbindung gebracht werden. Dieser besagt, dass Personen ein Objekt, das sie besitzen wesentlich höher bewerten, als dies Leute tun, die das Objekt nicht besitzen (Jost 2000, S. 210f). Auf Wissen übertragen würde das bedeuten, dass eigenes Wissen als so einmalig und wertvoll wahrgenommen wird, dass man es lieber nicht weitergibt, um die subjektive Einmaligkeit nicht zu verlieren, obwohl andere Mitarbeiter das Wissen zwar gebrauchen können, aber ihm keinen übermäßig großen Wert zusprechen würden.

Bender und Fish halten dem entgegen, dass die klassische Formel „Wissen gleich Macht, also horte es“ in der heutigen Zeit durch die Gleichung „Wissen gleich Macht, also teile es, damit es sich vervielfacht“ ersetzt werden müsste (Bender, Fish 2000, S. 134). Trotz der Erkenntnis, dass Ideen den größten Einfluss ausüben, wenn sie weit verbreitet werden, wird Wissen traditionell eher gehortet. Die Abneigung zur Wissensweitergabe ist kulturell geprägt und kann nicht leicht abgebaut werden (ibid.). Dies wird durch Ängste vor Verlust

des Arbeitsplatzes oder vor mangelnder Beförderung noch verstärkt (Bender, Fish 2000, S. 134).

Auch die Einführung von Informationssystemen (IS) kann zum Aufbau von Macht genutzt werden, indem die Daten im IS dazu genutzt werden, die Arbeit und Leistung der Mitarbeiter zu kontrollieren, die mit diesem IS arbeiten (Gray 2001, S. 374). Im Falle von Wissensrepositories ist der Fall jedoch etwas anders, da nicht die Automatisierung sondern der Wissensaustausch im Vordergrund steht und die Beiträge zum einen qualitativ schwierig zu beurteilen sind und andererseits die Quantität nur sehr begrenzte Aussagekraft hat (ibid.). Dennoch ermittelt Gray einen Machtverlust für Mitarbeiter einer Organisation durch die Einführung von Wissensrepositories, da ihre Einzigartigkeit sinkt und ihre Ersetzbarkeit steigt (Maier 2004, S. 380).

#### **3.6.4 Reputation**

Reputation ist eng mit dem Konzept der Macht verknüpft. Hohe Reputation bedeutet oft auch große Macht (Weinert 1998, S. 596). In diesem Abschnitt soll weniger der mit Macht zusammenhängende Anteil von Reputation untersucht werden, sondern die Reputation als Experte. Synonym zu Reputation finden sich in der Literatur auch die Begriffe Ansehen und Status.

Reputation bezeichnet die Wahrnehmung, die ein Individuum von den Intentionen und Normen anderer hat, und ist eng mit Vertrauen verbunden (Mui 2003, S. 20). Reputation ist ähnlich wie Expertise i. d. R. auf einen engen Bereich begrenzt, also kontextgebunden (ibid. S. 25). Sie kann entweder durch direkte Beobachtungen des Verhaltens eines Individuums entstehen oder durch indirekte Erfahrungen (ibid. S. 27). Diese indirekte Reputation kann aus Vorurteilen, Berichten anderer, oder auch aus der Mitgliedschaft in einer sozialen Gruppe abgeleitet werden (ibid. S. 29f).

Reputation hat drei verschiedene Aspekte (Mui 2003, S. 23). *Individuelle Reputation* bezeichnet die Einschätzung der Eindrücke von einem Individuum durch andere. *Soziale Reputation* bezeichnet die Eindrücke von Individuen, die vor dem Hintergrund der Gruppenzugehörigkeit von anderen eingeschätzt werden. *Ontologische Reputation* bezeichnet die kontextspezifische Facette der Reputation. In der qualitativen Soziologie wird dagegen zwischen zugeschriebener, erreichter und verdienter sowie erzwungener Reputation unterschieden (Mui 2003, S. 25).

Im ökonomischen Kontext wurde Reputation v.a. im Zusammenhang mit spieltheoretischen Modellen untersucht (Mui 2003, S. 22). Dabei entwickelt sich ein Konstrukt, das als Reputation bezeichnet werden kann in öffentlich beobachtbaren, teilweise öffentlich überwachten und z.T. auch in privat überwachten, wiederholten Gefangenendilemma Spielen (ibid.).

In der Soziologie wird Reputation meist als Parameter von sozialen Netzwerken untersucht und dementsprechend als eines von verschiedenen Zentralitätsmaßen konzeptualisiert (Mui 2003, S. 25).

Die in Abschnitt 3.3.6 beschriebene Attribution von Ursachen auf die Person oder Situation kann auch weitergehend verstanden und auf die Attribution möglicher Motive für das Zeigen eines bestimmten Verhaltens ausgedehnt werden (vgl. Staw 1975). Diese Motive können bei mehrfachem Auftreten des Verhaltens als Persönlichkeitszüge interpretiert werden. Eine mit der Attributionstheorie verwandte Theorie ist die *Personal Construct Theory* von Kelly (Fincham, Rhodes 2003, S. 174ff). Ein Konstrukt besteht aus zwei gegensätzlichen Adjektiven, die von Personen benutzt werden, um ihre Wahrnehmungen einzuordnen. Einer Person werden demnach anhand des gezeigten Verhaltens, aber auch schon anhand ihrer äußeren Erscheinung einzelne Adjektive zugeschrieben, die diese Person in der Wahrnehmung des Beobachters charakterisieren. Dabei sollten die vergebenen Adjektive im weiteren Verlauf der sozialen Interaktion ständig zur Disposition stehen, damit vorschnell gebildete Urteile das Bild des anderen trotz widersprüchlicher neuer Informationen nicht weiterhin bestimmen (ibid.). Für diese Zuschreibung von Adjektiven sind die Gemeinsamkeit der Zuschreibungsprozesse (communality) und der bewusste Versuch, die Denkprozesse des Anderen bei der eigenen Meinungsbildung zu berücksichtigen, wichtig (sociality, Fincham, Rhodes 2003, S. 176).

Die Aspekte einer Person die zur Bildung von Reputation führen sind je nach Branche sehr unterschiedlich. So erlangt ein Forscher hauptsächlich durch Veröffentlichungen in hochrangigen Fachzeitschriften und der anschließenden häufigen Zitation dieser Veröffentlichungen Reputation (Mui et al. 2002, S. 2; Mui 2003, S. 22).

Reputation kann sich aber nicht nur auf Individuen, sondern auch auf ganze Berufsgruppen beziehen. Johnson unterscheidet drei Klassen von Fachleuten mit hoher Reputation, nämlich Berufsgruppen in denen Experten dominieren (collegiate type), wie z.B. Ärzte, Anwälte oder Wissenschaftler, Berufsgruppen in denen das Ansehen des Arbeitgebers den Status bestimmt (state mediation), wie z.B. Richter, Lehrer oder Polizisten und schließlich

Berufsgruppen, bei denen die vom Arbeitgeber vorgegebene Tätigkeit das Ansehen bestimmt (corporate patronage), wie z.B. Ingenieure, Computerfachleute oder Wirtschaftsprüfer (Fincham, Rhodes 2003, S. 347). Computerfachleute versuchen aber im Gegensatz zu anderen (z.B. Wirtschaftsprüfern) weniger ihren Status durch Institutionalisierung zu behaupten sondern haben eher individuelle Strategien zu dessen Erhalt basierend auf kontinuierlichem Wissenserwerb (ibid. S. 348).

Auch mit den Rollen, die ein Mitglied einer Gruppe erfüllt, ist eine gewisse Reputation verbunden. Die Grundlage dafür können persönliche Leistung, Länge der Organisationszugehörigkeit, Alter oder Erfahrung sein (Edelmann 1996, S. 364). Reputation wirkt als Motivator (siehe Abschnitt 3.3.2) und kann in Form von Titeln, hoher Bezahlung, großem Verantwortungsbereich, großzügigen Büroausstattungen oder anderen Annehmlichkeiten übertragen werden (ibid.). Neben den von Edelmann genannten Faktoren spielen nach der Erfahrung des Autors in der Softwareentwicklung v.a. Firmenwägen und technische Geräte (sog. Gadgets, z.B. ein PDA, Personal Digital Assistant) eine Rolle. Reputation kann aber auch informell erworben werden, z.B. durch besondere Expertise oder Zusatzausbildungen (Edelmann 1996, S. 365).

***Auswirkungen auf die Wissensweitergabe:*** Reputation ist neben dem Tausch Wissen gegen Wissen der wichtigste Einflussfaktor in elektronischen Foren (Waltert 2002, S. 96). Dies ist auch konform mit den Beobachtungen von Boer & Berends (2003) die den Expertenstatus als ein Motiv für die Wissensweitergabe identifizieren, wie es von der Relation Model Theorie als Teil des Authority Ranking Modells vorhergesagt wird. Nicht nur die Quelle der Wissensweitergabe kann dabei im Status steigen sondern auch der Empfänger, da er Umgang mit Experten hat und dementsprechend selber als wissend eingeschätzt wird (vgl. Kilduff 1994, S. 89). Dies entspricht dem Expertenbegriff der soziologischen Schule (vgl. Abschnitt 3.3.3). Im Kontext von Wissensnetzwerken kann Reputation auch als das Ausmaß aufgefasst werden, in dem andere Teilnehmer einen Teilnehmer als wertvoll für das Netzwerk einstufen (Bush, Tiwana 2005, S. 68). So eine Reputation wird nur stückweise über einen längeren Zeitraum aufgebaut, in dem der Teilnehmer immer wieder Erfahrungen und Ideen beiträgt und dadurch als hilfreich und wertvoll angesehen wird (ibid.).

### 3.6.5 Vertrauen

Vertrauen kann als vierdimensionales Konzept definiert werden (Mishra 1996, S. 261). „Vertrauen ist die Bereitschaft einer Partei sich für eine andere Partei verwundbar zu machen, im Glauben dass diese Partei (a) kompetent, (b) offen, (c) betroffen und (d) zuverlässig ist“<sup>21</sup>. So geht man z.B. das Risiko ein, dass ein Kollege eine Idee für sich selbst beansprucht, wenn man eine eigene Idee mit ihm teilt (Chowdhury 2002, S. 3).

Zentrale Elemente in den meisten Definitionen von Vertrauen sind Verwundbarkeit, Risiko und positive Erwartungen über das Verhalten des Vertrauten (Luna-Reyes et al. 2004, S. 2). Ohne Unsicherheit über das zukünftige Verhalten und das damit verbundene Risiko, dass es für die eigene Person zum Schaden sein kann, ist das Konzept des Vertrauens unnötig. Vertrauen ist dann entscheidend, wenn das Verhalten des Vertrauten nicht unter der Kontrolle des Vertrauenden ist (ibid.). Die positiven Erwartungen über das Verhalten des Vertrauten können durch *rationale Überlegungen* (calculus-based trust) entstehen, etwa weil genügend Anreize existieren, die das erwartete Verhalten motivieren, oder durch *Erfahrungen mit dem früheren Verhalten* des Vertrauten entstehen (experience-based trust, ibid.). Die Erwartungen sind dabei auch von Persönlichkeitsmerkmalen abhängig, die zu eher pessimistischer oder zu optimistischer Einschätzung führen können (ibid.). Rationales Vertrauen setzt dabei nicht nur logische Überlegungen voraus sondern auch, dass der Vertrauensgeber den Vertrauensnehmer als rationalen Akteur betrachtet (Huotari, Iivonen 2004, S. 8).

Ökonomisch betrachtet wird Vertrauen eingesetzt, um unsichere Erwartungen zu stabilisieren und die mit der Unsicherheit einhergehende Komplexität zu reduzieren (Ripperger 2003, S. 13). Zum Verständnis der unterschiedlichen Aspekte von Vertrauen ist es daher nötig, verschiedene Ursachen für die Unsicherheit zu kennen (ibid. S. 14). Personen sind nicht dazu in der Lage, die Komplexität ihrer Umwelt zu beherrschen, sondern sind von ihr überfordert und müssen das Spektrum zukünftiger Möglichkeiten eingrenzen, um handlungsfähig zu bleiben (ibid. S. 35). Dieses Verständnis von Vertrauen setzt begrenzte Rationalität und die Möglichkeit für opportunistisches Verhalten voraus (ibid. S. 41f). Der extrinsische Wert von Vertrauen besteht in der Reduktion der Transaktionskosten

---

<sup>21</sup> Im Original: “Trust is one’s party willingness to be vulnerable to another party based on the belief that the latter party is (a) competent, (b) open, (c) concerned, and (d) reliable.”

(Nooteboom 2002, S. 3), da Vertrauen anstelle von Informationsbeschaffung, Überwachung und Kontrolle tritt.

Dabei stellt die alltägliche Definition von Vertrauen auf Annahmen über die Verlässlichkeit von Eigenschaften einer Person ab (Ripperger 2003, S. 36). Nooteboom legt einen erweiterten Vertrauensbegriff zugrunde und bezeichnet das hier behandelte Vertrauen als verhaltensbasiertes Vertrauen (Nooteboom 2002, S. 30). Weiterhin muss Vertrauen von Zuversicht abgegrenzt werden. Zuversicht ist eine generelle Reaktion auf die ständig präsenten Unsicherheiten im Leben (z.B. das Essen könnte vergiftet sein, Elektrosmog ist überall und macht krank), um nicht in ständiger Angst und Unsicherheit leben zu müssen (ibid.). Vertrauen dagegen ist eine mögliche Reaktion auf spezifische Risikosituationen. Vertrauen ist das Resultat einer bewussten Entscheidung nach Abwägen anderen Handlungsalternativen (Ripperger 2003, S. 37). Im Gegensatz zu Hoffnung, die im Umgang mit Risiken exogener Natur eingesetzt wird, bezieht sich Vertrauen auf Risiken endogener Natur (ibid. S. 38). Eine weitere Unterscheidung betrifft die Ursache der Unsicherheit. Vertrauen kann sich auf die Fähigkeiten einer Person beziehen, ein gewisses Verhalten zu zeigen, oder auf deren Motivation dazu (Ripperger 2003, S. 38; Nooteboom 2002, S. 9). Ersteres bezeichnet Ripperger als Zutrauen und ist für die Wissensweitergabe als ebenso relevant zu betrachten wie das Vertrauen auf das Wollen des Interaktionspartners.

In frühen Phasen einer sozialen Beziehung stehen noch keine Erfahrungen zur Verfügung, die zur Vertrauensbildung genutzt werden können. Deshalb ist in diesen frühen Phasen das rationale Vertrauen (calculus-based) wichtiger, während mit zunehmender Erfahrung mit der anderen Person das erfahrungsbasierte Vertrauen (knowledge-based) überwiegt (Luna-Reyes et al. 2004, S. 6). Die Rolle von geteiltem Verständnis und geteilten Werten für die Bildung von Vertrauen wird von mehreren Autoren betont (Huotari, Iivonen 2004, S. 9). Sie stellen den Rahmen für die Vertrauensbildung dar und erschaffen Vorhersehbarkeit des Verhaltens und damit Vertrauenswürdigkeit. Diese Art von Vertrauen wird aufgrund der zugrunde liegenden sozialen Normen auch als normatives oder wertbasiertes Vertrauen bezeichnet (ibid.). Geteiltes Verständnis ist aber auch für die Bildung von flüchtigem Vertrauen wichtig (swift trust, Huotari, Iivonen 2004, S. 17). Es kann relativ schnell entstehen, wenn bei sozialen Interaktionen die Rollen und Verantwortlichkeiten klar definiert sind und mit Meinungsverschiedenheiten effektiv umgegangen wird (ibid.).

Vertrauen wird auch als pfadabhängiges Phänomen beschrieben (Luna-Reyes et al. 2004, S. 4). Das bedeutet, dass kleine Ereignisse zu Beginn der sozialen Beziehung, die z.T. auch

zufällig auftreten können, über das Endergebnis, also Vertrauen oder Misstrauen entscheiden können (ibid.). Dies gilt jedoch vor allem für das Misstrauen, da die Vertrauensbildung ein langsamer Prozess ist, während sich Misstrauen sehr schnell schon bei kleinen Unregelmäßigkeiten im Verhalten der anderen Person bildet (ibid. S. 5). Die Pfadabhängigkeit führt dazu, dass Extremwerte auf der Vertrauens-Misstrauens-Skala häufiger vorkommen als erwartet.

Vertrauen kann auch aus entscheidungstheoretischer Sicht betrachtet werden und steht dann als Alternative zur Informationsbeschaffung und -analyse einerseits und Kontrolle andererseits (Volken 2002, S. 3). Ähnlich sieht auch Ripperger das Konzept, und definiert Vertrauen als die „freiwillige Erbringung einer riskanten Vorleistung unter Verzicht auf explizite vertragliche Sicherungs- und Kontrollmaßnahmen gegen opportunistisches Verhalten in der Erwartung, dass sich der andere trotz Fehlen solcher Schutzmaßnahmen nicht opportunistisch verhalten wird“ (Ripperger 2003, S. 45). Auf dieser Definition aufbauend kann das Vertrauensverhältnis als Principal-Agent-Beziehung dargestellt werden, in der der Principal das Vertrauen gegenüber dem Agent ausspricht (Vertrauensgeber) und damit der Agent zum Vertrauensnehmer wird (ibid. S. 76). Damit ist eine Art impliziter Vertrag geschlossen. Der Agent kann durch sein Verhalten das Vertrauen entweder honorieren oder enttäuschen (ibid. S. 74). Durch diese Betrachtung wird verdeutlicht, dass die klassischen Probleme einer Principal-Agent-Beziehung, nämlich Adverse Selection, Hold Up und Moral Hazard auch für das Vertrauensverhältnis eine Rolle spielen<sup>22</sup>.

Vertrauen ist auch eine reziproke soziale Beziehung und das Aussprechen von Vertrauen kann auch beim Vertrauensnehmer etwas verändern (Nooteboom 2002, S. 8). Ripperger betrachtet das implizit mit einer Principal-Agent-Perspektive, geht jedoch auf die sozialen

---

<sup>22</sup> Adverse Selection bezeichnet die ex ante existierenden Informationsasymmetrien, die dadurch auftreten, dass dem Principal wesentliche Eigenschaften des Agent vor dem eingehen des Vertrags nicht bekannt sind (hidden characteristics). Signalling und Screening werden als risikobegrenzende Mechanismen vorgeschlagen.

Moral Hazard bezeichnet die ex post anfallenden Informationsasymmetrien, die durch mangelnde Beurteilbarkeit des Anstrengungsniveaus des Agent entstehen (hidden action). Monitoring und explizite vertragliche Anreize können hier risikosenkend wirken.

Hold Up bezeichnet das Risiko von ex post opportunistischem Verhalten, welches durch einseitige Abhängigkeiten des Principal vom Agent entsteht (Ripperger 2003, S. 68ff).

Auswirkungen nicht näher ein. Vertrauen kann vom Vertrauensnehmer um des Vertrauens selbst willen gesucht werden, da es als positiver Nutzenbeitrag empfunden wird, als vertrauenswürdig zu gelten (Nooteboom 2002, S. 3). Hierbei wird auch die Verbindung zu sozialen Normen und zur Reputation deutlich. Vertrauen nicht zu enttäuschen ist eine weit verbreitete Norm und als vertrauenswürdig zu gelten kann als positive Reputation aufgefasst werden (ibid.). Für eine ausführlichere Betrachtung der angesprochenen Charakteristika siehe (Nooteboom 2002).

**Auswirkungen auf die Wissensweitergabe:** Chowdhury findet in der Literatur mehrere Belege dafür, dass interpersonales Vertrauen förderlich für die Kooperation ist (Chowdhury 2002, S. 4). Durch Vertrauen erhöhen sich die Anzahl der Interaktionen bei einer Kooperation und ist daher entscheidend für das Erschaffen neuen Wissens (Huotari, Iivonen 2004, S. 15).

Umgekehrt bietet kooperatives Handeln die Gelegenheit Vertrauen aufzubauen (Luna-Reyes et al. 2004, S. 4). Für die Weitergabe von Wissen im Rahmen unternehmensinterner Lernprozesse ist das gegenseitige Vertrauen eine elementare Voraussetzung (Schindler 2001, S. 82f). Dies betrifft insbesondere die Phasen eins und sieben des Wissensweitergabemodells (siehe Abschnitt 2.2.4). Vertrauen erleichtert Lernprozesse und viele Entscheidungen über den Austausch von Wissen erfordern ein bestimmtes Maß an Vertrauen (Huemer et al. 1998, S. 124 und 140; Nonaka 1994, S. 24f). Der subjektive Wert des Wissens bestimmt dabei die Bedeutung des Vertrauens (Zucker et al. 1996, S. 92). Vertrauen ist besonders in wissensintensiven Geschäftsbereichen eine essentielle Eigenschaft, da mit der dort vorhanden hohen Komplexität und Unsicherheit kaum anders umgegangen werden kann (Huotari, Iivonen 2004, S. 10).

Schindler unterscheidet für die Wissensweitergabe mehrere Ebenen des Vertrauens (Schindler 2001, S. 83), (1) empfängerzentriertes Vertrauen, (2) senderzentriertes Vertrauen und (3) konstituierendes Vertrauen. Empfängerzentriertes Vertrauen stellt auf die Qualität bzw. Echtheit des Wissens basierend auf seiner Einschätzung der Quelle des Wissens. Senderzentriertes Vertrauen bezieht sich auf die Erwartungen der Quelle, dass das Wissen nicht so eingesetzt wird, dass der Quelle in irgendeiner Weise Nachteile entstehen. Konstituierendes Vertrauen schließlich ist eine elementare Form des Vertrauens (eine Art Grundvertrauen), die wenig bewusst wahrgenommen wird, und bildet die Basis für die anderen beiden Vertrauensarten (ibid.).

Vertrauen kann nicht nur in einzelne Personen sondern auch in Gruppen oder soziale Systeme der Makroebene bestehen (Nooteboom 2002, S. 8). Dies kann für die Wissensweitergabe von Bedeutung sein, da das in enger Beziehung stehende Konzept der Reputation sowohl auf Individuen als auch auf Gruppenebene als Einflussfaktor empirisch belegt werden konnte (siehe Ensign, Hébert 2004). Phänomene der Makroebene wie Organisationen bilden aber auch einen Rahmen für Vertrauen (Nooteboom 2002, S. 30ff), so dass z.B. Mitgliedern der eigenen Organisation auch ohne Erfahrungen mit der speziellen Person vertraut wird.

Neben den Auswirkungen von Vertrauen auf die Wissensweitergabe gibt es auch einen Einfluss des Wissens und der Wahrnehmung auf Vertrauen. Es ist vielfach nicht so entscheidend, was passiert, sondern wie es vom Beobachter interpretiert wird und welche Schlüsse er daraus über die Motive oder Verantwortlichkeit beteiligter Personen zieht, ob Vertrauen gebildet oder zerstört wird (Nooteboom 2002, S. 24). Hier spielt wieder die Attributionstheorie eine Rolle (siehe Abschnitt 3.3.6). Nooteboom findet sogar Argumente für eine evolutionäre Verankerung von Vertrauen (analog dem konstituierenden Vertrauen von Schindler) im Gedächtnis (ibid. S. 26), was dem Grundvertrauen entspricht, von dem bei Kindern oft gesprochen wird. Vertrauen stärkt die kognitive und die relationale Dimension von sozialem Kapital, indem es hilft bei der Interpretation von Sprechakten nicht ständig die möglichen negativen Konnotationen und Hintergedanken zu suchen, sondern von einer positiven Interpretation auszugehen (Huotari, Iivonen 2004, S. 15). Aber auch die strukturelle Komponente sozialen Kapitals kann durch Vertrauen gestärkt werden, da sich die Häufigkeit von Kontakten erhöhen kann (ibid. S. 16).

#### **3.6.6 Diskussion**

Auch in diesem Kontextbereich fällt auf, wie eng die einzelnen Konzepte miteinander verzahnt sind. Vertrauen in die Fähigkeiten eines Kollegen ist nur analytisch von der Reputation als Experte des Kollegen zu unterscheiden. Soziale Normen regeln, dass Vertrauen nicht enttäuscht werden sollte. Wird man als Experte anerkannt, so besitzt man auch eine gewisse normative Macht. All diese Konzepte beeinflussen, wie sich soziale Netzwerke bilden, und aus strukturellen Indizien der Netzwerke kann man Rückschlüsse über die Macht und Vertrauensbeziehungen zwischen den Mitarbeitern ziehen. Die Wissensweitergabe als soziale Austauschbeziehung ist davon ebenso betroffen. Die Relation Model Theorie (Fiske 1991) kann als Rahmen für die Interpretation der Einflüsse dienen. Betrachtet man die Motive hinter den vier Beziehungsmodellen, so steht Authority

Ranking für Austauschbeziehungen die auf sozialem Status und Autorität, also z.B. der Reputation als Experte und Macht innerhalb einer Gruppe beruhen. Bei Boer und Berends werden diese beiden Facetten ebenfalls unterschieden und mit ARe (Authority Ranking basierend auf Expertise) und ARf (Authority Ranking basierend auf formeller Autorität) bezeichnet (Boer, Berends 2003). Wissen wird dabei weitergegeben um den eigenen Status zu untermauern, oder aber um vom Status des Empfängers zu profitieren. Communal Sharing steht für Austauschbeziehungen in kleinen Gruppen, in denen das Vertrauen zwischen den Mitgliedern sehr groß ist und man nach Fiske keine Gegenleistung erwartet. Der Autor ist aber der Auffassung, dass die weitere Gruppenmitgliedschaft, also soziale Zugehörigkeit durchaus als Gegenleistung aufgefasst werden kann. Es werden somit Geborgenheits- und Sicherheitsbedürfnisse aktiviert (siehe Abschnitt 3.3.2). In solchen Kleingruppen bilden sich auch eigene Gruppennormen, in denen Wissensweitergabe nicht nur positiv bewertet wird sondern auch unterlassene Wissensweitergabe negativ sanktioniert wird. Die Wirkung der Stärke der sozialen Beziehung auf die Art des weitergegebenen Wissens ist ebenfalls bemerkenswert und muss in einem Domänenmodell berücksichtigt werden.

### **3.7 Organisationskontext**

Dem Organisationskontext als letzten hier diskutierten Kontextbereich können eine Reihe der in der Empirie identifizierten Einflussfaktoren zugeordnet werden. Eine Kategorisierung der Faktoren ist somit nötig und wird in Tabelle 3-12 versucht.

Song et al. identifizieren Langzeitorientierung und Organisationskrisen als förderlich für Wissensweitergabe und Redundanz als hinderlich für Wissensweitergabe in US High-tech Organisationen (Song et al. 2003). Langzeitorientierung war aber in den vergangenen Jahren für Organisationen in der Softwareentwicklungsbranche schwierig, da die Auftragslage von 2002 bis 2006 für viele sehr schlecht war und einige Organisationen (insbesondere kleine und mittelständische) um das unmittelbare Überleben kämpfen mussten.

Tabelle 3-12: Einflussfaktoren im Organisationskontext

Kategorie	Einflussfaktor
Organisations- kultur	Offene Atmosphäre (Handzic)
	Gemeinschaftsgefühl (Handzic), kollektive Identität (B/K)
	(Organisations-)Kultur (Sarker/Joshi, Boer)
	Aufgeschlossene Kultur für Wandel und Innovation (Ladd/Ward)
	Kultur mit Aufgabenorientierung (Ladd/Ward)
	Kultur der Herausforderung und des Wettbewerbs (Ladd/Ward)
	Vorherrschendes Relation Model (Boer)
Organisations- struktur	Arbeitsteilung (Boer)
	Feedback-Mechanismen (Song)
	Formelle Organisation (Struktur, Autonomie, Kontrolle) (B/K)
	Teams (Song)
Strategie	Langzeitorientierung (Song)
	Organisatorische Redundanz (überlappende Skills/Aufgaben) (Song)
	Wettbewerb zwischen Organisationseinheiten um Ressourcen oder Marktanteile (B/K) => Profitcenter
	Kongruenz zwischen individuellen und organisatorischen Zielen (Ladd/Ward)
Organisations- krise	Externer Druck, starke Konkurrenz, schwacher Markt (Song)
Management- instrumente	Aufstiegschancen (Hendriks)
	Anreizsysteme (Wah, B/K)
	Organisatorische WM-Instrumente (B/K)
	Arbeitspraktiken, die Kommunikation und Imitation von Routinen fördern (B/K)

Organisatorische Redundanz andererseits ist für kleine und mittelständische Organisationen untypisch und eher für große Organisationen zu erwarten. Organisationskrisen sind hingegen aus den oben genannten Gründen in den letzten Jahren in der Softwarebranche vermehrt aufgetreten. Ein bekanntes Beispiel ist die Firma Intershop. Ähnliche Faktoren wie oben für Wissensweitergabe allgemein dargestellt ermitteln auch Hanne und Neu für Lernen und Produktivität in der Softwareentwicklung (siehe Abbildung 3-14).

Der Grad der Hierarchie und die Gruppengröße sind strukturelle Merkmale der Organisation. Der Führungsstil macht sich in der Strategie und in der Anwendung von Managementinstrumenten bemerkbar, während kooperative oder wettbewerbliche

Zusammenarbeit zwischen verschiedenen Abteilungen ebenfalls der Strategie zuzuordnen ist. Der Zusammenhalt schließlich ergibt sich zum großen Teil aus Normen und kann als Facette der Organisationskultur interpretiert werden.

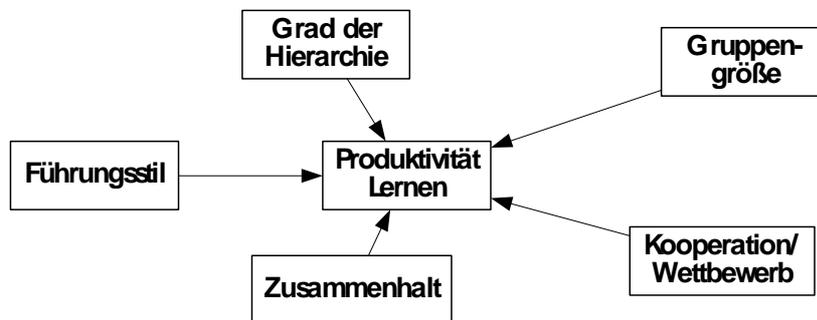


Abbildung 3-14: Organisatorische Einflussfaktoren für Softwareentwicklung (Hanne, Neu 2004, S. 6)

Der Begriff Organisationskultur ist in der Literatur nicht einheitlich belegt (Hendriks 2004, S. 3; Maier 2004, S. 189). Über einige Eigenschaften von Organisationskultur herrscht in der Literatur jedoch weitgehende Übereinstimmung (Maier 2004, S. 189). Demnach ist die Organisationskultur ein implizites Phänomen, das von den Organisationsmitgliedern gelebt wird und ihnen deswegen natürlich und offensichtlich erscheint. Es besteht aus kollektiven Werten, die verhaltensrelevant sind und durch den Umgang mit der internen und externen Umgebung gelernt wurden. Organisationskultur wird in einem sozialen Prozess weitergegeben und bietet Anhaltspunkte für die Interpretation von Verhalten. In der vorliegenden Arbeit wird die Organisationskultur als vorherrschendes Relation Model in der Organisation aufgefasst (siehe Abschnitte 3.3.2 und 3.6.6) und darüber hinaus auf das schwer fassbare Konzept nicht weiter eingegangen. Die folgenden Abschnitte fokussieren daher die Organisationsstruktur, insbesondere Projektarbeit in Teams, sowie das Human Resource Management als ein für Wissensweitergabe besonders relevantes Management-instrument.

#### 3.7.1 Organisationsstruktur

Im Bereich der Organisationsstruktur gibt es nicht nur viele Einflussfaktoren, sondern auch eine Fülle an theoretischen Ansätzen. Eine der klassischen Organisationstheorien ist das *bürokratische Modell von Weber*. Es besteht aus den fünf Elementen Personal, Arbeitsteilung, Hierarchie, Kompetenz und Objektivität (Fincham, Rhodes 2003, S. 331). Die Regeln der Arbeitsteilung schreiben vor, welcher Mitarbeiter welche Aufgaben übernimmt, während die Hierarchie Weisungsbefugnisse festlegt. Beides soll objektiv und einzig basierend auf den Kompetenzen erfolgen (ibid.). Webers Modell wurde als unrealistisch,

unflexibel und unpersönlich kritisiert, da gelebte Bürokratie eher zum genauen Gegenteil dessen führt, was Weber eigentlich bezwecken wollte, nämlich zu unnötiger Schreiarbeit, langen Entscheidungswegen und blinder Regelbefolgung (Fincham, Rhodes 2003, S. 333f). Eine bürokratische Organisationsstruktur wirkt sich auch negativ auf die gerade für wissensintensive Tätigkeiten wichtige Autonomie aus (ibid. S. 348). Webers Modell sieht genau wie Taylors „scientific management“ die Organisation als geschlossenes System (Weinert 1998, S. 519). Mit dieser Sichtweise sind Vernachlässigung von externen Faktoren wie dynamisch veränderlichen Marktbedingungen und internen Faktoren wie sozialen Prozessen zwischen Mitarbeitern verbunden (ibid.).

Die *Systemtheorie* stellt v.a. auf die Einbeziehung von Umweltfaktoren ab und sieht die Organisation als System, das aus Subsystemen besteht und Inputs aus der Umwelt durch Transformationsprozesse in Outputs verwandelt (Weinert 1998, S. 526). Die Grenzen der Organisation sind durchlässig für Einflüsse aus der sozialen, technischen, wirtschaftlichen und politischen Umwelt. Solche Organisationssysteme tendieren dazu, ihre Grundcharakteristika (z.B. Aufbaustruktur) gegen Umwelteinflüsse zu verteidigen (ibid. S. 527).

Die *verhaltensorientierte Organisationstheorie* geht von offenen Systemen aus, betont den einzelnen Mitarbeiter als Entscheidungsträger und betrachtet die Organisation in erster Linie als soziales System (Cyert, March 2001, S. 214f). Die Kernideen der Theorie sind begrenzte Rationalität (siehe Abschnitt 3.3.1), unvollkommene Anpassung an die Umgebung und nicht aufgelöste Konflikte (ibid.). Unvollkommene Anpassung besagt, dass Unterschiede in der inneren Struktur von Organisationen nicht ausschließlich mit unterschiedlichen Umgebungsbedingungen erklärt werden können sondern auch organisationsinterne Anpassungsprozesse betrachtet werden müssen, die suboptimal sein können. Nicht aufgelöste Konflikte existieren zwischen organisatorischen Akteuren, die individuelle Ziele verfolgen, welche in Konflikt zueinander stehen können, was auch durch den Arbeitsvertrag nicht vollständig aufgelöst wird (ibid.). Es müssen daher ständig Kompromisse ausgehandelt werden, die oft schwierig erzielbar und instabil sind (ibid.).

Eine einflussreiche Arbeit im Bereich der *Individuen-Theorie* stammt von March und Simon (1958). Sie betrachtet Organisationen als Systeme von informationsverarbeitenden und entscheidungsfällenden Individuen (Weinert 1998, S. 554; Levitt et al. 1999, S. 4). Dabei werden Entscheidungen auf Basis der Wahrnehmungen aus der Umgebung und den Erinnerungen der Person getroffen. Die Erinnerungen beinhalten Werte und Ziele der Person, die Beziehungen zwischen der Aktion und ihren Resultaten sowie mögliche

Alternativen (Weinert 1998, S. 555). Galbraith hat in der Empirie gezeigt, dass sich dieses Modell für die Analyse von Produktentwicklungsprozessen eignet (Galbraith 1974; Levitt et al. 1999, S. 4). Die informationsverarbeitende Perspektive geht von drei wesentlichen Annahmen aus (Weinert 1998, S. 578). (1) Organisationen sind offene Systeme und mit Unsicherheit konfrontiert, die durch das Sammeln von Informationen reduziert werden kann. (2) Es muss eine Organisations- und Informationsstruktur geschaffen werden, die das Sammeln, Interpretieren, Bewerten und Verarbeiten der Informationen erleichtert. (3) Für die Informationsverarbeitung sind v.a. die Untereinheiten (z.B. Abteilungen, Gruppen) von Bedeutung (ibid.). Kritik an der Individuen-Theorie gilt dem Vernachlässigen von nicht individuen-basierter Einflussfaktoren der Organisationsstruktur (Weinert 1998, S. 551).

Die *ökonomische Theorie* konzeptualisiert Organisationen als aktiven Teil eines ökonomischen Prozesses, der danach strebt, sich ständig weiterzuentwickeln, auszudehnen und zu verbessern (Weinert 1998, S. 562f). Diese Theorie trifft allerdings vorwiegend auf Industrie- und Geschäftsunternehmen zu und ist demnach nicht allgemeingültig (ibid. S. 563). Innerhalb der ökonomischen Organisationen müssen Informationen verarbeitet, Entscheidungen getroffen, Ergebnisse evaluiert und Prozesse gelenkt werden (ibid. S. 565). Die dafür relevanten Konzepte, die es zu untersuchen gilt, sind Einfluss- und Kontrollmechanismen, Autoritätsstrukturen, Kommunikationssysteme und Entscheidungsmechanismen sowie die organisatorischen Aktivitäten (ibid. S. 566). Wichtig sind dabei v.a. die Abhängigkeiten zwischen Rollen, Rolleninhabern, Mitarbeitern und Organisation sowie der Entscheidungsprozesse (ibid. S. 571).

In *Institutionentheorien* werden Organisationen als Systeme konzeptualisiert, die durch normativen Druck der Umgebung (z.B. dem Staat) und die Bestrebung zur Stabilisierung von Strukturen im Inneren bestimmt werden und durch Institutionalisierung auf diese beiden Einflüsse reagieren (Zucker 1987, S. 443). Institutionalisieren bedeutet dabei entweder die regelartige soziale Verankerung von organisatorischen Aktionsmustern oder das Einbetten von Akteuren und Artefakten in formelle Strukturen (ibid. S. 444). Institutionalisierung führt dabei zu geringeren Fehlerraten und Zentralisierung, aber auch zu größeren Beharrungskräften gegen Veränderungen (ibid. S. 448). Durch Institutionalisierung werden auch neue soziale Fakten geschaffen, z.B. durch Erschaffen einer neuen Rolle (ibid. S. 449). Der Fokus der Institutionenökonomik liegt dabei auf Konzepten wie Gesetzen, Verträgen, Hierarchien, sowie Berufsethos und sozialen Normen (Zucker 1987, S. 459).

Die *Lernende Organisation* als letzte hier betrachtete Theorie geht von Argyris aus und hat mit Senge einen bekannten Vertreter. Sie postuliert, dass Organisationen sich nicht nur reaktiv an veränderte Umgebungsbedingungen anpassen dürfen, um langfristig erfolgreich zu sein, sondern auch solche Veränderung antizipieren müssen, um schon vor der Veränderung entsprechende Anpassungsprozesse starten zu können (Weinert 1998, S. 579). Dies ist für Softwareunternehmen besonders wichtig, da sie in einem sehr dynamischen Umfeld agieren (Althoff et al. 2000, S. 349). Lernen tritt dabei entweder als Lernen erster Ordnung (single loop learning) auf, indem die Fähigkeiten der Organisation bestehende Ziele zu erreichen verbessert werden und aus den Erfahrungen der Vergangenheit gelernt wird, oder als Lernen zweiter Ordnung (double loop learning, siehe auch Abschnitt 3.3.4), bei dem auch grundlegende Ziele, Werte und Einstellungen der Organisation hinterfragt und neu bewertet werden (Weinert 1998, S. 580). In Softwareunternehmen tritt meist nur Lernen erster Ordnung auf, da in traditionellen Entwicklungsprojekten das Ziel eine termin- und budgetgerechte Lieferung einer Software mit den spezifizierten Eigenschaften ist und keine Aufwände für Wiederverwendbarkeit von Komponenten oder Überdenken der grundlegenden Entwicklungsprozesse investiert werden können (Althoff et al. 2000, S. 350). Um trotzdem zukünftige Kundenwünsche antizipieren zu können, muss in Forschung investiert werden und es sind Führungspersönlichkeiten mit systemischem Denken erforderlich (Weinert 1998, S. 582). Wissensweitergabe kann dazu zwar einen Beitrag leisten, ist aber alleine nicht ausreichend, um eine Organisation zu einer Lernenden Organisation zu machen.

**Konkrete Organisationsstrukturen:** Neben der oben diskutierten Theorie sollen im folgenden Abschnitt auch einige Beispiele für konkrete Aufbauorganisationen betrachtet werden. Die klassische Organisationsstruktur ist nach funktionalen Aspekten hierarchisch gegliedert und unterscheidet sich v.a. durch die Anzahl der Hierarchieebenen und die Größe der Verantwortungsbereiche einzelner Führungskräfte in Bezug auf die Anzahl unterstellter Mitarbeiter (Weinert 1998, S. 602f). Modernere Alternativen dazu sind in erster Linie die projektorientierte Organisation und die Matrixorganisation (ibid. S. 633). Projektorientierte Organisationen sind v.a. dort sinnvoll, wo ein hoher Grad an Planung, Forschung und Koordination erforderlich ist (ibid.). In einem Projekt werden spezialisierte Mitarbeiter unterschiedlicher Bereiche zusammen mit den benötigten Ressourcen unter einem Projektleiter versammelt, um in einem festgelegten Zeitraum ein spezifisches Ziel zu erreichen (ibid. S. 634, siehe auch Abschnitt 3.7.3). Matrixorganisationen verbinden die

funktionale Struktur klassischer Organisationen mit der Projektstruktur und versuchen dadurch die Vorteile beider Strukturen zu nutzen (Weinert 1998, S. 634f). Mitarbeiter sind dabei einem Funktionsbereich zugeordnet, also dem entsprechenden Bereichs- oder Abteilungsleiter unterstellt, und gleichzeitig einem Projekt zugeordnet, also auch dem Projektleiter unterstellt. Dadurch soll das Wissen von Spezialisten für alle Projekte gleichermaßen verfügbar gemacht und die Kommunikation durch häufigeren Kontakt auch über Abteilungsgrenzen hinaus verbessert werden (ibid. S. 635f). Vorteilhaft sind Matrixorganisationen v.a. wenn große Informationsverarbeitungskapazitäten benötigt werden und das Teilen von Ressourcen wichtig ist. Die Nachteile der Matrixorganisation sind Rollen- und Kompetenzkonflikte durch die doppelte Vorgesetzten-Untergebenen-Beziehung, Machtkämpfe zwischen Projektleitern um die besten Experten, sowie Koordinationsprobleme (ibid. S. 636f). Neben der Matrix aus Funktion und Projekt gibt es in der Praxis der Erfahrung des Autors nach auch solche die einen fachlich Vorgesetzten und einen Personal- oder Regionalverantwortlichen vorsehen.

Organisationen können anhand der drei grundlegenden Produktionssysteme Einzelfertigung, Massenfertigung und Prozessfertigung unterschieden werden (Balzert 2001, S. 84). Die Softwareentwicklung beinhaltet aber Elemente aller drei Systeme (ibid. S. 85). „Am besten lässt sich die Software-Entwicklung als standardisierte Einzelfertigung mit anspruchsvollen, komplexen, teilweise formalisierten Routinetätigkeiten bezeichnen, die in definierten Produktionsstufen erstellt wird“ (ibid.). Dies muss sich auch in der Organisationsstruktur niederschlagen, um erfolgreich zu sein.

Eine Theorie, die Zusammenhänge zwischen hierarchischer, Projekt- und informeller Organisation sehr anschaulich erklärt ist die *Hypertextorganisation*, die von Nonaka und seiner Forschungsgruppe entwickelt wurde (Nonaka 1994, S. 32ff; Maier 2004, S. 139). Sie stellt diese drei, in vielen Organisationen koexistierenden Organisationsstrukturen als verschiedene Ebenen dar, die über die Mitarbeiter miteinander vernetzt sind, analog einem Hypertextdokument, das über Links mit anderen Dokumenten vernetzt ist. Während auf der Geschäftsebene (business systems layer) hauptsächlich Routinetätigkeiten in hierarchischen Strukturen erledigt werden, sind die selbstorganisierenden Teams in der Projektebene (project-systems layer) vorwiegend damit beschäftigt, Neuerungen in der Organisation einzuführen. Auf der Wissensbasisebene (knowledge-base layer) wird die Struktur durch informelle soziale Beziehungen etabliert und Macht durch soziales Kapital und Expertise erworben. Hier erfolgt der größte Teil der Wissensweitergabe. Auch die im

Wissensmanagement viel diskutierten Communities sind auf dieser Ebene anzusiedeln (vgl. Maier 2004, S. 160ff). In der vorliegenden Arbeit wird diese Ebene als soziales Netzwerk diskutiert (siehe Abschnitt 3.6.1).

**Auswirkungen auf Wissensweitergabe:** Die Auswirkungen von Eigenschaften eines Phänomens der Makroebene auf das Arbeitsverhalten, insb. die Wissensweitergabe auf Mikroebene, zu untersuchen ist schwierig, da es eine Vielzahl moderierender Variablen gibt, die signifikante Auswirkungen haben (ibid. S. 615). Nicht zuletzt deshalb ist dieser Bereich bisher wenig beforscht (ibid.). Softwareentwicklungsorganisationen formen typischerweise ein neues Projektteam für jedes Entwicklungsprojekt und sind damit ein Beispiel für projekt- und teambasierte Wissensarbeit (Faraj, Sproull 2000, S. 1554). In der Organisationsforschung wird als abhängige Variable neben der Arbeitsleistung meist die Arbeitszufriedenheit verwendet. Ein direkter Einfluss von Organisationsgröße auf die Arbeitszufriedenheit ist aber nicht belegt. Vielmehr scheint die Größe der Subeinheit (Gruppe, Abteilung, Team) Einfluss auf die Arbeitszufriedenheit zu haben. Verschiedene Indizien weisen darauf hin, dass mittelgroße Subeinheiten positiv für die Arbeitszufriedenheit sind, während große Subeinheiten durchweg negativ beurteilt wurden und kleine Subeinheiten auch in manchen Studien als weniger gut eingeschätzt wurden (Weinert 1998, S. 625f). Da keine Klarheit über die Zusammenhänge zwischen Organisationsstruktur und Wissensweitergabe herrscht, bietet es sich für das Domänenmodell an, sich auf eine einzige Struktur zu konzentrieren und Aussagen nur für diese eine Organisationsstruktur zu machen, ohne Anspruch auf Verallgemeinerbarkeit zu erheben. *Für die weitere Arbeit wird daher von einer projektbasierten Abwicklung des Kerngeschäfts, also der Entwicklung von Individualsoftware, ausgegangen.*

#### **3.7.2 Gruppen- und Teamarbeit**

Eine Gruppe besteht aus mehreren Personen, die zeitlich überdauernd in direkter Interaktion stehen, durch Rollendifferenzierung und gemeinsame Normen gekennzeichnet sind und ein Wir-Gefühl verbindet (Wegge 2001, S. 485). Von Gruppenarbeit spricht man bei kooperierenden Mitarbeitern dann, wenn ein gemeinsamer Auftrag für mindestens zwei Personen vorliegt, der eine gemeinsame Handlungsorganisation und gemeinsame Entscheidungen auf Grundlage von zeitlichem und inhaltlichem Tätigkeitsspielraum der Gruppe verlangt (Wegge 2001, S. 486). Einige Autoren unterscheiden weiterhin zwischen Gruppen und (Projekt-)Teams (Carmel 1999, S. 51), wobei in den meisten Arbeiten ein Konsens besteht, dass eine Unterscheidung zwischen Gruppen- und Teamarbeit selten nützlich ist

(Wegge 2001, S. 486). Will man Gruppen noch genauer unterteilen oder klassifizieren, so bezieht sich die vorliegende Arbeit auf task forces nach O'Connor oder Projektgruppen nach Antoni (vgl. Wegge 2001, S. 486), wobei zu beachten ist, dass die Arbeit davon ausgeht, dass die Zusammenstellung der Gruppe zwar zeitlich beschränkt ist, aber trotzdem der überwiegende Teil des Kerngeschäfts in diesen Projektgruppen abgewickelt wird (siehe oben). Neben diesen formellen Gruppen oder Teams existieren auch informelle Gruppen (Edelmann 1996, S. 351), die hier jedoch nicht gesondert diskutiert sondern im Abschnitt 3.6.1 als soziales Netzwerk behandelt werden.

Gruppen entwickeln sich in mehreren Stufen, wobei vier- bis sechsstufige Modelle am weitesten verbreitet sind (Edelmann 1996, S. 356; Fincham, Rhodes 2003, S. 195). Zu Beginn steht die Phase der (1) Formierung der Gruppe (forming). Diese Phase ist von Unsicherheit gekennzeichnet, v.a. über die Struktur und Führung innerhalb der Gruppe. Das gegenseitige Vertrauen ist niedrig. Anschließend findet sich oft eine Phase des (2) Konflikts und der Herausforderung (storming), in der die Führungsqualität der Führungspersönlichkeit getestet wird aber auch die übrigen Annahmen und Vorgehensweisen einer Belastungsprobe unterzogen werden. Am Ende der zweiten Phase hat sich eine klare Führungshierarchie entwickelt oder die Gruppe ist zerbrochen. In der (3) Normierungsphase (norming) bilden sich Zusammenhalt und Geschlossenheit der Gruppe aus (Kohäsion). Die (4) Funktions- und Leistungsphase (performing) beginnt, wenn die Gruppe ihre volle Leistungsfähigkeit erreicht hat und die gemeinsame Anstrengung voll auf das Erreichen des Zieles ausgerichtet ist. Temporäre Arbeitsgruppen enden in der (5) Phase des Abschließens und Beendens (Edelmann 1996, S. 357).

Es gibt eine Reihe von Einflussfaktoren, die den Erfolg und die Ausprägung einer Gruppe beeinflussen. Mit zunehmender *Gruppengröße* wächst zwangsläufig die physische Distanz zwischen den Mitgliedern, die Homogenität und das Vertrauen sinken, Offenheit und Kommunikationsprozesse werden verkompliziert (Edelmann 1996, S. 360). Kleine Gruppe (N = 5 bis 6) bringen Aufgaben meist schneller zum Abschluss und sind produktiver. Größere Gruppen (N = 10 bis 12) können dafür vielschichtiger Informationen beim Problemlösen berücksichtigen, werden aber oft von wenigen Mitgliedern dominiert, neigen zur Untergruppenbildung und brauchen lange zur Entscheidungsfindung (ibid.).

Die *Kommunikationsstruktur* beeinflusst die Verteilung von Informationen und kann in Form von Stern-, Ring-, Ketten- oder Y-Form auftreten oder es können alle Gruppenmitglieder direkt miteinander kommunizieren (Edelmann 1996, S. 362).

**Rollen:** Jedes Mitglied einer Gruppe muss mindestens eine Rolle erfüllen. Eine *Rolle* besteht aus drei Teilen, (1) der Wahrnehmung der Rolle durch den Rolleninhaber, (2) dem von den anderen Gruppenmitgliedern erwarteten Rollenverhalten und (3) dem tatsächlich gezeigten Rollenverhalten (ibid.). Man spricht von Rollenidentität, wenn bestimmte Einstellungen oder Verhaltensweisen von allen Rolleninhabern geteilt werden, sie also konsistent über alle Rolleninhaber sind. Von Rollenkonflikt spricht man, wenn unterschiedliche Personen im Umfeld des Rolleninhabers unterschiedliche Erwartungen an das Rollenverhalten signalisieren, oder die Rollenwahrnehmung mit verinnerlichten Werten des Rolleninhabers in Konflikt steht (ibid.). Rollenüberladung liegt vor, wenn die Erwartungen an den Rolleninhaber nicht erfüllbar sind, und Rollenambiguität, wenn die Erwartungen nicht klar formuliert sind und der Rolleninhaber somit im Unklaren über diese ist (Edelmann 1996, S. 366).

Ein Rollensystem hat auch Einfluss auf die Entscheidungen und Arbeitsabläufe in Organisationen (Weinert 1998, S. 566). Es erfordert die Spezialisierung der Mitarbeiter auf einzelne, wiederkehrende Aktivitäten, die gegenseitige Kenntnis der Rollen und setzt Mittel zur Sicherung einer gewissen Kontinuität der Rolle voraus (ibid.).

Belbin hat in einer einflussreichen Arbeit neun prototypische Rollen herausgearbeitet, die in einer Gruppe wahrgenommen werden (Fincham, Rhodes 2003, S. 191f). (1) Der Koordinator (coordinator) versucht die Ziele und das Vorgehen der Gruppe abzustimmen. Er verteilt Aufgaben und Verantwortlichkeiten und zeichnet sich durch Selbstbewusstsein, Gewissenhaftigkeit und Zugehörigkeitsgefühle aus. (2) Das Kraftwerk (plant) der Gruppe ist der Ideengenerator. Es sucht permanent nach neuen, innovativen Ansätzen zur Lösung der Probleme der Gruppe und ist meist intelligent, introvertiert und erfinderisch. (3) Der Umsetzer (implementer) ist ein Gruppenmitglied, der Ideen und Vorgaben in praktikable und umsetzbare Vorgehensweisen überführt. Umsetzer sind aufgabenorientiert, gewissenhaft und fühlen sich der Gruppe verbunden. (4) Der Begutachter (monitor evaluator) analysiert die Probleme der Gruppe und bewertet die Beiträge der Mitglieder. Er ist oft nüchtern sachlich, introvertiert und scharfsinnig. (5) Der Former (shaper) ist darauf bedacht die Gruppenaktivitäten zu strukturieren und zu priorisieren. Er ist meist extrovertiert, angespannt und defensiv. (6) Der Teamarbeiter konzentriert sich auf das zwischenmenschliche Verhalten der Gruppenmitglieder. Er motiviert zur Mitarbeit, schlichtet und versucht Harmonie herzustellen ohne selbst dominant zu erscheinen. (7) Der Mittelbeschaffer (resource investigator) identifiziert Ideen und Ressourcen in der

Umgebung und versucht sie für die Gruppe zu allokalieren. Er ist oft extrovertiert, einfallreich und solide. (8) Der Fertigsteller (completer) ist bemüht, die Aufgaben termingerecht abzuschließen. Er hat ein Gespür für Bereiche die mehr Arbeit oder Aufsicht benötigen und wird durch Anspannung und Pflichtgefühl getrieben. (9) Der Spezialist schließlich ist darauf konzentriert sein spezielles Wissen oder seine einmaligen Fertigkeiten in die Gruppe einzubringen. Er tendiert zur Engstirnigkeit und ist selbstmotiviert.

Auch WM-spezifische Rollen sind für die Wissensweitergabe relevant. Aus den von Maier gesammelten Rollen erscheinen v.a. Coach, Mentor, Autor, Themenverantwortlicher (subject matter specialist), Boundary Spanner und Knowledge Broker relevant zu sein, da sie nicht so stark auf Technologie fokussiert sind wie der Wissens(-basis-)administrator und weniger auf administrative Aufgaben konzentriert sind als der Community-Manager oder der WM-Koordinator (Maier 2004, S. 142).

**Gruppenkennzeichen:** Die *Kohäsion* der Gruppe wird von mehreren der bereits genannten Faktoren beeinflusst und darüber hinaus stark von der Zusammensetzung determiniert (Edelmann 1996, S. 368). So weisen z.B. reine Frauengruppen eine höhere Kohäsion auf als Männergruppen. Auch die *Homogenität* ist förderlich für die Gruppenkohäsion, ebenso kleine Gruppengröße und gemeinsame Erfolge in der Vergangenheit (ibid.). Für optimale Produktivität wird allerdings empfohlen neben Hochleistungsgruppen, die nur aus Leistungsträgern bestehen, auch je einen Leistungsträger in Gruppen mit leistungsschwächeren Mitarbeitern zusammenzubringen, um letzteren ein Vorbild zu geben, das sie motiviert (Edelmann 1996, S. 370). Heterogene Arbeitsgruppen, also solche die hinsichtlich, Geschlecht, Alter, Nationalität, aber auch Fachwissen oder Intelligenz ungleich zusammengesetzt sind, zeigen bei Entscheidungs- und Kreativaufgaben bessere Leistungen als homogene (Wegge 2001, S. 489f). Andererseits wird jedoch der Entscheidungsprozess oft verzögert und die beste Entscheidung nur gefunden, wenn eine gewisse Überlappung der Fachkenntnisse und Fähigkeiten, also redundantes Wissen vorliegt (ibid. S. 490). Zudem kann sich zu hohe Heterogenität nachteilig für die Arbeitszufriedenheit, die Fluktuation und die Kommunikationshäufigkeit auswirken, sowie zu einer geringeren Gruppenkohäsion führen. Für die Leistung einer Gruppe bei der Lösung kognitiver Probleme ist der Mittelwert der Fertigkeiten der Gruppenmitglieder sowie der höchste Fertigkeitwert entscheidend, nicht jedoch die Heterogenität, es sei denn, dass die Aufgabe aus unterschiedlichen Teiltätigkeiten besteht. Trotz aller Schwierigkeiten sind Gruppen in den meisten Situationen insgesamt positiv zu bewerten

und zeigen bessere Leistung als das beste Gruppenmitglied allein es könnte (Fincham, Rhodes 2003, S. 196). Dieses Phänomen ist als Synergieeffekt bekannt.

Neben der Kohäsion ist vor allem die *Reflexivität* der Gruppe eine Determinante für deren Leistung (Fincham, Rhodes 2003, S. 197). Reflexivität bedeutet das bewusste und offene Analysieren der Gruppenziele, -strategien und -prozesse mit dem Ziel der Anpassung an die aktuellen Umstände und Gegebenheiten. Es kann auch helfen, dem Phänomen des Gruppendenkens entgegenzuwirken (ibid. S. 200).

Schließlich entwickeln Gruppen im Laufe der Zeit eigene Normen (siehe auch Abschnitt 3.6.2). Diese *Gruppennormen* sind Verhaltensmuster, von denen erwartet wird, dass sie von jedem Gruppenmitglied anerkannt und eingehalten werden (Edelmann 1996, S. 371). Sie beruhen auf geteilten Einstellungen, Meinungen und Gefühlen, die das soziale Verhalten lenken. Die Akzeptanz kann freiwillig durch persönliches Annehmen erfolgen, aber auch durch Gruppendruck von außen induziert werden. Häufig beziehen sich Normen auf die Herangehensweise an den Arbeitsprozess, das äußere Erscheinungsbild, den informellen Umgang oder die Zuweisung von Ressourcen und Aufgaben (ibid.).

Gruppenarbeit bietet auch Gelegenheit zur Befriedigung persönlicher Bedürfnisse, v.a. Anreize die Leistungsmotive (Stolz auf eigene Leistung, bzw. Furcht vor Blamage, siehe Abschnitt 3.3.2), und Anschlussmotive (soziale Zugehörigkeit) aktivieren (Wegge 2001, S. 492). Räumliche Verteilung macht sich durch physische Distanz zwischen den Mitarbeitern bemerkbar, was schon bei geringen Distanzen in einer wesentlichen Reduzierung der Wahrscheinlichkeit von regelmäßiger Kommunikation (mindestens einmal pro Woche) resultiert. Während bei Distanzen bis 10 Metern die Wahrscheinlichkeit noch über 30% liegt, sinkt sie bei Distanzen über 25 Metern auf unter 15% (teamness, Carmel 1999, S. 41ff).

Wenn konfliktäre individuelle Interessen von Mitarbeitern gruppenübergreifend auftreten, wird in der Literatur dafür der Begriff politische Auseinandersetzungen gebraucht, die wiederum als pathologisch bezeichnet werden (vgl. z.B. Schindler 2001). „Hidden Agenda“ ist ein Begriff der Principal-Agent-Theorie, der in diesem Zusammenhang gebraucht wird (vgl. Jost 2001) und die nicht öffentlich bekannten Ziele bezeichnet, die ein Individuum verfolgt, so dass getroffene Entscheidungen für Außenstehende nicht nachvollziehbar und irrational erscheinen. Politik ist in letzter Instanz mit der Neuverteilung von Macht beschäftigt und führt oft zu Konflikten (Warne 2003, S. 107f). Obwohl die Konflikte zwischen Entwicklern eine weitaus geringere Rolle spielen (in rund

10% der Fällen), als die Konflikte zwischen Entwicklern und Benutzern des Systems (30-40%), oder gar Benutzern untereinander (50-60%), haben immerhin noch 35-40% der Konflikte zwischen Entwicklern einen erheblichen (significant or major) Einfluss auf den Erfolg des Systementwicklungsprojekts (Warne 2003, S. 121). Zudem scheinen Konflikte zwischen Entwicklern und Projektmanager häufiger und störender zu sein, als Konflikte zwischen Entwicklern selbst (Warne 1997). Generell können Konflikte auch als Folge von falsch eingestufte Übereinstimmung zwischen den persönlichen und den Gruppenzielen entstehen (Edelmann 1996, S. 372f).

**Auswirkungen auf die Wissensweitergabe:** (Gruenfeld et al. 2000) haben gezeigt, dass temporäres Wechseln der Gruppenzugehörigkeit subtile Auswirkungen auf die Wissensweitergabe zwischen Gruppen hat. Mit dem Wechsel der Gruppenzugehörigkeit wurden die Ideen nicht automatisch mit von einer Gruppe zu einer anderen getragen. Die Ideen des neuen Gruppenmitglieds wurden in der Studie signifikant weniger beachtet als die von langfristigen Gruppenmitgliedern, nachdem das temporäre Mitglied die Gruppe wieder verlassen hatte. Darüber hinaus wurde neues Wissen, welches das temporäre Mitglied in dieser Zeit erworben hatte, nach seiner Rückkehr zu seiner ursprünglichen Gruppe dort ebenfalls wenig beachtet. Obwohl also durch die Maßnahme viel Wissen ausgetauscht wurde, wurde es kaum genutzt, da ihm in der Evaluationsphase (siehe Abschnitt 2.2.4) zu wenig Bedeutung zugemessen wurde.

(Stasser et al. 2000) haben Belege dafür gefunden, dass Metawissen über die Aufgaben von Gruppenmitgliedern einen positiven Einfluss auf die Wissensweitergabe hat. Während Gruppendiskussionen fokussierten sich in der Studie Gespräche stets auf Informationen, die mehreren Gruppenmitgliedern bereits vorher bekannt waren, während Informationen, die nur ein einzelnes Gruppenmitglied besaß und während der Diskussion zur Sprache brachte, meist wenig beachtet wurden. Dieser Bias wurde reduziert, wenn zu Beginn der Diskussion die Aufgaben- und damit auch Expertisebereiche der einzelnen Mitglieder zur Sprache gebracht wurden.

#### **3.7.3 Projektarbeit**

Für die Entwicklung von Individualsoftware ist die Projektstruktur als Organisationsstruktur relevant (Balzert 2001, S. 87). Eine Projektstruktur entwickelt mit Projektteams, die aus Mitarbeitern unterschiedlicher Disziplinen bestehen, anspruchsvolle Produkte und hat folgende Charakteristika (ibid.).

- Organische Struktur mit geringer Verhaltensformalisierung
- Horizontale Aufgabenspezialisierung auf Basis formeller Ausbildung
- Projektorientierte Matrixstruktur durch Gruppierung der Mitarbeiter in funktionale Einheiten und Einsatz in marktorientierten Projektteams
- Einsatz von Kommunikationsinstrumenten zur gegenseitigen Abstimmung

Die Projektarbeit bildet den Rahmen für die untersuchten Wissensweitergabeaktivitäten. Der Begriff des Projekts wird in der Literatur vielfach leicht unterschiedlich definiert. Schindler sammelt eine Reihe von Definitionen in seiner Arbeit (Schindler 2001, S. 372f). Davon werden hier einige zitiert, um dem Leser einen Eindruck des Spektrums zu geben.

- Ein Vorhaben mit zeitlicher Begrenzung, definiertem Ziel, Einmaligkeit (in Abgrenzung zu Routineaufgaben) und einer Komplexität, die eine Unterteilung in wechselseitig voneinander abhängige Teilaufgaben erforderlich macht (Haberfellner et al. 1999, S. 241).
- Ein Projekt ist ein einmalig durchzuführendes Vorhaben, das durch seine zeitliche Befristung, besondere Komplexität und interdisziplinäre Aufgabenstellung zu beschreiben ist (Krcmar 2000, S. 116).
- Abgrenzbare Einzelvorhaben mit definiertem Anfang und Ende (Ziel), Neuartigkeit, Risikoreichtum (technisch, wirtschaftlich, terminlich), Komplexität, sich im Laufe der Abwicklung ständig ändernde Bedürfnisse sowie großer Bedeutung für die beteiligten Organisationen und Termindruck (Litke 1995, S. 17).
- Ein Vorhaben mit definiertem Beginn und Abschluss, das sich im Gegensatz zu den regelmäßig wiederkehrenden Arbeitsabläufen eines Unternehmens durch folgende Merkmale beschreiben lässt: einmaliger und zeitlich begrenzter Lebenszyklus und relativ hohe technologische und/oder manageriale Komplexität und Neuartigkeit (Madauss 2000, S. 538).

Obwohl diese Definitionen leicht unterschiedliche Foki haben, wurden einige Merkmale mehrfach angeführt. So wird die zeitliche Befristung überall genannt, ebenso Einmaligkeit bzw. Neuartigkeit des Vorhabens und hohe Komplexität. Als Folge daraus oder auch Ursache davon ergeben sich einige weitere Merkmale, die nur in einzelnen Definitionen auftauchen, z.B. die Unterteilung in wechselseitig voneinander abhängige Teilaufgaben, interdisziplinäre Aufgabenstellung, Termindruck und sich ändernde Bedürfnisse.

Die traditionelle Perspektive auf Projekte und ihre Wirkungszusammenhänge ist auf die Messgrößen Zeit, Kosten, Qualität ausgerichtet, die auch als magisches Dreieck der Projektarbeit bekannt sind (Schindler 2001, S. 21). Man kann Projekte jedoch auch unter dem Gesichtspunkt der Systemtheorie und Kybernetik betrachten (siehe auch Abschnitt 3.7.1), wie das im Systems Engineering geschieht (Haberfellner et al. 1999). Diese Betrachtungsweise basiert auf der Annahme, dass kein einzelnes Individuum über das Gesamte für die Abwicklung eines Projektes erforderliche Wissen verfügt und die Projektarbeit durch eine hohe Dynamik und Komplexität geprägt ist (Schindler 2001, S. 22). Daraus resultiert eine Wahrnehmung der Projekte als komplexe dynamische soziale Systeme. Ein Projektteam kann abstrakt ausgedrückt als ein Konstrukt bestehend aus konsistenten Akteuren mit folgenden Merkmalen betrachtet werden (Schindler 2001, S. 62).

- Unterschiedliche fachliche Hintergründe der Mitarbeiter
- Spezialisiertes Expertenwissen
- Voll- oder Teilzeitmitgliedschaft der Teammitglieder
- Sich im Projektverlauf ändernde Teamzusammensetzungen
- Die mögliche Existenz eines Kernteams

Projektteams werden von einem Projektmanager (PM) geleitet der koordinierend tätig ist, darüber hinaus aber keine klassische Führungskraft darstellt, die Arbeitsanweisungen gibt (Balzert 2001, S. 87).

Projekte weisen hinsichtlich der Kriterien personelle und organisatorische Verteilung, Informationsverteilung, verteilter Bearbeitung von Aufgaben und der Verteilung von Kompetenzen und Wissen einen unterschiedlichen Verteilungsgrad auf (Schindler 2001, S. 26). In der vorliegenden Arbeit liegt der Fokus auf intraorganisationaler Projektarbeit innerhalb eines Gebäudes oder Gebäudekomplexes. Dies stellt einen einfacheren Fall dar als organisationsübergreifende Projektarbeit, die regional verteilt ist. Die Einschränkung dient demnach zur Komplexitätsreduktion des Gesamtproblems.

***Auswirkungen auf die Wissensweitergabe:*** Projektarbeit kann aufgrund der angeführten Merkmale grundsätzlich als wissensintensiv eingestuft werden und beruht auf der Entstehung und Weitergabe von Wissen und Erfahrungen (Schindler 2001, S. 34). Ein weiterer Hinweis dafür ist, dass in der Literatur häufig Projekte als Orte des Lernens postuliert werden (Schindler 2001, S. 46). Schindler erläutert dies anhand der Merkmale der Projektarbeit. So führt Interdisziplinarität zu neuen Impulsen, die das Lernen anregen.

Die Komplexität erfordert ein weit reichendes und gemeinsames Problemverständnis, das nur durch Lernen erreicht werden kann. Das gemeinsame Ziel ermöglicht ein fokussierteres Lernen, was durch den hohen Problemlösedruck aufgrund von Terminvorgaben noch unterstützt wird. Größere Autonomie des Teams im Gegensatz zur Stab-Linienorganisation erzeugt ein lernfreundliches Klima. Schnelle Rückkopplung und hohe Interaktionsdichte im Team wirkt weiterhin begünstigend. Die flache Hierarchie im Team fördert Transparenz von Entscheidungen und gibt damit Lernimpulse. Im Team besteht oft höheres gegenseitiges Vertrauen, das eine Grundlage für Lernen darstellt.

Die Projektorganisationsform hat spezifische Vor- und Nachteile für die Wissensweitergabe (Schindler 2001, S. 67ff). Bei reiner Projektorganisation neigen Abteilungsleiter dazu ihre besten Mitarbeiter zurückzuhalten, um einen Verlust der Wissensträger zu vermeiden. Im Projekt neu gewonnenes Wissen wird oft in der Abteilung nicht akzeptiert, wie Schindler mit dem Hinweis auf das bekannte „Not-invented-here“-Syndrom mutmaßt und Gruenfeld et al in ihrer Studie empirisch belegen (Gruenfeld et al. 2000). Weiterhin besteht die Gefahr, dass Wissen aufgrund der primären Innensicht der Mitarbeiter an mehreren Stellen gleichzeitig erzeugt wird. Als Lösung dieses Problems schlägt Schindler Vollzeitprojektleiter vor, die als „Wissensintegratoren“ zwischen verschiedenen Projektteams fungieren können. Solche Vollzeitprojektleiter werden auch für das Domänenmodell und die Simulation vorgeschlagen, da sie leichter zu konzeptualisieren sind als gemischte Rollen. Eine weitere Gefahr besteht in der übermäßig langen und frühzeitigen Allokation von Experten. Um im Bedarfsfall keinen Mangel an Wissen zu haben werden Experten ab ihrer Verfügbarkeit dauerhaft in das Projektteam integriert, obwohl nur sehr kurzzeitig echter Bedarf bestand. In der restlichen Zeit stehen sie aber nicht für andere Projekte zur Verfügung, wo sie ebenfalls dringend gebraucht werden.

Bei der Einflussprojektorganisation wird die Wissensweitergabe zwischen Teammitgliedern aus unterschiedlichen Abteilungen durch weiterhin bestehende Abteilungszuordnung behindert und die Experten sind oft gar nicht verfügbar, da sie von den abteilungsinternen Aufgaben zu stark beansprucht werden und der Projektleiter über keine Weisungsbefugnis verfügt (Schindler 2001, S. 69). Von Vorteil ist der einfache Rückfluss von Projektwissen in die Linienorganisation, da die Projektmitarbeiter ständig in der Linie eingebunden bleiben und so keine Akzeptanzprobleme entstehen. Dadurch verbessert sich auch die Übertragbarkeit auf neue Projekte. Eine Matrixorganisation verbessert dies noch einmal.

### 3.7.4 Personalführung und -entwicklung

Die aus Sicht der Wissensweitergabe relevanten Aufgaben des Personalwesens sind die Personalentwicklung mit Planung von Weiterbildungsmaßnahmen und Kompetenzentwicklung der Mitarbeiter einerseits und Entwicklung von Anreizsystemen und Karriereplanung andererseits.

Paauwe stellt in seiner Arbeit eine Reihe von Modellen vor, die den Zusammenhang zwischen den Aktivitäten des Personalwesens und der Leistungsfähigkeit einer Organisation herstellen (Paauwe 2004, S. 55ff). Dabei scheint das Modell von Guest (1997) besonders geeignet für die vorliegende Arbeit, da es sowohl die Makroebene (Personalstrategie, finanzielle Ergebnisse) als auch die Mikroebene (Verhaltensergebnisse) berücksichtigt und beide relativ stringent voneinander trennt (vgl. Abbildung 3-15).



Abbildung 3-15: Verbindung von Personalwesen und Leistung (vgl. Paauwe 2004, S. 57)

Ausgehend von der Personalstrategie werden Personalpraktiken priorisiert und angewendet. Status meint dabei die Einordnung in der Organisationshierarchie, Sicherheit zielt auf die Garantie eines sicheren Arbeitsplatzes ab, was gerade in der heutigen Zeit der Entlassungswellen für viele Arbeitnehmer ein entscheidender Faktor ist. Dadurch wirkt das Personalwesen positiv auf Engagement und Verbundenheit der Mitarbeiter, die Qualität der Mitarbeiter und ihre flexible Einsetzbarkeit. Das Verhalten der Mitarbeiter ändert sich in Folge dessen. Sie zeigen höhere Einsatzbereitschaft und Motivation, kooperieren stärker miteinander und fühlen sich als aktiver Teil des Unternehmens (organizational citizenship). Dadurch steigt die Produktivität ihrer Arbeit, die Produkt- und Dienstleistungsqualität steigt und es werden mehr Innovationen gemacht. Die Mitarbeiter fehlen weniger häufig, was sowohl auf die hohe Motivation als auch den geringeren psychischen Druck zurückzuführen ist, der durch Arbeitsbelastung und negatives Arbeitsklima andernfalls entstehen kann. Durch Zufriedenheit mit der eigenen Arbeit und Leistung sinkt die Fluktuation,

Konflikte unter Kollegen bleiben aus, Kunden werden zuvorkommend behandelt und haben weniger Anlass zu Beschwerden. In Folge all dessen erzielt das Unternehmen gesunde Gewinne und ist rentabel (zu diesen Interpretationen siehe Paaue 2004, S. 57f).

Die Karriere als Anreiz für Mitarbeiter ist insofern besonders interessant, als mit einem beruflichen Aufstieg eine höhere Bezahlung und meist auch höheres Ansehen oder auch Macht verbunden sind. Zusätzlich treten negative Effekte wie längere Arbeitszeiten oder unangenehmere Arbeiten, wie sie bei manuellen Tätigkeiten häufig mit höherem Gehalt verbunden sind, bei einem Aufstieg nicht zwangsläufig auf (Fincham, Rhodes 2003, S. 350). Dennoch werden in höheren Positionen oft Überstunden erwartet, so dass trotz nominell gleicher Arbeitszeit eine höhere zeitliche Belastung vorliegt. Gerade für gut ausgebildete Mitarbeiter mit entsprechend konkreten Vorstellungen von ihrer beruflichen und privaten Zukunft wird die Balance zwischen beruflichem Engagement mit dem dafür notwendigen Arbeitszeiteinsatz, den erwünschten Freizeitbeschäftigungen und Zeit für die Familie ein Problem (vgl. Hyman, Summers 2004, S. 418). Gemäß der mikroökonomischen Nutzentheorie streben Mitarbeiter v.a. nach Einkommen und Freizeit, wobei Einkommen und Arbeitszeit wie oben argumentiert zumindest teilweise korrelieren, so dass die beiden Größen widerstreitende Ziele sind. Die aktuellen Debatten über Verlängerung der tariflich geregelten Arbeitszeit zusammen mit den Statistiken<sup>23, 24, 25</sup> über geleistete Überstunden zeigen, dass eine ausgewogene Verteilung der Ressource Zeit auf Arbeit und Freizeit schwierig ist. Zeitmangel kann auch Stress bewirken (Weinert 1998, S. 242) und sich dadurch ungünstig auf Arbeitsleistung und Wissensweitergabe auswirken.

---

<sup>23</sup> Nach einer Studie des Statistischen Bundesamtes leisteten 2001 17% der männlichen und 12% der weiblichen Arbeitnehmer in der EU Überstunden. Die durchschnittliche Anzahl an Überstunden betrug 8 Stunden pro Woche. [http://www.eds-destatis.de/de/downloads/sif/nk\\_04\\_11.pdf](http://www.eds-destatis.de/de/downloads/sif/nk_04_11.pdf), letzter Zugriff 14.03.2006

<sup>24</sup> Zwar ist die Zahl der bezahlten Überstunden von 1,94 Mrd. in 1995 auf 1,44 Mrd. in 2005 gesunken, die unbezahlten Überstunden werden davon jedoch nicht erfasst, so dass die Folgerung es würden generell weniger Überstunden geleistet nicht abgeleitet werden kann. <http://www.welt.de/data/2005/12/03/812204.html>, letzter Zugriff 14.03.2006

<sup>25</sup> Nach einer Umfrage der Online-Stellenbörse stellenanzeigen.de vom November 2005 unter über 1.800 Arbeitnehmern in Deutschland leisten 79% regelmäßig Überstunden, 26% der Beschäftigten bleiben praktisch jeden Tag länger am Arbeitsplatz, weil sie sonst mit ihrer Arbeit nicht fertig würden. 53% machen nur dann Überstunden, wenn es unbedingt sein muss. Für 15 Prozent sind Überstunden eigenen Angaben zufolge gar kein Thema. <http://www.stellenanzeigen.de/asp/ueberuns/presse/anzeige.asp?id=149>, letzter Zugriff 14.03.2006

Flexible Arbeitszeitregelungen können motivierend auf die Mitarbeiter wirken (Weinert 1998, S. 187), ein direkter Zusammenhang zur Wissensweitergabe ist aber nur schwer herstellbar und bisher in keiner dem Autor bekannten Studie untersucht.

Eine wesentliche Aufgabe der Personalführung und -entwicklung ist funktionierende Be- und Entlohnungssysteme zu etablieren, die Mitarbeiter für ihre geleistete Arbeit entlohnen und so für eine anhaltende Leistungsmotivation sorgen (vgl. Weinert 1998, S. 567). Damit ökonomische Anreize funktionieren müssen die Mitarbeiter den Zusammenhang zwischen der Verbesserung seiner Arbeitsleistung und entsprechendem Anstieg seiner Be- und Entlohnung erkennen, die Rollenstruktur (siehe Abschnitt 3.7.2 und 3.7.1) muss diese Erkenntnis auch ermöglichen und die Mitarbeiter müssen entsprechende Motive haben, die sich durch ökonomische Anreize aktivieren lassen (ibid.). Dafür ist es erforderlich, den Zusammenhang zwischen persönlicher Leistung und Einkommen deutlich zu machen, klare Regeln für Beförderungen aufzustellen und die mit einer neuen Rolle verbundenen Privilegien sowie die Autonomie und Entscheidungsspanne zu kommunizieren (ibid. S. 568). Um die richtigen Anreize setzen zu können, müssen die Valenzen der Mitarbeiter bekannt sein. Eine empirische Studie zeigt, dass Vorgesetzte trotz ähnlicher eigener Bewertung von Anreizen die Gewichtung von Anreizen ihrer Mitarbeiter z.T. völlig falsch einschätzen (Weinert 1998, S. 600). So wurde z.B. die Wichtigkeit von Arbeitsplatzsicherheit und v.a. der Entlohnung überschätzt, während soziale Motive und interessante Tätigkeiten als Anreiz stark unterschätzt wurden (ibid.).

#### **3.7.5 Diskussion**

Im Organisationskontext treffen Faktoren, die weitgehend von der Unternehmensleitung gesteuert werden können, wie die Organisationsstruktur und der Einsatz von Managementinstrumenten wie Anreizsysteme und Personalausbildung auf Faktoren, die nicht unmittelbar von der Unternehmensleitung beeinflussbar sind, wie z.B. die Organisationskultur und Gruppennormen. Soll die Organisation so gestaltet werden, dass Wissensweitergabe gefördert wird, so muss den Mitarbeitern genügend Autonomie eingeräumt werden, damit sich auf freiwilliger Basis Austauschbeziehungen zwischen den Mitarbeitern etablieren können. Gruppenarbeit in Projektteams ist dafür eine gute Voraussetzung. Allerdings ist die oft genutzte Matrixorganisation aufgrund ihrer doppelten Hierarchie und den damit verbundenen Kompetenzproblemen zumindest für die Wissensweitergabe entlang der Hierarchie problematisch. Für die Weitergabe auf gleicher Ebene sind keine Nachteile zu erwarten. Der Wechsel einzelner Mitarbeiter zwischen etablierten

Gruppen hat sich in der Empirie als wenig hilfreich erwiesen. Wie sich das allerdings beim Wechsel in vollständig neu zusammengesetzte Projektteams darstellt, ist in keiner dem Autor bekannten Studie untersucht worden. Eine Hypothese, die im später zu erstellenden Simulationsmodell untersucht werden könnte, ist dass solche dynamisch formierten Projektteams sich positiv auf die Wissensweitergabe auswirken. Die Auswirkung unterschiedlicher Besetzung solcher Teams mit Experten und Novizen ist eine weitere Frage, die bisher nicht beantwortet wurde. Für die Leistungsfähigkeit sind reine Expertenteams zusammen mit Novizenteams, denen ein Experte zugewiesen wird, vorteilhaft. Ob das auf Wissensweitergabe übertragbar ist, ist unklar und könnte auch in der Simulation untersucht werden.

Zur Modellierung von Organisationen ist es laut Carley nötig, eine Reihe von Netzwerken zu berücksichtigen, die in typischen Organisationen existieren (Carley 2001, S. 3). Die Netzwerke ergeben sich aus den Verbindungen zwischen den Mitgliedern der Organisation (in Tabelle 4-7, S. 283 als Agenten bezeichnet), dem Wissen, den Ressourcen, den Aufgaben und den Organisationseinheiten bzw. Organisationen, wenn interorganisationale Phänomene untersucht werden. Daraus ergeben sich 15 netzwerkartige Beziehungen. Carley benutzt dabei selbst in ihren Arbeiten nicht immer alle Elemente. So werden je nach Simulation Ressourcen (Carley 2002b) oder Organisationen (Carley, Tsvetovat 2004) weggelassen. Diese Elemente sind insofern von besonderer Bedeutung, weil sie nicht nur in der Simulation abgebildet werden müssen sondern die Agenten auch eine Wissensrepräsentation davon haben müssen. Sie müssen also wissen, wen sie kennen (soziales Netzwerk), was andere Agenten wissen, welche Ressourcen es gibt und wie diese eingesetzt werden können, welche Aufgaben es gibt, wer sie erledigt, welches Wissen und welche Ressourcen er dazu braucht und in welcher Reihenfolge die Aufgaben abgearbeitet werden müssen. Schließlich ist auch Wissen über Organisationseinheiten relevant, z.B. welche Agenten in welcher Abteilung arbeiten, welches Wissen (z.B. in Dokumenten) in den Abteilungen vorgehalten wird, welche Ressourcen sie besitzen, welche Aufgaben in der Abteilung anfallen und erledigt werden und wie sie mit anderen Abteilungen zusammenarbeiten.

### **3.8 Domänenmodell der Wissensweitergabe**

Ein integriertes Modell der Wissensweitergabe scheint nötig zu sein, um neben Einflussfaktoren der Mikroebene auch Faktoren der Makroebene berücksichtigen zu können, ohne die Wirkung von moderierenden Variablen zu vernachlässigen (vgl. Weinert

1998, S. 628). So ein integriertes Modell sollte sich an bestehenden Modellen orientieren, die Arbeitsverhalten im Allgemeinen oder andere der Wissensweitergabe verwandte Konzepte modellieren. Hier wird daher auf die Modelle der Einflussfaktoren für Arbeitsverhalten in Organisationen von (Weinert 1998, S. 590), das psychologische Modell der Informationsverarbeitung von (Edelmann 1996, S. 4) und das Modell des Arbeitsverhaltens von (Jost 2000, S. 96-111) aufgebaut. Abbildung 3-16 zeigt eine grafische Übersicht des eigenen Domänenmodells der Wissensweitergabe.

Als Gelegenheiten werden verschiedene Typen unterschieden. Als Pull-Gelegenheit dient im Modell ausschließlich die aktive Nachfrage nach Wissen durch Fragen eines Kollegen. Die Push-Gelegenheiten ergeben sich durch vorherige Wissensweitergaben, Lernen im Lernfeld, oder Tätigkeitsfeld sowie in externen Trainings erworbenes Wissen. Den dritten Typ von Gelegenheit stellen organisatorisch verankerte Gelegenheiten dar, wie z.B. interne Trainings oder Meetings. Der Geschäftsprozess, in diesem Fall der Softwareentwicklungsprozess, der aus den Phasen Analyse, Entwurf, Implementation, Test und Roll-Out besteht, kann durch Lernen im Tätigkeitsfeld sowohl den Anfang einer Wissensweitergabeaktivität sein als auch durch Anwenden des neu erworbenen Wissens das Ende des Weitergabeprozesses darstellen.

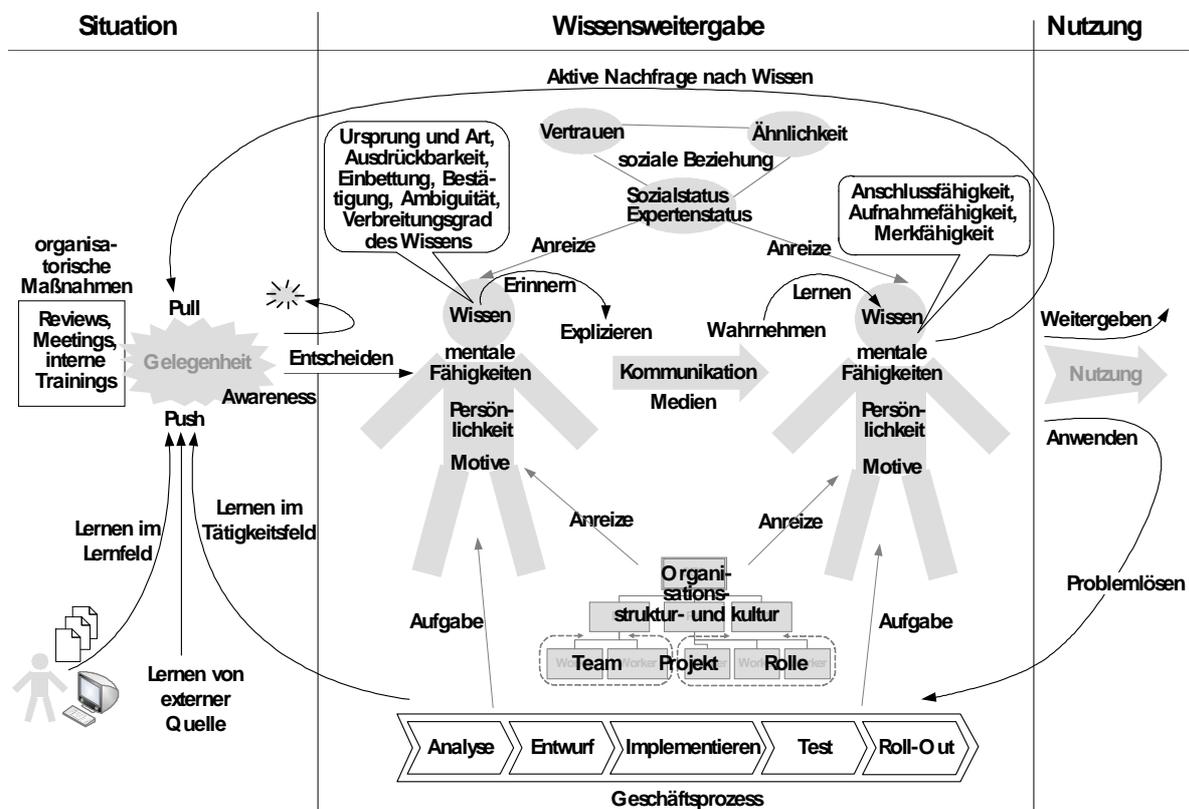


Abbildung 3-16: Domänenmodell der Wissensweitergabe grafisch (eigene Darstellung)

Die Wissensweitergabe selbst läuft so ab, wie schon im Modell in Kapitel 2 beschrieben. Die Organisation stellt in erster Linie den Rahmen dar, in dem die Wissensweitergabe abläuft. Organisatorische Randbedingungen können die Wissensweitergabe erleichtern oder behindern. Dem gegenüber steht die Einbettung in das soziale Netzwerk, welche maßgeblichen Einfluss auf den Erfolg oder Misserfolg der Wissensweitergabe hat. Für diesen Kontextbereich kommt im Modell die Relation Model Theory von Fiske zum Einsatz. Sie verbindet die Aspekte Sozialstatus, Expertenstatus, Streben nach Wissen und Streben nach Einkommen und schlägt somit die Brücke von der ökonomischen Theorie zu den sozialen Netzen.

Sowohl die Wirkungen des sozialen Netzwerks als auch der Organisation werden als Anreize modelliert, die auf Motive in der Person wirken und motivieren oder demotivieren. Für die Motivation wird von einer Prozesstheorie ausgegangen. Am besten geeignet scheint dafür Vrooms VIE-Theorie, da sie mit ökonomischen Modellen über Erwartungswerte und Nutzenfunktionen kompatibel ist. Der Inhaltsaspekt darf aber nicht vernachlässigt werden. Hier wird aber nicht auf eine Motivationstheorie zurückgegriffen, sondern wiederum die Relation Model Theory mit den fünf Motiven Zugehörigkeit zu sozialen Netzen, Streben nach Geld, Wissen, Expertenstatus und Macht verwendet.

Andere Faktoren aus sozialem Netzwerk und Organisation wie Vertrauen und Organisationskultur wirken als eine Art „Hygienefaktoren“ und stellen eine Grundvoraussetzung für Wissensweitergabe dar oder senken die „Hemmschwelle“ zur Weitergabe. Spezifische Persönlichkeitsmerkmale führen dazu, dass trotz gleicher Umweltbedingungen unterschiedliche Entscheidungen für oder gegen Wissensweitergabe getroffen werden. Dies wird im Modell über unterschiedliche Gewichtungen der Motive abgebildet.

Im Bereich Entscheidungsfindung koexistieren mehrere konkurrierende Alternativen. Die Kritik an klassischen Rational-Choice-Ansätzen ist so stark, dass in jüngerer Zeit meist von beschränkter Rationalität ausgegangen wird. Aber auch an diesen Ansätzen wird kritisiert, dass Menschen in vielen Situationen überhaupt keine Auswahl im entscheidungstheoretischen Sinne treffen, sondern einfach die zuletzt erfolgreiche Aktion in ähnlicher Situation wieder auswählen ohne auch nur eine einzige Alternative zu berücksichtigen. Weiterhin gibt es neurobiologische Untersuchungen, die zu dem Ergebnis kommen, dass Handlungen sogar durchgeführt werden, bevor es überhaupt zu Gehirnaktivität kommt. Dieser scheinbare Widerspruch lässt sich mit der Aktivitätstheorie auflösen, wenn deren lerntheoretische Aspekte betrachtet werden (siehe Abschnitt 3.3.4).

Auf der Ebene der Aktivität bestehen viele Unsicherheiten, vorteilhafte Lösungsansätze sind noch nicht bekannt, oft müssen zusätzliche Informationen beschafft werden. In diesem Fall, besonders bei wichtigen Aktivitäten mit weit reichenden Wirkungen die ohne großen Zeitdruck ausgeführt werden können, ist zu erwarten, dass Entscheidungen nach dem Modell der beschränkten Rationalität getroffen werden und viele, wenn auch vermutlich nicht alle relevanten Alternativen bewertet werden. Kommt der Entscheider wieder in eine ähnliche Situation, so kann er auf die erste Entscheidung zurückgreifen wenn sie sich bewährt hat und evtl. noch einige Modifikationen vornehmen, wenn er beim Durchführen der beschlossenen Aktionen zusätzliches entscheidungsrelevantes Wissen erworben hat. Kommen ähnliche Situationen häufiger vor, so wird überhaupt keine Alternative mehr erwogen, sondern sofort die bewährte Strategie gewählt, wenn der Entscheider nicht zwischenzeitlich neue Informationen bekommen hat, die diese Strategie in Frage stellen. Das Verhalten wurde gelernt und man befindet sich auf der Ebene der Aktionen innerhalb der Aktivitätentheorie. Muss die Aktivität regelmäßig oder oft durchgeführt werden, so wird das Verhalten routinisiert und überhaupt keine bewusste Entscheidung mehr getroffen, sondern reflexhaft die bewährten Aktionen ausgeführt. Die Aktivitätentheorie nennt Handlungen auf dieser Ebene Operationen (vgl. dazu auch Wiig 2003, S. 12).

Das Festhalten an bewährten Strategien, das in der Empirie häufig zu beobachten ist (Paauwe 2004, S. 40; Mandl et al. 1988, S. 128) kann über risikoscheues Verhalten und Opportunitätskosten erklärt werden, da der Erfolg neuer Alternativen ungewisser erscheint als der Erfolg bewährter Alternativen und die Kosten für Informationsbeschaffung und Entscheidungsfindung zu vermeiden versucht werden.

Auch opportunistisches Verhalten muss bei der Entscheidungsfindung berücksichtigt werden. Dieses ist mit dem Modell der beschränkten Rationalität vereinbar, da bei opportunistischem Verhalten andere Ziele verfolgt werden, als dies im Sinne des Prinzipals ist und damit eine andere Nutzenfunktion vorliegt. Innerhalb dieser Präferenzen kann jedoch für die Entscheidungsfindung weiterhin beschränkte Rationalität unterstellt werden.

Für Theorien zum Gedächtnis und Lernen ist die Einordnung der konkurrierenden Alternativen noch schwieriger. Dort bietet der symbolische Ansatz eine konsistente und gute Erklärungsbasis, während die Theorien zu Situated Cognition dem in vielen Bereichen fundamental widersprechen. Trotzdem scheinen sie nicht gänzlich unvereinbar zu sein, da sowohl Clancey (Clancey 1997, S. 2,4) als auch Law (Law 1998a) nicht die Existenz symbolischer Repräsentation an sich in Frage stellen sondern nur deren Auswirkungen auf

das Handeln neu bewerten. Die vollständige Abkehr von symbolischen Wissensrepräsentationen, die teilweise vielleicht auch durch unbefriedigende Ergebnisse von Computerimplementierungen der Modelle initiiert wurde (vgl. Clancey 1997, S. 1) erscheint in jedem Fall fragwürdig. Die Literatur zu Expertisestufen und zu Wissensarten lässt den Schluss zu, dass prozedurales und deklaratives Wissen (wie von Vertretern des symbolischen Ansatzes postuliert) vor allem auf den unteren Expertise-Stufen (bis etwa Stufe 3 in dem Modell von Dreyfus und Dreyfus) einen großen Anteil hat. Für höhere Stufen kann durch Vermittlung von Faktenwissen und Regeln kein Wissenszuwachs mehr erzielt werden (siehe dazu auch die Darstellung der Cognitive Flexibility Theory in Sonntag, Schaper 2001, S. 247). Law hat gezeigt, dass der Anteil von situativen ad-hoc Aktionen im Gegensatz zu geplanten Aktionen auch stark von der Tätigkeit abhängt (Codieren vs. Debuggen bei der Programmierung). Der Erfahrung des Autors nach spielen situative Faktoren wie von Gibson postuliert gerade bei mangelndem Wissen (z.B. bei der Konfiguration von komplexen Softwaresystemen) eine große Rolle. Ist die genaue Bedeutung von Parametern oder Bedienelementen nicht bekannt, dann wird oft einfach nach dem Trial-and-Error Prinzip ausprobiert, was das Ausführen einer Aktion oder eine Konfigurationsänderung bewirkt. Dies gilt besonders, wenn die Erfahrung mit der Dokumentation vermuten lässt, dass sich dort keine weiterführenden Informationen finden. Andererseits kann den Verfechtern einer ausschließlich situierten Kognition entgegen gehalten werden, dass z.B. Programmierprobleme oft in Situationen gelöst werden, wo die Wahrnehmung der Umgebung fast vollständig ausgeblendet wird, wie auf dem (Fuß-)Weg zur Arbeit oder wenn überhaupt keine Interaktion mit der Umgebung stattfindet, wie z.B. vor dem Einschlafen im Bett, oder sogar im Schlaf. Diese empirisch beobachtbaren Phänomene lassen sich mit den Ansätzen zur Situated Cognition nur schwer erklären und sprechen eher für einen symbolischen Ansatz (siehe mentale Simulation).

Ein Aspekt der in der Wissensmanagement-Literatur nach Meinung des Autors bisher zu wenig berücksichtigt wird, ist die Art des betrachteten Wissens. Oft wird auf Polanyi verwiesen und argumentiert, dass der implizite Anteil des Wissens ein wichtiger Faktor ist, ohne dass deutlich gemacht wird, welcher Teil des Domänenwissens nicht explizierbar ist. Auch Polanyi selbst betrachtet Wissen nach Meinung des Autors zu undifferenziert, indem er die in den Abschnitten 3.3.3, 3.3.4 und 3.3.5 diskutierten Unterschiede zwischen sprachlichem, bildhaftem und Handlungswissen überhaupt nicht erwähnt, künstlerische,

athletische und technische Fertigkeiten zusammen betrachtet<sup>26</sup> und anschließend Wissen und Können, sowie „know what“ und „know how“ unter dem Begriff des Wissens subsumiert, den er dann im Weiteren undifferenziert verwendet (Polanyi 1966, S. 7). Die Übertragbarkeit von Erkenntnissen, die unter spezifischen Bedingungen gewonnen wurden, muss genau geprüft werden. So ist die von Polanyi öfters angeführte Erkennung von Gesichtern und deren Emotionen, deren Merkmale nicht sprachlich expliziert werden können zwar noch ähnlich zu dem in der Expertiseforschung gut untersuchten Erkennen von Schachstellungen, da in beiden Fällen visuelle Aspekte eine große Rolle spielen. Aber schon hier gibt es große Unterschiede, da sich Gesichtserkennung über Jahrtausende evolutionär entwickelt hat und schon für Menschen in der Steinzeit wichtig war<sup>27</sup>, was für Schachstellungen offensichtlich nicht zutrifft. Ein weiterer Übertragungsschritt ist vom Schachgroßmeister auf den Experten in der Softwareentwicklung nötig, der innerhalb kurzer Zeit die kritischen Aspekte der zu entwickelnden Softwarelösung aus einem Pflichtenheft mit mehreren hundert Seiten herausdestillieren kann. Zwar sind beide Analyseaufgaben hauptsächlich kognitiv, aber das Pflichtenheft ist überwiegend sprachlich kodiert, während beim Schach primär eine bildhafte Kodierung vorliegt. Zudem ist das Pflichtenheft nicht auf einen Blick überschaubar.

Auch die Schlussfolgerung Polanyis von durch Reiz-Reaktions-Lernen erworbenen, körperlichen Reaktionen auf bestimmte Reize (nach Pawlow), zu denen kein kognitiver Zugang besteht (Polanyi 1966, S. 7f), auf die Aussage, dass wir mehr wissen, als wir sprachlich ausdrücken können (ibid. S. 8) erscheint bei so einer umfassenden Auffassung von Wissen wie Polanyi sie vertritt als zu pauschal. Das soll nicht heißen, dass der Autor Vorhandensein impliziten Wissens an sich bestreitet, sondern nur, dass von Fall zu Fall genau geprüft werden muss, wie groß und relevant der implizite Anteil des Domänenwissens tatsächlich ist.

Die meist genannten Beispiele für Domänen, in denen implizites Wissen einen großen Anteil hat sind Domänen, in denen sprachlich kodiertes Wissen nicht wichtig ist. Ein Pianist kann einem Klavierschüler nur schwer erklären, wie man gut Klavier spielt, weil das Wissen, bzw. die motorischen Fähigkeiten zum Spielen im Kleinhirn kodiert sind, was

---

<sup>26</sup> „we may put in the same class the performance of skills, wether artistic, athletic or technical“ Polanyi, M. (1966): The tacit dimension, Routledge & Kegan Paul Ltd., London.

<sup>27</sup> V.a. Frauen, da die darauf angewiesen waren sich den ganzen Tag in der Gemeinschaft zu behaupten und denen es Vorteile brachte kleinste Gefühlsschwankungen richtig zu deuten

hauptsächlich durch Übung passiert. Für einen Softwareentwickler ist Übung, sprich praktische Erfahrung mit z.B. der Programmierung, ebenfalls wichtig, aber er kann durch Lesen von Programmcode und Analysieren von Software mit entsprechender Anleitung durch einen Experten und Lehrbücher sicher mehr Lernen, als ein Klavierschüler durch Studium von Noten. Auch das oft zitierte Beispiel des Friseurlehrlings, der durch Zuschauen und Üben lernt, wie man richtig Haare schneidet, ohne dass er je ein Buch darüber gelesen hat oder viele Fachgespräche mit dem Friseurmeister führen musste ist nicht ohne weiteres auf den Softwareentwickler übertragbar, da Haare schneiden vorwiegend eine Kombination aus motorischen Abläufen und einem Blick für ästhetische Frisuren ist. Diese Beispiele geben einige Hinweise darauf, dass bildhaftes Wissen, Handlungswissen und v.a. motorische Fähigkeiten nicht oder nur sehr schwer sprachlich übertragbar sind, also einen hohen impliziten Anteil haben. Polanyi selbst weist darauf hin, dass die bildhafte Erkennung von Gesichtern einer bildhaften Methode der Beschreibung, nämlich des Zusammenstellens von Phantombildern aus einzelnen Gesichtsteilen, wie das bei der polizeilichen Ermittlung getan wird, durchaus zugänglich ist. Die Hypothese des Autors ist daher, dass Wissen vorwiegend entsprechend seiner Kodierung im Gedächtnis expliziert werden kann, also bildhaftes Wissen durch bildhafte Methoden, Handlungswissen durch Demonstration und sprachliches Wissen durch Sprache. Dies ist auch im Einklang mit multimodalen Gedächtnismodellen, die von einer Trennung von verbalen und nonverbal kodierten Gedächtnisinhalten ausgehen (Meyer, Sugiyama 2007, S. 23). Darauf aufbauend kann gefolgert werden, dass in der Softwareentwicklung aufgrund der Fundierung der Programmierung in der Logik der Mathematik und der überwiegend sprachlichen Auseinandersetzung mit der Thematik der Anteil impliziten Wissens vergleichsweise gering ist. Die Arbeit geht deswegen von weitgehender Explizierbarkeit des relevanten Wissens in der Softwareentwicklung aus.

Es ist dem Autor bewusst, dass es für den Erfolg von Softwareprojekten durchaus Faktoren gibt, bei denen ein hoher impliziter Anteil vorliegt, z.B. beim Erkennen von Widerständen gegen die Einführung einer neuen Software durch die Benutzer, oder Finden der Argumente, welche die Entscheidungsträger davon überzeugen, warum die Software der eigenen Firma den Konkurrenzprodukten vorzuziehen ist. Diese sind jedoch vorwiegend organisationsübergreifend bei der Kommunikation mit dem Kunden oder auch mit Partnern relevant und werden daher in der vorliegenden Arbeit nicht betrachtet. Dies stellt eine Einschränkung der Arbeit dar, erscheint aber aus Gründen der Komplexitätsreduktion

notwendig. Es wird also explizites oder artikulierbares Wissen fokussiert (deklaratives Wissen und Skills bei Meyer, Sugiyama 2007, S. 27).

## **4 Multiagentensimulation in den Sozial- und Wirtschaftswissenschaften**

Nachdem das Problemfeld in den vorherigen Kapiteln umfassend beschrieben wurde und eine Reihe von Theorien und empirischen Befunden zur Erklärung herangezogen wurden, nähert sich das folgende Kapitel der Operationalisierung des Untersuchungsgegenstandes an. Dazu wird zuerst das Gebiet der Computersimulation vorgestellt und insbesondere auf das Teilgebiet Multiagentensimulation (Multi Agent-Based Simulation, MABS) eingegangen. Anschließend werden die Grundlagen dieses Simulationsansatzes, die Agententechnologie, weiter detailliert und verschiedene Agentenarchitekturen ausführlich vorgestellt und auf ihre Eignung für die Simulation von Menschen hin überprüft. Im darauf folgenden Abschnitt werden weitere MABS Arbeiten analysiert, um Ergänzungen zu den generischen Architekturen zu finden. Schließlich werden verwandte Simulationsarbeiten geschildert, bevor im letzten Abschnitt das Domänenmodell operationalisiert und zum Konzeptmodell verfeinert wird.

### **4.1 Computersimulation als Forschungsansatz**

Das Problemfeld WM kann aus unterschiedlichen Perspektiven beleuchtet werden und es wurden verschiedene Lösungsmöglichkeiten vorgeschlagen. So existieren zwar mehrere empirische Studien, die WM-relevante Fragestellungen beleuchten (vgl. Maier 2002, S. 283ff; insb. Trittman, Brössler 2000), aber sie müssen aufwändig vorbereitet werden und ihre Durchführung ist sehr zeitintensiv. Außerdem muss bei der Interpretation der so gewonnenen Daten berücksichtigt werden, dass bereits die Anwesenheit des Wissenschaftlers die Entwicklung im Unternehmen beeinflussen kann. Schließlich kann man nur jeweils ein Szenario beobachten. Was unter selben Vorbedingungen in einem anderen Szenario passiert wäre, bleibt Spekulation. Die analytische Trennung einzelner Einflussfaktoren ist kaum möglich. (Pidd 2004, S. 9f) nennt als weitere Nachteile von Forschungen am realen System, dass die Kosten dabei hoch sind, Replikation der Ergebnisse kaum möglich ist und in einigen Fällen Experimente am realen System aus rechtlichen oder Sicherheitsgründen nicht möglich oder zu gefährlich sind.

Laborexperimente sind eine alternative Möglichkeit zur Datengewinnung. So gibt es z.B. im Bereich Groupware und Group Decision Support Systems Arbeiten, die sich mit WM-relevanten Fragestellungen beschäftigen (z.B. Dennis 1996; für einen Überblick siehe

Chun, Park 1998). Der Untersuchungsgegenstand ist dabei jedoch immer losgelöst von der normalen Arbeitssituation und es können i. d. R. nur einzelne Faktoren und weniger die Zusammenhänge untersucht werden.

(Pidd 2004, S. 9) nennt das Aufstellen eines exakten mathematischen Modells, also ein analytisches Vorgehen, als weitere Möglichkeit zur Problemlösung, was aber im Falle der Wissensweitergabe mit seiner Komplexität und den schlecht quantifizierbaren Größen kaum machbar scheint (vgl. Law, Kelton 2000, S. 91). Pidd führt weiterhin als Nachteil von mathematischen Modellen an, dass es nur eingeschränkt möglich ist mit nicht normalverteilten Wahrscheinlichkeiten zu arbeiten (Pidd 2004, S. 10).

Simulation als weitere Alternative zur Analyse von Sachverhalten bietet die Möglichkeit bei geringen Kosten und guter Wiederholbarkeit schnell verschiedene Szenarien durchzuspielen und zu analysieren (Brooks, Robinson 2001, S. 8f; vgl. auch Banks 1998 S. 10f für weitere Vorteile). Mit Hilfe der Simulation werden Daten generiert, die dann induktiv analysiert werden können (Axelrod 1997, S. 24). Ein weiterer Vorteil ist, dass die Simulation auch zur Schulung von Mitarbeitern eingesetzt werden kann (vgl. z.B. Pfahl et al. 2003 für eine Simulation zur Schulung von IT Projektmanagern; Vogel 2005, für eine Simulation zur Schulung von Fabrikleitern). Allerdings ist auch bei der Simulation die Vorbereitung (Erstellen und Implementieren des Simulationsmodells) aufwändig und die gewonnenen Daten müssen immer vor dem Hintergrund der getroffenen Annahmen und Vereinfachungen beurteilt werden (vgl. Banks 1998, S. 13).

Vor allem in den Sozialwissenschaften wird Multiagentensimulation als Spezialfall der Simulation immer häufiger als Alternative zu empirischen Erhebungen und Laborexperimenten eingesetzt (Axelrod 1997, S. 25). Gerade im sozialwissenschaftlichen Bereich sind Langzeitstudien über mehrere Jahre üblich, so dass der Vorteil von Simulationen, lange Untersuchungszeiträume in kurzer Zeit simulieren zu können voll zum Tragen kommt (vgl. Law, Kelton 2000, S. 92). MABS basiert auf den Forschungen im Bereich Multiagentensysteme (MAS), das sich aus der Künstlichen Intelligenz und speziell dem verteilten Problemlösen heraus entwickelt hat (vgl. Ferber 2001, S. 31), und erlaubt es nicht nur wie traditionelle Simulation, Phänomene auf einer Ebene zu untersuchen sondern sowohl die Mikroebene zu modellieren und zu analysieren als auch emergentes Verhalten auf der Makroebene zu beobachten (Klügl 2001, S. 82; vgl. Gershenson 2002, Absatz 3.6). Gerade im stark von Interdisziplinarität profitierenden Forschungsgebiet WM scheint die Multiagentensimulation ein geeignetes Mittel um ein tieferes Verständnis der Erfolgs-

faktoren und Barrieren zu erreichen. Ein besonderer Vorteil ist, dass das Überprüfen von Hypothesen, die sich auf organisatorische Maßnahmen beziehen, auch in der Simulation durch Änderungen am Umfeld erfolgen kann. Die Agenten selber können unverändert bleiben. Dadurch sind das Isolieren einzelner Faktoren und eine gezielte Untersuchung der erzielten Ergebnisse möglich.

#### **4.1.1 Klassische Simulationsansätze**

Ein großer Teil der klassischen Simulationsliteratur beschäftigt sich mit Simulationstechniken, die alternativ zum Agentenansatz sind. Häufig in der Literatur diskutierte Simulationstechniken sind z.B. System Dynamics, Monte-Carlo Simulation, Discrete Event Simulation und Neuronale Netze. Obwohl diese Arbeiten nur bedingt hilfreich für das angestrebte Simulationsprojekt sind, werden in diesem Abschnitt kurze Abrisse all dieser Simulationstechniken gegeben, um die Einordnung von MABS deutlich zu machen.

Die historische Entwicklung von Simulationsstudien in Management- und Organisationsforschung kann wie folgt geschildert werden (Canals et al. 2004, S. 7). Zuerst wurden Simulationstechniken in der Unternehmensforschung und der Unternehmensführung eingesetzt (z.B. von Law & Kelton, oder Pidd). Später wurden Simulationen basierend auf Gleichungssystemen in der Volkswirtschaftslehre und für Managementprobleme verwendet (Berends & Romme 1999; Nelson & Winter 1982; Krugman, 1996; Prietula, Carley, Gasser 1998; Carley & Gasser 1999; Levinthal 1997). Seit einiger Zeit werden agentenbasierte Simulationen im Management eingesetzt, wenn Probleme in komplexen Systemen simuliert werden sollen, wie z.B. Organisationsdesign (Rivkin & Siggelkow, 2003), Strategie (Rivkin, 2000; Rivkin & Siggelkow, 2002) oder strategisches Wissensmanagement (Rivkin, 2001; Boisot et al. 2003).

Troitzsch stellt diese Entwicklung anschaulich in einer Grafik zusammen (siehe Abbildung 4-1). Klügl teilt Simulationstechniken anhand der implementierten Simulationsmodelle in die Klassen Makro- und Mikromodelle ein, wobei individuenbasierte Modelle einen häufig verwendeten Subtyp von Mikromodellen darstellen (Klügl 2001, S. 49ff). Die Makromodelle finden sich dabei in

Abbildung 4-1 in der Spalte ganz links mit System Dynamics, World Dynamics und STELLA als Vertretern (siehe auch Engel, Möhring 1995, S. 48). Mikroanalytische Modelle und Zelluläre Automaten werden als Vertreter der Mikromodelle angeführt

werden. *Mikroanalytische Modelle* besitzen keine räumliche Komponente und simulieren Individuen mit Attributen und Zustandsübergangsfunktionen, die z.B. in Interaktionsmatrizen dargestellt werden können. Für Mikrosimulationen gab es lange Zeit keine universellen Simulationssysteme sondern nur an spezielle Anwendungsfälle angepasste Einzelentwicklungen und auch keinen geschlossenen formellen Modellierungsansatz (Engel, Möhring 1995, S. 49). *Mehrebenenmodelle* bilden sowohl die Mikro- als auch die Makroebene in einem einzigen Modell ab und ermöglichen es so, die Zusammenhänge zwischen den Individuen und den von ihnen gebildeten Populationen zu untersuchen (ibid. S.50). MIMOSE ist einer der ersten und bekanntesten Vertreter (ibid.). Auch Multiagentensimulationen sind diesem Bereich zuzuordnen (ibid. S. 51, siehe auch Abschnitt 4.1.4). Im Unterschied zu anderen Mehrebenenmodellen werden dort aber keine numerischen Verfahren angewendet.

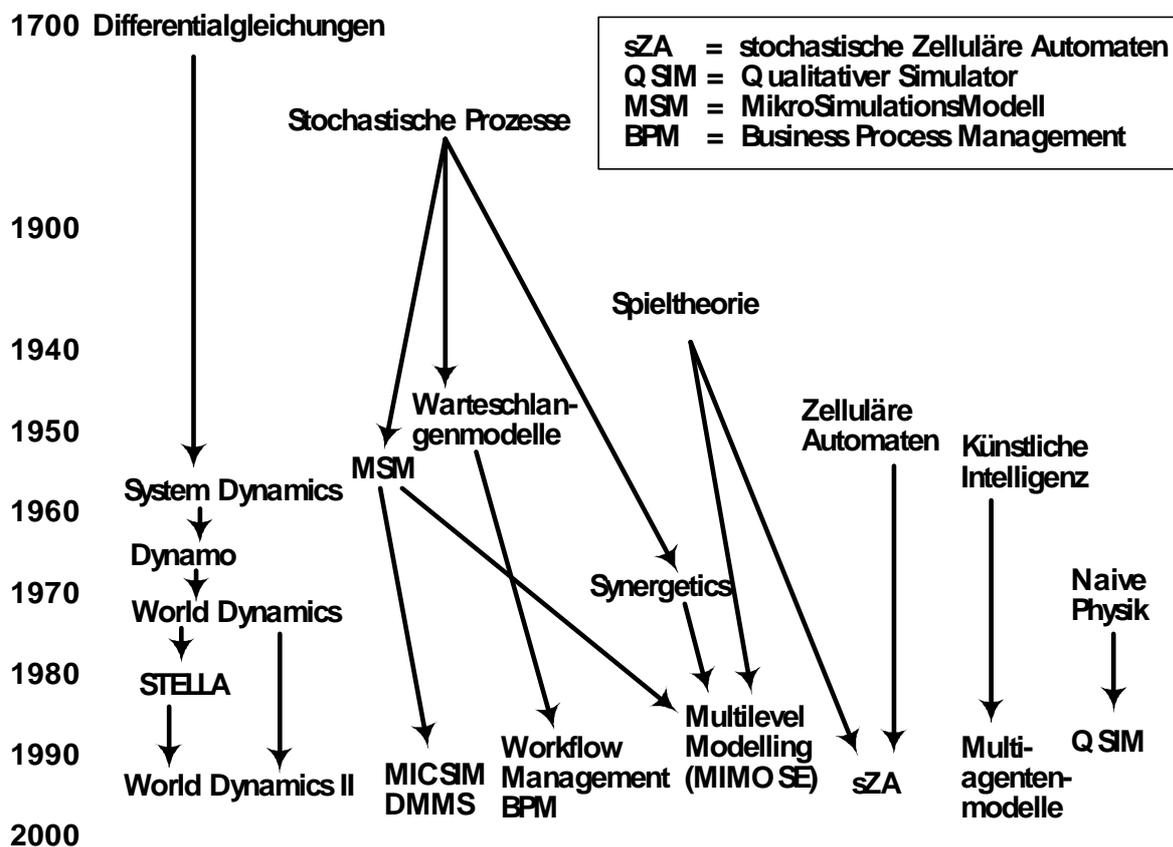


Abbildung 4-1: Historische Entwicklung von Simulationstechniken nach (Troitzsch 1997, S. 42)

*Qualitative Simulationen* beruhen auf der am menschlichen Urteilsvermögen ausgerichteten Repräsentation von Sachverhalten in nicht numerischen Werten (Engel, Möhring 1995, S. 51). Variablen sind also nominal oder ordinal skaliert, die Zeit wird ebenfalls nicht metrisch exakt abgebildet. Vorteile dieser Art der Simulation, die z.B.

durch das Werkzeug QSIM unterstützt wird, ist das Entfallen der Parameterschätzung und einer quantitativen Validierung (ibid. S. 52).

Einen neueren Ansatz der Simulation, der vorwiegend für die Abbildung von kooperativem Verhalten von Organisationsmitgliedern in Handlungskontexten mit Computerunterstützung entwickelt wurde, stellt die **Modellierung von Arbeitsprozessen** dar (in Abbildung 4-1 als Workflowmanagement und BPM bezeichnet, ibid.). Hierzu wurden vor allem Methoden aus dem Software Engineering übernommen, insbesondere solche zur grafischen Darstellung von Softwaresystemen wie z.B. Petri Netze.

**System Dynamics** geht laut Pidd auf die Arbeiten von Forrester aus dem Jahr 1961 zurück. Ursprünglich mussten Differenzgleichungen geschrieben werden, die dann von einem Computerprogramm in einer rundenbasierten Simulation benutzt wurden (Pidd 2004, S. 263). Mittlerweile können meist grafische und interaktive Modellierungsumgebungen wie z.B. Stella dafür verwendet werden. Das Problem wird dazu in eine flussorientierte Darstellung gebracht, wobei heutzutage in erster Linie Material- und Informationsflüsse betrachtet werden (ibid. S. 264). Für jede Variable des Modells wird für jeden Zeitschritt der aktuelle Wert in der Form

$$(6) \text{ Wert}_t = \text{Wert}_{t-1} + d_t * (\text{Zuflüsse} - \text{Abflüsse})$$

berechnet.

Die **Monte-Carlo Simulation** bedient sich stochastischer Verfahren zur Modellierung deterministischer Probleme (Liebl 1995, S. 55). Die Probleme müssen dabei auf die Schätzung von Flächen zurückgeführt werden, bei denen dann die Trefferquote von zufällig generierten Koordinaten im Raum  $[0; 1]$  als Schätzer dient (Hit-or-Miss Monte Carlo). Durch die Kombination von Monte-Carlo Simulation mit Markov-Ketten können auch voneinander abhängige Zufallsvariable simuliert werden (Ross 2002, S. 223ff). Monte-Carlo Simulationen sind im Allgemeinen eher statisch als dynamisch (Law, Kelton 2000, S. 90). In der Literatur wird der Begriff Monte-Carlo Simulation oft fälschlicherweise für jedwede Art von Simulation verwendet, die Zufallsgrößen benutzt (ibid.).

**Discrete Event Simulation** (Brooks, Robinson 2001, S. 87ff; Law, Kelton 2000, S. 6ff) ist eine ereignisgesteuerte Simulation (siehe auch den Absatz über „Modellierung der Zeit“ weiter unten). Es werden Entitäten simuliert, die im Fokus des Interesses stehen und auf Ressourcen zugreifen können (Pidd 2004, S. 64). Entitäten können permanent zu Klassen oder temporär zu Mengen (Sets) zusammengefasst werden und besitzen Attribute (ibid. S. 65). Auf diesen Systemobjekten werden Operationen ausgeführt, die zu Zustands-

änderungen führen. Diese Operationen werden als Aktivitäten bezeichnet (ibid. S. 66). Ereignisse werden generiert, wenn eine für die Simulation bedeutsame Zustandsänderung auftritt, wobei der Modellierer entscheidet, was bedeutsam ist. Sie können zu Gruppen sequentiell auftretender Ereignisse zusammengefasst werden, die dann als Prozess bezeichnet werden. Ein weiteres wichtiges Element ist die Simulationsuhr. Da die Ereignisse zu unregelmäßigen Zeitpunkten auftreten können wird die Simulationsuhr verwendet, um die aktuelle Simulationszeit abfragen zu können (ibid.). Schließlich beinhalten auch viele Discrete Event Simulationen einen Warteschlangenmechanismus. Damit werden sowohl offensichtliche Warteschlangen (z.B. LKWs, die auf das Beladen warten) als auch weniger offensichtliche Warteschlangen (z.B. Feuer, das in einer Simulation von Feuerwehreinsätzen auf das Löschen wartet) modelliert (ibid. S. 63).

Die *objektorientierte Simulation* überträgt die Konzepte aus der objektorientierten Programmierung auf die Simulation (Brooks, Robinson 2001, S. 99). Es werden also Konzepte wie Spezialisierung/Generalisierung, sowie Aggregation durch die Basismechanismen der Objektorientierung, Vererbung und Kapselung unterstützt (vgl. Banks 1998, S. 397ff). Es wurden auch Simulationswerkzeuge entwickelt, die eine grafische Modellierung erlauben, wie z.B. MOOSE (Cubert, Fishwick 1997). Für eine Übersicht siehe auch (Roberts, Dessouky 1998).

Eine weitere Simulationstechnik ist die Verwendung von *neuronalen Netzen*. Diese übertragen die Funktionsweise biologischer Gehirne auf Computersysteme (Müller 2002a, S. 55ff). Ein Gehirn oder allgemeiner ein Nervenzentrum besteht aus einer Ansammlung miteinander verbundener Nervenzellen. Diese Neuronen können Reize aus der Umwelt über Synapsen aufnehmen und das dadurch entstandene Signal über Axone und Dendriten zu den verbundenen Neuronen weiterleiten. Dabei kann die Verbindung sowohl verstärkend als auch hemmend sein. Dadurch sind komplexe Verschaltungen von Signalen möglich. Weiterleitungen erfolgen nur, wenn eine gewisse Reizschwelle überschritten wird. Mathematisch lässt sich dies als gewichtete Summe von Signalen darstellen, die bei Überschreiten eines Schwellenwerts weitergeleitet werden. Derart gestaltete neuronale Netze können lernfähig gemacht werden, wenn die Gewichte im Laufe der Zeit angepasst werden. Damit können z.B. soziale Netzwerke (Müller 2002a) oder auch risikoreiche Projekte (Badiru, Sieger 1998) simuliert werden.

*Zelluläre Automaten* beruhen auf einer expliziten Raumdarstellung als n-dimensionales Gitter aus Zellen (Klügl 2001, S. 54f). Jede Zelle ist mit ihren Nachbarzellen verbunden,

wobei je nach Nachbarschaftsform Verbindungen zu unterschiedlich vielen Nachbarzellen bestehen (siehe Abbildung 4-2).

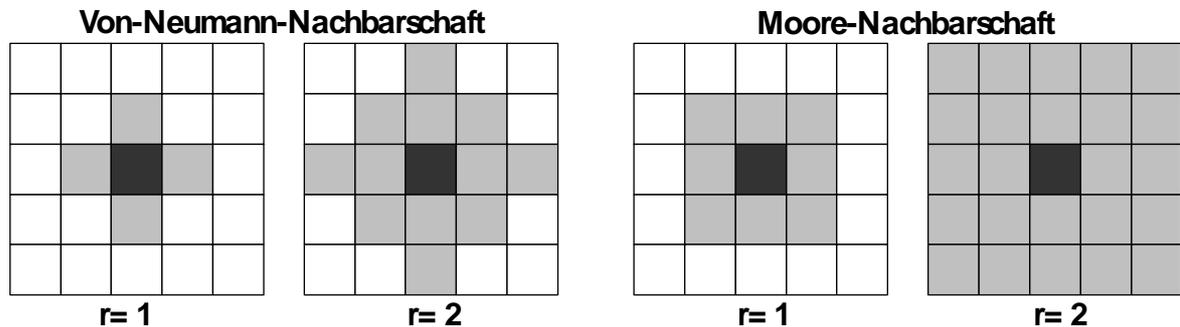


Abbildung 4-2: Verschiedene Formen von Nachbarschaft nach (Klügl 2001, S. 55)

Jede Zelle besitzt einen diskreten Zustand, der auf Basis des Zustands der Nachbarzellen in jeder Zeiteinheit neu berechnet wird. Die Aktualisierung berücksichtigt dabei nur lokale Informationen und ist für alle Zellen einheitlich. Die Zustandsüberföhrungsfunktionen können sowohl deterministische als auch stochastische Komponenten enthalten. Zelluläre Automaten haben eine feste Position im Gitter und können nicht wandern. Sie sind darüber hinaus von der Logik her meist sehr einfach gehalten.

#### 4.1.2 Grundlagen der Simulation

Einige grundlegende Überlegungen der klassischen Simulationsliteratur sind unabhängig von der konkreten Simulationstechnik und damit auch für Multiagentensimulationen relevant. (Engel, Möhring 1995) listen aufbauend auf Troitzsch eine Reihe von Merkmalsdimensionen anhand derer sich Simulationen klassifizieren lassen (S.41f). Diese sollen herangezogen werden, um einige Simulationsgrundlagen vorzustellen.

1. Bildbereich: formelle vs. nicht formelle Modelle
2. Verwendungszweck: Beschreibung, Erklärung, Prognose, Entscheidung
3. Zeitabhängigkeit: statisch oder dynamisch
4. Zeitstruktur: diskret oder kontinuierlich
5. Determiniertheit: deterministisch oder stochastisch
6. Anzahl der Modellebenen: Mikro-, Makro-, Mehrebenenmodelle
7. Beziehungen zwischen den Modellelementen: linear oder nicht linear
8. Begründungszusammenhang: daten- oder konzeptbasiert, empirisch oder theoretisch
9. Wertebereich der Merkmale von Modellelementen: qualitativ oder quantitativ

Die Modellierung der *Zeit* als gleiche Zeitintervalle (rundenbasierte Simulation) ist alternativ zum Ansatz der ereignisgesteuerten Simulation, bei der das Modell nur nach Auftreten eines neuen Ereignisses aktualisiert wird, was nach variablen Zeiteinheiten passieren kann (Pidd 2004, S. 16ff). Der Vorteil ereignisgesteuerter Simulationen ist, dass längere Zeiten der Inaktivität keine Rechenzeit kosten und Perioden mit hoher Aktivität mit entsprechend kleinen Zeitintervallen durchlaufen werden können (S. 18). Zudem ist dem Beobachter schneller klar, wann bedeutsame Ereignisse in der Simulation aufgetreten sind. Deswegen kommt Pidd zu dem Schluss, dass ereignisgesteuerte Simulationen rundenbasierten vorzuziehen sind. Für Multiagentensimulationen ist jedoch die Gleichzeitigkeit der Aktionen zentral, so dass Interaktion und gegenseitige Beeinflussung zu emergenten Makrophänomenen führen können, weshalb dort eine rundenbasierte Umsetzung notwendig ist.

Eine weitere Grundsatzentscheidung bei Simulationen ist, ob *stochastische Einflüsse* modelliert werden oder die Simulation deterministisch ist. Kann das Verhalten eines Systems nicht exakt vorhergesagt werden sondern nur Wahrscheinlichkeitsverteilungen für zukünftige Zustände angegeben werden, so muss es mit stochastischen Variablen abgebildet werden (Pidd 2004, S. 19). Für die Simulation von Wissensweitergabe trifft das zu.

Weiterhin muss entschieden werden, ob *Zustandsänderungen* sich nur in diskreten Schritten ereignen oder auch kontinuierlich erfolgen können. In gängigen Computersystemen wird zwar generell nur mit diskreten Werten gerechnet, die Annäherung, die über Gleitkommazahlen mit hoher Genauigkeit erzielt wird kann jedoch für die meisten Fälle als ausreichend gelten (Pidd 2004, S. 26). Auch können einige Zustände diskret und andere kontinuierlich veränderlich sein. Für die Modellierung von Wissen wird dieser Frage später in dieser Arbeit noch genauer untersucht werden (siehe Abschnitt 4.5.2).

Für stochastische Simulationen müssen *Zufallsvariablen* generiert werden, die einer bestimmten empirisch beobachteten oder vermuteten Verteilung entsprechen. Die Güte eines Zufallsgenerators bemisst sich nach folgenden Kriterien (Liebl 1995, S. 26f).

- Unabhängigkeit: Elemente jeder Teilfolge müssen paarweise unabhängig sein
- Entsprechung der Verteilung: kein Intervall innerhalb der Grenzen sollte häufiger besetzt sein als das durch die Verteilungsfunktion vorgegeben ist.
- Besetzungsdichte: für Gleitkommavariablen sollte eine entsprechende Genauigkeit und damit genügend viele verschiedene Zahlen innerhalb eines Intervalls möglich sein

- Effizienz: die Zahlen sollten schnell und mit geringem Speicherverbrauch erzeugt werden
- Reproduzierbarkeit: es sollte möglich sein, aus einer definierten Startsituation immer wieder die gleichen Zufallszahlen zu erzeugen, um Simulationsergebnisse reproduzieren zu können.

Die Grundlage für Zufallszahlen bieten gleichverteilte Zufallszahlen, wie sie von den meisten Programmiersprachen leicht erzeugt werden können. In Java können z.B. Zufallszahlen mit der Klasse `java.util.Random` erzeugt werden. Sie wird standardmäßig mit der aktuellen Systemzeit in Millisekunden initialisiert, kann aber auch mit einem festen Wert initialisiert werden, um reproduzierbare Zufallsreihen (Pseudozufallszahlen) zu erhalten<sup>28</sup>. Daraus lassen sich dann Zufallszahlen in anderen Verteilungen ableiten (Liebl 1995, S. 47). So erzeugt das Verfahren von Muller aus zwei Zufallszahlen  $y_1$  und  $y_2$  aus dem Intervall  $[0; 1]$  zwei unabhängige, standardnormalverteilte Zufallszahlen  $x_1$  und  $x_2$  mit folgender Formeln (ibid.):

$$(7) \quad x_1 = \sqrt{-2 * \log y_1} \cos(2 * \pi * y_2); \quad x_2 = \sqrt{-2 * \log y_1} \sin(2 * \pi * y_2)$$

Um statt der Standardnormalverteilung eine Normalverteilung  $N(\mu, \sigma^2)$  mit Erwartungswert  $\mu$  und Standardabweichung  $\sigma$  zu erhalten, müssen die Formeln wie folgt abgewandelt werden.

$$(8) \quad x_1 = \mu + \sqrt{-2 * \sigma^2 * \log y_1} \cos(2 * \pi * y_2); \quad x_2 = \mu + \sqrt{-2 * \sigma^2 * \log y_1} \sin(2 * \pi * y_2)$$

Zur Generierung von Zufallszahlen, die der  $\chi^2$ -, F-, t-, Poisson-, oder Exponentialverteilung genügen, siehe (Liebl 1995, S. 48f; Ross 2002, S. 37ff, 45ff, 63ff; Banks 1998, S. 93ff).

### 4.1.3 Social Simulation

Das Forschungsgebiet Social Simulation entstand Ende der 80er und Anfang der 90er Jahre durch verstärkte Bemühungen der Sozialwissenschaftler, soziale Phänomene mittels Computersimulation zu erforschen (Conte et al. 1997, S. 13), und hat die Vorherrschaft von Differentialgleichungen zur Abbildung der Dynamik sozialer Systeme gebrochen (Engel, Möhring 1995, S. 47). Dabei wurden sowohl Mikro- als auch Makromodelle zur Untersuchung herangezogen die dank fortgeschrittener Technologien im Hard- und

---

<sup>28</sup> Siehe <http://java.sun.com/j2se/1.5.0/docs/api/java/util/Random.html>, JavaDoc Dokumentation der Klasse `java.util.Random` im Java SDK, letzter Abruf am 29.12.2006

Softwarebereich trotz einer angemessenen Komplexität der Modelle schnell genug berechnet werden konnten. Bereits Mitte der 90er Jahre, aber v.a. seit Ende der 90er, wurden immer mehr agentenorientierte Konzepte für die Simulation sozialer Phänomene eingesetzt (Axelrod 1997, S. 25; Troitzsch 1997, S. 42). Simulationen in sozialwissenschaftlichen Gebieten können zur Herleitung mathematischer Modelle verwendet werden oder als eigenständiges Forschungsinstrument (Troitzsch 1997, S. 41). Im zweiten Fall können Vorhersagemodelle von Erklärungsmodellen unterschieden werden (ibid. S. 46). Während Vorhersagemodelle schnell komplex werden, um eine möglichst gute Annäherung an die tatsächlich eintretenden Sachverhalte zu erreichen (z.B. Wettervorhersage), müssen Erklärungsmodelle so einfach wie möglich gehalten werden. Eine Simulation der Wissensweitergabe muss soziale Aspekte zwingend berücksichtigen, da ein isolierter Mensch nur einen Bruchteil des Wissens lernen kann, das er in sozialen Gruppen lernt (Gershenson 2002, Absatz 5.1).

Terán identifiziert zwei verschiedene Typen von sozialwissenschaftlichen Simulationsmodellen, solche, die den Fokus auf die Entscheidungen von Individuen legen und solche, die die Verbindungen der Individuen untereinander fokussieren (Terán 2004, Absatz 4.3). Je nach Anwendungsgebiet wird der eine oder der andere Aspekt genauer modelliert und stärker berücksichtigt. Die Agenten können als rationale Agenten modelliert werden, wenn soziale Entscheidungen fokussiert werden, oder als adaptive, verhaltensbasierte Agenten, wenn soziales Verhalten fokussiert wird (Gershenson 2002, Absatz 4.7). Die vorliegende Arbeit geht von den Entscheidungen der Individuen aus, versucht jedoch auch die sozialen Aspekte der Verbindungen zwischen den Individuen genügend zu berücksichtigen, da sie für die Wissensweitergabe wichtig sind (siehe Abschnitt 2.2.3).

Goldspink stellt fest, dass von den von McKelvey identifizierten Ordnungskräften, physikalische, organische, rationale Ordnung und Komplexität (im Sinne der Komplexitätstheorie, vgl. z.B. Byrne 1998; McKelvey 1999), für sozialwissenschaftliche Simulationen zumindest die letzten drei von Bedeutung sind (Goldspink 2000, Absatz 1.4). Es sollten also die Ergebnisse natürlicher Selektion (z.B. Freundschaftsbeziehungen, aber auch genetische Algorithmen), rationaler Entscheidungen von Individuen (z.B. Wissen weiterzugeben) und die Strukturen, die sich aus der Komplexität des Modells ergeben berücksichtigt werden. Nach Goldspink sollte v.a. vermieden werden, die Ordnung, die sich aus der Komplexität ergibt mit der rationalen Ordnung zu verwechseln (ibid.). Die Komplexität eines (sozialen) Systems steigt mit der Anzahl der Elemente, der

Interaktionen zwischen den Elementen und der Komplexität der Interaktionen (Gershenson 2002, Absatz 3.3). Weiter stellt Goldspink fest, dass der Begriff „sozial“ nicht zu eng gefasst werden sollte, um nicht interessante Phänomene auszuschließen (Goldspink 2000, Absatz 2.1). Zusammenfassend identifiziert Goldspink folgende Elemente, die in sozialwissenschaftlichen Simulationen vorkommen sollten (ibid. Absatz 4.1).

- Ein Medium, in dem die sozialen Akteure interagieren können
- Autonome, operational geschlossene Individuen, die als Agenten bezeichnet werden. Dabei werden primäre Agenten, die im Mittelpunkt des Interesses stehen und biologische Individuen repräsentieren von sekundären Agenten unterschieden, die biologisch sein können, aber nicht müssen und sogar passive Objekte verkörpern können
- Systeme von Agenten, in denen die primären Agenten organisiert sind
- Systeme von Systemen von Agenten, die koexistieren, sich aber auch überschneiden können

Für die Wissensweitergabe in der Softwareentwicklung ist das Medium der physische Raum, in dem sich die Mitarbeiter bewegen und der ihnen Gelegenheit zur Interaktion gibt. Für räumlich getrennte Standorte bilden elektronische Medien teilweise Ersatz für den physischen Raum und damit einen weiteren Teil des Mediums. Die primären Agenten sind die Mitarbeiter der Organisation. Sekundäre Agenten stellen z.B. Computer und Anwendungssysteme dar. Systeme, in denen die Agenten organisiert sind, stellen z.B. Abteilungen oder Teams dar. Diese können auch koexistieren und Teams können aus Mitgliedern unterschiedlicher Abteilungen gebildet werden. Die Systeme ergeben sich dabei als emergente Makro-Phänomene aus den Individuen und ihren Interaktionen (Gershenson 2002, Absatz 3.5).

### 4.1.4 Multiagentensimulation

Beide Zweige der verteilten künstlichen Intelligenz (VKI, auch DAI, distributed artificial intelligence), Multiagentensimulation (multi-agent based simulation, MABS) und verteiltes Problemlösen (distributed problem solving, DPS), bedienen sich in ihrer Arbeit der Multiagentensysteme. Die Weiterentwicklung der klassischen Simulation mit Hilfe von Softwareagenten wurde maßgeblich von sozialwissenschaftlichen Forschungen beeinflusst. (Davidsson 2002) grenzt begrifflich die Gebiete Social Aspects of Agent Systems (SAAS), welches sich mit der Ausgestaltung von Regeln und Normen für Multiagentensysteme

beschäftigt, Multi-Agent Based Simulation (MABS), welches Agentenkonzepte auf Simulationen ohne sozialen Bezug anwendet, Social Simulation, welches der Erforschung sozialer Phänomene mit klassischer Simulation nachgeht und Agent-Based Social Simulation (ABSS) voneinander ab, welches sozialwissenschaftliche Fragestellungen mit Agententechnologie modelliert.

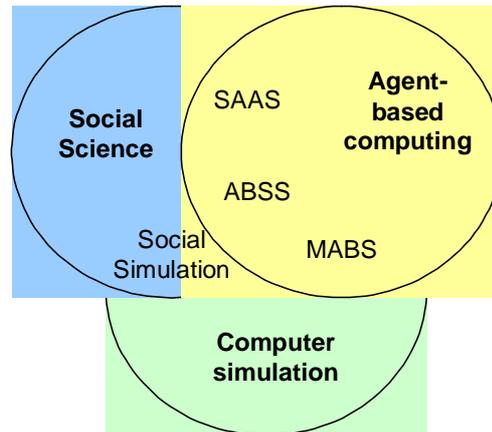


Abbildung 4-3: Abgrenzung der Forschungsgebiete nach (Davidsson 2002)

In der Praxis hat sich jedoch vor allem der Begriff MABS durchgesetzt, obwohl streng genommen viele Forschungsarbeiten dem Gebiet ABSS zuzuordnen wären. „Echte MABS Systeme“ ohne sozialwissenschaftlichen Charakter sind z.B. die Simulation von fahrerlosen Transportsystemen (z.B. Günther et al. 2001), die Simulation von Multiagentensystemen selbst (z.B. Horling et al. 2000) und diverse Simulationen in der Biologie, z.B. mit Ameisen (Drogoul, Ferber 1992) oder Bienen (Dornhaus et al. 1998), die einerseits nahe an den Forschungen zu Artificial Life liegen, andererseits aber auch soziale Aspekte berücksichtigen, da es sich ja um staatenbildende Tiere handelt, die für ihre sozialen Verhaltensweisen bekannt sind. Interessant für die vorliegende Arbeit sind aber hauptsächlich Agentenarchitekturen für MABS von Menschen (z.B. Urban 2000) und die Herangehensweise an diese Art der Simulation (Drogoul et al. 2002).

MABS setzen sich heute in vielen Bereichen durch, z.B. der Soziologie, Biologie, Physik, Chemie und Ökonomie, und lösen andere Simulationstechniken ab (Drogoul et al. 2002, S. 1). Gründe dafür sind die einfache Abbildung unterschiedlichster Individuen von einfachen reaktiven bis hin zu komplexen kognitiven Agenten sowie die gute Untersuchungsmöglichkeit unterschiedlicher Abstraktionsstufen (Individuen, Gruppen und Gesamtgesellschaft/Organisation, ibid., S.1). Weitere Details zu MABS finden sich in den Kapiteln 4.2 und folgende.

#### 4.1.5 Vorgehen bei der Entwicklung von Simulationsmodellen

Wissenschaftler sind i. d. R. mehr daran interessiert, ein Modell zu entwickeln, als an der Durchführung von Simulationsexperimenten und der Auswertung der Ergebnisse, da diese beiden Tätigkeiten weniger als wissenschaftliche Arbeit betrachtet werden (Drogoul et al. 2002, S. 3). Andererseits sind die Arbeitsschritte eng miteinander verbunden und der Simulationsprozess besteht aus einer Reihe von Iterationen der Schritte.

Nach (Troitzsch 1990, S. 6f) kann man vier Phasen für die Simulation unterscheiden (ursprünglich nach Zeigler 1976, S.4, siehe Abbildung 4-4). Man betrachtet ein reales System und identifiziert den Ausschnitt, den man simulieren will. Dann sucht man nach geeigneten Theorien, die zur Repräsentation des Systems herangezogen werden können. Diese werden zur Modellierung benutzt, so dass ein (semi-)formelles Modell des Ausschnitts des realen Systems entsteht. Dieses muss weiter formalisiert werden, so dass es in einem Softwareprogramm abgebildet werden kann, das die Simulation letztendlich durchführt.

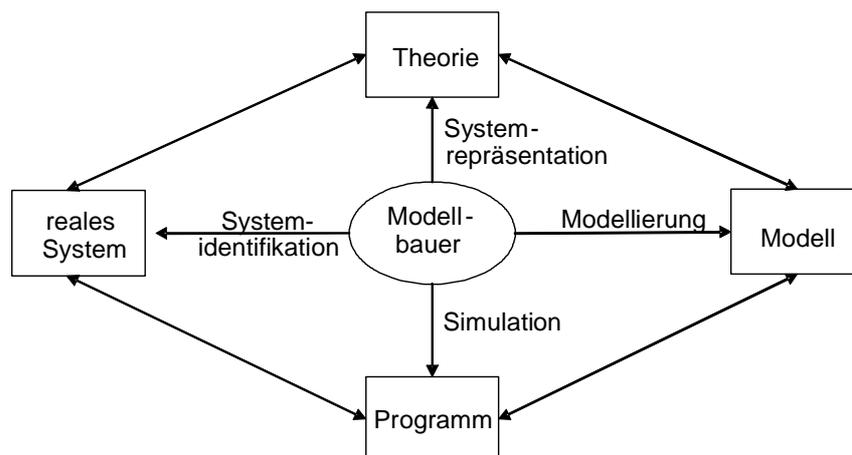


Abbildung 4-4: Vorgehen bei der Entwicklung einer Simulation nach (Troitzsch 1990, S. 6)

Ähnlich wie Troitzsch beschreiben auch Brooks und Robinson das Vorgehen, sie subsumieren aber die Systemidentifikation und -repräsentation in der Erstellung eines konzeptuellen Modells und betonen dafür die Analyse der Ergebnisse und deren Anwendung im realen System (Brooks, Robinson 2001, S. 63ff).

Drogoul et al. bauen auf die Arbeiten von Fishwick, Gilbert und Troitzsch auf und entwickeln ein detaillierteres Vorgehensmodell, das in Abbildung 4-5 dargestellt ist (vgl. Drogoul et al. 2002). Es ist speziell auf die Anforderungen von MABS abgestimmt. Auf Basis von Beobachtung des realen Systems werden Annahmen auf Mikroebene getroffen. Diese werden in ein Domain Modell überführt, das eine agentenorientierte

Beschreibung des realen Systems darstellt. Daraus kann ein Design Modell abgeleitet werden, welches formaler ist und von einem Teil der Komplexität des Domain Modells abstrahiert. Die Umsetzung in ein operationales Modell ist die Vorstufe der eigentlichen Programmierung und in der Praxis oft identisch damit (Drogoul et al. 2002, S. 7f), was vor dem Hintergrund aktueller Bemühungen um Model-driven Development auch nachvollziehbar ist (Beydeda et al. 2005). Nach der Implementierung müssen Tests durchgeführt werden, wodurch die fehlerfreie Implementierung nachgewiesen werden soll (Verifikation). Anschließend können Simulationsläufe absolviert werden, welche die inhaltliche Eignung des Modells überprüfen (Validierung).

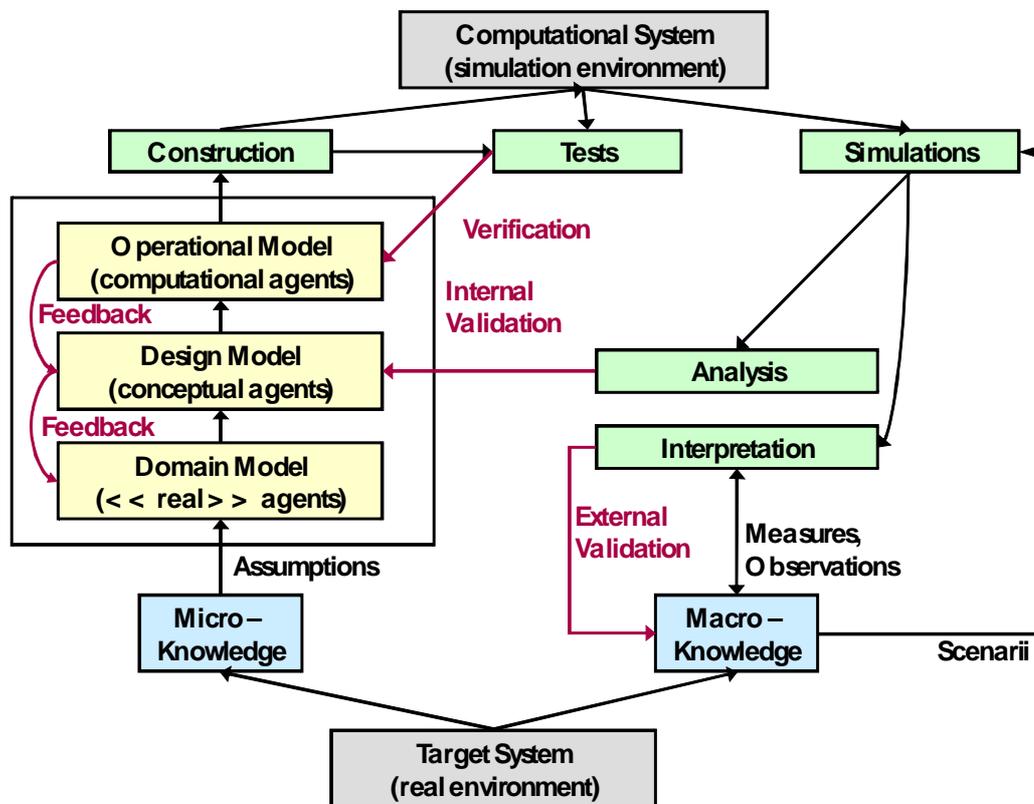


Abbildung 4-5: Vorgehensmodell MABS nach (Drogoul et al. 2002, S. 9)

Opwis und Plötzner gehen nur von drei Schritten bis zum Simulationssystem aus. Sie sehen das Entwickeln einer Konzeptualisierung der Anwendungsdomäne mit Hilfe einer Ontologie als ersten Schritt. Dieses wird durch Formalisierung in ein formelles Modell überführt, welches in einer formellen Sprache mit Hilfe von Axiomen repräsentiert werden kann und im dritten Schritt in Datenstrukturen und Prozeduren implementiert wird (Opwis, Plötzner 1996, S. 6). Terán argumentiert unabhängig davon, dass in vier Sprachbereichen wichtige Entscheidungen zu treffen sind. Er identifiziert die Bereiche (Terán 2004 Absatz 3.6)

1. Natürliche Sprache und Kultur (z.B. Lateinamerikanisch, Angelsächsisch)
2. Theorie- und Modellierungsparadigmen (z.B. Simons Bounded Rationality, Cyert und Marchs Behavioral Theory of the Firm)
3. Modellierungssprache (z.B. MABS, Ereignis-orientierte Simulation)
4. Simulationsprogrammiersprache (z.B. Arena, Java)

Terán weist dabei besonders darauf hin, dass es durchaus üblich ist, sich im Bereich zwei auf mehrere Theorien zu stützen. Dies ist laut Terán solange in Ordnung, wie das Simulationsmodell nicht im Widerspruch zu den Grundannahmen der Theorien steht.

Einen etwas anderen Fokus legt Banks (Banks 1998, S. 15ff). Er beschreibt zwölf Phasen eines Simulationsprojekts.

1. Problem formulieren
2. Ziele setzen und Gesamtplan erstellen
3. Modell konzeptualisieren
4. Daten sammeln
5. Modell übersetzen (in ein Simulationsprogramm)
6. Verifizieren
7. Validieren
8. Experimente planen
9. Simulationsläufe durchführen und analysieren
10. Weitere Simulationsläufe bei Bedarf durchführen
11. Dokumentation der Ergebnisse
12. Implementieren der simulierten Maßnahmen im realen System

Inhaltlich sehr ähnlich, aber anders zusammengefasst präsentieren Law und Kelton das Vorgehen bei einem Simulationsprojekt in zehn Phasen (Law, Kelton 2000, S. 84ff). Bemerkenswert ist einzig die Validierung des konzeptuellen Modells, die bei Law und Kelton auftaucht.

Als Erfolgsfaktoren für Simulationen werden in (Musselman 1998) neben allgemeinen Faktoren, die für jede Art von Softwareprojekt für Kunden zutreffen, auch einige simulationsspezifische genannt. Besonders hervorzuheben sind davon

- Standards zu verwenden

- Zu verstehen, was die Grenzen des Modells sind
- Die Ergebnisse zu hinterfragen
- Einfach zu starten
- Komplexität handhabbar zu gestalten
- Das Problem nicht aus den Augen zu verlieren
- Zu wissen, wann man aufhören muss
- Vor der Modellbildung das Ziel zu spezifizieren

Bruderer und Maiers (1997) erarbeiten Vorschläge, wie Simulationsprogramme populärer gemacht werden können. Diese können ebenfalls als Hinweise für erfolgreiche Simulationen interpretiert werden. Sie fordern

- Anbieten einer einfach zu bedienenden Benutzerschnittstelle
- Verwenden der Simulation als Schulungsinstrument
- Entwickeln von modularen und objekt-orientierten Simulationen
- Verfügbarmachen der Simulation für andere
- Replizieren und Erweitern anderer Arbeiten
- Verwenden angemessener statistischer Methoden zum Vergleich von Modellen

Die vorliegende Arbeit folgt dem Vorgehensmodell von Drogoul et al. (siehe auch Abbildung 4-5) und versucht darüber hinaus, die Erfolgsfaktoren, die sich bei anderen Autoren finden, zu berücksichtigen.

### **4.2 Softwareagenten in Multiagentensystemen**

Bevor das Gebiet der Multiagentensimulation genauer analysiert wird, werden in den folgenden Abschnitten die Grundlagen der Agententheorie erläutert. Dazu wird zu Beginn der Begriff des Softwareagenten definiert und anschließend Merkmale diskutiert, die Softwareagenten von anderen Softwaresystemen oder -komponenten unterscheiden. Anschließend werden Multiagentensysteme vorgestellt, in denen mehrere Agenten interagieren.

#### **4.2.1 Definition und Eigenschaften**

Pragmatisch und knapp kann ein Softwareagent als eine autonom agierende Softwareeinheit betrachtet werden. Vor diesem Hintergrund werden im Folgenden einige

Definitionen aus der Literatur vorgestellt. Eine sehr einfache und wenig diskriminierende Definition stammt von Russel und Norvig, die einen Agenten ganz allgemein als irgendetwas definieren, das seine Umwelt durch Sensoren wahrnimmt und durch Aktuatoren beeinflusst (Russell, Norvig 2003, S. 32). Sie schränken diese Definition für rationale Agenten weiter ein: „Für jede mögliche Reihe von Wahrnehmungen sollte ein rationaler Agent eine Aktion auswählen, die sein Leistungsmaß voraussichtlich bei der gegebenen Reihe an Wahrnehmungen und dem eingebauten Wissen des Agenten maximiert“ (Russell, Norvig 2003, S. 36)<sup>29</sup>. Eine populäre Definition stammt von Wooldridge und Jennings: „Ein Agent ist ein Computersystem, welches in einer Umgebung situiert ist und in der Lage ist, autonome Aktionen in dieser Umgebung durchzuführen, um seine vorgegebenen Ziele zu erreichen“ (Jennings, Wooldridge 1998, S. 4)<sup>30</sup>. Ähnlich lautet auch die Definition von Kirn. Er definiert einen Agenten als „[...] ein Softwaresystem, welches Autonomie sowie die Fähigkeit zu problemlösendem Verhalten aufweist und innerhalb einer Gruppe von Agenten deshalb auf einem Rationalitätskalkül basierende Interaktionen planen und ausführen kann“ (Kirn 2002, S. 57). Eine Definition, die sehr viele Aspekte berücksichtigt stammt von Ferber. Er definiert einen Agenten als „eine physische oder virtuelle Entität, (a) die in einer Umgebung handeln kann, (b) direkt mit anderen Agenten kommunizieren kann, (c) durch eine Menge von Tendenzen, in Form von individuellen Zielen oder von Befriedigungs- oder Überlebensfunktionen, die sie zu optimieren versucht, (d) die über eigene Ressourcen verfügt, (e) die in der Lage ist, ihre Umgebung (zu einem gewissen Grad) wahrzunehmen, (f) die nur eine partielle Repräsentation ihrer Umgebung hat [...], (g) die Fähigkeiten besitzt und Dienste anbieten kann, (h) die in der Lage sein kann, sich selbst zu reproduzieren und (i) deren Verhalten dazu tendiert, ihre Ziele zu erreichen, soweit das ihre Fähigkeiten und Ressourcen zulassen und in Abhängigkeit ihrer Wahrnehmungen, ihrer Repräsentation und den Nachrichten, die sie erhält“ (Ferber 1999, S. 9).

Neben diesen Definitionen haben sich zwei verschiedene Sichten auf den Agentenbegriff herausgebildet, die sich durch das Zuschreiben verschiedener Attribute unterscheiden. Der

---

<sup>29</sup> Im Original: “For each possible percept sequence, a rational agent should select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has.”

<sup>30</sup> Im Original: „First, an agent is a computer system situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.“

schwache und der starke Agentenbegriff (weak and strong notion of agency) bauen beide auf gleicher Grundlage auf. Demzufolge zeichnet sich ein Agent durch Autonomie, Reaktivität, Proaktivität und Sozialität, also dem Interagieren mit anderen Agenten aus (für eine ausführliche Diskussion vgl. Wooldridge, Jennings 1994, S. 2ff; für eine Zusammenfassung vgl. Wooldridge, Jennings 1995, S. 4f; Luck, d'Inverno 2001, S. 4). Besitzt eine Softwareeinheit diese Eigenschaften so kann sie nach dem schwachen Agentenbegriff als Agent bezeichnet werden. Der starke Agentenbegriff fordert darüber hinaus, dass Agenten anthropomorphe Eigenschaften wie Überzeugungen, Wünsche, Intentionen, Wissen und ähnliche mentale Konzepte besitzen (vgl. auch Gandon 2002, S. 139).

Einige Autoren machen die Abgrenzung zwischen Objekten in der objektorientierten Programmierung und Agenten deutlich, indem sie einige plakative Gegenüberstellungen vornehmen. So finden sich die Aussagen: „objects do it for free, agents for money“ (Jennings et al. 1998, S. 283), und „ein Objekt kapselt seinen Zustand, ein Agent sein Verhalten“ (Jennings, Wooldridge 1998, S. 4). Diese weisen darauf hin, dass Agenten eigene rationale Ziele verfolgen und ein erwünschtes Verhalten nicht ohne weiteres von außen induziert werden kann, wie das bei Objekten der Fall ist. Eine umfangreichere Gegenüberstellung findet sich in (Odell 2002).

Die meisten Autoren stimmen darin überein, dass über eine grundlegende Definition hinaus die den Agenten zugeschriebenen Eigenschaften entscheidend sind und definieren eine Vielzahl solcher Eigenschaften die im folgenden aufgeführt werden (Wooldridge, Jennings 1995, S. 4f; Nwana 1996; Ferber 1999, S. 9; Klügl 2001, S. 14f; Kirn 2002):

- autonom: Ein Agent soll selbst darüber entscheiden können, welche Aktionen er ausführt und nicht durch Funktionsaufrufe zu einem bestimmten Verhalten gezwungen werden können.
- benevolent: Ein Agent soll sich anderen Agenten gegenüber wohlwollend verhalten und diese nicht korrumpieren.
- deliberativ: Ein Agent soll seine Aktionen dynamisch zu Plänen zusammenstellen können, um seine Ziele zu erreichen. Er soll selbständig erkennen, wann ein Plan nicht mehr verwirklichtbar ist und ihn dann zugunsten eines anderen aufgeben.
- emotional: Ein Agent soll analog zu Menschen verschiedene Emotionen haben können, die sein Verhalten beeinflussen.

- intentional: Ein Agent soll absichtsvoll handeln.
- kommunikativ: Ein Agent soll mit anderen Agenten oder einem Benutzer kommunizieren können und dadurch seine Wissensbasis erweitern und aktualisieren.
- kooperativ: Ein Agent soll mit anderen Agenten zusammenarbeiten können, so dass Aufgaben, die von einem Agenten alleine nicht erfüllt werden können zusammen erledigt werden.
- lernend: Ein Agent soll lernen, welche Aktionen unter welchen Bedingungen Erfolg versprechend sind und welche vermutlich scheitern, und sein Verhalten entsprechend anpassen.
- mobil: Ein Agent soll sich in seiner Umgebung bewegen können oder sogar seine Umgebung verlassen und in eine neue Umgebung eintreten können (auch als Migration bezeichnet, siehe offene MAS im Abschnitt 4.2.3).
- pro-aktiv: Ein Agent soll seine Ziele auch ohne expliziten Stimulus von Außen selbstständig verfolgen können.
- rational: Ein Agent soll nach bestem Wissen versuchen seine Ziele zu erreichen und seine Aktionen basierend auf einem nachvollziehbaren Calculus auswählen.
- reaktiv: Ein Agent soll auf Änderungen in seiner Umgebung reagieren können.
- situiert: Ein Agent agiert in einer Umgebung, die er über Sensoren wahrnehmen und über Effektoren (auch als Aktoren bezeichnet) beeinflussen kann.
- sozial: Ein Agent soll mit anderen Agenten interagieren können.

Auf die Agenten, die in dieser Arbeit modelliert werden treffen alle Eigenschaften bis auf „mobil“ zu. Die Mobilität ist nicht erforderlich, da nur ein Simulationssystem vorhanden ist und damit keine Notwendigkeit besteht, sich zwischen verschiedenen Systemen zu bewegen.

### 4.2.2 Wissensbasis

Ein zentrales Element für einen Agenten ist die Wissensbasis. Sie enthält formal eine Reihe von Sätzen, die in einer Wissensrepräsentationssprache ausgedrückt werden (Russell, Norvig 2003, S. 195). Die Sätze stellen die Fakten der Wissensbasis dar. Zusätzliches Wissen kann durch Inferenz aus bestehenden Fakten und Regeln abgeleitet werden (ibid.). Fakten können durch Wahrnehmung der Welt oder durch Kommunikation mit anderen Agenten erworben werden. Initial durch den Programmierer vorgegebenes

Wissen des Agenten wird als Hintergrundwissen bezeichnet (ibid., S. 196). Wissen kann deklarativ oder prozedural repräsentiert werden. Konkrete Ansätze dazu sind Prädikatenlogik (First-Order Logic), Frame Logic und Description Logic (Fensel 2004, S. 68). Eine Wissensrepräsentationssprache in der Künstlichen Intelligenz (KI) muss

- qualitatives Wissen beschreiben
- aus Fakten und Regeln neues Wissen ableiten
- spezielle Situationen und allgemeine Prinzipien gleichermaßen repräsentieren
- komplexe semantische Bedeutungen wiedergeben
- und Metawissen darstellen können (Luger 2001, S. 60).

Für viele Agentensysteme ist es aber nicht zwangsläufig nötig eine Wissensrepräsentation zu besitzen, die alle Forderungen von Luger erfüllt. Selbst im starken Agentenbegriff ist z.B. keine Forderung nach automatischem Schlussfolgern vorhanden, sondern nur eine Umsetzung des Konzeptes Wissen gefordert. Andererseits ist für die Modellierung von menschlichen Akteuren insbesondere das Metawissen von Interesse, da nur durch die Kenntnis der Fähigkeiten anderer Agenten eine dezentrale Zuweisung von Aufgaben möglich ist (siehe Abschnitt 4.2.5).

### 4.2.3 Multiagentensysteme (MAS)

Multiagentensysteme sind Systeme, in denen zwei oder mehrere Agenten interagieren oder zusammenarbeiten, um eine Menge von Aufgaben zu erledigen, oder eine Menge von Zielen zu erreichen (Lesser 1999, S. 133). Mathematisch betrachtet besteht ein MAS aus folgenden Elementen (Ferber 1999, S. 11): (1) eine Umgebung  $E$ , (2) eine Menge von Objekten  $O$ , die in der Umgebung  $E$  situiert und passiv sind, (3) eine Menge von Agenten  $A$ , die sich von Objekten dadurch unterscheiden, dass sie aktiv sind, (4) eine Menge von Beziehungen  $R$  zwischen Objekten, Agenten und zwischen einander, (5) eine Menge von Operationen  $Op$ , die es Agenten ermöglichen Objekte wahrzunehmen und zu manipulieren und (6) Operatoren, welche die Reaktionen der Umgebung auf Manipulationsversuche durch Agenten abbilden.

Drogoul et al. stellen so ein System grafisch und weniger formal dar, wie in Abbildung 4-6 zu sehen ist.

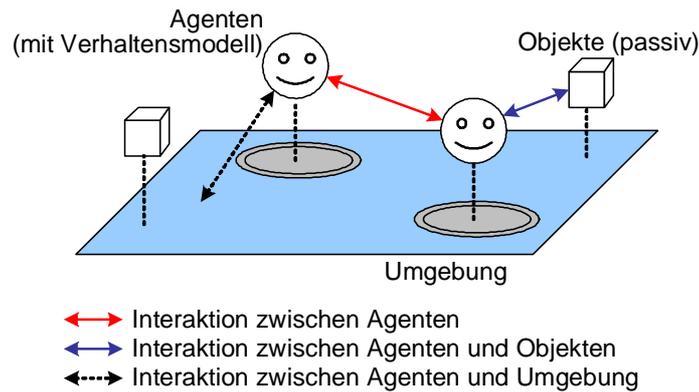


Abbildung 4-6: Grafische Darstellung eines MAS nach (Drogoul et al. 2002, S. 7)

Von einem anderen Blickwinkel aus lassen sich spezielle Eigenschaften identifizieren, die ein MAS auszeichnen (Klügl 2001, S. 17). Jeder Agent hat eine beschränkte Sicht auf das Gesamtsystem, die aus unvollständiger Information und beschränkter Problemlösefähigkeit resultiert. Jeder Agent verwaltet seine Daten lokal und ist gleichzeitig aktiv, so dass sich im Idealfall eine nebenläufige Ausführung ergibt. Weiterhin gibt es idealerweise keine zentrale Kontrollinstanz, um die geforderte Kontrollautonomie von Agenten nicht zu gefährden.

Die Umgebung  $E$  in einem MAS lässt sich anhand von mehreren Eigenschaften klassifizieren (Russell, Norvig 2003, S. 41f). Zunächst kann sie *ganz* oder nur *teilweise beobachtbar* sein. Während es für Anwendungen des verteilten Problemlösens vorteilhaft sein kann, wenn die gesamte Umgebung jederzeit beobachtbar ist, ist es für Multiagentensimulationen i. d. R. zweckmäßig, dass Agenten nur die unmittelbare Umgebung, nicht aber die Umgebung des ganzen Systems zu jeder Zeit beobachten können. Umgebungen können *deterministisch* sein, so dass die Umweltzustände nur von den vergangenen Zuständen und den Aktionen des Agenten abhängt oder *stochastisch*, wenn der Agent zukünftige Zustände nicht mit Sicherheit vorhersehen kann. Dabei ist zu beachten, dass Russel und Norvig diese Eigenschaften aus Sicht eines einzelnen Agenten betrachten. Daraus folgt, dass in MAS die Umgebung immer stochastisch ist, wenn ein Agent die Aktionen anderer Agenten nicht mit Sicherheit vorhersehen kann. Weiterhin unterscheidet man *episodische* und *sequentielle* Umgebungen. Episoden sind abgeschlossene Einheiten, die nicht von den Aktionen in früheren Episoden abhängen. Russel und Norvig führen hier Klassifikationsaufgaben als ein typisches Beispiel an. Dementsprechend sind für Multiagentensimulationen sequentielle Umgebungen bedeutsamer, in denen Aktionen jederzeit alle zukünftigen Aktionen beeinflussen können. Die nächste Unterscheidung ist zwischen *statischen* und *dynamischen* Umgebungen. In statischen Umgebungen bleiben

die Umweltzustände während der Entscheidungsphase des Agenten gleich. Es spielt also zumindest auf die Umgebung bezogen keine Rolle, wie viel Zeit der Agent zum Entscheiden benötigt. In dynamischen Umgebungen hingegen kann sich die Umgebung permanent ändern, so dass Zeit für die Entscheidung eine wichtige Rolle spielt. Russel und Norvig sprechen von semidynamischen Umgebungen, wenn sich die Umweltzustände zwar nicht ändern, die Leistung eines Agenten aber trotzdem von der Zeit abhängt (z.B. beim Schachspielen mit Zeitbegrenzung). Weiterhin können Umgebungen mit *diskreten* oder *kontinuierlichen* Zustandsräumen unterschieden werden. Dies bezieht sich sowohl auf die Fortschreibung der Zeit (bei rundenbasierten Simulationen z.B. diskret) als auch auf die Umweltzustände im engeren Sinne (z.B. Entfernungen, Fertigstellungsgrade, etc.). Schließlich werden Umgebungen mit nur *einem Agenten* oder mit *mehreren Agenten* unterschieden, wobei sich die Frage stellt, wann andere Objekte der Umgebung als Agent betrachtet werden müssen. Russel und Norvig beantworten diese Frage nur teilweise, indem sie darauf hinweisen, dass Objekte, die Ziele verfolgen, die in Konkurrenz zu den Zielen des Agenten stehen, in jedem Fall als Agenten betrachtet werden müssen (z.B. Gegner beim Schach). Ob andere Verkehrsteilnehmer in einer Verkehrssimulation aus Sicht eines Taxifahreragenten als stochastische Objekte oder Agenten gelten müssen, lassen sie hingegen offen. Aus Sicht des Autors sollten aber zumindest alle Objekte, die sich ähnlich wie der Agent verhalten, als Agent betrachtet werden. Dadurch würden alle Verkehrsteilnehmer zu Agenten, da aus einer Außenperspektive nicht plausibel ist, wieso ein Verkehrsteilnehmer anders zu behandeln ist als ein anderer.

Wie schon zu Beginn angesprochen, können generell für MAS die Anwendungsgebiete verteiltes Problemlösen und Multiagentensimulation unterschieden werden. Ferber differenziert noch feingranularer (Ferber 1999, S. 31ff; siehe auch Lesser 1999, S. 134), was aber für die vorliegende Arbeit nicht relevant ist. Innerhalb des Anwendungsgebiets verteiltes Problemlösen identifiziert Luger in den Bereichen Fertigung, Prozesssteuerung, Telekommunikation, Verkehrssysteme, Informationsmanagement und E-Commerce inhärent verteilte Probleme, die sich gut für die Lösung durch Multiagentensysteme eignen und für die auch schon Forschungsarbeiten zum Einsatz von MAS existieren (Luger 2001, S. 268f). Dabei sind vor allem die richtige Agentenarchitektur (vgl. Sloman, Scheutz 2002, S. 1) und effiziente Methoden zur Koordination der einzelnen Agenten relevante Forschungsthemen (Lesser 1999, S. 134). In jüngerer Zeit werden auch Herausforderungen thematisiert, die mit der Nutzung offener MAS-Plattformen einhergehen, angefangen von

Fragestellungen bei der Migration von Agenten (Kendall et al. 2000) über die Einigung auf eine gemeinsame Sprache (z.B. Zelewski, Siedentopf 1999; Steels 2005) bis hin zu den ökonomischen und sicherheitstechnischen Fragestellungen bei der Nutzung von Rechenzeit durch Agenten aus nicht vertrauenswürdigen Quellen (Schoder, Eymann 2000; Sycara et al. 2003).

Zusätzlich werden mit zunehmender Reife und Komplexität der Agentensysteme auch Fragestellungen wie Unterstützung des Softwareentwurfs und Verifikation der MAS immer wichtiger, was in dieser Arbeit in einem eigenen Abschnitt behandelt wird (siehe Abschnitt 5.1).

#### **4.2.4 Kommunikation**

Agentenkommunikation ist die Voraussetzung für Zusammenarbeit. Die Basis für existierende Agentenkommunikationssprachen (ACL, Agent Communication Language) bildet die Sprechakttheorie, die von Austin entwickelt und von Searle verfeinert wurde (siehe Abschnitt 3.5.1). Sie wurde 1986 von Winograd und Flores auf Computersysteme übertragen (Winograd, Flores 2000). Sprechakte eines bestimmten Typs werden in der Agentenkommunikation Performative genannt (Cohen, Levesque 1990), obwohl das vor dem Hintergrund der Sprechakttheorie streng genommen nicht vollständig korrekt ist (Labrou 2001, S. 5). Das Äußern von Sprechakten kann auch als Aktion betrachtet werden (Labrou, Finin 1996, S. 5). Die illokutionären Teile von Sprechakten bestehen wiederum aus mehreren Bestandteilen, der illokutionären Kraft, dem illokutionären Punkt und dem propositionellen Inhalt (Flores-Méndez 2002, S. 14). Der illokutionäre Punkt lässt sich in die Klassen Assertiva (jemanden über etwas informieren), Direktiva (jemanden um etwas bitten oder es befehlen), Kommissiva oder Promissiva (eine Verpflichtung, etwas zu tun), Deklarativa (eine Erklärung, mit der ein neuer Fakt geschaffen wird) und Expressiva (einen inneren Zustand zum Ausdruck bringen) unterteilen (Flores-Méndez 2002, S. 14f; Ferber 1999, S. 320). Diese Klassen sind ähnlich der Klassifizierung von Searle, aber z.B. mit den Assertives speziell auf das Agentenumfeld angepasst.

Die derzeit vorherrschenden ACLs sind die vom Agentenstandardisierungsgremium FIPA (Foundation for Intelligent Physical Agents) spezifizierte FIPA ACL (FIPA 2002) einerseits und der vom Knowledge Sharing Effort (KSE) entwickelte Knowledge Query and Manipulation Language (KQML Finin et al. 1994; Labrou, Finin 1996; Labrou, Finin 1997) andererseits. Letztere wurde ursprünglich für die Interoperabilität von Wissensbasen entworfen. Andere bekannte Ansätze sind die Joint Intention Theory (Cohen, Levesque

1990), Singhs soziale Semantik (Singh 1998) und Columbettis Albatross, wobei die beiden letzten Semantik über soziale Verpflichtungen definieren, während die Joint Intention Theory genau wie FIPA ACL und KQML die Semantik über mentale Attribute wie Überzeugungen, Wünsche und Absichten definieren (Flores-Méndez 2002, S. 15f). Die beiden konkurrierenden Standards sind im Laufe der Zeit immer ähnlicher geworden (vgl. Labrou 2001, S. 10). Im Detail bestehen jedoch einige Unterschiede die hier nicht näher betrachtet werden (genauere Informationen dazu finden sich bei Labrou 2001, S. 11ff).

Konzeptionell gesehen besteht eine Nachricht in einer ACL aus der Nachrichtenebene, der Kommunikationsebene und der Inhaltsebene (Labrou 2001, S. 5). Die Nachrichtenebene wird durch die ACL spezifiziert, indem festgelegt wird, welche Performatives, also welche Nachrichtentypen oder sprachliche Primitive, möglich sind. Die Kommunikationsebene ist für den Transport der Nachricht zuständig. Hier muss über identifizierende Attribute für Sender, Empfänger und Nachricht sichergestellt werden, dass die Nachricht den Empfänger auch erreicht und er darauf reagieren kann. Als Transportprotokolle kommen alle Alternativen für den Nachrichtenaustausch in verteilten Systemen in Frage, angefangen von TCP/IP-Sockets basierten Verfahren bis hin zu Standards wie Java RMI, DCOM, CORBA oder SOAP über HTTP (siehe z.B. Alonso et al. 2004). Sollen Agenten nur innerhalb eines lokalen Systems Nachrichten austauschen, so können auch direkte Funktionsaufrufe zur Kommunikation genutzt werden, die jedoch über einen expliziten Kommunikationskanal mediiert sein sollten. Die Inhaltsebene schließlich muss dafür sorgen, dass die Inhalte in einer einheitlichen Sprache repräsentiert werden (z.B. KIF, Knowledge Interchange Format) und es ein gemeinsames Verständnis der referenzierten Konzepte gibt, was i. d. R. über Ontologien sichergestellt wird (siehe z.B. Fensel 2004).

Der Aufbau einer Nachricht in FIPA ACL und KQML stellt sich folgendermaßen dar Es wird zuerst der Sprechakt definiert, dann folgen Sender und Empfänger, der eigentliche Inhalt und Metadaten über den Inhalt, die die verwendete Sprache und Ontologie angeben.

Im dargestellten Beispiel wird nach Informationen über den Prozessor (CPU) eines bestimmten Laptop-Modells (libretto50) gefragt (Performative `ask-one`). Die Antwort wird mit dem Performative `tell` gegeben. Die Inhaltssprache ist KIF. Weitere Performatives in KQML sind z.B. `advertise` (Anbieten von Diensten), `ask-if` (Bestätigung von Informationen), `broker-one` (Vermitteln von Agenten mit bestimmten Fähigkeiten oder Informationen) und `sorry` (eine Frage kann nicht beantwortet werden) (Labrou, Finin 1996).

```
(ask-one
  :sender labrou
  :receiver laptop-center
  :content (CPU libretto50 ?processor)
  :ontology electronics
  :language kif)
```

```
(tell
  :sender laptop-center
  :receiver labrou
  :content (CPU libretto50 pentium)
  :ontology electronics
  :language kif)
```

*Beispiel 1: Sprechakte in KQML (Labrou 2001, S. 6)*

Die gelisteten Performatives lassen schon erkennen, dass KQML (genau wie FIPA ACL) für die Kommunikation beim verteilten Problemlösen entwickelt wurden. In Multiagentensimulationen werden zwar ähnliche Sprechakte benötigt wenn Menschen simuliert werden, da die Sprachen ja an die natürliche Sprache angelehnt sind, die Interpretation ist aber wesentlich weniger formal. Deswegen wurde dem Autor in Gesprächen mit Agentenforschern<sup>31</sup> auf Konferenzen empfohlen, eine eigene ACL in Anlehnung an die vorgestellten Standards zu definieren, die jedoch keinen Anspruch auf Allgemeingültigkeit oder gute Wiederverwendbarkeit in anderen MABS-Systemen erheben soll.

Aufbauend auf diesen elementaren Sprechakten können Regeln für Konversationen modelliert werden, in denen festgelegt wird, welche Sprechakte mit welchen anderen beantwortet werden können oder müssen. Solche Kommunikationsprotokolle können z.B. als endliche Automaten oder mit Petrinetzen abgebildet werden (Ferber 1999, S. 325ff). Im Rahmen der Agentenkommunikation sind v.a. Kommunikationsprotokolle für die Zusammenarbeit relevant, wie es z.B. das verbreitete Contract Net Protokoll darstellt (siehe Abschnitt 4.2.5)

---

<sup>31</sup> u.a. Franziska Klügl und Torsten Eymann

### 4.2.5 Interaktion, Koordination und Kooperation

Interaktion ist ein zentraler Bestandteil von MAS und tritt auf, wenn zwei oder mehr Agenten durch eine Menge von reziproken Aktionen in eine dynamische Beziehung zueinander treten (Ferber 1999, S. 59). Damit dies möglich wird müssen eine Reihe von Voraussetzungen erfüllt sein: (1) die anwesenden Agenten müssen zu Aktionen und/oder Kommunikation fähig sein, (2) es muss eine Situation existieren, die als Treffpunkt für die Agenten dienen kann, (3) es muss dynamische Elemente geben, die temporäre Beziehungen ermöglichen und (4) es muss eine gewisse Unstetigkeit in den Beziehungen geben, so dass die Autonomie der Agenten gewahrt bleibt (Ferber 1999, S. 60). Dementsprechend definiert Ferber eine Interaktionssituation als eine Zusammenstellung von Verhalten, das aus der Gruppierung von Agenten resultiert, die agieren müssen um ihre Ziele zu erreichen. Die Aufmerksamkeit liegt dabei besonders auf begrenzt vorhandenen Ressourcen und den individuellen Fähigkeiten der Agenten (Ferber 1999, S. 61). Aus dieser Definition werden zwei problematische Fälle ersichtlich: wenn die interagierenden Agenten konfliktäre Ziele haben und wenn sie um begrenzte Ressourcen konkurrieren. Sind die Ziele dagegen einander förderlich und die Ressourcen in ausreichender Menge vorhanden oder die interagierenden Agenten benötigen unterschiedliche Arten von Ressourcen, so ist die Interaktion unproblematisch. Die dritte Komponente von Interesse für die Interaktion ist, ob ein Agent seine Ziele alleine erreichen kann oder ob die Interaktion und nachfolgende Kooperation für das Erreichen der Ziele eines Agenten erforderlich sind, weil der Agent alleine nicht über ausreichende Fähigkeiten oder Arbeitskapazität verfügt (Ferber 1999, S. 65). Ferber leitet daraus acht verschiedene Interaktionssituationen ab, die er in die Kategorien Indifferenz, Kooperation und Antagonismus einteilt.

Für Mitarbeiter einer Organisation wird oft Benevolenz angenommen, was nachvollziehbar ist, da alle Mitarbeiter den Firmenzielen verpflichtet sind. Andererseits verfolgen Mitarbeiter aber auch individuelle Ziele, die konfliktär sein können (vgl. z.B. Principal-Agent-Theorie und hidden action), und es kann auch vorkommen, dass Mitarbeiter unterschiedlicher Abteilungen durch Management-Vorgaben konfliktäre Ziele haben, z.B. wenn die Vertriebsmitarbeiter am Umsatz gemessen werden und die Softwareentwickler am Deckungsbeitrag, so dass der Vertrieb für aufwändige Großprojekte unbedingt den Zuschlag bekommen will, auch wenn mit dem zugesagten Budget die Umsetzung kaum möglich ist und dadurch niedrige Deckungsbeiträge entstehen.

Bestehen Konflikte, so können sie in MAS durch systemweite Regelungen oder Verhandlung zwischen den betroffenen Agenten aufgelöst werden (Ferber 1999, S. 81f)

Für den Fall der Kooperation ist zu beachten, dass man nicht nur dann von Kooperation spricht, wenn Agenten absichtsvoll ein gemeinsames Ziel identifizieren und dessen Erreichung gemeinsam verfolgen, sondern auch, wenn der Betrachter den Eindruck erhält, dass die Agenten kooperieren, unabhängig davon, in welchem mentalen Zustand die Agenten sich befinden (Ferber 1999, S. 70f). Dies wird am Beispiel der Ameisen deutlich, denen wir kooperatives Verhalten unterstellen ohne zu fragen, ob ihr Nervensystem überhaupt in der Lage ist das Ziel der Kooperation zu verfolgen.

Eine Form der Kooperation, die in Organisationen besonders wichtig erscheint ist die Arbeitsteilung bei Spezialisierung. In MAS stellt sich dabei die Frage, wie die Arbeit auf die verfügbaren Agenten verteilt wird. Prinzipiell kann zwischen zentraler und dezentraler Verteilung unterschieden werden (Ferber 1999, S. 343f). Im zentralen Fall kann die Arbeit entweder entlang einer bestehenden Hierarchie oder durch spezialisierte Agenten (Trader oder Broker) verteilt werden. Im dezentralen Fall sind die Alternativen ein Acquaintance Network und das Contract Net Protokoll. Bei Acquaintance Networks brauchen Agenten Informationen über die Fähigkeiten anderer Agenten (aller oder zumindest einer genügend großen Anzahl), so dass sie gezielt um Hilfe bitten können, während im Contract Net Protokoll nach Angeboten für das Erledigen einer Teilaufgabe gefragt wird und dann das beste Angebot angenommen wird.

In Organisationen wird für Arbeitsaufträge i. d. R. eine zentrale Verteilung über Abteilungsleiter oder Projektmanager etabliert sein, während der dezentrale Fall bei der Hilfe unter Kollegen zur Anwendung kommt. Hier kann jedoch die Voraussetzung des Wissens über die Fähigkeiten anderer Agenten nicht als selbstverständlich gegeben angesehen werden.

Clancey et al. differenzieren weiterhin zwischen Koordination, Kooperation und Zusammenarbeit (Clancey et al. 2004, S. 26f). Koordination liegt dann vor, wenn Agenten den Zugriff auf gemeinsam genutzte Ressourcen abstimmen oder sicherstellen, dass ihre Aktionen keinen negativen Einfluss aufeinander ausüben. Kooperation liegt dann vor, wenn eine Ressource gemeinsam so benutzt wird, dass die Arbeitsinhalte aufeinander abgestimmt werden müssen. Zusammenarbeit liegt dann vor, wenn die gemeinsame Arbeit auch ein gemeinsames Ziel hat, zum Beispiel ein gemeinsames Projekt. In Organisationen

treten alle drei Formen auf, so dass sie auch in der vorliegenden Arbeit berücksichtigt werden.

### 4.3 Agentenarchitekturen für die Simulation von Menschen

Die Agentenarchitektur bildet das zentrale Glied, das Entscheidungsmöglichkeiten, Wissensbasis und Aktionen der Agenten definiert. Sie lassen sich nach verschiedenen Gesichtspunkten klassifizieren. Klügl zieht Wissensrepräsentation und Entscheidungsmöglichkeiten dafür heran und kommt zu den in Abbildung 4-7 dargestellten vier Klassen (Klügl 2001, S. 22ff).

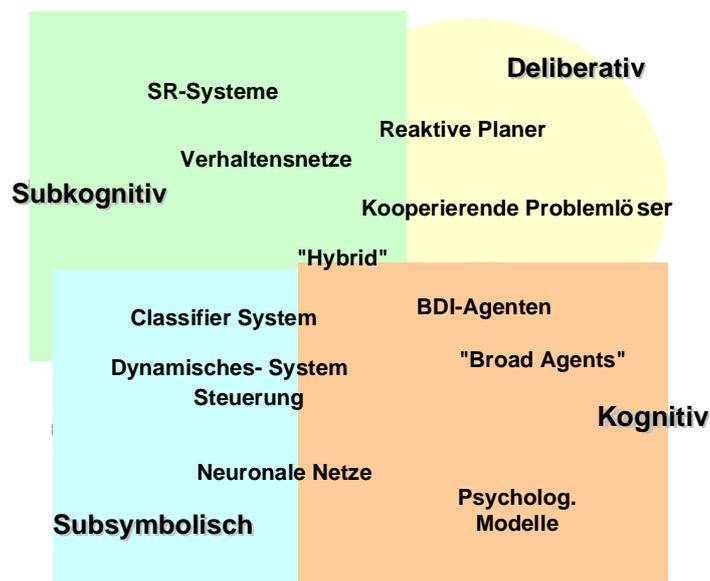


Abbildung 4-7: Klassifikation von Agentenarchitekturen nach (Klügl 2001, S. 22)

Subsymbolische Architekturen wählen die nächste Aktion des Agenten über Mechanismen aus, die keine symbolische Repräsentation des Problems benötigen, sondern direkt mit messbaren Aspekten des realen Systems arbeiten können. Subkognitive Architekturen werden auch als reaktiv bezeichnet und benutzen Reiz-Reaktionsregeln (Stimulus-Response, SR) oder Verhaltensnetze zur Auswahl der Aktion. Die oft zitierte Subsumptionsarchitektur von Brooks (z.B. Brooks 1991) ist ein Vertreter dieser Klasse. Deliberative Agenten benutzen ein internes Weltmodell und eine explizite Repräsentation der Ziele, um eine Abfolge von Aktionen zu planen. Klügl unterscheidet in dieser Klasse weiterhin klassische Planer, die einen Plan aufstellen und anschließend ohne Rücksicht auf Umweltänderungen abarbeiten, reaktive Planer, die flexibel auf Änderungen der Umwelt reagieren können, und hybride Architekturen, die reaktive und deliberative Komponenten

enthalten (Klügl 2001, S. 26). Kognitive Architekturen beruhen auf kognitionswissenschaftlichen oder psychologischen Modellen der menschlichen Informationsverarbeitung.

Für die Modellierung von Wissensweitergabe erscheinen deliberative und kognitive Architekturen prinzipiell geeignet, während subsymbolische und subkognitive Architekturen über keine ausreichende Wissensrepräsentationsmöglichkeiten dafür verfügen.

Die bekannteste Klassifikation unterscheidet in reaktive und deliberative bzw. intentionale Agenten. Verhagen findet in der Literatur zusätzlich noch die Klasse der glaubwürdigen Agenten und die der sozialen Agenten (Verhagen 2000, S. 51). Soziale Agenten haben über die Funktionen der deliberativen Agenten hinaus ein explizites Modell anderer Agenten, so dass soziales Verhalten entstehen kann. Glaubwürdige Agenten stellen eher den Gegenansatz zu kognitiven Architekturen dar. Während letztere postulieren, dass menschenähnliche Agenten nur über das möglichst direkte Umsetzen der vermuteten Gehirnstruktur in die interne Struktur des Agenten zu erreichen sind, geht es bei glaubwürdigen Agenten ausschließlich um das beobachtbare Verhalten. Die interne Struktur kann dabei ganz anders aussehen und wird meist auf Einfachheit und Effizienz hin optimiert (siehe z.B. Bates 1994; Reilly 1996; Loyall 1997).

Eine andere Klassifikation von Agenten stammt von Verhagen. Er unterscheidet vier Typen von Agenten anhand des Grads der Autonomie (Verhagen 2000, S. 45f) und überarbeitet dabei eine Klassifikation von Conte, Castelfranchi und Dignum (Conte et al. 1999). Reaktive Agenten haben keinerlei Autonomie, sie reagieren deterministisch auf Wahrnehmungen aus der Umgebung. Plan-autonome Agenten haben zwar fest vorgegebene Ziele, können aber frei entscheiden, wie sie diese erreichen. Ziel-autonome Agenten können darüber hinaus auch selbst entscheiden, welches ihrer Ziele sie wann verfolgen. Die größte Unabhängigkeit haben laut Verhagen Norm-autonome Agenten, die selbst Ziele generieren können und deren Legitimität anhand von Normen evaluieren, die vorgegeben sein aber auch innerhalb der Agentengesellschaft verhandelt werden können. Die Autonomie schlägt sich auch in der Architektur nieder, wie der folgende Abschnitt zeigt.

#### **4.3.1 Generische Architekturen**

Grundlegende Architekturen und Klassifikationen zu Agenten finden sich in (Russell, Norvig 2003, S. 44ff). Sie beginnen dabei mit einfachen Architekturen und führen nach und nach Komplexität ein, um deliberatives Verhalten zu ermöglichen. Diese sind in den folgenden Abbildungen dargestellt, wobei die neu eingeführten Komponenten in grau abgebildet werden.

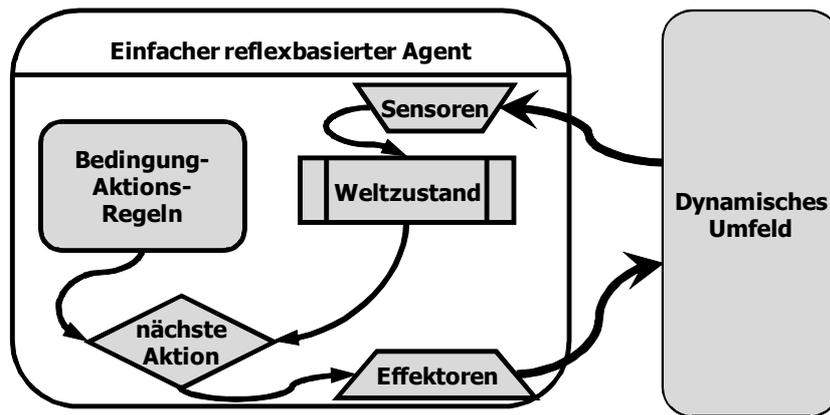


Abbildung 4-8: Reflexbasierter Agent (nach Russel, Norvig 2003, S. 47)

Den einfachsten Fall stellt ein einfacher reflexbasierter Agent dar, der kein internes Weltmodell und auch sonst keine internen Zustände besitzt. Er nimmt permanent Daten über sein Umfeld über die Sensoren auf und fügt diese zu einem temporären Weltzustand zusammen. Aus diesem und internen Regeln leitet er die nächste Aktion ab, die dann über die Effektoren umgesetzt wird.

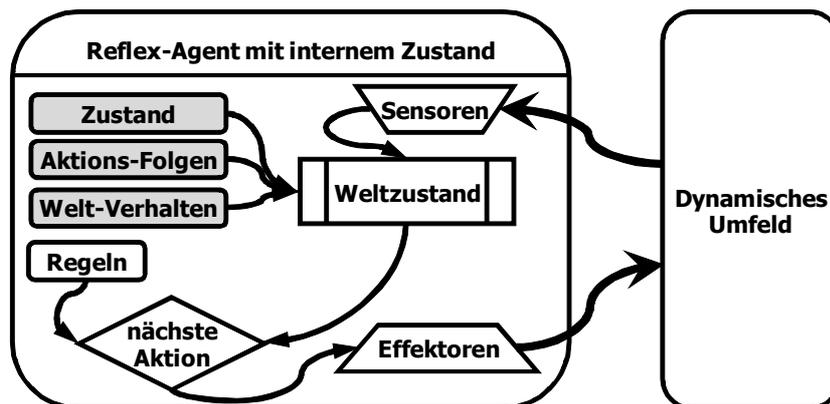


Abbildung 4-9: Modellbasierter Agent (nach Russel, Norvig 2003, S. 49)

Modellbasierte Agenten erweitern diese einfache reflexbasierte Architektur um interne Zustände. Diese Zustände können verwendet werden, um über Veränderungen des Umfelds und Auswirkungen der eigenen Aktionen zu reflektieren. So kann auch bei nur teilweise beobachtbarem Umfeld ein umfassenderes Weltmodell aufgebaut werden und Anpassung des Verhaltens in Abhängigkeit von Erfolg und Misserfolg der eigenen Aktionen erfolgen.

Eine zusätzliche Erweiterung stellt das Einführen einer expliziten Zielrepräsentation dar. Dadurch kann der Erfolg des Verhaltens in der Vergangenheit anhand des Zielerreichungsgrads gemessen werden und zukünftige Aktionen durch Antizipation des damit verbundenen Zielerreichungsgrads ausgewählt werden.

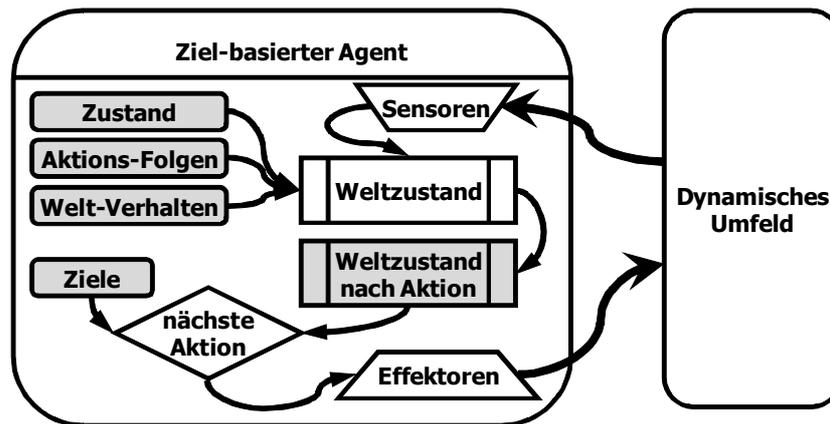


Abbildung 4-10: Modell-basierter Agent mit Zielen (nach Russel, Norvig 2003, S. 50)

Eine letzte Erweiterung des Modells stellt die Einführung einer Nutzenfunktion dar. Dadurch können mehrere Aktionen, die alle zielführend sind mit gewichteten Nutzenbeiträgen bewertet und damit die Aktionsauswahl weiter verbessert werden. Weiterhin können konfliktäre Ziele gegeneinander abgewogen werden und das als nächstes zu verfolgende Ziel in Abhängigkeit der Wahrscheinlichkeit seiner Erreichung und seiner Wichtigkeit ausgewählt werden.

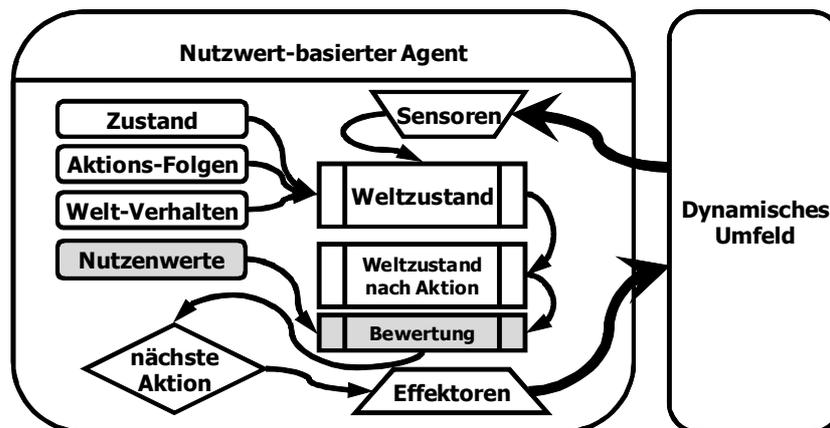


Abbildung 4-11: Nutzen-basierter Agent (nach Russel, Norvig 2003, S. 52)

Als weitere Eigenschaft von Agenten diskutieren Russel und Norvig das Lernen. Es kann prinzipiell in allen Agentenarchitekturen mit internem Zustand eingeführt werden, um weitere Verbesserungen zu erreichen, kann aber auch als fundamentale Architektur verstanden werden, wenn Agenten mit keinem oder nur geringem Vorwissen ausgestattet werden sollen und die Auswirkungen von Aktionen und andere relevante Konzepte von Grund auf lernen müssen (Russell, Norvig 2003, S. 51ff).

### 4.3.2 Anforderungen an Agentenarchitekturen

Bevor weitere in der Praxis verbreitete Architekturen vorgestellt werden, sollen Kriterien zu deren Vergleich und Bewertung erarbeitet werden. Angesichts der Vielzahl an Architekturen die seit Beginn der 90er Jahre vorgeschlagen wurden (vgl. Wooldridge, Jennings 1994 für einen Überblick) und deren Reihen auch im 21. Jahrhundert immer noch Zuwachs finden (z.B. Bach 2003), ist es überraschend, dass es keine aktuellen Vergleiche zu geben scheint. Trotz der dementsprechend großen Auswahl an Architekturen scheint sich keine Konsolidierung abzuzeichnen. Für die vorliegende Arbeit soll keine neue Architektur entwickelt werden sondern aus den vorhandenen eine geeignete ausgewählt werden bzw. wünschenswerte Elemente bestehender Architekturen geeignet kombiniert werden. Sloman und Scheutz haben begonnen, eine Reihe von Kriterien für den Vergleich von Agentenarchitekturen zu erarbeiten (Sloman, Scheutz 2002), die auf den Vorarbeiten der beiden Autoren zu realistischem Verhalten von simulierten Menschen (Scheutz et al. 2000) und Agentenarchitekturen (Sloman 2000) basieren. Die dort vorgestellten Vergleichskriterien sind an das von Sloman entwickelte Framework CogAff angelehnt, welches wiederum stark auf affektive mentale Konzepte abstellt. Deshalb sollen noch weitere Kriterien hinzugezogen werden, um die Diskussion neutraler zu gestalten. Carley et al. beschäftigen sich schon seit Beginn der 90er Jahre mit Anforderungen an Agentensysteme. Einen ersten Anforderungskatalog stellen sie 1992 vor (Carley et al. 1992, S. 91). Diese Anforderungen werden später im Rahmen der ACTS Theorie<sup>32</sup> von Carley und Prietula weiter ausgebaut und in Form von neun Axiomen formuliert (Carley, Prietula 1994, S. 5). Diese gehen über die Anforderungen an einzelne Agenten hinaus, berücksichtigt aber nur die Auswirkungen auf die Agentenarchitekturen. In Tabelle 4-1 findet sich eine Zusammenfassung der Anforderungen aller drei Quellen. Anhand dieser Kriterien sollen einige bestehende Architekturen verglichen werden. Die vorgestellten Architekturen stellen keineswegs eine vollständige Liste dar. Die Auswahl erfolgte zum einen über die Dokumentation in der Literatur durch eigene Arbeiten und die Verbreitung der Architekturen, die über die Zitation in anderen Publikationen ermittelt wurde. Deswegen wurden BDI, Soar, ACT-R und InteRRaP untersucht.

---

<sup>32</sup> Agents who are Cognitively-restricted, Task-oriented and Socially-situated

Tabelle 4-1: Anforderungen an soziale Agenten

<b>Bereich</b>	<b>Anforderungen</b>
Wahrnehmung und Aktionen	Nimmt die Umwelt durch eine oder mehrere Sensoren wahr Manipuliert Objekte der Umwelt mit Aktoren Kann sich selbst an unterschiedliche Orte bewegen
Wissensbasis	Erinnert Orte, Personen, Aufgaben Verwendet ein physisches Symbolsystem mit logischer, prozeduraler, oder neuronaler Repräsentation Hat Lernmechanismen, z.B. Neuronale Netze, erweiterbare Wissensbasen, Generieren neuer Regeln
Innere Architektur	Hat mehrere Ebenen die sequentiell durchlaufen werden, oder gleichzeitig aktiv sind Die Zuständigkeit von Ebenen wird durch eine Hierarchie bzw. Priorisierung oder durch funktionale Eignung festgelegt Übergeordnete Ebenen kontrollieren untergeordnete Ebenen direkt oder können diese durch Training beeinflussen Emotionen werden in eigenem „Kontrollzentrum“ modelliert, oder entstehen emergent aus anderen Eigenschaften
Aufgaben	Kommt mit unvollständiger Aufgabenstellung zurecht Zerlegt Aufgaben selbständig Koordiniert seine eignen Teilaufgaben Hat explizite Zielrepräsentation Kommt mit mehrere gleichzeitigen auch konfliktären Zielen zurecht
Kommunikation	Kann Fragen stellen und Anforderungen geben Kann Befehle erteilen und entgegen nehmen Kann zu einzelnen Agenten und zu Gruppen sprechen
Soziales Modell	Hat Modelle anderer Agenten Hat ein Modell der Organisation Besetzt eine formelle oder informelle Position in der Organisation und kann damit eine oder mehrere sozial-situierte Rollen einnehmen

Die Subsumptionsarchitektur von Brooks (z.B. Brooks 1991) wurde trotz weiter Verbreitung nicht berücksichtigt, da sie kein deliberatives Verhalten modelliert, was für die Wissensweitergabe als essentiell angesehen wird wie oben diskutiert. Zum anderen wurden solche Architekturen ausgewählt die explizit für die Simulation von Menschen entworfen wurden, weshalb PECS, BRAHMS, CogAff und MicroPSI ausgewählt wurden, obwohl sie in der Literatur bisher nur geringere Beachtung finden.

### 4.3.3 BDI: Belief, Desire, Intention

Die BDI-Architektur (Rao, Georgeff 1991; Rao, Georgeff 1995) ist eine der bekanntesten Agentenarchitekturen. Sie unterteilt die Planungskomponente eines Agenten in Beliefs, Desires und Intensions, wobei Beliefs die Ansichten des Agenten über die Umwelt widerspiegeln (kognitive Komponente), Desires sind Wünsche des Agenten, die auch miteinander konkurrieren können (affektive Komponente), wogegen Intensions konkrete Absichten sind, die der Agent demnächst verwirklichen will und die deshalb konfliktfrei sein müssen (konative Komponente). Diese Einteilung geht auf die Arbeiten des Philosophen Bratman zurück (Bratman 1984; Bratman 1987; Bratman et al. 1988).

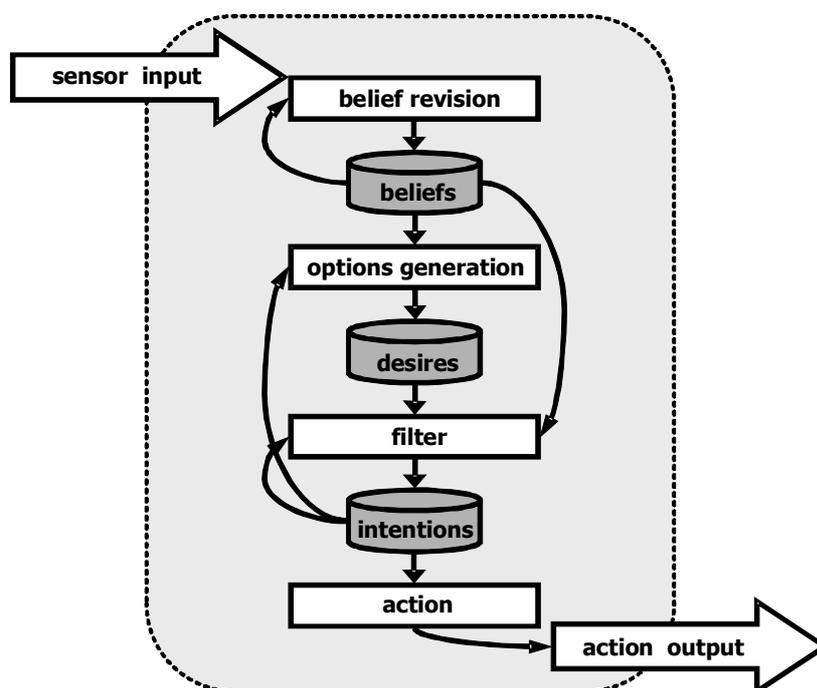


Abbildung 4-12: BDI Verarbeitungsmodell nach (Rao, Georgeff 1995, S. 7)

Die Vergleichskriterien greifen hier nur zum Teil, da es sich nicht um eine mehrschichtige Architektur handelt. Die Wissensrepräsentation erfolgt logisch in Form der Beliefs. Rao und Georgeff arbeiten mit einer an die Prädikatenlogik und Modallogik angelehnten BDI Logik (Rao, Georgeff 1995, S. 5), es sind aber auch andere Repräsentationen denkbar. Lernen im Sinne der KI ist nicht vorgesehen. Es gibt aber eine erweiterbare Wissensbasis. Die Auswahl von Motiven, in der BDI-Begriffswelt Desires genannt, erfolgt über die Option-Generation Komponente anhand der in der Umgebung vorgefundenen Situation. Für das Auflösen von Konflikten zwischen Desires ist der nachgeschaltete Filter zuständig (siehe Abbildung 4-12). Die von Rao und Georgeff vorgeschlagene BDI Logik ist aber nicht gut für diese Konfliktauflösung geeignet (Lang et al. 2001, S. 1). Im BDI Modell

wird die sog. Possible-World Semantik verwendet, in der Entscheidungsbäume aufgestellt werden, die zur Zielerreichung führen. Emotionen oder andere „höhere Eigenschaften“ sind nicht vorgesehen. Davon abgesehen sind aber spezialisierte Zentren eingeplant. Sprache ist nicht explizit vorgesehen und daher keine Kommunikation möglich. Folglich gibt es auch kein soziales Modell.

Die einflussreichste BDI-Architektur ist das Procedural Reasoning System (PRS), das im dMars System implementiert wurde und in vielen Anwendungen zum Einsatz kommt (Klügl 2001, S. 28)

#### **4.3.4 InteRRaP: Integration of Reactivity and Rational Planning**

InteRRaP (Fischer et al. 1994; Müller 1996; Müller, Pischel 1993) teilt die innere Struktur eines Agenten in drei Schichten: Behavior-based Layer, Local Planning Layer und Cooperative Planning Layer. Diese Schichten umfassen jeweils eine Planungskomponente und das zugehörige Wissen (siehe Abbildung 4-13). Auf unterster Ebene wird ein reflexbasierter Agent (auch reaktiver Agent genannt) realisiert. Sind die Fähigkeiten der untersten Schicht nicht ausreichend für die gegebene Situation, so wird die nächst höhere Schicht involviert. In der mittleren Schicht werden alle Pläne erarbeitet, die der Agent selber ausführen kann, und auf der höchsten Schicht werden Pläne gemacht, die andere Agenten zur Ausführung benötigen.

Die Ebenen werden nacheinander aktiviert, höhere Schichten werden nur bei Bedarf aktiv. Niedrigere Schichten kontrollieren die Aktivierung höherer Schichten. Sind diese aktiv, so wird die dort ausgewählte Aktion oder der Plan auch ausgeführt. Konflikte mit anderen Ebenen können nicht auftreten. Der Entscheidungsmechanismus ist auf jeder Ebene gleich. Unterschiede ergeben sich durch den Zugriff auf unterschiedliche Wissensbasen. Wissensrepräsentation, Lernen und Konfliktauflösung zwischen Wünschen sind analog dem BDI Modell, da dieses die Grundlage für InteRRaP bildet.

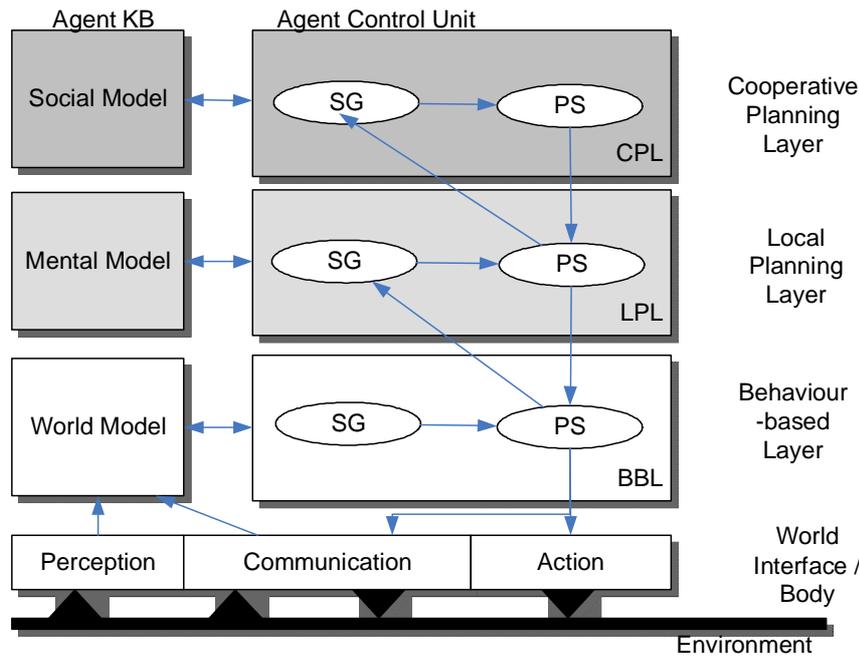


Abbildung 4-13: Agentenarchitektur InterRaP nach (Fischer et al. 1994, S. 3)

Wahrnehmungen gelangen durch das World Interface in die unterste Schicht und nehmen einen einzigen definierten Weg von der untersten Schicht nach oben (bottom-up Aktivierung), während Aktionssignale von der aktiven Schicht nach unten weitergeleitet werden (top-down Ausführung). Die SG Komponenten stehen für Situationserkennung und Zielaktivierung, PS-Komponenten für Planung und Scheduling. Alle Eigenschaften sind in spezialisierten Zentren verortet. Sprache ist vorgesehen, der Zusammenhang mit der internen Funktion ist jedoch nicht näher spezifiziert. Ein soziales Modell mit Repräsentation anderer Agenten ist explizit vorgesehen. Über eine Repräsentation von Rollen oder der Organisation ist nichts bekannt.

#### 4.3.5 PECS: Physis, Emotion, Cognition, Status

Das PECS-Modell (Urban 1997 (hier noch als PEC Modell); Urban 2000; Urban, Schmidt 2001) beschreibt eine Agentenarchitektur, die weitgehend auf die Zustandsvariablen eines Agenten abstellt. Es basiert auf systemtheoretischen Grundlagen und zielt auf die Simulation von Menschen. Das Modell sieht Variablen vor, die den physischen Zustand des Agenten beschreiben (Physis), solche, die die Emotionen des Agenten widerspiegeln (Emotion), den kognitiven Zustand abbilden (Cognition) und die soziale Stellung des Agenten repräsentieren (Social Status). Die konkrete Ausgestaltung der jeweiligen Komponenten bleibt jedoch dem Modellierer der Agenten überlassen. Hier gibt Urban nur Beispiele.

Das Modell sieht ein komplexes Zusammenspiel zwischen den einzelnen Komponenten vor, die untereinander wechselwirken, und auf die beiden Verhaltensebenen mit deliberativem und reaktivem Verhalten einwirken (siehe Abbildung 4-14). Die einzelnen Zustandskomponenten sind dabei gleichzeitig aktiv. Die Einflüsse der einzelnen Ebenen werden über Gewichte und/oder Regeln miteinander kombiniert. Bemerkenswert ist dabei, dass im Gegensatz zu allen anderen Architekturen die physischen Sensoren und die kognitive Wahrnehmungskomponente getrennt sind und eine Beeinflussung beider Komponenten durch die internen Zustände vorgesehen ist, so dass z.B. bei Übermüdung die Wahrnehmung getrübt werden könnte. Damit lassen sich z.B. Verkehrssimulationen durchführen, die Verkehrsunfälle in Abhängigkeit von verschiedenen Ursachen wie Müdigkeit, Alkoholeinfluss und Ähnlichem untersuchen.

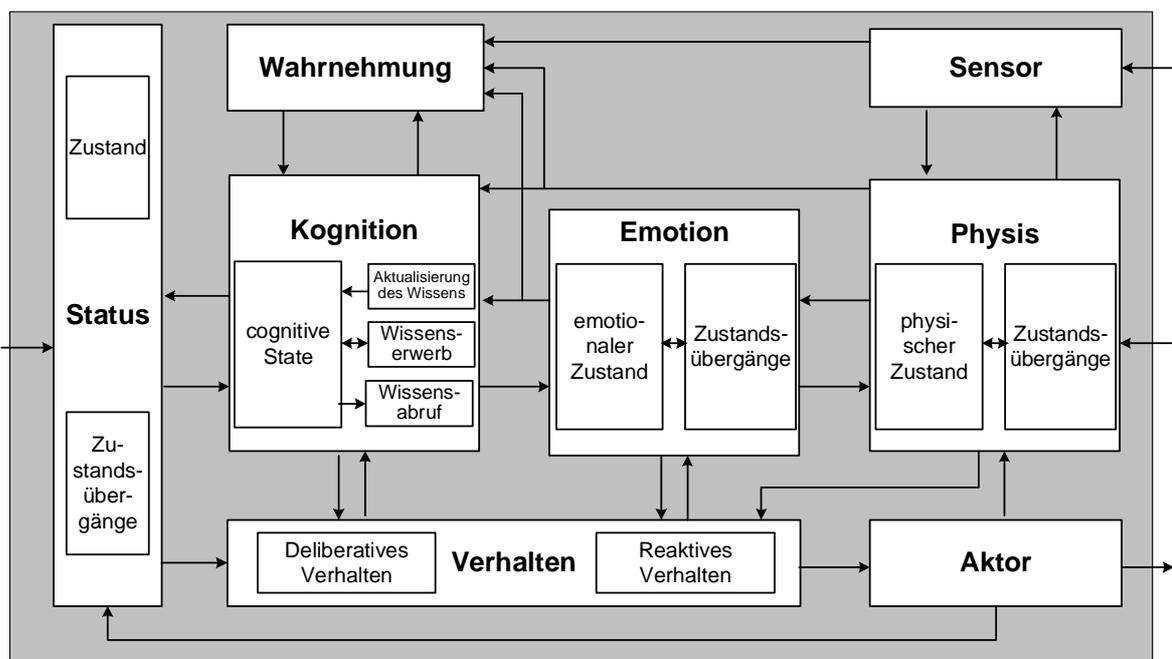


Abbildung 4-14: PECS Agenten-Architektur (Urban 2000, S. 93)

Als Entscheidungsmechanismen sieht Urban einfache Wenn-Dann-Regeln für das reaktive Verhalten und etwas komplexere Regeln nach dem SSA-Modell von Schmidt und Toussaint für das deliberative Verhalten vor (Urban 1997, S. 4). Die Wissensrepräsentation erfolgt durch Zustandsvariablen. Logikbasierte Wissensrepräsentation, wie sie in der KI bevorzugt wird, ist nicht vorgesehen. Lernen im KI-Sinne ist ebenfalls nicht vorgesehen, es gibt aber eine erweiterbare Wissensbasis und Reflexion auf einer Meta-Ebene ist angedacht. Das Lernen, welches in (Urban, Schmidt 2001, S. 4) beschrieben wird, ist nur eine der simulierten Aktivitäten, aber kein Bestandteil der Architektur. Motivauswahl und Konfliktauflösung erfolgt über eine Nutzenfunktion. Die Wahrnehmung erfolgt über einen

zentralen Eintrittspunkt, den Sensor. Von dort ist eine Verzweigung zur Komponente Physis und Wahrnehmung vorgesehen. Die Aktionssignale werden in der Verhaltensebene generiert und von dort an die Aktionskomponente weitergeleitet. Für jede Eigenschaft des Agenten gibt es ein spezialisiertes Zentrum. Kommunikation ist explizit vorgesehen. Die Nachrichten sind dabei ähnlich den oben beschriebenen KQML Nachrichten, obwohl Urban nicht explizit darauf referenziert. Es ist direkte Kommunikation über einen externen Connector möglich, sowie indirekte über einen Blackboard-Mechanismus. Ein soziales Modell mit expliziter Repräsentation der eigenen Rolle und anderer Agenten ist ebenfalls integriert.

### 4.3.6 ACT-R: Adaptive Control of Thought - Rational

Das von Anderson geprägte ACT-R Modell (Lebiere, Anderson 1993; Anderson et al. 1997; Anderson, Schunn 2000; Anderson et al. 2004) entstammt dem Gebiet der kognitiven Psychologie und orientiert sich dementsprechend an Theorien über die Funktion menschlichen Gedächtnisses. Es beruht auf einer Lerntheorie, die von deklarativem Wissen (Faktenwissen) und prozeduralem Wissen (Regelwissen) ausgeht. Das Modell ist eines der ältesten und geht auf die Arbeiten Andersons in den 70er und 80er Jahren zurück (Anderson 1977; Anderson 1987). Die Vorgänger des ACT-R Systems, das mittlerweile in Version 6.0 vorliegt (o.V. 2005) und von Anderson zusammen mit seinen Mitarbeitern (insbesondere Lebiere und Taatgen) kontinuierlich weiterentwickelt wurde, sind unter den Namen ACTE, ACT und PUBS in der Literatur dokumentiert<sup>33</sup>. Die vorliegende Literatur zu ACT-R bezieht sich aber noch auf die Version 5.0, so dass auch die folgenden Ausführungen diese Version fokussieren. Das System ist in LISP implementiert und bildet eine kognitive Architektur im Computer ab. Eine kognitive Architektur stellt eine grundlegende Spezifikation eines Informationsverarbeitungssystems dar. Dieses System enthält Mechanismen für intelligentes Verhalten. Vereinfacht ausgedrückt stellt sie eine Programmierumgebung dar, die es erlaubt, kognitive Vorgänge mit einem Computer zu simulieren. Dabei wird versucht, die Eigenarten der menschlichen Kognition zu berücksichtigen. Anderson trifft deshalb Annahmen über die Funktionsweise der menschlichen Informationsverarbeitung, die auf aktuellen Forschungsergebnissen der Neurobiologie und Kognitionspsychologie beruhen (Anderson et al. 2004, S. 1037f).

---

<sup>33</sup> Sämtliche Publikationen von Andersons Forschungsgruppe seit 1971 sind unter <http://act-r.psy.cmu.edu/> online abrufbar.

Deklaratives Wissen ist in ACT-R in Form von Chunks modelliert, prozedurales liegt in Form von Regeln, sog. Produktionsregeln, vor. Chunks haben einen definierten Typ und beliebig viele Slots. Obwohl LISP eine funktionale Sprache ist, haben Chunks große Ähnlichkeit mit Frames aus der Frame-Logik und Objekten aus der objektorientierten Programmierung.

ACT-R besitzt nicht direkt eine Ebenen-Architektur, sondern unterscheidet einzelne Module, die gleichzeitig aktiv sein können. Neben den in Abbildung 4-15 dargestellten Modulen sind noch weitere Module möglich. So werden z.B. ein Audio Modul als weitere Eingabe- und ein Sprachmodul als weitere Ausgabeschnittstelle explizit erwähnt (Anderson et al. 2004, S. 1040). Die Module sind funktionell spezialisiert. Es besteht keine direkte Kontrolle zwischen den Modulen. Die Produktionsregeln stellen das zentrale Element dar, welches alle anderen Module miteinander verbindet. Eine direkte Verbindung zwischen einzelnen Modulen wird jedoch diskutiert, da es Hinweise für solche Verbindungen im menschlichen Gehirn gibt (Anderson et al. 2004, S. 1041). Als Entscheidungsmechanismus fungieren die Produktionsregeln. Wissen wird deklarativ im entsprechenden Modul und prozedural in Form von Produktionsregeln gespeichert.

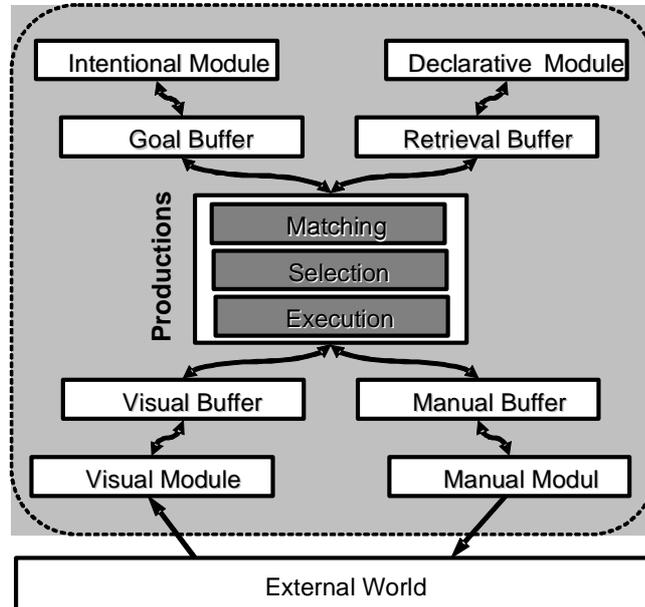


Abbildung 4-15: ACT-R Architektur nach (Anderson et al. 2004, S. 1037)

Beide Wissensbasen können erweitert werden. Es gibt aber auch Lernen im Sinne der KI. Der Erfolg von Aktionen wird überwacht und Aktionen werden neu gewichtet, so dass in der Vergangenheit erfolgreichere Aktionen mit höherer Wahrscheinlichkeit in Zukunft wieder ausgewählt werden. Motive werden im Goal Buffer abgebildet. Derzeit kann nur

ein Ziel gleichzeitig verfolgt werden, so dass es keine Konflikte geben kann. Wahrnehmungen können in mehreren Modulen in das System gelangen. Derzeit werden jedoch alle Wahrnehmungen zentral verarbeitet. Ein direktes Weiterleiten an andere Module ist derzeit wie schon erwähnt nicht vorgesehen. Sprache ist zwar explizit vorgesehen, die Zusammenhänge zwischen internen Funktionen und der Sprache sind allerdings unklar. Ein soziales Modell ist nicht implementiert.

#### 4.3.7 SOAR: States, Operators And Reasoning

Soar (Laird et al. 1987; Laird, Rosenbloom 1996; Young, Lewis 1999) ist eine Agentenarchitektur mit Wurzeln in der Kognitionspsychologie. Das Grundprinzip beruht auf der Trennung der Problemlösekomponente von den dynamisch erweiterbaren Wissens-elementen im Gedächtnis die in den Arbeiten von Newell entwickelt wurden (Newell, Simon 1972; Newell 1990). Abhängig von dem zur Verfügung stehenden Wissen werden sehr allgemeine aber schlechte (wenig Wissen) oder sehr spezielle und gut geeignete (viel Wissen) Problemlösealgorithmen verwendet. Die Problemlösekomponente verwendet Repräsentationen des Problemraums, der als eine Art Entscheidungsbaum abgebildet wird (vgl. Lehman et al. 1996, S. 12).

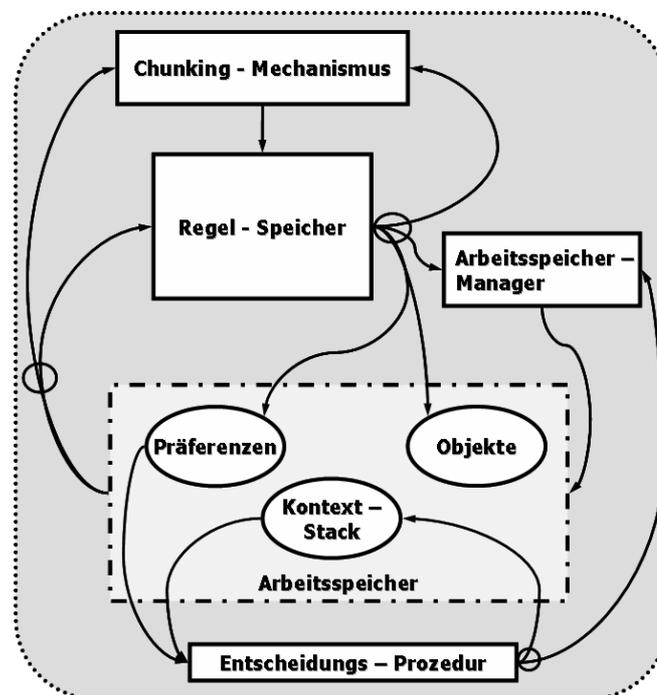


Abbildung 4-16: SOAR Architektur nach (Laird et al. 1987)

Die SOAR-Architektur ist nicht in Ebenen unterteilt. Dementsprechend lassen sich einige Kriterien nicht anwenden. Die Wissensbasis ist in Langzeitgedächtnis und Arbeitsspeicher unterteilt. Im Langzeitgedächtnis ist Wissen in Form von Regeln gespeichert. Der Arbeits-

speicher ist um das aktuelle Ziel herum aufgebaut und wird daher auch als Zielkontext bezeichnet (Lehman et al. 1996, S. 17). Er besteht aus Fakten und geladenen Regeln. Alle Regeln, die mit den Fakten im Arbeitsspeicher erfüllt werden, feuern und lösen Aktionen aus, die interne und externe Wirkung haben können. Lernen ist als Deduktion neuer Regeln aus vergangenen Aktionen und Situationsbeschreibungen implementiert.

Der Lernmechanismus wird immer dann ausgelöst, wenn die bestehenden Regeln nicht ausreichen, eine einzige Aktion auszuwählen. In so einer Situation wird ein neues Teilziel „Wissen ergänzen“ erzeugt, dessen Verfolgung zum Laden vorher als „nebensächlich“ eingestufte Fakten führt, die dann episodische Erinnerungen triggern können und so zur Auswahl einer Aktion führen können. Die Motivauswahl ist in der Literatur nicht detailliert beschrieben. Newells Theorie sieht zwar Ziele und Teilziele vor, in Veröffentlichungen wird aber oft nur von Zuständen gesprochen (Young, Lewis 1999, S. 3), die vom Ausgangszustand mit Hilfe diverser Operatoren in den nächsten Zustand überführt werden. Regeln werden gegen diese Zustände geprüft und feuern bei erfüllten Bedingungen. Über die Auflösung konfliktärer Motive wird nichts berichtet. Es gibt nur einen Ein- bzw. Austrittspunkt für Wahrnehmungen und Aktionen. Verschieden Wege sind nicht vorgesehen. Es sind keine spezialisierten Komponenten vorgesehen. Höhere Eigenschaften müssen also über die Interaktion der Komponenten entstehen. In den Publikationen ist aber nichts darüber zu finden. Kommunikation ist in der SOAR-Basisarchitektur nicht vorgesehen, auch ein soziales Modell existiert nicht.

#### **Weiterentwicklungen von SOAR:**

Eine Linie der Weiterentwicklungen von SOAR führt über Plural-SOAR (Carley et al. 1992; Carley, Prietula 1993) zu den über eine Agentenarchitektur hinaus gehenden Arbeiten von Carley et al. zu Computational Organization Theory (Carley 1994; Carley 1996; Carley, Gasser 1999; Carley 2000b; Carley 2001; Carley 2002b), bei denen neben den einzelnen Agenten auch das Umfeld berücksichtigt wird. In Plural-SOAR erweitern Carley et al. die SOAR Agenten um Kommunikationsfunktionen und ein explizites Ortsgedächtnis und zeigen, dass dadurch Agenten in einem Lagerhaus eine bessere Leistung erbringen. Die ACTS Theorie fordert darüber hinaus eine explizite Repräsentation der Aufgabe und anderer Agenten, so dass soziale Beziehungen nicht nur entstehen, sondern auch reflektiert werden können (Carley, Prietula 1994, S. 3). Implementierungen und konkrete Architekturvorschläge kommen aber nicht von Carleys

Forschungsgruppe selbst, sondern von Verhagen (Verhagen 1998; Verhagen 2000, S. 90-109).

Eine andere Weiterentwicklung ist unter dem Namen Team-SOAR in der Literatur dokumentiert (Kang et al. 1998; Kang 2001). Die entsprechenden Veröffentlichungen waren jedoch nicht zugänglich, so dass keine Details über die Richtung der Weiterentwicklung in Erfahrung gebracht werden konnten.

### 4.3.8 BRAHMS

BRAHMS (Sierhuis, Clancey 1997; Clancey et al. 1998; Sierhuis et al. 2000; Sierhuis 2001; Clancey 2002; Sierhuis, Clancey 2002; Clancey et al. 2004) ist in erster Linie eine Sprache zur Implementierung von Multiagentensimulationen. Sie eignet sich besonders zur Modellierung von Menschen die als Individuen in Organisationen und Teams zusammen arbeiten (Sierhuis et al. 2002, S. 2). Mit der Sprache verbunden sind allerdings auch Charakteristika einer Agentenarchitektur, die sich in zahlreichen Publikationen abzeichnet, aber leider nirgends als explizites Architekturbild abgedruckt ist. Die Architektur basiert auf dem BDI-Paradigma, welches um eine Vielzahl von Konzepten erweitert wurde. So ist Kommunikation möglich und ein komplexes Ortsmodell vorhanden. Eine Organisationshierarchie und die Aufmerksamkeit der Agenten wird ebenfalls explizit abgebildet (Clancey et al. 1998, S. 19). Das Verhalten der Agenten ist als Problemraumhierarchie modelliert. Auch die Entscheidungen werden etwas anders als im BDI-Vorbild als Produktionsregeln implementiert. Neben dem BDI-Modell liegen BRAHMS die Aktivitätstheorie (z.B. Engeström 1987; Kuutti 1996) und die Theorien zu situativen Aktionen und situativer Kognition zugrunde (Sierhuis et al. 2002, S. 2). Die Entwickler betonen, dass neben Menschen auch die Aktivitäten von Systemen und anderen Artefakten (im Sinne der Aktivitätstheorie) abgebildet werden, mit dem Ziel, reale Situationen möglichst detailgetreu nachzustellen. Im Gegensatz zu den anderen vorgestellten Architekturen dokumentieren die BRAHMS Entwickler ausschließlich die nachgestellten Situationen von ethnographischen Studien (hauptsächlich im Zusammenhang mit der Raumfahrt, z.B. FMARS, Apollo) und benutzen es bisher nicht um Vorhersagen zu machen, die anschließend empirisch validiert werden.

Die Architektur von BRAHMS Agenten ist nicht mehrschichtig. Insofern sind einige Kriterien nicht anwendbar. Zur Wissensrepräsentation wird First-Order Logik verwendet. Die Notation in BRAHMS weist jedoch große Ähnlichkeit mit objektorientierten Sprachen auf (z.B. `initial_beliefs: (current.groupMembership =`

ScienceOperations Team); wobei `current` den aktuellen Agenten repräsentiert, ähnlich wie `this` in Java das aktuelle Objekt bezeichnet). Diese Fakten werden durch Produktionsregeln in Thoughtframes ergänzt, die zur Deduktion neuer Fakten eingesetzt werden. Dadurch und durch Wahrnehmungen aus der Umwelt kann die Wissensbasis erweitert werden. Echtes Lernen im Sinne der KI ist nicht vorgesehen. Motivauswahl erfolgt ausschließlich auf Basis von erfüllten Vorbedingungen. Konflikte zwischen Motiven werden durch vorgegebene Priorisierung aufgelöst. Wahrnehmungen kommen an einer zentralen Stelle ins Agentensystem. Aktionen werden in sog. Workframes abgebildet, in die Vorbedingungen und die tatsächlich durchgeführten Arbeitsschritte spezifiziert sind. Workframes können dabei wiederum aus anderen Workframes bestehen, so dass eine Hierarchie entsteht. Der Workframe auf oberster Ebene bleibt dabei aktiv, solange Workframes auf unterer Ebene abgearbeitet werden. Dieser Teil der BRAHMS Architektur basiert auf Brooks Subsumptionsarchitektur. Spezialisierte Zentren für einzelne Agenteneigenschaften sind nicht vorgesehen. Auf Emotionen wird nicht explizit eingegangen. Sprache ist vorgesehen und wird als Basisaktivität abgebildet. Es können in der Wissensbasis des Agenten vorhandene Fakten kommuniziert werden. Die BRAHMS Entwickler betonen explizit die Modellierung des Gesamtsystems anstelle einer ausschließlich agenteninternen Repräsentation, wie es von der Distributed Cognition Theory vorgeschlagen wird (Sierhuis et al. 2002, S. 6). Das soziale Modell ist stark ausgeprägt und enthält sogar eine explizite Repräsentation der Organisationsstruktur.

### 4.3.9 H-CogAff

Die Architektur CogAff (Cognition and Affection, Sloman et al. 1994; Scheutz et al. 2000; Sloman 2000; Sloman, Logan 2000; Sloman 2001; Sloman et al. 2004) wird von Sloman et al. schon seit Anfang der 90er Jahre entwickelt, um menschenähnliche Agenten zu modellieren. In der aktuellen Version ist sie intern als eine Art Matrix aufgebaut, in der die Zeilen aus der reaktiven, der deliberativen und der Meta-Management Ebene bestehen und die Spalten nach Nilsons 3-Türme-Modell (Nilsson 1998) durch die Wahrnehmungs-, die Verarbeitungs- und die Aktionstürme gebildet werden (Sloman 2001, S. 10). Besonderes Augenmerk wird dabei auf die Modellierung affektiver Zustände gelegt, da sie für Sloman untrennbar mit höherer Intelligenz zusammenhängen (Scheutz et al. 2000, S. 1) und außerdem zu überlegener Performanz führen können (Scheutz et al. 2000, S. 7). Die reaktive Schicht ist die Quelle primärer Emotionen (z.B. Erstaunen, Angst), die deliberative Schicht Quelle sekundärer Emotionen (z.B. Scham, Erleichterung) und die

Meta-Management Schicht ist Quelle tertiärer Emotionen (z.B. Bewunderung, aufgeregte Erwartung, Scheutz et al. 2000, S. 4; Sloman 2001, S. 13). Die H-CogAff Architektur ist eine Spezialisierung der CogAff Architektur und speziell für die Simulation von geistig normalen entwickelten Erwachsenen erstellt worden (Sloman 2001, S. 11), während CogAff allgemein auch Kleinkinder und Menschen mit Geisteskrankheiten abbilden können soll (Sloman 2001, S. 6). Das Konzept der Persönlichkeiten dient dazu, zwischen verschiedenen Kontrollmodi der Meta-Management Schicht zu unterscheiden, so dass z.B. zwischen einem „Arbeitsmodus“, in dem die Person als Mitarbeiter einer Organisation auftritt, und einem „Freizeitmodus“, in dem die Person z.B. als Ehefrau oder Vater auftritt, unterschieden werden kann.

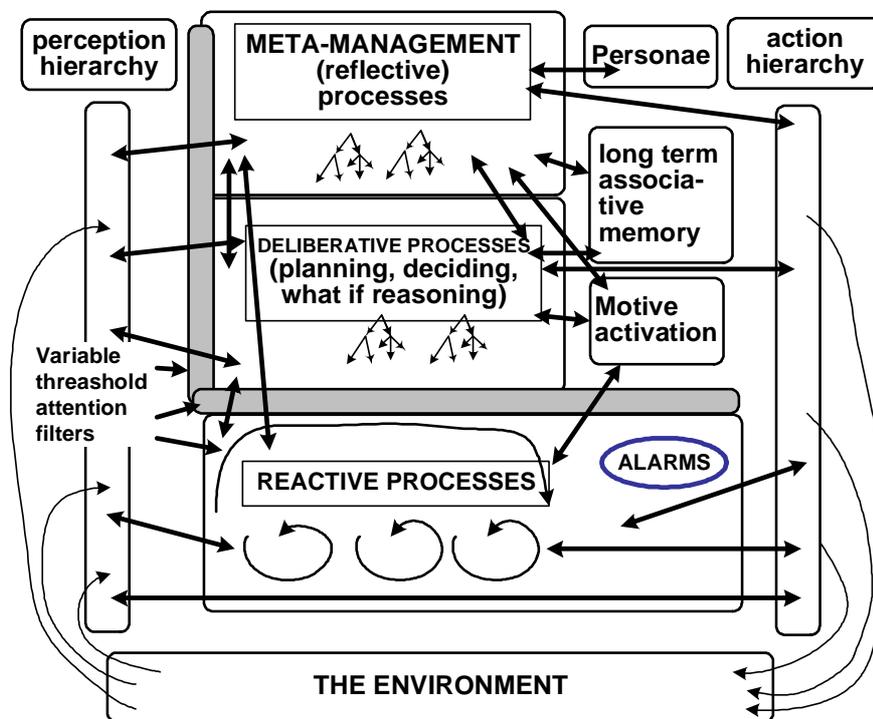


Abbildung 4-17: H-CogAff Architektur nach (Sloman et al. 2004, S. 32)

In CogAff sind alle Schichten gleichzeitig aktiv. Ein Aufmerksamkeitsfilter hilft dabei, wichtige deliberative Prozesse nicht durch reaktive Signale zu unterbrechen. Die einzelnen Schichten sind funktional differenziert. Es gibt keine direkte Kontrolle, die einzelnen Schichten können sich aber gegenseitig über sog. Alarme beeinflussen (siehe Abbildung 4-17). Auf jeder Ebene werden unterschiedliche Mechanismen eingesetzt. Wissen scheint in Form von Zuständen repräsentiert zu werden. Es werden wunschartige Zustände und überzeugungsartige Zustände unterschieden (Sloman et al. 2004, S. 20). Genauere Ausführungen zur Art der Wissensrepräsentation finden sich jedoch in den angegebenen Literaturquellen nicht. Lernen ist durch die Meta-Management Schicht möglich. Wie

genau das Lernen implementiert ist, wird ebenfalls nicht näher ausgeführt. Es kann aber in jedem Fall von einer erweiterbaren Wissensbasis ausgegangen werden.

Die Motivauswahl erfolgt über sog. Filter. Durch die mehreren gleichzeitig aktiven Ebenen, die direkt mit dem Aktionsturm verbunden sind, ist auch das gleichzeitige Aktivieren mehrerer Aktionen möglich. Wahrnehmungen können direkt an alle Schichten geleitet werden. Auch die Aktionssignale können von allen Schichten direkt an den Aktionsturm geleitet werden. Für Emotionen ist kein eigenes Funktionszentrum vorgesehen. Sie entstehen durch die Wechselwirkung zwischen den existierenden Komponenten. Über Kommunikationsmechanismen finden sich keine expliziten Aussagen. Auch über die Verwendung externer Objekte zur Erweiterung des Gedächtnisses findet sich nichts. Eigene Abstraktionen sind möglich, wie das passiert wird aber nicht erwähnt.

#### **4.3.10 MicroPSI**

Die MicroPSI Architektur (Bach 2003) stellt eine Computerimplementierung von Dörners PSI Theorie dar (Dörner 1999). Sie legt besonderes Augenmerk auf Motivation und Emotion und basiert zum einen auf der BDI Architektur und zum anderen auf CogAff. Im Gegensatz zu CogAff werden allerdings nur zwei Schichten unterschieden, die Verhaltensebene und die Meta-Management Ebene. Eine Unterscheidung zwischen reaktiver und deliberativer Ebene existiert nicht.

In MicroPSI sind wie bei CogAff mehrere Ebenen gleichzeitig aktiv, die funktional differenziert sind. Die Meta-Management-Ebene kann kontrollierend auf die Verhaltensebene einwirken. Die Entscheidungsmechanismen auf Verhaltensebene und Meta-Management Ebene sind unterschiedlich, da auf der Verhaltensebene eine vollständige BDI-Architektur umgesetzt wird, während die Meta-Management Ebene nur in Ausnahmefällen regulierend in diesen BDI-Reasoning Prozess eingreift. Sämtliches Wissen wird in MicroPSI als Knotennetz abgebildet, was an eine Mischung aus neuronalen und semantischen Netzen erinnert und als konnektionistische Wissensrepräsentation einzustufen ist. Unterschiedliche Typen von Konzeptknoten sind miteinander und mit Situationen oder Aktionen verknüpft. Aktivierungsschwellen und Gewichte können durch Lernen angepasst werden. Die Wissensbasis ist in Langzeit- und Kurzzeitgedächtnis aufgeteilt (siehe Abbildung 4-18).

Lernen ist durch Erweitern der Wissensbasis und Anpassen der Gewichte und Aktivierungsschwellen möglich. Motive werden durch körperliche, emotionale oder kognitive Bedürfnisse ausgewählt. Konflikte werden analog den BDI-Mechanismen

behandelt. Wahrnehmungen von der Umgebung gelangen über den PerceptSensor in das Agentensystem und können direkt in das Kurzzeitgedächtnis, die Verhaltensebene und die Meta-Management Ebene geleitet werden. Ein zweiter Sensor sorgt für die Überwachung des „Agentenkörpers“. Aktionssignale werden nur von der Verhaltensebene generiert und an die Komponente für externes Verhalten geleitet.

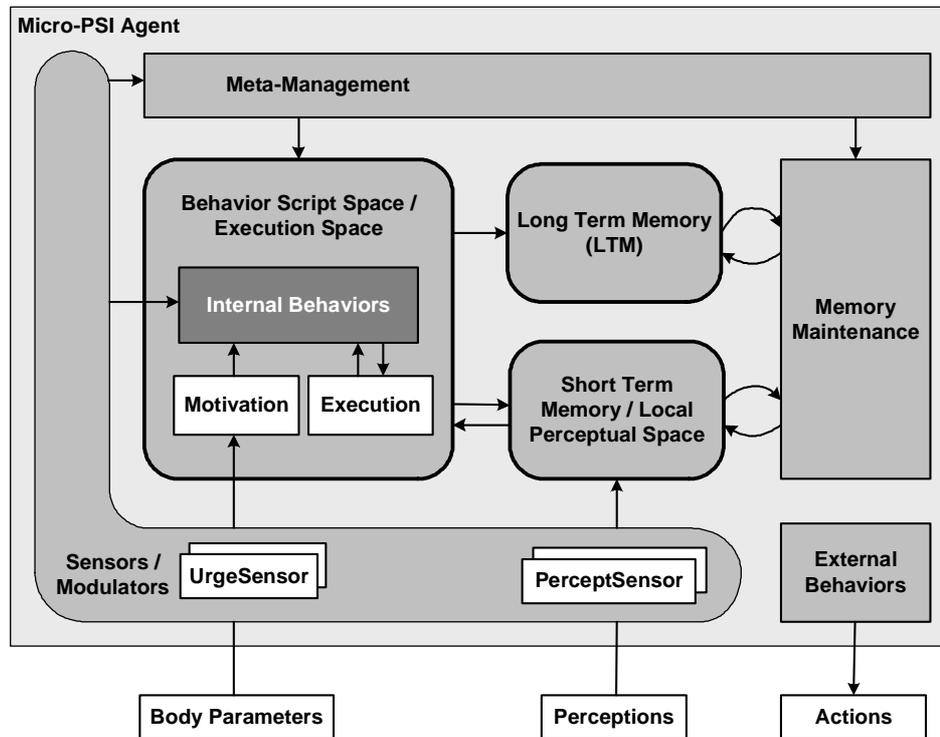


Abbildung 4-18: MicroPSI Agenten-Architektur nach (Bach 2003, S. 16)

Es gibt für Motivation und Emotionen eigene spezialisierte Zentren. Kommunikation ist bisher noch nicht implementiert, soll aber als Folge der internen Repräsentation entstehen. Ebenso fehlt die Möglichkeit zu Sozialverhalten. Eigene Abstraktion von Wahrnehmungen ist über Klassifikation und hierarchische Knotennetze möglich.

#### 4.3.11 Ergebnisse des Vergleichs

Die kriteriengestützte Vorstellung der Agentenarchitekturen bringt sowohl Unterschiede als auch Gemeinsamkeiten zum Vorschein. Zum einen stehen Schichtenarchitekturen, den Architekturen, die auf einer Kognitionstheorie basieren, gegenüber. Bei den Schichtenarchitekturen besteht über eine reaktive und eine deliberative Schicht weitgehende Einigkeit, während eine mögliche dritte Schicht in ihrer Funktion (Meta-Management, kooperative Planung) und Notwendigkeit unterschiedlich diskutiert wird. Weiterhin besitzen die meisten Architekturen eine Aufteilung der Wissensbasis in deklaratives Faktenwissen und prozedurales Regelwissen. Es lassen sich andererseits deutliche Unter-

schiede zwischen den Schichtenarchitekturen (H-CogAff, InteRRaP, MicroPSI und PECS) und den übrigen untersuchten Architekturen (BDI, SOAR, BRAHMS, ACT-R) erkennen.

Ob Carleys Anforderungen von den einzelnen Architekturen erfüllt werden ist z.T. nur schwer aus der Literatur zu den einzelnen Architekturen abzuleiten. Die Wahrnehmung der Umwelt ist überall gegeben. Auch Manipulation der Umwelt ist überall möglich. Die Fähigkeiten zum Ortswechsel innerhalb der Umgebung ist nicht direkt Teil der Architekturen, da die Bewegung als eine von mehreren möglichen Aktionen angesehen werden muss, die in der Architektur keiner besonderen Hervorhebung bedarf. Die Wissensbasen sind in allen Architekturen als universeller Speicher für Inhalte aller Art konzipiert, so dass sowohl Orte als auch Aufgaben und Personen abgebildet werden können. Trotzdem ist in einigen Architekturen explizit erwähnt, dass kein Memorieren anderer Agenten vorgesehen ist (z.B. MicroPSI). Dies zeigt, wie auch das häufige Fehlen von Kommunikationsfunktionen, dass viele der vorgestellten Architekturen aus der Zeit der Einzelagentenanwendungen stammen (ACT-R, BDI, SOAR). Die Zerlegung von Teilaufgaben und deren Koordination ist in allen Architekturen vorgesehen. Die Behandlung unvollständiger Aufgabenstellungen ist kaum explizit erwähnt. Modelle anderer Agenten sind v.a. in der InteRRaP Architektur offensichtlich vorgesehen. Ein Modell der Organisation ist nur bei BRAHMS erwähnt. Der Status-Bereich von PECS könnte aber auch dafür verwendet werden.

*Tabelle 4-2: Zusammenfassende Übersicht der Agentenarchitekturen*

	<b>BDI</b>	<b>SOAR</b>	<b>ACT-R</b>	<b>BRAHMS</b>
<b>Architekturtyp</b>	Kognitiv	Kognitiv	Kognitiv	Hybrid
<b>Wahrnehmung</b>	ein Sensor	ein Sensor	ein Sensor	ein Sensor
<b>Wissensbasis</b>	prozedural und deklarativ	prozedural und deklarativ	prozedural und deklarativ	prozedural und deklarativ
<b>Innere Architektur</b>	eine Ebene	eine Ebene	eine Ebene	eine Ebene
<b>Kommunikation</b>	nicht vorgesehen	nicht vorgesehen	nicht vorgesehen	explizit möglich
<b>Aufgaben</b>	keine Aussage möglich	keine Aussage möglich	keine Aussage möglich	keine Aussage möglich
<b>Soziales Modell</b>	nicht explizit	nicht explizit	nicht explizit	explizit vorgesehen

Wie Tabelle 4-2 zeigt sind sich die Architekturen, die keine Schichten aufweisen recht

ähnlich. Einzig BRAHMS sticht durch seine Kommunikationsmöglichkeiten und das explizit berücksichtigte Sozialmodell heraus.

*Tabelle 4-3: Zusammenfassende Übersicht der Schichtenarchitekturen*

	<b>H-CogAff</b>	<b>InteRRaP</b>	<b>MicroPSI</b>	<b>PECS</b>
<b>Architekturtyp</b>	Hybrid	Hybrid	Hybrid	Hybrid
<b>Wahrnehmung</b>	auf mehreren Ebenen	wandert durch die Ebenen	getrennt in externe und körperliche	Wahrnehmung durch Physis beeinflussbar
<b>Wissensbasis</b>	nicht näher beschrieben	3-schichtig prozedural und deklarativ	konnektionistisch	
<b>Innere Architektur</b>	Reaktiv, Deliberativ, Meta-Mgmt	Reaktiv, Deliberativ, Kooperativ	- Deliberativ, Meta-Mgmt	Reaktiv, Deliberativ -
<b>Soziales Modell</b>	Nein	Ja	Nein	Ja
<b>Kommunikation</b>	Nein	Ja	Nein	Ja
<b>Ziel-autonom</b>	Ja	Ja	Ja	Nein
<b>Aktionsauswahl</b>	Motive & Alarme	Desires & Intentions	Motive	PECS-Zustände
<b>Emotionen</b>	Ja	Nein	Ja	Ja

Insgesamt erscheinen InteRRaP, wegen der expliziten Behandlung von Kommunikation und Sozialmodell und PECS, wegen der Selbstreflexion des eigenen Status, als die geeignetsten Modelle für das untersuchte Problemgebiet. In der weiteren Arbeit wird daher auf diese beiden Architekturen aufgebaut.

#### **4.4 Verwandte Arbeiten**

Nachdem die Agentenarchitekturen diskutiert wurden, soll im Folgenden durch eine Analyse verwandter Arbeiten untersucht werden, wie andere Forscher ähnliche Simulationsvorhaben umsetzen, um für die eigene Simulation davon zu lernen. Verwandte Arbeiten ergeben sich v.a. aus zwei Grundgebieten: WM-nahe Fragestellungen, die mit klassischen Simulationstechniken bearbeitet wurden, und MABS Studien, die jedoch meist thematisch mehr oder weniger weit vom Thema Wissensmanagement entfernt sind. Tabelle 4-4 gibt einen Überblick über die verwandten Arbeiten, die dann in den folgenden Abschnitten genauer vorgestellt werden. Für eine Diskussion weiterer Simulationsstudien

zu WM-Themen wird der Leser auf (Rich 2002, S. 30ff) verwiesen, wobei Rich z.T. ältere Arbeiten aufgreift und den Schwerpunkt auf System Dynamics Arbeiten legt, aber auch einige dem Autor nicht vorliegende Arbeiten diskutiert. Weitere Arbeiten mit Schwerpunkt auf der Simulation von Softwareentwicklung werden in (Kellner et al. 1999, S. 11f; Hanakawa et al. 2002, S. 404f; Wickenberg, Davidsson 2002, S. 174f; Hanne, Neu 2004, S. 9f) vorgestellt. Neuere Arbeiten mit Schwerpunkt Softwareentwicklung finden sich auch in den Proceedings der ProSim Workshops, z.B. (Acuña, Juristo 2003; Conradi et al. 2003; Birkhölzer et al. 2004; Misic et al. 2004; Cau et al. 2005; Madachy 2005).

Tabelle 4-4: Überblick über verwandte Arbeiten

Quelle	Stichwort	Simulationstechnik	Einsatzgebiet	Analyseobjekt
(Mi, Scacchi 1990)	SE Prozesse	MABS	Softwareentwicklung	Mitarbeiter
(Carley 1992)	OL + Turnover	Monte Carlo	Org. Lernen	Mitarbeiter
(Cataldo et al. 2000)	Personnel Selection	Individuenbasiert	Personal Management	Mitarbeiter
(Electronic Arts 2000)	Die Sims	MABS	Soziologie	Person
(Mickeler et al. 2000)	OLSim	System Dynamics	Org. Lernen	Lernprozess
(Auer, Norris 2001)	ArrierosAlife	MABS	Soziologie	Selbständige
(Ben Said et al. 2001)	CUBES	MABS	Marketing	Kunde
(Gilbert et al. 2001)	Innovation Networks	MABS	Wissensmanagement	Firma
(Take 2 Interactive 2001)	Tropico	MABS	Makroökonomie	Person
(Anjewierden et al. 2002)	KMsim	System Dynamics	Wissensmanagement	Geschäftseinheit
(Dal Forno, Merlone 2002)	Arbeitseinsatz	MABS	Unternehmenskultur	Mitarbeiter
(Hanakawa et al. 2002)	Lernen im SE	Mikrosimulation	Softwareentwicklung	Mitarbeiter
(Martinez-Miranda, Aldea 2002)	Teamarbeit	MABS	Projektarbeit	Mitarbeiter

#### 4 Multiagentensimulation in den Sozial- und Wirtschaftswissenschaften

Quelle	Stichwort	Simulationstechnik	Einsatzgebiet	Analyseobjekt
(Rich 2002)	Wissensdynamik	System Dynamics	IT Berater	Firma
(Turner et al. 2002)	WM Strategie	MABS	Wissensmanagement	Team
(Inuzuka 2003)	Wissensweitergabe	MABS	Wissensmanagement	Mitarbeiter
(Schwaiger, Stahmer 2003)	SimMarket	MABS	Marketing	Kunde
(Moreno et al. 2003)	Teamarbeit	MABS	Projektarbeit	Mitarbeiter
(Canals et al. 2004)	SimISpace	MABS	Wissensmanagement	Firma
(Clancey et al. 2004)	BRAHMS	MABS	Soziologie, Raumfahrt	Mitarbeiter
(Hanne, Neu 2004)	Softwarequalität	Discrete Event	Softwareentwicklung	Mitarbeiter
(Nissen, Levitt 2004)	Wissensflüsse	COT / BS	Softwareentwicklung	Wissensfluss
(Blaschke 2004)	Demographie	Mikroanalyse	Wissensmanagement	Mitarbeiter
(Elliman et al. 2005)	Wissensintensive Tätigkeiten	Discrete Event	Schiedsgericht	Mitarbeiter
(Cowan, Jonard 2005)	Tauschmarkt	Individuenbasiert	Forschung	Mitarbeiter

Im Folgenden werden die verwandten Arbeiten näher vorgestellt. Die Darstellung behandelt dabei die folgenden Aspekte:

- Ziel der Simulation,
- Historie des Simulationsmodells,
- eingesetzte Simulationstechnik,
- Branche, in der das Simulationsmodell eingesetzt werden soll,
- Disziplin der Forscher, die das Modell entwickelt haben,
- verwendete Theorien,
- Analyseobjekt und Granularität,
- abhängige und unabhängige Variablen,
- sowie das erzielte Ergebnis.

Aufgrund fehlender Informationen können jedoch nicht immer alle Punkte behandelt werden. Abschließend wird kritisch Stellung zu der diskutierten Arbeit genommen und deren Ansatz und Ergebnis im Verhältnis zur eigenen Arbeit eingeordnet. Um die Übersicht zu wahren, werden ähnliche Arbeiten gruppiert dargestellt.

#### **4.4.1 Simulation von Organisatorischem Lernen**

Der Einfluss von unterschiedlich hoher Mitarbeiterfluktuation auf die Entwicklung der organisationsweiten Fähigkeit, Entscheidungen korrekt zu treffen, ist Untersuchungsgegenstand einer Simulationsstudie von Kathleen Carley (Carley 1992). Dabei werden einzelne Entscheidungsträger vom Analysten über das mittlere Management bis zum CEO modelliert, die ausgehend von einer Problemsituation, die als Binärwort vorliegt, eine binäre Entscheidung (ja/nein) fällen müssen. Die Entscheidung wird anhand eines einfachen Majoritätssystems (Anzahl Einser > Anzahl Nullstellen im Binärwort) und anhand der Erfahrung mit ähnlichen früheren Fällen getroffen. Je drei Analysten bilden zusammen eine Gruppe, die von einem Entscheider im mittleren Management betreut wird. Die Entscheidungen einzelner Analysten, die jeweils ein Teilproblem analysieren werden vom mittleren Management zu einer einzigen Entscheidung aggregiert und an den CEO weitergegeben, der wiederum basierend auf diesen Vorgaben die endgültige Entscheidung trifft. Lernen wird als Verbesserung der Entscheidungsqualität im Bezug auf die bestmögliche Entscheidung interpretiert, die sich durch Kenntnis des Gesamtproblems (gesamtes Binärwort) ergibt. Die Studie ist keine direkte Umsetzung einer Theorie, stützt sich aber auf den breiten theoretischen Hintergrund der Organisations- und Entscheidungstheorie. Insbesondere wird das intentionale, beschränkt rationale Verhalten von Individuen als Grundlage von Organisationen hervorgehoben (S. 21), wobei Carley sich u.a. auf (Simon 1947; March, Simon 1958; Cyert, March 1963; March, Olsen 1975) stützt.

Die Fluktuation ist als Poisson-verteilter Zufallsprozess modelliert, wobei die Rate je nach Position im Unternehmen variiert (z.B. 0,01 für Analysten, 0,0033 für Manager und 0,0011 für CEO) und in verschiedenen Szenarien niedrig, mittel oder hoch angesetzt wird (0,01, 0,02, oder 0,1). Die Komplexität der Aufgabe wird über die Länge des Binärworts abgebildet und variiert von sehr niedrig (9 Bit, je eines pro Analyst) über ungerade Vielfache von 9 bis hoch (63 Bit, 7x9). Als weitere unabhängige Variable wird die Organisationsstruktur eingeführt, die entweder hierarchisch oder teamorientiert sein kann. Beide Formen unterscheiden sich allerdings nur in der Aggregation der Entscheidungen. Die Organisationsgröße ist auf 13 Personen festgelegt (9 Analysten, 3 Manager, 1 CEO).

Als abhängige Variable fungiert die Lernrate, definiert als Anzahl der Zyklen bis zur Erhöhung der Wahrscheinlichkeit einer korrekten Entscheidung um 0,1, sowie die daraus ableitbare Anzahl korrekter Entscheidungen in den letzten 50 der insgesamt 2500 Zyklen pro Simulationslauf.

Die Ergebnisse sind im Einklang mit Aussagen aus der Literatur, dass je nach Komplexität der Aufgabe und Qualifikation neu eingestellter Mitarbeiter der einhergehende Erfahrungsverlust negativ auf die Leistung der Organisation wirkt. Im Gegensatz zur Literatur wird aber in jedem der getesteten Szenarien ein negativer Effekt festgestellt, was z.B. an der starr modellierten Umwelt liegen kann.

Eine weitere Simulationsstudie zur Untersuchung des Organisatorischen Lernens konzentriert sich auf die einzelnen Prozesse, die dies ermöglichen (Mickeler et al. 2000). Die Lernprozesse werden mit Hilfe eines System Dynamics Ansatzes abgebildet. Das Modell sieht sowohl Single-Loop als auch Double-Loop Learning vor. Lernen wird dabei nicht nur im Sinne einer Verhaltensänderung gesehen sondern allgemeiner als Veränderung der organisatorischen Wissensbasis aufgefasst. Das Modell ist so allgemein gefasst, dass es nicht auf eine konkrete Branche angewiesen ist. Als das Modell entwickelt wurde, waren die Forscher im Bereich Betriebswirtschaftslehre der Universität Mannheim tätig, und beschäftigten sich hauptsächlich mit Lernen und Simulation. Als theoretische Basis werden die Arbeiten von March und Olsen zum Organisatorischen Lernen einerseits und von Daft und Weick die Sicht auf Organisationen als Interpretationssystem andererseits verwendet (S. 4). Die Kontingenztheorie wird herangezogen, um den Zusammenhang zwischen Anpassung der Organisation an ihre Umwelt und den damit verbundenen Gewinn zu begründen (S. 5). Analysiert werden dabei die Lernprozesse in ganz abstrakter Weise. Die abhängige Variable ist der Periodengewinn, der umso höher ist, je geringer die Differenz von Umweltzustand und Organisationszustand ist, also je besser die Organisation die zukünftigen Umweltzustände vorhersagen und sich anpassen kann. Die Auswahl der Anpassungsstrategie hängt vom letzten erzielten Gewinn und von der Wissensbasis ab. Das Ergebnis der Simulation stellt sich folgendermaßen dar. Double-Loop-Learning, was die Autoren offenbar als radikale Veränderung der Organisation verstehen, ist sowohl bei statischer Umwelt (der eingangs gewählte Umweltzustand bleibt unverändert) als auch bei dynamischer Umwelt dem Single-Loop-Learning überlegen. Letzteres wird als Generieren neuer Anpassungsstrategien verstanden und ist daher an die geringe maximal mögliche Anpassung gebunden. Die Schlussfolgerungen der Autoren auf

die Relevanz von aktivem Vergessens und einer großen Wissensbasis erscheinen aufgrund des hohen Abstraktionsniveaus fragwürdig.

Welche Auswirkungen unterschiedliche demographische Faktoren auf die Wissensentstehung in Organisationen haben zeigt die Arbeit von Blaschke (2004). Er berücksichtigt dabei die Beschäftigungsdauer der Mitarbeiter sowie Fluktuation. Das Wissen wird dabei als Vektor binärer Variablen definiert, die jeweils für einen Skillwert bestimmen, ob der Mitarbeiter diesen Skill besitzt oder nicht, was zusätzlich durch den Umweltzustand beeinflusst wird, der angibt, ob der jeweilige Skill relevant ist. Blaschke stützt sein Modell im Wesentlichen auf systemtheoretische Kommunikationsstrukturen in Organisationen, wie sie von Luhmann postuliert werden. Simuliert werden dabei einzelne Mitarbeiter in einem Zeitraum von fünf Jahren mit einem Monat als atomare Zeiteinheit. In den Simulationsstudien werden fünf Unternehmenstypen mit je 50 Mitarbeitern verglichen, die sich durch die Struktur der Beschäftigungsdauer ihrer Mitarbeiter unterscheiden. Das Modell bildet zwar die Mikrostruktur des Unternehmens ab, bleibt aber bzgl. der Detaillierung sehr abstrakt, da scheinbar nur die Kommunikation zwischen Mitarbeitern simuliert wird aber kein täglicher Arbeitsablauf oder eine andere betriebswirtschaftliche Kenngröße. Als Ergebnis ermittelt Blaschke, dass „jüngere“ Organisationen schneller Wissen aufbauen als „ältere“ und eine moderate Fluktuationsrate von 5% besser für die Wissensentstehung ist als ein hohe oder gar keine.

#### **4.4.2 Sozial- und Wirtschaftssimulationen als Computerspiele**

Zu Unrecht werden in „seriösen“ Simulationsprojekten Ergebnisse aus der Forschung und Entwicklung von Computerspielen vernachlässigt. Dabei blickt das Genre Simulation innerhalb der Computerspiele nicht nur auf eine lange Tradition zurück und freut sich dank immer realistischerer Szenarien anhaltend großer Beliebtheit, sondern es gibt auch hier Grundlagenforschung, deren Ergebnisse sich auf andere Simulationsgebiete übertragen lassen. Für die vorliegende Arbeit sind vor allem zwei Gebiete interessant: grundlegende Lösungen für Probleme die sich im Zusammenhang mit der Berücksichtigung von räumlichen Anordnungen ergeben (z.B. Wegfindung von A nach B) und die Modellierung von Individuen in Wirtschafts- und Sozialsimulationen (z.B. Tropicco, Die Sims). Dort geht es im Gegensatz zu den kognitiven Agentenarchitekturen (z.B. SOAR, ACT-R) nicht darum, eine möglichst realitätsnahe Abbildung der Entscheidungswege bei Menschen zu erreichen, sondern ein möglichst glaubwürdiges Verhalten zu erzeugen (z.B. Reilly 1996). Da es für die hier angestrebte Simulation nicht auf die korrekte Abbildung kognitiver

Prozesse sondern auf realitätsnahes Verhalten der Agenten ankommt, ist dieser Bereich als hinreichend relevant einzustufen, um hier zwei erfolgreiche Vertreter solcher Computerspiele vorzustellen. Deren interner Aufbau wurde versucht, so weit wie möglich zu ermitteln. Anfragen an die Entwickler blieben leider unbeantwortet, so dass keine Implementierungsdetails geschildert werden können.

*Die Sims*<sup>34</sup> ist eines der erfolgreichsten Computerspiele der letzten Jahre<sup>35</sup> und hat nicht nur zahlreiche Add-Ons, Nachfolger und Nachahmer gefunden sondern auch einen erheblichen Frauenanteil an den Käufern (20% laut einer Umfrage). Dies kann als Indiz gewertet werden, dass die simulierten Menschen relativ glaubhaft sind und das Spiel eine emotionale Beziehung erlaubt, die für weibliche Spieler besonders wichtig ist. Die Angaben beziehen sich auf die englische Version des Spiels.

Es wird der Alltag von Menschen simuliert, wobei der Schwerpunkt auf der Freizeit liegt, während die Arbeitszeit zwar im Spiel vorkommt, aber vom Spieler nicht aktiv miterlebt werden kann. Jede simulierte Person kann dabei als Agent aufgefasst werden und wird als ein Sim bezeichnet. Ein Sim wird grafisch repräsentiert und über eine Reihe von dynamisch veränderlichen Zuständen (Needs), die als Motiv für Aktionen dienen, einige überdauernde Eigenschaften (Personality), die grundlegende Einstellungen und Vorlieben darstellen, sowie arbeitsrelevante Fertigkeiten (Job) abgebildet. Das Ziel des Spiels ist es ein erfolgreiches Leben zu führen, indem Geld verdient wird und gleichzeitig die Bedürfnisse des Sims bestmöglich befriedigt werden.

*Tabelle 4-5: Zustandsvariablen im Spiel Die Sims*

<b>Zustandsbereich</b>	<b>Zustände</b>
Needs	Bladder, Comfort, Energy, Fun, Hunger, Room, Social
Personality	Active, Neat, Nice, Outgoing, Playful
Job	Body, Charisma, Cooking, Creativity, Mechanical, Logic

Tabelle 4-5 zeigt die Zustandsvariablen, die in den einzelnen Bereichen gruppiert sind. Als Bedürfnisse werden sowohl zu minimierende Zustände modelliert (Bladder, Hunger), als auch zu maximierende Zustände (Comfort, Energy, Fun, Room, Social). Jedem Bedürfnis sind ein oder mehrere positiv beeinflussende Tätigkeiten und ein oder mehrere negativ

---

<sup>34</sup> <http://www.diesims.de/products.view.asp?id=11>, siehe auch [http://de.wikipedia.org/wiki/Die\\_Sims](http://de.wikipedia.org/wiki/Die_Sims)

<sup>35</sup> mehr als 35 Mio. verkaufte Exemplare samt Add-Ons und bereits 1 Million Verkäufe des Nachfolgers in den ersten 10 Tagen, Stand September 2005, <http://www.heise.de/newsticker/meldung/51563>

beeinflussende Tätigkeiten zugeordnet. Energie wird durch Schlafen regeneriert, Hunger durch Essen gestillt, das Bedürfnis nach Gemeinschaft (Social) durch Gespräche mit anderen Personen befriedigt und der Spaß durch Fernsehen, Computerspielen oder Tanzen erhöht. Dadurch, dass Tätigkeiten gleichzeitig mehrere Zustände beeinflussen, ergibt sich bereits eine relativ komplexe und auch glaubhafte Simulation. So steigert z.B. Basketball spielen im Hof zwar den Fun-Level, erzeugt aber auch gleichzeitig Hunger, senkt die Energie und wirkt förderlich auf das Sozialbedürfnis, wenn Freunde mitspielen. Als grundlegende Beschränkung für alle Aktivitäten wirken die knappen Ressourcen verfügbare Zeit und verfügbares Geld, das im Beruf auf Tagesbasis verdient wird und für Wohnungseinrichtung und Gartenausstattung, Haushaltshilfen zum Aufräumen und Putzen sowie Essen ausgegeben werden kann. Die Ausübung eines Berufes setzt entsprechenden Zeiteinsatz voraus, die ganz im Mikroökonomischen Sinne (Nutzen = Einkommen + Freizeit) dann nicht mehr für Freizeitaktivitäten zur Verfügung steht. Zudem werden für eine berufliche Weiterentwicklung bestimmte Fertigkeiten vorausgesetzt. Dies bringt höheres Einkommen, kann aber auch zu längeren und veränderten Arbeitszeiten führen, so dass es z.B. schwieriger wird Freunde zu treffen, wenn z.B. bis 20 Uhr gearbeitet wird. Die Fertigkeiten können durch Üben mit bestimmten Gegenständen aufgebaut werden, die jedoch erst gekauft werden müssen und Platz im Haus benötigen. So fördert z.B. Schachspiel logisches Denken, Malen mit der Staffelei die Kreativität und die Benutzung des Bücherregals (lesen) die mechanischen Kenntnisse. Für einige Berufe wird auch eine Anzahl an Freunden benötigt, die quasi die Popularität eines Sims widerspiegeln. Freundschaften werden durch regelmäßige Treffen und gleiche Interessen aufgebaut. Pro Spiel werden rund 20 Personen simuliert.

Die Persönlichkeitswerte werden zu Beginn eines Spiels einmalig festgelegt und sind unveränderlich. Sie wirken sich auf das Verhalten und die Interpretation von Situationen aus, z.B. wenn ein Sim einen geringen Wert in Active hat, so wird sein Fun-Level durch Basketball-Spielen und Tanzen weniger steigen, als beim Fernsehen; bei einem Sim mit hohem Active-Wert genau umgekehrt.

Zusammenfassend kann man sagen, dass das Zusammenspiel aus einer überschaubaren Anzahl an dynamisch veränderlichen und permanenten Zuständen und Aktionen, die einen Nutzenbeitrag bringen aber auch Kosten (im Sinne von negativem Nutzenbeitrag) verursachen, in gewissem Rahmen glaubwürdiges menschliches Verhalten erzeugt.

Ein Simulationsspiel mit ganz anderem Ziel, aber ähnlicher Realisierung ist die Wirt-

schaftssimulation *Tropico*<sup>36</sup>. Auf den ersten Blick wirkt Tropico wie eine von vielen Aufbausimulationsspielen, in denen Gebäude gebaut werden müssen, die gewisse Güter produzieren oder sonstige positive Auswirkungen haben (z.B. Ausbildung) während eine zentrale Simulationseinheit vorgibt, welche Bedürfnisse die Bevölkerung hat. Bei näherer Betrachtung stellt sich jedoch heraus, dass jeder einzelne Einwohner des simulierten Wirtschaftsraums (eine Insel) mit einer eigenständigen Persönlichkeit modelliert und damit als eigener Agent realisiert wurde. Dabei sind einige ähnliche Zustände vorgesehen wie bei Die Sims (z.B. Hunger, Energie, Spaß), aber darüber hinaus existieren noch eine Vielzahl weiterer Zustände. Tabelle 4-6 zeigt die dem Spieler zugänglichen Informationen über einen Agenten. Die genannten Bereiche entsprechen den Überschriften der Tabs im Spiel.

Die Bedürfnisse der Gesamtbevölkerung, die im Laufe des Spiels von anfangs aus rund 20 Individuen gebildet wird und auf über 200 anwachsen kann, werden durch Aggregation der Einzelbedürfnisse gebildet. Die Agenten suchen sich selbständig einen Beruf nach ihren Fähigkeiten und Stärken. Vom erzielten Einkommen mieten sie eine Wohnung und ernähren ihre Familie. Das übrige Geld können Sie für Freizeitvergnügungen (z.B. Barbesuche) ausgeben. Die Agenten können auch miteinander interagieren. So können Agenten mit hoher Führungsqualität zum Meinungsführer werden und die Ansichten anderer Agenten über Politik oder Religion beeinflussen. Dadurch entsteht eine erhebliche Komplexität, die zur Glaubwürdigkeit des Szenarios beiträgt. Andererseits wird der Spieler davon auch teilweise überfordert, weil er die Zusammenhänge nicht mehr überblicken kann und bei wirtschaftlichen Problemen in der simulierten Ökonomie mit den offensichtlichen Gegenmaßnahmen nicht zum Ziel kommt. Zu hohe Komplexität kann also auch im Fall von Computerspielen für eine Simulation von Nachteil sein.

---

<sup>36</sup> <http://www.tropico.de/>

Tabelle 4-6: Zustandsvariablen im Spiel Tropicco

Bereich	Zustände
Übersicht	Hunger, Ruhe (Energie), religiöses Bedürfnis, Entertainment Bedürfnis (Spaß), medizinisches Bedürfnis
Zufriedenheit	Essen, Wohnen, Religion, Entertainment, Medizin, Sicherheit, Umwelt, Freiheit, Job, Achtung
Politik	Unterstützung für: Religiöse, Kapitalisten, Intellektuelle Achtung vor politischer Führung
Job	Gehalt, relatives Lohnniveau, Jobqualität
Wohnung	Miete, Wohnungsqualität
Familie	Eltern, Ehepartner, Kinder
Fähigkeiten	Bildung, Intelligenz, Courage, Führungsqualität (7-stufige Lickert-Skala)
Stärken	Erfahrung in bisher ausgeübten Berufen (0.000 – 1.000)

#### 4.4.3 Simulation von Wissensflüssen und Wissensweitergabe

In der Arbeit (Cataldo et al. 2000) wird der Einfluss von gezielter Personalbeschaffung, die entweder Generalisten oder Spezialisten bevorzugt auf die Wissensweitergabe untersucht. Die Simulation baut dabei auf frühere Modelle von Kathleen Carleys Forschungsgruppe auf (z.B. Carley 1992; Carley, Prietula 1993; Carley 1994; Carley 1996), die aus dem Bereich organisatorisches Lernen und mathematische Organisationstheorie stammen. Das Modell ist individuenbasiert. Ob es sich um eine Multiagentensimulation im engeren Sinne handelt, bei der die Individuen Entscheidungsfreiheit über ihre Aktionen haben kann auf Basis der Veröffentlichung nicht eindeutig entschieden werden. Es hat aber den Anschein, dass die Individuen nur auf Basis von Wahrscheinlichkeitsverteilungen zufallsgesteuert eine von wenigen verschiedenen Aktionen ausführen (Entscheiden, Raten, Routine-tätigkeit, Cataldo et al. 2000, S. 27). Das Modell kann branchenübergreifend eingesetzt werden, fokussiert aber Büroarbeit. Es basiert auf der von Carley selbst entwickelten Construction Theorie. Dabei handelt es sich um eine sozialwissenschaftliche Theorie, die Gleichgewichte und Veränderungen in sozialen Systemen auf Veränderungen in Verteilung von Wissen zurückführt, welche durch Interaktionen zwischen Personen hervorgerufen werden, bei denen Wissen erworben und ausgetauscht wird (Cataldo et al. 2000, S. 6). Das Analyseobjekt sind Mitarbeiter einer Organisation und das Wissen, das sie untereinander austauschen. Unabhängige Variablen sind dabei die Organisationsstruktur (hierarchisch, Team, hybrid), Personalaustausch zwischen Gruppen (in Routineaufgaben,

nie), Fluktuation und Wissenstiefe (Generalist, Spezialist). Für das erste Experiment wurden vier unterschiedliche Aufgaben benutzt, die empirisch belegt sind, nämlich Dienstleistungsaufgaben, Gruppenbesprechungen, Ersatzteilverwaltung und Training (ibid, S. 28). Für jede Aufgabe besteht eine gewisse Wahrscheinlichkeit zur Interaktion und unterschiedliche Anteile an Wissensarten, die für diese Aufgaben benötigt werden. Unterschieden wird zwischen Allgemeinwissen, aufgabenspezifischem Wissen und Gruppenwissen. Letzteres würde in der Terminologie der vorliegenden Arbeit als Metawissen über das Wissen der Gruppenmitglieder bezeichnet werden. Das Wissen selbst wird als Sammlung diskreter Wissensbausteine modelliert, von denen es 1000 verschiedene gibt. Die Ergebnisse zeigen, dass in Team-orientierten Organisationen am meisten Wissen weitergegeben wird und in hierarchischen Strukturen am wenigsten. Fluktuation wirkt sich dabei positiv auf die Menge an weitergegebenem Wissen aus. Bei Generalisten wird geringfügig, jedoch statistisch signifikant mehr Wissen weitergegeben als bei Spezialisten. Ohne Fluktuation fällt der Unterschied erheblich größer aus, weil es für Spezialisten weniger neues zu lernen gibt. Die Leistung der Gruppe ist bei Routineaufgaben deutlich höher ohne Fluktuation, für alle anderen Aufgaben aber mit Fluktuation geringfügig höher. Am häufigsten wird aufgabenspezifisches Wissen ausgetauscht und das meiste Wissen wird in Entscheidungsaufgaben ausgetauscht.

In einem weiteren Experiment wird die Umwelt mit einbezogen, indem Patente angemeldet und Artikel bezogen werden können, sowie Interaktionen zwischen Organisationen bestehen. Dabei ergab sich, dass Spezialisten mehr Patente erteilt bekommen, als Generalisten.

Die Beziehungen zwischen Geschäftsprozessen und Wissensprozessen werden von einem niederländischen Forscherteam aus Twente und Amsterdam simuliert (Anjewierden et al. 2002; Shostak et al. 2002). Als Wissensprozesse werden Erwerb, Entwicklung, Nutzung, Weitergabe und Erhalt von Wissen berücksichtigt. WM-Interventionen beeinflussen, wie schnell, effektiv und effizient die Wissensprozesse ablaufen und damit die organisatorische Kompetenz, diese zusammen mit der Wissensnutzung die Ergebnisse der Geschäftseinheiten. Als Simulationstechnik wird Systems Dynamics eingesetzt. Das Modell ist prinzipiell branchenneutral konzipiert und wurde für eine fiktive Firma der chemischen Industrie namens Coltec detailliert. Die Forscher haben teils sozialwissenschaftlichen und teils Informatik-Hintergrund und arbeiten hauptsächlich im Wissensmanagement-Umfeld. Es werden keine Theorien für das Modell herangezogen sondern von eigenen Basisannahmen ausgegangen. Sie postulieren, dass Wissen quantifizierbar ist und auf einer

Relativskala gemessen werden kann, die Leistung einer Organisation von ihrem Wissen und dessen Nutzung abhängt, Wissen veraltet und neues Wissen in der Organisation entwickelt werden oder von außen in die Organisation kommen kann. Analyseobjekt sind die Wissens- und Geschäftsprozesse auf der Ebene von Geschäftseinheiten. Die abhängige Variable ist die Leistung der Organisation, die sich in Gewinn, Marktanteil und Kundenzufriedenheit niederschlägt. Die unabhängigen Variablen sind WM-Interventionen und Ereignisse aus der Umwelt der Organisation. Für Coltec wurden 50 Typen externe Ereignisse mit positiven oder negativen Auswirkungen spezifiziert und 56 Typen von WM-Interventionen in den Bereichen Organisation, Training, Entlohnung, IKT und Kooperation. Eine konkrete Intervention ist z.B. „Durchführen eines externen Trainings für Marketing und Vertrieb“. Als Ergebnis der Simulation werten die Autoren, dass die Leistung der Organisation von der Leistung aller Geschäftseinheiten abhängt, die Leistung der Geschäftseinheiten von der Wissensnutzung und der Effizienz der Wissensprozesse abhängt und die Kompetenz von den WM-Interventionen abhängt. Dies ist allerdings nicht überraschend, da die Ergebnisse direkt aus den zugrunde liegenden Annahmen folgen.

Die Arbeit von Nissen und Levitt simuliert Wissensflüsse zwischen Mitarbeitern während des IS Entwicklungsprozesses (Nissen, Levitt 2004). Sie basiert auf den Vorarbeiten der Autoren zu Wissensflüssen und Wissensprozessen (Nissen et al. 2000; Nissen 2002) sowie der Simulation von Projektarbeit (Kunz et al. 1998; Levitt et al. 1999) mit dem Tool VDT (Virtual Design Team). VDT wurde entwickelt, um Projektarbeit zu simulieren und den Einfluss von zentralistischen Entscheidungsprozessen und Koordinationsproblemen durch Abhängigkeiten zwischen Subaufgaben innerhalb von Projekten auf das Projektergebnis bzgl. Zeit, Kosten und Qualitätsabweichungen zu untersuchen. Die Entwicklung begann bereits 1988 mit der Doktorarbeit von Cohen, die 1992 fertig gestellt wurde (Cohen 1992). Seitdem wurde die Software kontinuierlich weiterentwickelt und die Annahme von Zielkongruenz aller Mitarbeiter fallen gelassen. Die Erweiterung um Wissensflüsse steht noch am Anfang (Nissen, Levitt 2004, S. 5). Simuliert werden einzelne Akteure, die in den neueren Arbeiten z.T. auch als Agenten bezeichnet werden. Es scheint sich trotzdem nicht um eine MABS im engeren Sinne zu handeln. Die Akteure agieren wie von Galbraith postuliert (Galbraith 1974) als informationsverarbeitende Einheiten, die in ein Kommunikationsnetzwerk eingebunden sind und beschränkte Verarbeitungskapazitäten haben (bounded rationality Simon 1955; Simon 1997b). In verschiedenen Studien, v.a. im Flugzeug- und Fahrzeugbau, wurden gute Übereinstimmungen zwischen Vorhersagen über

Engpässe und tatsächlichem Projekterfolg erzielt. Untersuchungsergebnisse, die auch Wissensflüsse berücksichtigen sind dem Autor nicht bekannt.

Das VDT Aktivitätsmodell besteht aus den folgenden Komponenten:

- Zeitdauer (in nominaler Zeit)
- Fehlerabhängigkeit der Aktivitäten
- Komplexität der Anforderungen (niedrig, mittel, hoch)
- Benötigte Fertigkeit (z.B. Finanzabrechnung, Stahlverarbeitung)
- Komplexität der Lösung (niedrig, mittel, hoch)
- Größe der Teilaufgaben (Zeit um eine Teilaufgabe zu erledigen, wobei alle Teilaufgaben gleich groß sind und die kleinste Arbeitseinheit darstellen)
- Nachfolger/Vorgängerbeziehungen der Aktivitäten
- Grad der Unsicherheit (niedrig, mittel, hoch)
- Arbeitsvolumen (ausgedrückt in der Zeit, die ein Arbeiter mit mittlerer Fähigkeit benötigt um die Aktivität auszuführen, unter der Annahme, es wäre keine Nacharbeit nötig)

Die Arbeit, die dem hier vorgestellten Ansatz am nächsten kommt, stammt von den japanischen Forschern Inuzuka und Nakamori (Inuzuka, Nakamori 2002a; Inuzuka, Nakamori 2002b; Inuzuka 2003; Inuzuka, Nakamori 2004). Das Ziel dieser Simulation ist es, die ökonomische Überlegenheit von Kodifizierungs- oder Personalisierungsstrategien (Hansen et al. 1999) für das Wissensmanagement in Organisationen zu ermitteln. Dazu werden Mitarbeiter einer Organisation als Agenten modelliert. Jeder Mitarbeiter ist in ein Team eingebunden, das aus 12 Mitgliedern besteht. Die Gesamtorganisation besteht aus 2-5 Teams (24-60 Personen). Die Simulation ist rundenbasiert. In jeder Runde wird ein Agent zufällig ausgewählt und ein Bedarf nach Wissen induziert. Er sucht dann nach dem Gesetz des kleinsten Aufwands (Zipf 1949) und dem Satisfizierungsprinzips (Simon 1997a) nach Wissensquellen, wobei erst das eigene Wissen, dann das Wissen anderer Agenten und schließlich kodifiziertes Wissen in einer Datenbank getestet wird. Das gesamte Wissen wird als 120 diskrete Objekte modelliert, von dem ein Agent ein oder mehrere besitzen kann. Hat er neues Wissen erworben, wird er es mit einer gewissen Wahrscheinlichkeit dokumentieren. Die abhängige Variable ist die verbrauchte Zeit als Kostengröße, die dem erfolgreich im eigenen Gedächtnis, bei Kollegen oder in der Datenbank gefundenes Wissen

als Erfolgsgröße gegenüber gestellt wird. Der Zeitverbrauch für die einzelnen Aktivitäten wurde dafür empirisch ermittelt (n=22). Die unabhängigen Variablen sind die Wahrscheinlichkeiten, mit denen Wissen zuerst bei Kollegen oder in der Datenbank nachgefragt wird und mit der Wissen dokumentiert wird. In einer Erweiterung wurden weiterhin so genannte Gatekeeper eingeführt, die eine höhere Nachfrage nach Wissen haben und auch ein größeres Meta-Wissen über das Wissen ihrer Kollegen besitzen. Zudem wurden Experimente mit Veralten von Wissen angestellt. Die Ergebnisse zeigen, dass die Personalisierungsstrategie kostengünstiger Erfolge erzielt und Gatekeeper diese nochmals verbessern, wobei ein optimales Verhältnis bei einem Gatekeeper auf vier Mitarbeiter liegt. Kritisch kann angemerkt werden, dass die Arbeit der Teams nicht im Modell berücksichtigt wird und Wissensweitergabe damit losgelöst von der Arbeitsaufgabe betrachtet wird.

Rich stellt in seiner Dissertation ein System Dynamics Modell zusammen, das die Effekte von Pflege der Wissensbasis und Veralten von Wissen auf den Erfolg von WM-Initiativen abbildet (Rich 2002; siehe auch Rich, Duchessi 2004). Er berücksichtigt dabei Wissen der Mitarbeiter sowie Wissen in einer elektronischen Wissensbasis. Die Arbeit stützt sich auf Fallstudien in zwei großen IT-Beratungsunternehmen mit mehreren Tausend Mitarbeitern. Es wird ein Modell entwickelt, in dem viele Abhängigkeiten zwischen den Mitarbeitern, Dokumenten und der Umwelt abgebildet sind. So werden bei den Mitarbeitern Junior- und Seniorberater unterschieden und bei den Dokumenten die Relevanz und die Themenabdeckung berücksichtigt. Aus der Nachfrage nach Wissen und der Erfolgsrate beim Lernen aus Dokumenten wird die Zufriedenheit mit der WM-Initiative berechnet. Bei hoher Zufriedenheit ist umgekehrt auch der Beitrag zur Wissensbasis wieder hoch. Analyseobjekt ist die Wissensmenge die unternehmensweit zur Verfügung steht. Als ökonomische Vergleichsgröße wird der Zeiteinsatz der Mitarbeiter herangezogen. Die atomare Zeiteinheit ist ein Monat bei einer Simulationszeit von 10 Jahren. Es wird eine Reihe von Ergebnissen aus mehreren Simulationsexperimenten generiert.

Im Basismodell zeigt sich, dass das Veralten der Wissensbasis eine natürliche Folge des Sammelns vieler Dokumente ist. Zudem ist durch Einsatz von 5% der Arbeitszeit von Seniorberatern zur Pflege der Wissensbasis das Wissen bei den Juniorberatern um 35% gestiegen (Rich 2002, S. 125).

Im zweiten Experiment wird die Alterungsrate von Dokumenten halbiert, so dass neues Wissen nach 6 Monaten in Allgemeinwissen übergeht und nach 18 Monaten veraltet ist.

Dadurch erreicht die Dokumentenbasis nie den hohen Relevanzwert von 0,83 des Basisszenarios, was sich durch die implementierte Feedbackschleife in niedrigerer Bereitschaft zur Pflege der Wissensbasis auswirkt, so dass die Seniorberater gegen Ende der Simulation kaum noch Zeit dafür allokierten. Dadurch sinkt die Relevanz noch schneller, so dass schließlich nur noch gut 30% erreicht werden gegenüber 55% im Basisexperiment. Die Abdeckung sinkt von über 90% auf 55% und damit auch der Wissenszuwachs bei den Juniorberatern von 35% auf 5% (S. 128f). Wird die Simulationszeit auf 20 Jahre verlängert zeigt sich, dass die Relevanz immer weiter sinkt und irgendwann die Juniorberater nichts mehr aus der Wissensbasis lernen können. Die WM-Initiative ist also nur ein kurzfristiger Erfolg und langfristig gesehen als Fehlschlag einzustufen (S.132).

Im dritten Experiment wird von einer Unterfinanzierung ausgegangen, was zu einem anfänglichen Einsatz von nur 2,5% (statt 5%) der Arbeitszeit der Seniorberater zur Pflege der Wissensbasis führt. Dadurch entwickelt sich die Wissensbasis ähnlich wie im Experiment 2, die Relevanz sinkt allerdings nicht ganz so schnell, so dass zum Schluss noch 38% erreicht wird bei einer Abdeckung von 48%. Langfristig ist die Entwicklung jedoch analog Experiment 2.

Experiment 4 geht von hohen Erwartungen der Nutzer der Wissensbasis aus, die dann nicht erfüllt werden. Dadurch sinkt die investierte Zeit in Pflege der Wissensbasis sehr schnell, was schon nach 6-8 Jahren zum Erliegen der WM-Initiative führt. Nach 10 Jahren ist die Relevanz schon unter 20% und die Abdeckung bei 50%. Im fünften und letzten Experiment werden die Erwartungen des Managements an die Auswirkungen der WM-Initiative von 2 auf 6 gesetzt, so dass diese nicht erreicht werden und in Folge dessen Ressourcen von der Pflege der Wissensbasis abgezogen werden. Dies wirkt sich zwar kurzfristig weniger dramatisch auf den Erfolg der Initiative aus (50% Relevanz und 70% Abdeckung), führt aber langfristig auch zum Scheitern, da ab Jahr 16 keine Pflege der Wissensbasis mehr stattfindet. Die Ergebnisse wurden ausgewählten Experten der Firmen vorgelegt. Diese sollten die Plausibilität der Ergebnisse bewerten. Abgesehen vom Basis-szenario dessen Ergebnisse nur zum Teil als plausibel eingeschätzt wurden, wurden die Ergebnisse der erweiterten Szenarien alle für sehr plausibel gehalten.

Auch Cowan und Jonard beschäftigen sich mit der Analyse von Wissensflüssen und konzentrieren sich dabei auf Forschungsinstitute (Cowan, Jonard 2004; Cowan, Jonard 2005). Die Forscher stammen aus den Bereichen Wirtschaftswissenschaften und

Versicherungsmathematik. Bemerkenswert an den Arbeiten ist v.a. die Offenlegung der verwendeten Funktionsgleichungen für die Zustandsübergänge. Ihre Basishypothese besteht darin, dass Wissen nur dann weitergegeben wird, wenn mittelfristig auch mit einer Gegenleistung in Form von Wissen, das zurückfließt, zu rechnen ist. Sie betrachten dabei  $l=10$  Wissensbereiche, das Wissen in jedem Bereich wird als kontinuierliche Variable gemessen, deren Skala scheinbar nach oben offen ist, deren Werte sich aber in den Experimenten zwischen 0 und 3 bewegen. Es werden 15.000 Zeitperioden durchlaufen (entspricht 7,2 Jahren bei einer atomaren Zeiteinheit von 1 h, 8h/d und 5d/w).

In zufälligen Abständen wird neues Wissen durch Kombination des Wissens in einem Gebiet mit dem Wissen in zwei angrenzenden Gebieten erzeugt. Dazu wird eine Cobb-Douglas Produktionsfunktion eingesetzt. Wissen wird genau dann weitergegeben, wenn die Individuen  $i$  und  $j$  beide voneinander lernen können, also  $i$  in einem Bereich mehr Wissen hat als  $j$  und  $j$  in einem anderen Bereich mehr Wissen hat als  $i$ . Dabei können sie einen gewissen Prozentsatz des Unterschieds im Wissen von  $i$  und  $j$  in dem bestimmten Bereich voneinander lernen, also z.B. hat  $i$  Wissen in Höhe von 2,5 im Bereich 3 und  $j$  Wissen in Höhe von 1,5 im Bereich 3, dann lernt  $j$   $\alpha * (2,5 - 1,5)$  von  $i$ , also z.B. 0,7 bei  $\alpha=0,7$ . Weiterhin berücksichtigen Cowan und Jonard soziale Netzwerke, die zwischen Mitarbeitern einer Abteilung, aber auch zwischen Mitarbeitern verschiedener Abteilungen bestehen. Letztere werden als permanente Verbindungen bezeichnet, die auch bestehen bleiben, wenn ein Mitarbeiter die Abteilung wechselt. Abteilungswechsel erfolgen, wenn ein Mitarbeiter über oder unter einer gewissen Schwelle vom Abteilungsdurchschnitt des Gesamtwissens pro Mitarbeiter liegt. Neu geschaffenes Wissen wird an alle Bekannten (mit Abstand 1) innerhalb des sozialen Netzes kommuniziert (z.B. durch Präsentation auf einer Konferenz), über verwendete Theorien wird jedoch nichts berichtet. Das Analyseobjekt ist das Wissen auf Abteilungs- und Gesamtebene. Als Simulationstechnik wird eine individuenbasierte Simulation angewendet, die rundenbasiert arbeitet, aber nicht als agentenorientiert im engeren Sinne bezeichnet werden kann.

Die abhängige Variable ist das Wissen in den zehn Bereichen. Die unabhängigen Variablen sind die Größe und Struktur der sozialen Netze und die Bereitschaft zum Wechsel der Abteilung. Als Ergebnis der Simulationsstudie ergibt sich, dass eine mittlere Anzahl von permanenten Verbindungen (3-4) zu höherem Wissen führt als mehr oder weniger permanente Verbindungen. Je mehr permanente Verbindungen existieren, desto homogener ist die Verteilung von Wissen, desto stärker wird jedoch Wissen in wenigen

Wissensbereichen gesammelt (Spezialisierung). Was das Wechseln der Abteilung anbelangt, so ist auch hier die Anzahl der permanenten Verbindungen entscheidend. Je mehr permanente Verbindungen, desto schneller der Wechsel in eine andere Abteilung, wobei sich auch schnell eine Konzentration von wissensreichen Mitarbeitern in einigen wenigen Abteilungen abzeichnet.

### **4.4.4 Simulation von Wissensmanagement-Strategien**

Zwei Arbeiten unterschiedlicher Forscherteams beschäftigen sich mit der Untersuchung von Wissensmanagement-Strategien. (Canals et al. 2004) simulieren zwei Faktoren einer Wissensmanagementstrategie, die miteinander gekoppelt werden können: Strukturierung von Wissen und Blockieren der Diffusion von firmeninternem Wissen nach außen. Als Erfolgsgröße dienen dabei sowohl das Einkommen der Organisation aus dem Verkauf oder der Lizenzierung von Wissen als auch die Wissensgüter (knowledge assets, S. 9). Die simulierten Firmen gehören alle einer Branche an und konkurrieren um Kunden und Ressourcen. Eine Firma wird dabei durch einen Softwareagenten repräsentiert. Ausgehend von einer initialen Gleichverteilung der Strategien zeigt sich schon nach 100 Perioden, dass Blockieren der Wissensdiffusion keine erfolgreiche Strategie ist, während die Strukturierung von Wissen kaum Vor- oder Nachteile zu bringen scheint (40 zu 60 Verteilung der beiden Strategiealternativen).

(Turner et al. 2002) konzentrieren sich bei ihrer Untersuchung auf Teams und untersuchen den Einfluss von spezialisiertem bzw. generalisiertem Lernen auf den Erfolg eines Teams. Weiterhin werden für Generalisten drei verschiedene Entscheidungsfindungsstrategien untersucht. Die simulierten Teams bestehen jeweils aus 4 Personen, die neue Produkte entwickeln, welche aus jeweils vier voneinander unabhängigen Komponenten bestehen. Der Erfolg des Teams wird gemessen über den Vergleich der Komponenten des Produktes mit den Wünschen des Marktes. Je qualifizierter die Mitarbeiter desto genauer können Sie die Wünsche des Marktes einschätzen. Der Markt kann sich aber auch während der Produktentwicklungszeit (100 Zyklen) ändern. Simuliert werden 50 aufeinander folgende Produktentwicklungen. Die Simulationsstudien zeigen, dass eine Spezialisierungsstrategie in allen Situationen der Generalisierungsstrategie überlegen oder gleichwertig ist.

### **4.4.5 Simulation sozialer Netzwerke**

Auer und Norris untersuchen bei Ihren Experimenten den Einfluss sozialer Netzwerke auf den Erfolg von Selbstständigen. Sie simulieren die sozialen Netzwerke zwischen

Eselreitern, sozusagen die Logistikdienstleister Perus (Auer, Norris 2001). Die Eselreiter transportieren Touristen, die sich in Touristenbüros melden oder irgendwo in der Simulationsumgebung entdeckt werden. Das Modell basiert auf empirischen Daten von dreijähriger Feldforschung des Anthropologen Auer. Das Ziel war es, Verständnis von Änderungen in den Netzwerken im Zeitverlauf zu erlangen. Zugrunde liegende Theorien werden in der Publikation nicht explizit erwähnt. Es werden jedoch einige sozialwissenschaftliche Konzepte wie Normen berücksichtigt und in das Modell integriert (Abschnitt 4.1). Das Analyseobjekt sind selbständige Eselreiter, die Touristen transportieren. Dadurch erzeugen sie ein Einkommen, das sie in weitere Esel oder in Land investieren können (auf dem dann Touristen getroffen werden können) und für die Grundversorgung ihrer Familie brauchen. Die atomare Zeiteinheit scheint ein Tag zu sein. Der Simulationszeitraum sind 3-10 Jahre. Die abhängigen Variablen sind das Vermögen der Eselreiter, das aus kumulierten Einkünften abzüglich der Lebenshaltungskosten besteht und die Bekanntschafts-, bzw. Ausleihbeziehungen zwischen Eselreitern, die sich gegenseitig mit Eseln aushelfen. Das Ausleihen von Eseln ist als einfaches spieltheoretisches Modell implementiert. Hat ein Eselreiter freie Esel und der Nachfrager hat in der Vergangenheit kooperiert, dann wird der Esel verliehen, andernfalls nicht. Die unabhängigen Variablen sind die Anzahl der Touristen, die Transportdienste benötigen, die Größe des Geländes und die „Sichtweite“ einzelner Agenten, die für das Erkennen von potentieller Kundschaft relevant ist. Als Ergebnis halten die Autoren fest, dass die beobachteten Netzwerke zum großen Teil reproduziert werden konnten, jedoch Teambildung (zwei Eselreiter transportieren gemeinsam ein Gruppe Touristen) und Kooperation (Absprachen für die zukünftige Touristenzuteilung) im Modell nicht abgebildet sind.

Gilbert et al. simulieren Netzwerke auf Organisationsebene die durch Kooperationen zur Unterstützung des Innovationsprozesses entstehen (Gilbert et al. 2001). Ziel ist es dabei, die Strukturen existierender Netzwerke nachzubilden. Die Simulationsstudien basieren auf den empirischen Studien von Gilbert und Pyka in der Mobilkommunikation und Biotechnologie sowie den theoretischen Modellen von Gilbert. Die Simulation ist als Multiagentensimulation angelegt. Das generische Modell wurde in zwei Studien auf die Mobilkommunikation- und Biotech-Branche angepasst. Die Forscher stammen aus den Disziplinen Soziologie und Wirtschaftswissenschaften. Theorien gehen nicht direkt in das Simulationsmodell ein, zur Modellierung wird jedoch ein von Gilbert entwickelter mathematischer Formalismus namens Kene verwendet. Das Analyseobjekt sind Organisationen,

die (in F&E Abteilungen) Innovationen generieren. Die abhängigen Variablen sind die Anzahl am Markt erfolgreicher Innovationen und die Kooperationen mit anderen Organisationen. Unabhängige Variablen sind die Technologiebereiche und Innovationsstrategie. Die empirischen Beobachtungen werden durch die Simulationsergebnisse reproduziert. Eine vollständige Validierung steht laut den Autoren aber noch aus.

### **4.4.6 Simulation von Teamarbeit**

Die optimale Zusammensetzung von Teams für die Bearbeitung von technischen Projekten untersucht die Arbeit (Moreno et al. 2003). Die spanischen Forscher simulieren mit einem MAS einzelne Mitarbeiter, die verschiedene Erfahrung haben (good, medium, low), abhängig davon in einer von drei Vergütungsgruppen eingruppiert sind (professional level) und sich weiterhin durch verschiedene Offenheit für Zusammenarbeit unterscheiden. Das optimale Team kann das Projekt schnell (in time) und kostengünstig (in budget) bearbeiten, was konfliktäre Ziele sind, da nur gut geschulte Mitarbeiter schnelle Erfolge erzielen, diese aber mehr kosten als weniger qualifiziertes Personal. Die Projekte setzen sich aus Arbeitspaketen zusammen, die durch Vorgänger/Nachfolger Beziehungen voneinander abhängen. Mitarbeiter können auch Unterstützung von Hilfskräften (unskilled workers) für Routinetätigkeiten anfordern. Dies ist mittels des Contract Net Protokolls realisiert. Konkrete Ergebnisse lassen sich aus der Veröffentlichung nicht erkennen. Die Software soll in Unternehmen zur Unterstützung der Projektmanager bei der Teambildung eingesetzt werden.

(Martinez-Miranda, Aldea 2002) ist eine frühere Arbeit derselben spanischen Universität und untersucht ebenfalls Teamarbeit, der Fokus liegt jedoch auf sozialen und emotionalen Aspekten. Die Simulation ist als MABS realisiert. Die Architektur der Agenten, die einen Mitarbeiter repräsentieren, ist eine Vereinfachung der PECS-Architektur, der die Physis-Komponente fehlt. Die Mitarbeiter haben einen von vier Persönlichkeitstypen (amiable, expressive, analytical, driver) und können vier emotionale Zustände einnehmen (desire, interest, disgust, anxiety) sowie Stress erfahren. Auf Typebene wird unterschieden zwischen Projektmanagern, Ingenieuren, Technikern und Assistenten. Die Erfahrung in der jeweiligen Rolle wird ebenso wie die Ausprägung der Persönlichkeiten, Emotionen und Stress auf einer dreistufigen Skala (niedrig, mittel, hoch) gemessen. Die Arbeitsaufgaben werden mit den Attributen Anzahl benötigter Mitarbeiter, Reihenfolge der Aufgaben, Schwierigkeit (komplex, simpel), Typ (generisch, spezialisiert), Termin, Priorität, Kosten und Qualität (hoch, mittel, niedrig) modelliert. Das Modell wurde mit der JADE

Agentenplattform umgesetzt, die Agenten kommunizieren über FIPA ACL miteinander. Als Branchen werden die chemische Industrie und andere Branchen mit empirischer Forschung angesprochen. Theorien werden für die Definition der Persönlichkeitstypen und Emotionen verwendet. Analyseobjekt ist das Projektteam, simuliert wird auf Ebene der Mitarbeiter. Abhängige Variable ist der Erfolg des Projekts, unabhängige Variable ergeben sich aus den geschilderten Freiheitsgraden. Ergebnisse existierten zum damaligen Zeitpunkt noch nicht. Da die beiden vorgestellten Arbeiten scheinbar aufeinander aufbauen und das jüngere Modell wesentlich weniger komplex ist, wurden scheinbar im Verlauf der Forschung ursprünglich vorgesehene Details weggelassen. Dies kann sowohl auf Schwierigkeiten bei der Implementierung als auch auf eine Fokussierung zur Verbesserung der Nachvollziehbarkeit der Ergebnisse hindeuten.

### **4.4.7 Simulation von Konsumentenverhalten**

Das CUBES Projekt (CUsomer BEhavior Simulator) versucht das Verhalten von Konsumenten basierend auf Theorien mehrere Disziplinen in einem kompetitiven Marktumfeld nachzubilden, so dass empirisch beobachtbare Phänomene wie Segmentierung und Evolution der Marktanteile einzelner Marken in der Simulation ebenfalls entstehen (Ben Said et al. 2001). Multiagentensimulation wird als Simulationstechnik eingesetzt. Es ist branchenunabhängig einsetzbar, da es nicht auf konkrete Produkte abstellt. Die Forscher haben Informatikhintergrund und arbeiten ausschließlich im Bereich der Multiagentensimulation. Die theoretischen Grundlagen der Arbeit bilden Marketingtheorien über Meinungsführer und Innovationsdiffusion, angereichert um sozialwissenschaftliche, psychologische und ökonomische Theorien über Gruppenverhalten, Meinungsbildung, Lernen, rationales Verhalten und Nutzen. Es werden mehrere tausend Konsumenten simuliert, deren Kaufverhalten durch Innovation, Opportunismus und Misstrauen bestimmt wird. Weiterhin beeinflussen die sozialen Prozesse Imitation und Konditionierung den Agenten. Die abhängige Variable ist der Marktanteil der simulierten Marken. Die Ergebnisse von zwei Experimenten zeigen, dass sich sowohl die Dominanz einer Marke über längere Zeit als auch stetiger Wettstreit zwischen mehreren Marken mit häufig wechselnden Marktanteilen reproduzieren lässt. Die Autoren erklären das mit unterschiedlichen sozialen Prozessen in der Zielgruppe.

Ein ähnliches Projekt entstand am DFKI in Saarbrücken. Dort wurden die Auswirkungen von Marketingaktionen auf die Verkäufe von Produkten in Supermärkten simuliert (Schwaiger, Stahmer 2003). Die Simulation basiert auf Verkaufsdaten mehrerer Jahre eines

großen SB-Warenhauses. Die Konsumenten werden als Agenten modelliert, die bestimmte Präferenzen und Kaufgewohnheiten besitzen. Als Supermarkt-externe Größen werden Wettbewerber, Lieferanten und ausgewählte Umwelteinflüsse wie Wetter und Aktienentwicklung jeweils als eigene Agenten modelliert. Auch Produkte werden als Agenten abgebildet, obwohl sie sich weitgehend passiv verhalten. Die beiden Forscher sind (Wirtschafts-)Informatiker. Verwendete Theorien werden nicht explizit genannt. Das Verhalten der Agenten wird als Verhaltensnetz mit widerstreitenden Präferenzen abgebildet (z.B. Qualitäts- und Preissensitivität), wobei psychologische Modelle, Marketingmodelle und die vorliegenden empirischen Daten als Ausgangsbasis dienen. Analyseobjekt sind Konsumenten, die Individuen oder Gruppen mit gleichen Präferenzen repräsentieren können. Werden Individuen modelliert, so werden die Gruppenzugehörigkeiten zusätzlich mit berücksichtigt. Die abhängige Variable ist der Umsatz mit den simulierten Produkten. Unabhängige Variable sind die Preisgestaltung der Produkte, product placement und Werbeaktionen. Von besonderem Interesse sind dabei cross selling Potentiale. Ergebnisse werden in der Quelle nicht genannt. In einem persönlichen Gespräch mit einem der Autoren wurde aber bestätigt, dass die Simulation nach Abstimmung mit den Trainingsdaten die aufgezeichneten Verkaufsdaten weitgehend reproduzieren konnte.

#### **4.4.8 Simulation von Arbeitsverhalten und Arbeitseinsatz**

(Clancey et al. 2004) untersuchen mit Hilfe des Simulationswerkzeugs BRAHMS, ob sich das Arbeitsverhalten von Mitarbeitern glaubwürdig simulieren lässt. Dazu werden einzelne Mitarbeiter als Agenten in einem Multiagentensystem auf einem extrem hohen Detaillierungsgrad modelliert. Das Modell dient dabei nicht der Voraussage sondern vielmehr der detailgetreuen Nachbildung von aufgezeichnetem Verhalten (Video und Notizen eines Soziologen). Das Modell wurde vor allem in der Raumfahrt eingesetzt, wo z.B. Apollo Astronauten auf dem Mond oder ISS Besatzungsmitglieder simuliert wurden. Die Grundlage bildet die Aktivitätentheorie. Aktivitäten werden als priorisierte Situation-Aktions-Regeln, so genannte Workframes, modelliert, die von bedingten Inferenz-Regeln, so genannten Thoughtframes, ergänzt werden (S. 4). Ein Workframe besteht aus Vorbedingungen, die der Agent für erfüllt halten muss, Aktionen, Bedingungen, die mit den Aktionen verknüpft sind, und den Konsequenzen des Handelns, die als Änderungen der Überzeugungen der Agenten und der Zustände der Umwelt abgebildet sind. Das Modell beinhaltet weiterhin Repräsentationen von Orten, Werkzeugen, Einrichtungsgegenständen, Computern, und robotischen Systemen. Eine 3D Visualisierung der Simulation unterstützt

die Auswertung und Beurteilung durch Domänenexperten. Das aus der Kognitionspsychologie heraus entstandene Modell berücksichtigt auch biologisch bedingte menschliche Bedürfnisse wie Hunger, Müdigkeit und hygienische Bedürfnisse, was als wesentlich erachtet wird (S. 18). Die Ergebnisse der Studie lassen sich folgendermaßen zusammenfassen (S. 34): Nicht direkt arbeitsbezogenen Aktivitäten wie Essen, Ausruhen und Freizeitbeschäftigungen Nachgehen müssen in Simulationen von Arbeitsverhalten berücksichtigt werden. Dementsprechend sind auch die oben angesprochenen menschlichen Bedürfnisse zu modellieren. Routinisiertes Verhalten wird oft gewissen Normen entsprechend ausgeführt aber auch flexibel den jeweiligen Umständen angepasst.

Um Arbeitsverhalten von Wissensarbeitern geht es auch in (Elliman et al. 2005). Dort werden die Arbeitsabläufe von Anwälten in Schiedsgerichten simuliert. Sie überwachen ihren Posteingang für neue Fälle und arbeiten an dem Fall eine Runde lang. Dann überprüfen sie wieder den Posteingang und die Prioritäten und arbeiten entweder weiter am letzten Fall oder beginnen die Arbeit an dem Fall mit der höchsten Priorität. Weiterhin kann die Arbeit von Sitzungen mit dem Kunden unterbrochen werden. Das Analyseobjekt sind die Anwälte, die vom System zufällig generierte Fälle bearbeiten. Relevante Variablen sowie Ergebnisse werden nicht dokumentiert. Dafür werden eine Reihe von Erweiterungen für das Simulationsmodell vorgeschlagen, die es branchenunabhängig einsetzbar machen sollen. Dazu wurden Wissensarbeiter in akademischen und nicht-akademischen Umfeldern befragt. In der Simulation sollten demnach drei verschiedene Typen von Unterbrechungen möglich sein: terminlich festgesetzte (scheduled), anforderungsbestimmte (on-demand) und freiwillige (at-will). Weiterhin wurden Strategien zur Priorisierung der Aufgaben ermittelt, wobei Termine, Wichtigkeit des Kunden und Länge der Aufgabe die wichtigsten Kriterien zum Festsetzen der Priorität waren. Auch die Aufgaben selber wurden in drei Typen unterteilt: einmalige Aufgaben, projekt-bezogene Aufgaben und regelmäßig wiederkehrende Aufgaben. Die Befragten waren selten in mehr als zwei großen Projekten gleichzeitig involviert.

(Dal Forno, Merlone 2002) simulieren eine Organisation mit fixer Struktur und ohne Hierarchie. Die Agenten sind zufällig über das Gebäude verteilt und bewegen sich zufällig. Treffen zwei Agenten aufeinander so arbeiten sie gemeinsam an einer Aufgabe, die in Form eines Interaktionsspiels abgebildet ist. Dabei entscheidet jeder Agent unabhängig über den Arbeitseinsatz, je nach seinem Typ, der die Strategie festlegt. Der Arbeitseinsatz und der Erfolg der gemeinsamen Aktion ist a posteriori für beide Agenten beobachtbar und

beeinflusst den zukünftigen Arbeitseinsatz des Agenten. Dal Forno und Merlone implementieren verschiedene Agententypen mit vollständig rationalen, beschränkt rationalen und fixen Strategien. Zudem wird die vollständige Beobachtbarkeit des anderen in einigen Experimenten durch Einführen von Umweltstörungen eingeschränkt, so dass der Arbeitseinsatz und das Ergebnis je nach Parameterwahl um bis zu 50% über- oder unterschätzt werden können. Das Ergebnis der Simulationsstudien ist, dass sich je nach Zusammensetzung der simulierten Belegschaft unterschiedliche Gleichgewichtszustände einstellen, die als Unternehmenskultur interpretiert werden. Interessant ist das Ergebnis, dass schon einzelne Agenten mit konstant hohem Arbeitseinsatz signifikant positiven Einfluss auf die Gesamtleistung der Organisation haben, da viele andere Agenten ihren eigenen Arbeitseinsatz entsprechend anpassen.

### **4.4.9 Simulation von Softwareentwicklungsprozessen**

Schon zu Beginn der 90er Jahre entstand eine Arbeit, die Softwareentwicklungsprozesse mit Hilfe von agentenbasierter Simulation untersucht (Mi, Scacchi 1990). Ein Agent repräsentiert einen Mitarbeiter, der in einem Team oder einer Arbeitsgruppe eingebunden ist und unterschiedliche Aufgaben abarbeitet. Er kann dabei auf Dokumente, Soft- und Hardware zurückgreifen, die wie das Budget als Ressource modelliert sind und mehrere Rollen innehaben. Das Verhalten der Agenten wird durch seine Fertigkeiten, die verfügbaren Ressourcen und die Arbeitsaufgaben, sowie einige andere Parameter der organisatorischen Situation, in der er sich befindet, bestimmt (S.7). Die Arbeitsaufgaben werden als Netzwerk von Aktionen abgebildet und es werden Entwicklungsaufgaben von Koordinationsaufgaben unterschieden. Prinzipiell können mit dem Simulationsmodell beliebige Softwareentwicklungsmodelle simuliert werden, z.B. das Wasserfallmodell. In der Quelle wird jedoch ein einfaches Modell mit einem Team aus drei Entwicklern beschrieben, die Architekturforschung und Detailentwurf für ein Informationssystem durchführen. Durch die kollektiven Aufgaben Kommunikation und Synchronisation können die Agenten Informationen austauschen und gemeinsam an einer Arbeitsaufgabe arbeiten. Aufgaben können erst durchgeführt werden, wenn alle Voraussetzungen dafür erfüllt sind. Insbesondere werden Reihenfolgebedingungen, Ressourcenbedingungen und Ausführungsrechte behandelt. Die beiden Forscher sind beide im Bereich Softwareentwicklung tätig. Verwendete Theorien werden nicht explizit erwähnt. Das Analyseobjekt sind Softwareentwickler. Abhängige Variable ist die Zeit, die zur Durchführung der vorgegebenen Aufgabe benötigt wird. Unabhängige Variable werden

nicht aufgeführt. Das Ergebnis besteht in einer Liste der durchgeführten Aktionen der einzelnen Agenten. Es werden keine weiteren Schlüsse daraus gezogen.

In (Hanakawa et al. 2002) wird der Einfluss von Wissen der Softwareentwickler auf den Projektfortschritt untersucht. Die Arbeit präsentiert die Weiterentwicklung des Modells in (Hanakawa et al. 1998) um die Berücksichtigung von Voraussetzungen für den Erwerb von Wissen, die in Form von Abhängigkeiten zwischen Wissensgebieten modelliert sind. Das Gesamtmodell ist in drei Teile gegliedert, das Aktivitätsmodell, das Wissensmodell und das Produktivitätsmodell. Zusammen stellen sie das Verhalten eines einzelnen Softwareentwicklers dar. Die Simulation ist damit der Mikrosimulation zuzuordnen und soll Projektfortschritte für Softwareentwicklungsprojekte vorhersagen. Das Aktivitätsmodell gibt an, welche Aufgaben bearbeitet werden sollen und welches Wissen hierzu nötig ist. Das Wissensmodell gibt an, welches Wissen der Softwareentwickler hat und wie schnell er durch Bearbeiten einer Aufgabe neues hinzu lernt. Das Produktivitätsmodell gibt an, wie das Verhältnis von vorhandenem zu benötigtem Wissen auf die Produktivität wirkt.

Hervorzuheben sind v.a. die detailliert ausgearbeiteten Funktionen für die einzelnen Teilmodelle. Während das Aktivitätsmodell noch von einer einfachen Normalverteilung ausgeht, ist der Wissenszuwachs als abschnittsweise definierte Exponentialfunktion modelliert und die Produktivität (siehe Abbildung 4-19, 4) besitzt einen S-Kurvenförmigen Verlauf (siehe Abbildung 4-19, 1, 2 und 3). Die Teilmodelle wurden nicht theoretisch fundiert sondern aus kleineren empirischen Studien destilliert. Das Analyseobjekt ist der Arbeitsfortschritt bei der Projektarbeit eines einzelnen Softwareentwicklers. Die abhängige Variable ist der Arbeitsfortschritt und das Wissen in mehreren Gebieten. Die unabhängigen Variablen sind die Lernfähigkeit des Entwicklers und die benötigten Wissensgebiete zur Erledigung der Arbeitsaufgabe (z.B. Softwareentwurf), welche aus einer Abfolge von Subaufgaben besteht (z.B. Architekturentwurf, Schnittstellenentwurf, Komponentenentwurf und Algorithmenentwurf, *ibid.* S. 388).

In den fortgeschrittenen Experimenten werden technische Reviews als Möglichkeit zum Lernen als weitere unabhängige Variable eingeführt. Die Produktivität wird gemessen in Anzahl der Seiten des Ergebnisdokuments (z.B. Entwurfsspezifikation) pro Stunde Arbeitszeit. Das Wissen wird abstrakt auf einer Skala von 1-100 gemessen. Als Ergebnis wird festgehalten, dass die Lernfähigkeit mangelndes Wissen im Projektablauf schnell ausgleichen kann (*ibid.* S. 393f) und dass technische Reviews (also Lerngelegenheiten oder Trainings) zum richtigen Zeitpunkt innerhalb des Projekts durchgeführt werden sollten, um

maximalen Erfolg zu erzielen (ibid. S. 404).

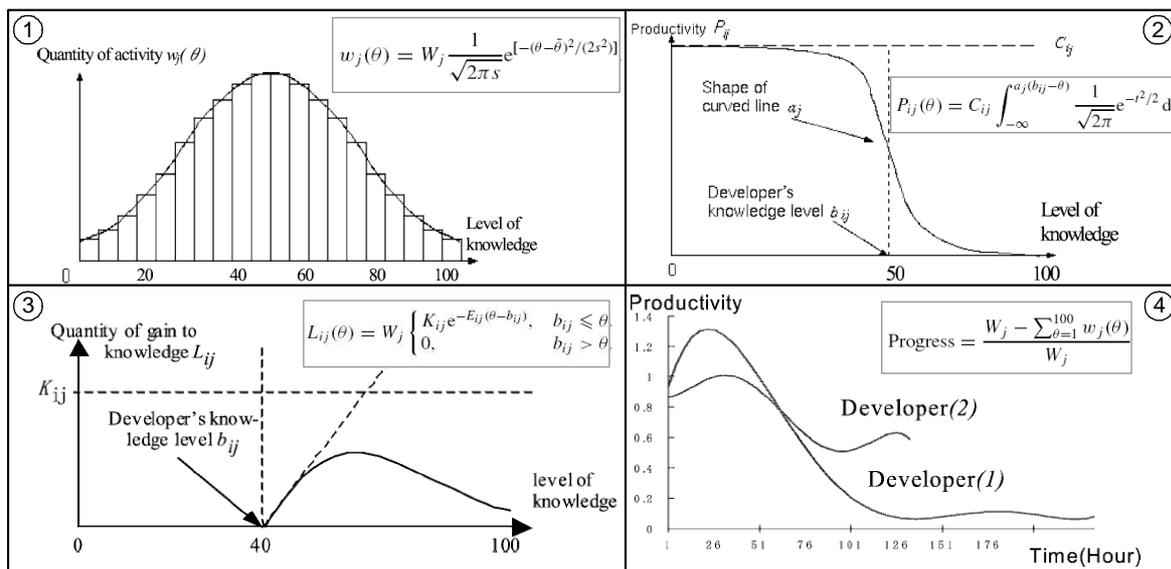


Abbildung 4-19: Aktivitäts-, Wissens-, Produktivitäts- und Gesamtmodell nach (Hanakawa et al. 2002)

(Hanne, Neu 2004) untersuchen den Einfluss von Wissen und Lernrate auf Softwarefehler und Arbeitsaufwand im Softwareerstellungprozess. Simuliert wird dabei ein Mitarbeiter, der im Softwareerstellungprozess involviert ist. Als Simulationstechnik wird Discrete Event Simulation eingesetzt. Die Forscher kommen aus den Bereichen Softwareingenieurswesen bzw. Mathematik/Optimierung. Im umfangreichen Theorieileil der Arbeit werden einige Arbeiten aus der Psychologie angeführt, die aber eher empirisch als theoriegeleitet zu sein scheinen. Die Diskussion bewegt sich auf Konzeptebene, Theorien werden nicht genannt. Die Simulation durchläuft die typischen Phasen eines Softwareentwicklungsprojekts, Entwurf, Implementierung und Test, wobei die Tests noch detailliert werden in Code Inspektion, Nacharbeit und Modultest. Der simulierte Softwareentwickler (oder die Gruppe) hat für jede Phase entsprechende Skills, die als kontinuierliche Variable im Intervall [0; 1] abgebildet sind. Dabei wird jeweils zwischen Produktivität und Qualität unterschieden. Hinzu kommen domänenspezifische Skills wie z.B. Datenbanken oder I/O, die aber in der Simulation keinen Einfluss zu haben scheinen. Beim Durchführen einer Arbeitsaufgabe steigt der entsprechende Skill (siehe Abbildung 4-20).

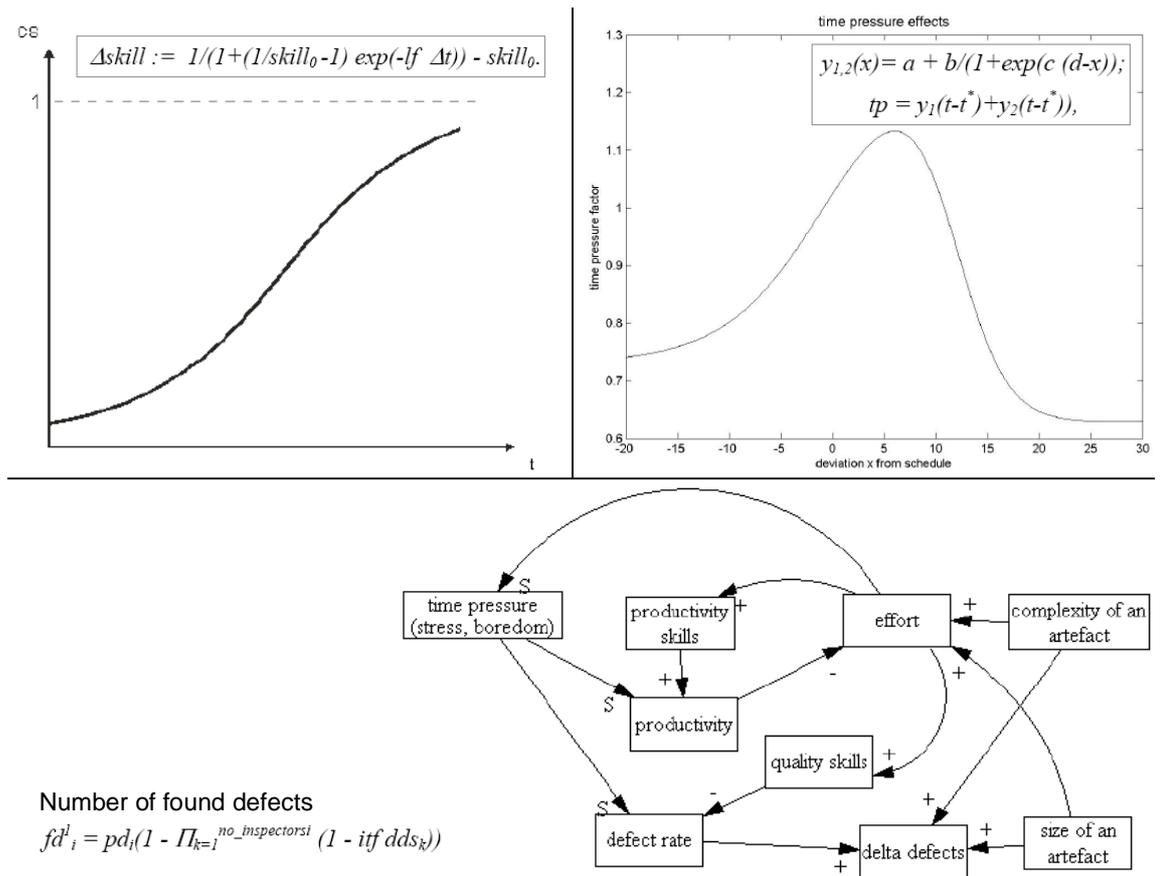


Abbildung 4-20: Modell der Softwareentwicklung nach (Hanne, Neu 2004)

Als abhängige Variablen fungieren der nötige Arbeitseinsatz in Personenstunden und die Anzahl an Fehlern im Quellcode vor und nach der Inspektion. Als unabhängige Variable werden Lernrate und Startskill verwendet. Als Startwerte für Skills sind drei Stufen vorgesehen: niedrig (0.3), mittel (0.5) und hoch (0.7). Die Lernraten für Programmieren und Inspektion wurden ebenfalls in drei verschiedenen Szenarien eingeordnet: kein Lernen (Rate = 0), mittlere Lerngeschwindigkeit (Rate = 0,0001 fürs Programmieren und 0,001 für die Inspektion) und hohe Lerngeschwindigkeit (Rate jeweils um Faktor 10 höher). Weiterhin ist als unabhängige Variable Zeitdruck vorgesehen (siehe Abbildung 4-20). Ergebnisse werden kaum diskutiert. Es wird aber ersichtlich, dass mit steigendem Startwert der Skills sowohl die Anzahl der Fehler im Quellcode als auch der nötige Arbeitseinsatz abnehmen. Die unterschiedlichen Lernraten machen bei niedrigem und mittlerem Startskill einen großen Unterschied, während bei hohem Startskill die Lernrate keinen signifikanten Einfluss mehr hat. Die Anzahl an Fehlern nach allen Tests ist bei hohem Startwert nur unwesentlich niedriger als bei mittlerem, wogegen die Anzahl der Fehler vor Durchführen der Tests wesentlich niedriger ist und der Arbeitsaufwand ebenfalls signifikant sinkt.

### 4.5 Konzeptmodell der Wissensweitergabe

Wie der vorherige Abschnitt zeigt gibt es eine Vielzahl an verwandten Arbeiten aus den Bereichen Simulation von Softwareentwicklungsarbeit, Wissensmanagement und MABS. Einige davon blicken auf eine lange Historie zurück und sind oft verfeinert und erweitert worden, so dass sie eine beachtliche Komplexität aufweisen (z.B. VDT). Aber auch in den jüngeren Simulationsarbeiten werden interessante Details modelliert, die auch für die vorliegende Arbeit relevant scheinen, z.B. Abhängigkeiten zwischen Arbeitsaufgaben, Fehler beim Codieren, die später nachgearbeitet werden müssen, oder nicht unmittelbar arbeitsrelevante Zustände wie Emotionen.

Ein eigenes Simulationsmodell muss sich neben dem realen System auch an diesen Simulationsmodellen messen lassen. Da einige der vorgestellten verwandten Arbeiten schon auf eine jahrelange Erprobung und Weiterentwicklung zurückblicken können, ist der Anspruch an eine eigene Simulation sehr hoch gesteckt. Da jedoch nur eine der angesprochenen Arbeiten Wissensweitergabe auf der Mikroebene modelliert und diese den Arbeitskontext vollständig ausblendet, erscheint es sinnvoll ein eigenes Modell für diesen speziellen Zweck zu entwickeln. Dem Drang, im eigenen Modell ein Superset aller existierenden Modelle zu schaffen, das z.B. die detaillierte Umsetzung von Teams und Arbeitsaufgaben von VDT und die detaillierte Modellierung von sozialem Verhalten von BRAHMS beinhaltet muss widerstanden werden, um das Modell handhabbar und validierbar zu machen. Andererseits sollte das zu entwickelnde Modell einfach erweiterbar sein, so dass nach erfolgreichen Experimenten mit dem einfachen Modell schrittweise Aspekte ergänzt werden können, um zu untersuchen, ob die Simulationsergebnisse dadurch realitätsnäher werden. „Das Modell muss so einfach wie möglich bleiben, während es andererseits in der Lage sein muss die relevanten Aspekte des Untersuchungsobjekts mit genügend Detail abzubilden“ (Canals et al. 2004, S. 4).

Da das hier vorgestellte Modell noch nicht das Implementierungsmodell ist, sondern ein konzeptuelles Designmodell darstellt, soll an dieser Stelle das in Kapitel 3 erarbeitete Domänenmodell noch nicht auf die später nötige Minimalversion reduziert werden, sondern eine hinreichend komplexe, aber schon mit Agentenkonzepten hinterlegte und operationalisierte Version des Domänenmodells diskutiert werden. Dazu wird zuerst aus Sicht der Softwareentwicklung aufgezeigt, welche Aspekte in der Simulation berücksichtigt werden müssen.

Nach Kellner et al. müssen bei der Simulation von Softwareprozessen zunächst die Modellgrenzen, die Ergebnisvariablen, der Prozessabstraktionsgrad und die Eingabeparameter bestimmt werden (Kellner et al. 1999). Als Modellgrenzen kommen verschiedene Bereiche in Frage, angefangen von einem Teil des Entwicklungsmodells (z.B. Anforderungsanalyse, Design, Test), über ein vollständiges Projekt, mehrere parallele Projekte bis hin zur langfristigen Evolution eines Softwaresystems oder der ganzen Organisation (ibid. S. 6f). Davon abhängig sind der zu betrachtete Zeithorizont (<12, 12-24, >24 Monate) und der zu betrachtende Ausschnitt der Organisation (weniger als ein Produkt, bzw. Projektteam, eines, oder mehrere). Als Ergebnisvariablen kommen Arbeitsaufwand/Kosten, Projektdurchlaufzeit, Anzahl der Fehler, Teambesetzung, Personalauslastung, ROI sowie die Produktivität oder Auftragsbestand in Betracht (ibid. S. 7). In Bezug auf die Prozessabstraktion sollte der Entwickler Schlüsselaufgaben und -aktivitäten, wichtige Objekte (z.B. Codeeinheiten, Fehlerberichte) und Ressource (z.B. Mitarbeiter, Hardware) identifizieren sowie die Abhängigkeiten zwischen diesen betrachten (z.B. Abhängigkeiten zwischen Aktivitäten, Objekt- und Informationsfluss zwischen Aktivitäten, Reihenfolge). Typische Eingabeparameter für Softwareprozesssimulationen sind z.B. (Kellner et al. 1999, S. 8):

- Die Menge an Arbeit (in Lines of Code oder Funktionspunkten)
- Der nötige Arbeitseinsatz als Funktion der Größe
- Die Effizienz der Fehlererkennung bei Test oder Inspektion
- Der nötige Arbeitseinsatz für Nachbesserung als eine Funktion von Größe und Anzahl identifizierter Fehler
- Die tatsächliche Fehlerbeseitigungsrate während der Nachbesserung
- Personalfluktuatation
- Personelle Kapazität und Motivation im Zeitablauf
- Menge und Effektivität von Trainings

Kirk und Tempero versuchen das Modell von Kellner et al. weiter zu operationalisieren, indem sie ein flexibel einsetzbares Modell für Softwareprozesssimulationen schaffen (Kirk, Tempero 2004). Sie verwenden dazu die Modellelemente Produkt, Ingenieur (z.B. Analyst), Aktivität (z.B. Entwurf), Methode (z.B. testen) und Kontext (z.B. verwendete Werkzeuge). Das Produkt wird dabei durch die Kennzeichen Funktionalität, Qualität und Kosten (zeitlich und monetär) beschrieben.

Neben den klassischen Phasen der Softwareentwicklung (siehe Abschnitt 3.2.2) werden in Softwareprozesssimulationen oft Nacharbeitsphasen (rework) vorgesehen, welche Iterationen zwischen Implementierung und Test zumindest teilweise abbilden (Kellner et al. 1999; Hanne, Neu 2004; Kirk, Tempero 2004).

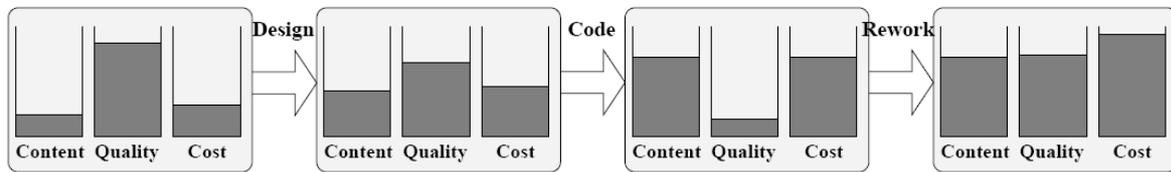


Abbildung 4-21: Zusammenhang zwischen Entwicklungsphasen und Ergebnis (Kirk, Tempero 2004, S. 5)

Wie in Abbildung 4-21 zu sehen ist, hängen die wichtigsten Ergebnisvariablen Zeit, Kosten, Qualität und Funktionalität von den Entwicklungsphasen ab (Zeit und Kosten in Abbildung zusammengefasst). Während Entwurf und Kodierung v.a. auf die Erhöhung der Funktionalität zielen, ist das Ziel der Nacharbeit die Erhöhung der Qualität. Die Kosten werden durch alle Phasen gleichermaßen (durch Personaleinsatz und Zeitverbrauch) erhöht. Nach dieser Einführung wird die verwendete Architektur vorgestellt.

### 4.5.1 Architekturüberblick

Die vorgestellten verwandten Arbeiten bauen aus nicht bekannten Gründen nicht auf etablierten Agentenarchitekturen auf und vernachlässigen dadurch Erkenntnisse des Forschungsbereichs VKI. Da dort aber auch Konzepte der Psychologie, Neurologie und den Sozialwissenschaften einfließen, geht somit ein wertvoller Beitrag verloren. Die vorliegende Arbeit versucht daher auf einer der vorgestellten Architekturen aufzubauen. Wie dort resümiert, kommen vor allem die InteRRaP- und die PECS-Architektur aufgrund ihrer Eigenschaften in Betracht. Da InteRRaP aus den Forschungen zu verteiltem Problemlösen stammt während PECS aus dem Bereich Simulation kommt, sind dort trotz aller Ähnlichkeiten unterschiedliche Erfahrungen implementiert. InteRRaP konzentriert sich stark auf die detaillierte Abbildung der Entscheidungsfindung, während PECS die Zustandsbereiche fokussiert, die Einflüsse auf das Verhalten des Agenten haben. Es bietet sich daher aus Sicht des Autors an, die beiden Architekturen miteinander zu verknüpfen. Abbildung 4-22 zeigt die SimKnowledge Agentenarchitektur als Synthese von InteRRaP-Schichten und PECS-Zustandsbereichen. Die englischen Bezeichnungen wurden gewählt, da der Code in Englisch gehalten ist und entsprechend auch die UML Diagramme englisch sind.

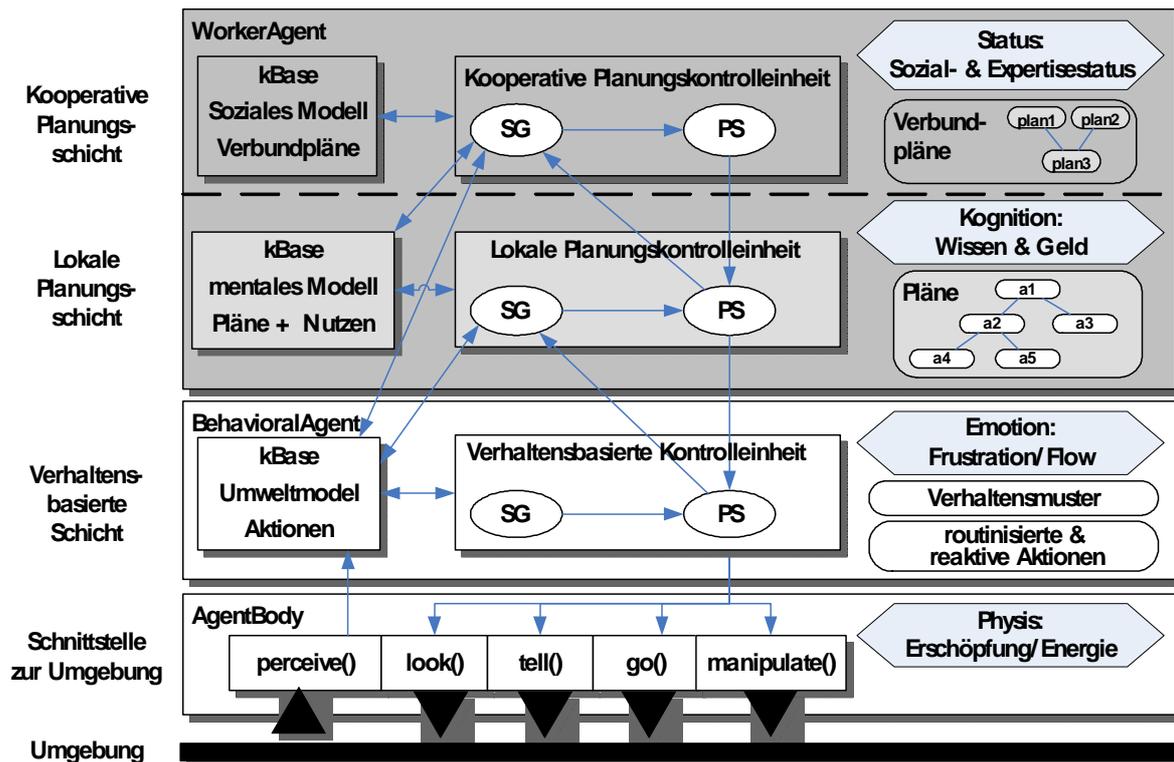


Abbildung 4-22: SimKnowledge Agentenarchitektur (eigene Darstellung)

Die strikte Trennung zwischen der lokalen und der kooperativen Planungsschicht wurde aufgeweicht und an die Zweiteilung in deliberative und reaktive Verhaltenskomponente von PECS angeglichen. Im Hinblick auf die Implementierung ist es aber zweckmäßig, Pläne, die das Ausführen von Aktionen durch andere Agenten beinhalten, von Plänen, die der Agent eigenständig ausführen kann zu trennen, wie es InteRRaP vorsieht. Die Wirkungen der Zustandsbereiche sind in der Abbildung aus Gründen der Übersichtlichkeit nicht berücksichtigt.

**AgentBody:** Die bei InteRRaP wahlweise als World Interface und Agent Body bezeichnete Schicht enthält die Sensoren und Aktoren (Effektoren), die unmittelbar auf die Umgebung wirken bzw. Wahrnehmungen aus der Umgebung aufnehmen. Als operationale Primitive sind nur Sehen (aktives Wahrnehmen der Umgebung), Sprechen (Versenden von Sprechakten), Gehen (Verändern des Standorts des Agenten) und Manipulieren (manueller Eingriff in die Umgebung) vorgesehen. Der PECS-Zustandsbereich Physis passt an dieser Stelle gut dazu. Hier könnten die ganze Reihe körperlicher Bedürfnisse abgebildet werden, die Clancey in den BRAHMS Simulationen modelliert (z.B. Müdigkeit, Harndrang, Hunger, siehe Clancey et al. 2004, S. 19 und 32) und die in ähnlicher Form auch in den Simulationsspielen aus der Sims Reihe wieder auftauchen. Clancey fordert ja explizit für die Simulation von Arbeitspraktiken das „off-task“-Verhalten mit zu berücksichtigen. Für

die Simulation von Wissensweitergabe wäre dies sicherlich ein Gewinn, da bekanntermaßen gerade in Arbeitspausen interessante Gespräche zwischen Kollegen in der Kantine, in Kaffeeküchen oder Raucherecken zustande kommen, in den neben nicht arbeitsrelevanten Geschichten auch relevantes Wissen oder zumindest Metawissen ausgetauscht wird. Trotzdem beschränkt sich diese Arbeit auf die Modellierung eines Parameters aus diesem Bereich, die physische Energie bzw. deren Mangel, der sich in Erschöpfung und Müdigkeit ausdrückt (vgl. auch Hanne, Neu 2004, S. 9) und auf die Wissensweitergabe z.B. durch schlechtere Konzentrationsfähigkeit wirken kann. Nachdem die Agenten versuchen Energiemangel zu vermeiden, wird dadurch z.B. auch bei gleichem restlichen Nutzen die „kräftesparende“ elektronische Kommunikation (Telefon, Email) dem physischen Aufsuchen des Kollegen vorgezogen.

**Verhaltensbasierte Schicht:** In der verhaltensbasierten Schicht werden Reaktionen auf Umgebungseinflüsse ausgelöst. Neben den aus der Biologie bekannten Reflexen, die keine planerischen Überlegungen im Gehirn benötigen sondern direkt auf Ebene des Rückenmarks verschaltet sind spielen hier auch routinisierte Aktionen und Verhaltensmuster eine Rolle. Dies sind nach der Aktivitätentheorie Operationen, die durch Übung und Lernen vom Stadium der bewussten Planung in das Stadium der unterbewussten Ausführung ohne Planung übergegangen sind (vgl. Abschnitt 3.3.4). Dieses Konzept ist nicht nur für eine einfache Implementierung sinnvoll (oft mit if-then-else-Blöcken), sondern wirkt auch vermittelnd zwischen den gegensätzlichen Meinungen der Vertreter von Rational Choice und Naturalistic Decision Making Ansätzen (siehe Abschnitt 3.8). Da ein großer Teil unseres täglichen Lebens aus einer Abfolge routinisierter Aktionen besteht erscheint es so, dass der in der homo oeconomicus gar nicht existiert. Tatsächlich spricht aber einiges dafür, dass (beschränkt) rationales Entscheiden eben nur selten und in neuen Situationen mit entsprechend viel Zeit zum Überlegen angewendet wird. Ist die Situation bekannt oder kann mit Analogieschlüssen erfolgreiches Verhalten in ähnlichen Situationen erinnert werden, wird diese Aktion erneut gewählt.

Auf dieser Schicht ist auch der PECS-Zustandsbereich für Emotionen einzuordnen. Was diesen anbelangt ist dem Autor kein überzeugender Beleg bekannt, der die essentielle Bedeutung von Emotionen für die realistische Simulation von menschlichem Verhalten bestätigt. Die breite Diskussion in der Community der Agenten- und Simulationsforscher muss aber als Indiz gewertet werden, dass Emotionen eine Bedeutung zukommt, auch wenn die Forschungsergebnisse diese Bedeutung noch nicht klar genug belegen. Deshalb

beinhaltet das konzeptuelle Modell auch diesen Bereich. Die Frage, welche emotionalen Zustände das sein können, ist allerdings schwierig zu beantworten und kann nur durch Experimentieren und anschließendes Validieren geklärt werden. Aus der Erfahrung des Autors spielt aber das von Csikszentmihalyi entwickelte Flow-Konzept (Csikszentmihalyi 2004) wie in Abschnitt 3.3.7 diskutiert eine Rolle für die Situationen die zu Wissensweitergabe führen, sowohl durch frustrierende Erlebnisse die zur Nachfrage von Wissen führen, als auch bei motivierenden Lernerfahrungen in Flow-Situationen, die oft zum aktiven Anbieten des neu erworbenen Wissens führen. Alternativ könnten auch die von Hanne und Neu diskutierten Faktoren Langeweile und Stress als Extremwerte einer Zustandsvariablen herangezogen werden (Hanne, Neu 2004, S. 8f). Da es keine überzeugenden Argumente für die eine oder andere Entscheidung gibt, wurde die Variable neutral als emotionale Energie bezeichnet und so modelliert, dass Sie durch Erfolge steigt und durch Misserfolge sinkt.

**Lokale Planungsschicht:** In der lokalen Planungsschicht werden alle Entscheidungen des Agenten getroffen, wenn er das avisierte Ziel alleine erreichen bzw. die Aufgabe alleine lösen kann. Aktionen werden zu Plänen zusammengesetzt und alternative Aktionen anhand des erwarteten Nutzens verglichen und diejenige mit dem höchsten Erwartungsnutzen ausgewählt. Der Nutzen wird dabei über eine Nutzenfunktion berechnet, welche die Zustände aller Schichten zur Gewichtung der erwarteten Nutzenbeiträge einer Aktion heranzieht. Die Arbeit zu Utilitarian Desire (Lang et al. 2000) stellt dar, dass die Desires der BDI-Architektur auch als Nutzengewinne oder -verluste interpretiert werden können und somit zu einer gemeinsamen Funktion der Präferenzen des Agenten kombiniert werden können (vgl. auch Wendelken, Shastri 2002 für einen ähnlichen Ansatz). Je weiter vom angestrebten Zustandsniveau der aktuelle Zustand abweicht, umso stärker wird ein Nutzenbeitrag für diesen Zustand bewertet. Darüber hinaus werden auch Gewichte vergeben, welche den Präferenzen eines Agenten für die einzelnen Zustände entsprechen. Diese Präferenzen können als Persönlichkeitsstruktur eines Agenten aufgefasst werden. Für die Ermittlung der Zustände im PECS-Zustandsbereich Kognition, der in dieser Schicht anzusiedeln ist, wird auf die Relation Model Theory von Fiske und deren Interpretation im Sinne der Wissensweitergabe von Boer et al. zurückgegriffen (siehe Abschnitt 3.3.2). Die vier Modelle lassen sich auch als Streben nach Geld (Market Pricing), Streben nach Ansehen oder Macht (Authority Ranking), Streben nach Wissen (Equality Matching) und Streben nach sozialer Zugehörigkeit (Communal Sharing)

interpretieren. Dementsprechend werden im Bereich Kognition die Zustände Wissen und Geld eingeführt. Der in der Nutzenfunktion verwendete Zustand Wissen kann dabei als einfache Summe des Wissens in allen Wissensbereichen modelliert werden. Durch höhere Gewichtung von neuem Wissen in bisher wenig ausgeprägten Wissensbereichen oder von neuem Wissen in schon stark ausgeprägten Wissensbereichen können Generalisten bzw. Spezialisten simuliert werden (siehe Abschnitt 3.3.5). Der Zustand Geld steht stellvertretend für monetäre Anreize jeder Art und ist unerlässlich für die Simulation der Auswirkungen von Anreizsystemen auf die Wissensweitergabe. Auch höherer Arbeitsinsatz durch bezahlte Überstunden oder Streben nach höheren Monatseinkommen ist damit simulierbar. Eine wichtige Frage in diesem Zusammenhang ist z.B. ob die empirisch beobachtbaren Fehlschläge von monetären Anreizen für die Wissensweitergabe mit Hilfe eines so konzipierten Simulationsmodells reproduziert werden können.

***Kooperative Planungsschicht:*** In der kooperativen Planungsschicht findet wie in der lokalen Planungsschicht deliberatives Planen statt. Der Unterschied besteht in der expliziten Berücksichtigung anderer Agenten und deren Beiträgen zur Zielerreichung. Dazu muss es explizite Repräsentationen anderer Agenten und deren Fähigkeiten bzw. Wissen geben (siehe Abschnitt 4.5.2). Der PECS-Zustandsbereich (sozialer) Status ist in dieser Schicht einzuordnen. Die Ausgestaltung erfolgt mit den Zuständen Sozialstatus und Expertisestatus, die das Streben nach sozialer Zugehörigkeit und Anerkennung repräsentieren, wie es von der Relation Model Theory in den Modellen Communal Sharing und Authority Ranking impliziert wird.

### 4.5.2 Wissensrepräsentation

Betrachtet man den Aufbau der Wissensbasis im Detail, so stellt sich die Frage, welche Inhalte abgebildet werden müssen. Eine mögliche Antwort auf die Frage liefert die Meta-Matrix von Carley (Carley 2000a; Carley 2001). Sie schlägt vor, das Gedächtnis als Matrix von relevanten Entitäten aufzubauen. Dazu zählen zum einen die Agenten selbst, deren Wissen und die Aufgaben der Agenten. Hinzu kommen je nach Anforderungen der Simulation die Organisation (je nach Modell inter- oder intra-organisational) und die Ressourcen. Ordnet man diese Entitäten in einer Matrix an, so ergeben sich eine Reihe von Netzwerken, wie in Tabelle 4-7 zu sehen ist. Ein Agent sollte demnach wissen, mit welchen anderen Agenten er interagiert hat. Weiterhin weiß er, welches Wissen er besitzt, und wie Wissensbausteine zusammenhängen. Bei den Ressourcen weiß der Agent, welche Fähigkeiten andere Agenten haben.

Tabelle 4-7: Meta-Matrix nach (Carley 2001, S. 3), (Carley 2002b, S. 211), (Carley, Tsvetovat 2004, S. 24)

	<b>Agent</b>	<b>Wissen</b>	<b>Ressourcen</b>	<b>Aufgabe</b>	<b>Organisation</b>
<b>Agent</b>	Interaktions- Netzwerk	Wissens- Netzwerk	Fähigkeits- Netzwerk	Zuweisungs- Netzwerk	Teammitglieder- Netzwerk
<b>Wissen</b>		Informations- Netzwerk	Fertigkeits- Netzwerk	Anforderungs- Netzwerk	Kompetenz- Netzwerk
<b>Ressource</b>			Substitutions- Netzwerk	Bedarfs- Netzwerk	Investitions- Netzwerk
<b>Aufgabe</b>				Vorgänger- Netzwerk	Teamaufgaben- Netzwerk
<b>Organisation</b>					Inter-Team- Netzwerk

Das Fertigkeiten-Netzwerk gibt an, welches Wissen benötigt wird, um mit Ressourcen umzugehen. Das Wissen über Substitutionsmöglichkeiten zwischen Ressourcen ist schon sehr fortgeschritten und wird nur in wenigen Simulationen notwendig sein. Zudem muss ein Agent wissen, welche Aufgabe von wem bearbeitet wird und welche Aufgabe welches Wissen und welche Ressourcen erfordert. Das Netzwerk zwischen Aufgaben bildet die Vorgänger-Nachfolger-Beziehungen ab. Im Organisationsbereich muss bekannt sein, welche Agenten zum eigenen Team gehören, welche Teams über welches Wissen und welche Ressourcen verfügen und welche Aufgaben sie bearbeiten. Schließlich kann auch noch Wissen über die Interaktionen zwischen Teams relevant sein<sup>37</sup>.

Bezüglich der Wissensrepräsentation finden sich in der Literatur sowohl Arbeiten, die simuliertes Wissen in Form von diskreten Wissens-elementen abbilden (Carley 1992; Cataldo et al. 2000; Inuzuka 2003), als auch Arbeiten, die Wissen als kontinuierliche Variable modellieren (Rich 2002; Hanne, Neu 2004). Eine dritte Variante der Wissensrepräsentationen ist zwar ebenfalls mit diskreten Variablen realisiert, benutzt jedoch im Gegensatz zu den oben zitierten Arbeiten keine Binärworte sondern qualitative Wissensstufen (z.B. hoch, mittel, niedrig, Martinez-Miranda, Aldea 2002; Moreno et al. 2003), was aber als zu grob für den Hauptuntersuchungsgegenstand erachtet wird. In einigen Publikationen wird auch zwischen Generalisten mit wenig tief greifendem Wissen

<sup>37</sup> Bei Carley ursprünglich Inter-Organisations-Netzwerk, was jedoch für die vorliegende Arbeit aufgrund der Abgrenzung auf intra-organisatorische Phänomene nicht relevant ist.

in vielen Wissensgebieten und Spezialisten mit tief greifendem Wissen in wenigen Wissensgebieten unterschieden. (Cataldo et al. 2000) modelliert diese Unterschiede in einem Binärstring als weit auseinander liegenden Einsen bzw. Clustern von zusammen liegenden Einsen (S. 12f). Es gibt Argumente für beide Abbildungen, so dass schließlich rein subjektiv eine kontinuierliche Repräsentation bevorzugt wurde.

### 4.5.3 Umgebung

Die Umgebung, in der sich die Agenten bewegen, wird durch zelluläre Automaten abgebildet, die durch Moore-Nachbarschaft mit Radius eins mit ihren Nachbarn verbunden sind. Dadurch ist es möglich eine räumliche, 2-dimensionale Umgebung bereitzustellen, die wesentliche Eigenschaften von realen Umgebungen abbilden kann. So können die zellulären Automaten akustische und visuelle Signale weiterleiten, so dass andere Agenten wahrgenommen werden können und Unterhaltungen zwischen Agenten auch von anderen, räumlich nahen Agenten verfolgt werden können. Durch kontinuierliche Abschwächung der Signale bei der Weiterleitung kann die Reichweite festgelegt werden. Durch Spezialisierung der Zellen können Wände von Gängen und Räumen unterschieden werden und mit unterschiedlichen Dämpfungen so auch die Akustik unterschiedlicher Bürolayouts simuliert werden. Somit könnten auch Ablenkungen durch Störgeräusche modelliert werden, was aber für die vorliegende Arbeit nicht relevant erscheint. Relevant ist hingegen das Wahrnehmen anderer Agenten, da durch visuelle Wahrnehmung soziale Kontakte initiiert werden können und durch Mithören von Unterhaltungen Wissen „aufgeschnappt“ werden kann, wobei das in erster Linie für Meta-Wissen gilt, da arbeitsrelevantes Wissen meist nicht durch Mithören einiger Sätze erlangt werden kann. Es kann aber sein, dass die Aufmerksamkeit auf das Gespräch gelenkt wird und der Mithörer sich daraufhin aktiv in das Gespräch einschaltet.

## 5 SimKnowledge

Nachdem das mit Agenten-Architektur hinterlegte Konzeptmodell vorgestellt wurde, wird in diesem Kapitel das eigentliche Simulationsmodell beschrieben. Zuerst werden Erfahrungen aus dem Agent-oriented Software Engineering diskutiert. Anschließend wird ein Überblick über das Simulationsmodell gegeben und eine detaillierte Beschreibung des Modells strukturiert nach der MASSIVE Methode vorgestellt. Abschließend werden die Analysen zur Auswahl der Simulationsprache und eines Simulationsframeworks präsentiert, bevor einige Besonderheiten der Implementierung vorgestellt werden.

### 5.1 Agent-oriented Software Engineering

Zur Entwicklung von Multiagentensystemen (MAS) können prinzipiell herkömmliche Entwicklungsmethoden verwendet werden. Durch speziell auf die agentenorientierte Softwareentwicklung (AOSE) angepasste Methoden kann der Entwicklungsprozess jedoch besser unterstützt werden (Weiß, Jakob 2005, S. 15). Zum Teil wird die Agentenorientierung sogar als allgemeine Weiterentwicklung bisheriger Softwareentwicklungsansätze gesehen (Timm et al. 2006, S. 342). Die AOSE Ansätze sind dementsprechend nicht als vollständig eigenständige Ansätze entwickelt worden, sondern stellen i. d. R. Weiterentwicklungen existierender Methoden dar.

Weiß und Jakob sehen vier Ausgangspunkte für AOSE Methoden, nämlich die Objektorientierung, das Requirements Engineering und Knowledge Engineering sowie Methoden, die sich direkt aus der Agententechnologie entwickelt haben (Weiß, Jakob 2005, S. 15). Eine Übersicht ausgewählter Methoden ist in Abbildung 5-1 zu sehen.

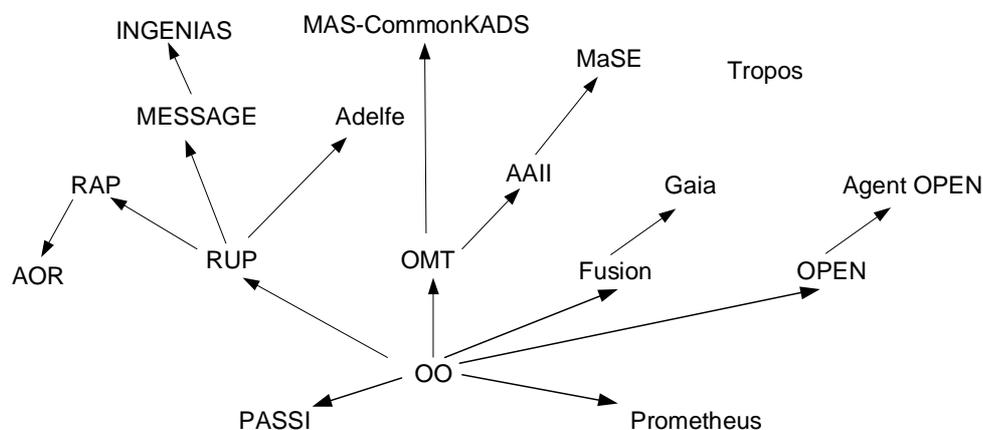


Abbildung 5-1: Genealogie einiger AOSE-Methoden in Anlehnung an (Henderson-Sellers, Giorgine 2006, S. 7)

Im Folgenden soll je ein Vertreter aus einem der Bereiche kurz vorgestellt werden. Für einen detaillierteren Vergleich sei der Leser auf (Weiß, Jakob 2005, S. 40-188) verwiesen, wo Gaia, MASSIVE, Zeus, MaSE und Aalaadin einander gegenübergestellt werden. Auch in (Henderson-Sellers, Giorgine 2006) werden einige Methoden detailliert vorgestellt (Tropos, MAS-CommonKADS, PASSI, Prometheus, Gaia, MESSAGE, INGENIAS, S. 20-276), welche in einem späteren Kapitel mit den Methoden Adelfe, MaSE und RAP kriteriengestützt verglichen werden (S. 341-367).

Der Fokus von Methoden wie Gaia liegt auf agentenspezifischen Abstraktionen wie Gruppen, Organisationen, Rollen und Methoden zur Spezifikation von Koordination und Kommunikation (Timm et al. 2006, S. 350). Gaia wurde von Wooldridge et al. entwickelt und zielte in seiner ursprünglichen Version auf benevolente Agenten in geschlossenen MAS. Die Methode sieht drei Phasen vor, Anforderungsspezifikation, Analyse (Rollen und Interaktionen) und Entwurf (Agentenmodell, Dienstmodell und Beziehungsmodell, *ibid.*). Spätere Phasen des Softwareentwicklungsprozesses werden nicht berücksichtigt. Organisatorische Abstraktionen wurden erst in späteren Erweiterungen durch Zambonelli et al. eingeführt. Andere Weiterentwicklungen sind ROADMAP und KGR (Weiß, Jakob 2005, S. 48).

Tropos fokussiert stark die Anforderungsanalyse und gliedert sich in die Phasen frühe Anforderungen, späte Anforderungen, Architektorentwurf, Detailentwurf und Implementierung. Die Methode benutzt objektorientierte Modellierungstechniken wie UML und wurde entwickelt um BDI Agenten für die JACK Agentenplattform zu entwickeln (Timm et al. 2006, S. 351). Tropos bietet darüber hinaus formale Analysetechniken zur Überprüfung der Korrektheit der Anforderungen. Die Spezifikation wird dazu in der temporalen Sprache KAOS angegeben. Tropos stellt auf soziale Anwendungsfälle ab und greift Konzepte aus Organisationstheorie, strategischen Allianzen und Sozialwissenschaften auf (*ibid.*).

Nachdem die Wissensbasis eines Agenten aus den Forschungen zur Wissensmodellierung hervorgegangen ist, überrascht es nicht, dass auch in Erweiterung zu der verbreiteten Wissensmodellierungsmethode CommonKADS agentenspezifische Methoden entwickelt wurden (Timm et al. 2006, S. 352). MAS-CommonKADS und CoMoMAS sind sich sehr ähnlich und unterstützen die Phasen Analyse und Entwurf. Es wird dabei eine multiperspektivische Sicht auf die Problemstellung aufgebaut, die aus Agenten-, Aufgaben-, Expertise-, Koordinations-, Kommunikations-, Organisations- und

Entwurfsmodell besteht (Henderson-Sellers, Giorgine 2006, S. 46). Trotz hoher Erwartungen wurden diese Methoden nicht weit verbreitet, da es versäumt wurde, Unterstützung in Entwicklungsumgebungen einzubauen (Timm et al. 2006, S. 352).

MASSIVE (MultiAgent SystemS Iterative View Engineering) ist ein Vertreter der objektorientierten Methoden und propagiert Konzentration auf die Anforderungen statt auf Technologien, Sichten statt eines Modells, iteratives statt sequentielles Vorgehen und ein Gleichgewicht von Entwurf und Implementierung (Weiß, Jakob 2005, S. 81). MASSIVE besteht aus dem Produktmodell mit den in Tabelle 5-1 dargestellten Sichten, dem Prozessmodell, welches die einzelnen Entwicklungsschritte beschreibt, und einem als „Erfahrungswerkstatt“ bezeichnetem organisatorischen Rahmen zur Nutzung der gewonnenen Erfahrungen aus einem Projekt in anderen Projekten (ibid., S. 82).

Neben umfassenden Methoden wurden auch auf Technologieebene Vorschläge zur Erweiterung objektorientierter Methoden gemacht. Insbesondere Erweiterungen zu UML wie AgentUML werden seit Jahren diskutiert (z.B. Odell et al. 2000). Einige der Vorschläge sind auch in die Entwicklung von UML 2.0 eingeflossen, so dass neben den Klassendiagrammen auch Sequenzdiagramme und Aktivitätsdiagramme für die Modellierung von Agentensystemen empfohlen werden (Klügl et al. 2002). Formale Überprüfbarkeit erhält man durch die Verwendung der OCL (Object Constraint Language). Zur Entwicklung des eigenen Simulationsmodells wurde nicht strikt einer Entwicklungsmethodik gefolgt sondern sich grob an der MASSIVE Methode orientiert, da ein iteratives Vorgehen favorisiert wurde.

## 5.2 Überblick

Um der Beschreibung des eigentlichen Simulationssystems eine gewisse Struktur zu geben, wurde auf die in der MASSIVE Vorgehensmethode definierte Gliederung zurückgegriffen. MASSIVE wurde deswegen ausgewählt, weil das dort beschriebene Vorgehen ebenfalls weitgehend bei der Erstellung der Arbeit verwendet wurde. In MASSIVE werden eine Reihe von Sichten definiert, die zur Anforderungsanalyse und zur Spezifikation des Systems dienen.

Bevor jedoch das SimKnowledge Modell aus der Perspektive der verschiedenen Sichten dargestellt wird, soll ein kurzer Überblick gegeben werden, damit die darauf folgenden Teilaspekte besser verständlich sind.

Tabelle 5-1: Sichten in MASSIVE

		<b>Analyse</b>	<b>Design</b>
Agentenspezifikation	Aufgabensicht	X	
	Systemsicht	X	X
	Umgebungssicht	X	
	Architektursicht		X
	Rollensicht		X
	Interaktionssicht		X
	Gesellschaftssicht		X

Simuliert werden soll ein mittelständischer IT-Dienstleistungsbetrieb, der Softwareentwicklungs- und Systemintegrationsprojekte für Kunden abwickelt. Dieses Beispiel wurde gewählt, da die IT-Branche generell und Dienstleistungen im Bereich Systemintegration als sehr wissensintensiv gelten und kurze Innovationszyklen aufweisen (Trittmann, Brössler 2000). Zudem hat der Autor eigene praktische Erfahrung in diesem Bereich. Es gibt weiterhin Überschneidungen zu den in der Literatur diskutierten Bereichen IS-Design und New Product Development.

Die Modellgrenzen entsprechen dem Unternehmensgebäude. Als externe Einflüsse wirken der Absatzmarkt (Kunden, Wirtschaftslage, Konkurrenten) und der Arbeitsmarkt. Der Finanzmarkt als dritte relevante Größe wird nicht betrachtet.

Das Unternehmen wird vom Geschäftsführer (CEO, Chief Executive Officer) geleitet, der die Kundenaufträge annimmt und das Firmenziel Gewinnmaximierung verfolgt. Der Gewinn ergibt sich aus Einkünften durch erfolgreich abgewickelte Projekte und den Kosten für Personal und Weiterbildung (Bücher, Schulungen, Hard- und Software, ...). Dem CEO untergeordnet sind eine Reihe von Projektleitern, die jeweils ein Projekt durchführen und versuchen dies möglichst schnell und erfolgreich abzuwickeln. Sie stellen ein Projektteam zusammen, das sie aus einem Pool von Mitarbeitern nach deren Qualifikation und freier Kapazität auswählen können. Jeder Projektmitarbeiter verfügt über Wissen in mehreren Bereichen, die sich im Laufe der Simulation durch Lernen, Üben und Vergessen verändern. Abhängig von diesem Wissen und den Anforderungen, die der Kundenauftrag vorgibt wird das Projekt schneller oder langsamer fertig gestellt. Eine Übersicht über die Zusammenhänge zwischen den Faktoren ist in Abbildung 5-2 dargestellt.

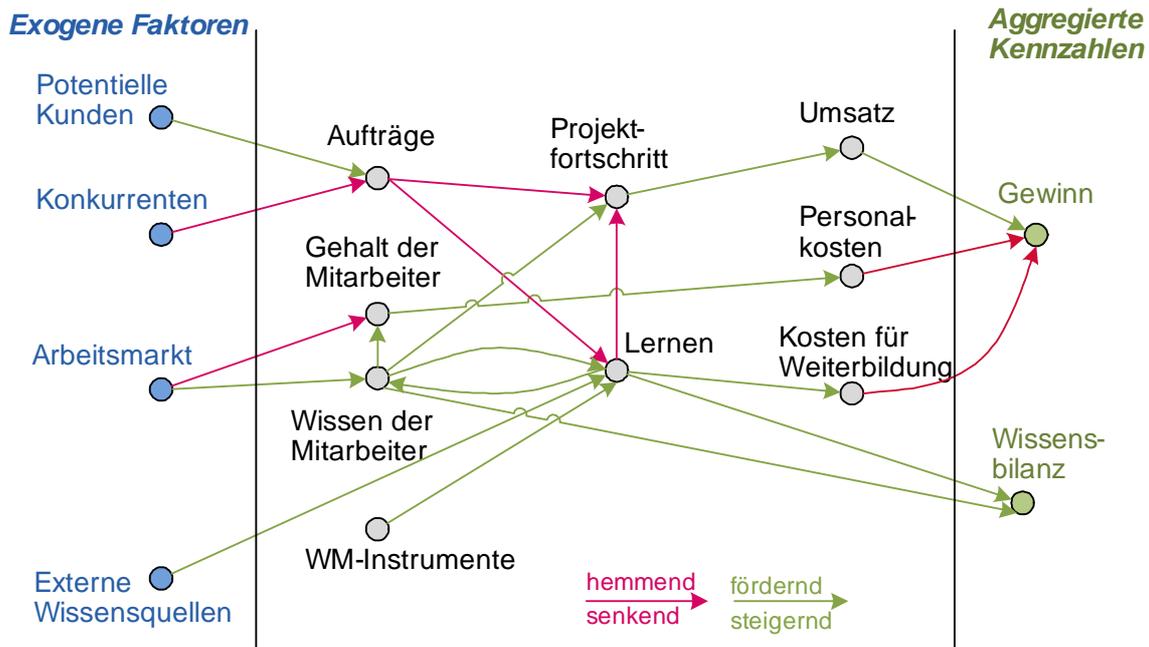


Abbildung 5-2: Wirkungszusammenhänge in SimKnowledge (eigene Darstellung)

Die grünen Pfeile in der Abbildung symbolisieren dabei einen positiven Zusammenhang, die roten Pfeile einen negativen. Einige der Zusammenhänge sind auf den ersten Blick nicht einleuchtend und sollen daher kurz erläutert werden. Der Arbeitsmarkt symbolisiert die Verfügbarkeit von qualifizierten Arbeitskräften auf dem Markt. Daher wirkt eine hohe Verfügbarkeit solcher Arbeitskräfte senkend auf das Gehalt der Mitarbeiter, da eine schlechte Verhandlungsbasis für die Mitarbeiter existiert und leicht Ersatz beschafft werden kann. Umgekehrt wirkt dies positiv auf das Wissen der Mitarbeiter, da neue Fachkräfte angeworben werden können, die Wissen mitbringen und es weitergeben können. Der negative Zusammenhang zwischen Aufträgen und Projektfortschritt erklärt sich durch die Ressourcenbeschränkung in Bezug auf Zeit und qualifizierte Mitarbeiter. Je mehr Aufträge parallel bearbeitet werden desto weniger Personal mit dem erforderlichen Wissen steht zur Verfügung. Daher wirken zusätzliche Aufträge hemmend auf den Projektfortschritt. Auf den ersten Blick nicht einsichtig ist auch der negative Zusammenhang zwischen Lernen und Projektfortschritt. Auch er erklärt sich durch die beschränkt verfügbare Arbeitszeit, die entweder für die Projektarbeit oder für Lernen verwendet werden kann. Der positive Zusammenhang, dass durch Lernen mehr Wissen zur Verfügung steht und dadurch schneller Fortschritt in der Projektarbeit erzielt wird ist über den Zwischenschritt des Faktors „Wissen der Mitarbeiter“ eingetragen.

## **5.3 Beschreibung nach der MASSIVE Methode**

### **5.3.1 Aufgabensicht**

Die Aufgabensicht wird für die Anforderungsanalyse verwendet. Hier werden die funktionalen und nicht-funktionalen Anforderungen definiert, auf deren Basis das Simulationsmodell und –programm entwickelt wurden.

Aus funktionaler Sicht soll in erster Linie ein mittelständischer IT-Dienstleister abgebildet werden, der Softwareentwicklungs- und Systemintegrationsprojekte für seine Kunden abwickelt. Das Modell muss dabei so gestaltet sein, dass es die Analyse der Wissensweitergabe zwischen Mitarbeitern erlaubt. Um die Wissensweitergabe im organisatorischen Kontext zu untersuchen, soll es typische Prozesse bei der Erstellung von IT-Systemen abbilden. Dies beinhaltet die Phasen der Softwareentwicklung mit Analyse, Entwurf, Implementierung, Test und Roll-Out (siehe Abschnitt 3.2.2). Die Rollentrennung zwischen den verschiedenen Phasen sowie zwischen Projektmanagement und Entwicklern soll ebenfalls abgebildet werden. Die dafür benötigten Wissensgebiete müssen ebenfalls abgebildet werden.

Wichtig für ein Simulationssystem ist weiterhin, dass alle interessierenden Variablen einfach auszuwerten sind. Dafür sollen standardisierte Schnittstellen zur Verfügung gestellt werden, die den Transfer der Daten zur Auswertung in marktüblichen Softwaresystemen ermöglicht (z.B. Excel, SPSS). Um diese Aufgabe zu vereinfachen, sollen die Daten in einem relationalen Datenbankmanagementsystem (DBMS) gespeichert werden. Darauf kann dann entweder direkt (z.B. über die ODBC-Schnittstelle) zugegriffen werden, oder Daten im CSV (Comma Separated Values) Format exportiert werden. Über SQL-Abfragen sind zudem einfachere Auswertungen direkt möglich. Darüber hinaus soll es jederzeit möglich sein, wichtige Daten im laufenden System über die grafische Benutzeroberfläche auslesen zu können.

An nicht-funktionalen Anforderungen ist in erster Linie die Performance zu nennen. Um eine Reihe von Simulationsläufen in begrenzter Zeit durchführen zu können, soll ein Lauf, in dem ein Zeitraum von 5 Jahren mit maximaler Agentenanzahl simuliert wird auf jeden Fall „über Nacht“ durchlaufen können, also maximal 12-14 Stunden dauern, so dass das Experiment am Abend vorbereitet und am nächsten Morgen ausgewertet werden kann. Zur Parallelisierung wird es als ausreichend erachtet, wenn ein Simulationslauf auf einem Rechner läuft und die Parallelisierung durch das Durchführen verschiedener Läufe auf

verschiedenen Rechnern erreicht wird. Eine Optimierung auf Cluster oder Mehrprozessormaschinen erscheint nicht nötig. Im Laufe der Entwicklung hat sich jedoch herausgestellt, dass dieses Ziel problemlos erreicht wird und ein Simulationslauf nur wenige Minuten pro Simulationsjahr dauert.

### 5.3.2 Umgebungssicht

In der Umgebungssicht soll der organisatorische Kontext und der technische Kontext beschrieben werden, in dem sich das Simulationssystem bewegt.

**Organisatorischer Kontext:** Simuliert werden soll ein mittelständisches IT-Dienstleistungsunternehmen, das Softwareentwicklungsprojekte als Auftragsarbeit für Kunden erledigt. Die Abbildung eines mittelständischen Unternehmens erfordert, dass die notwendige Anzahl von Mitarbeitern simuliert werden können. Das Institut für Mittelstandsforschung in Bonn definiert mittelständische Unternehmen in Deutschland als solche, die 10-499 Mitarbeiter haben und 1 bis unter 50 Mio. Euro Umsatz erwirtschaften<sup>38</sup>. Es müssen also bis zu 500 Agenten simuliert werden können. Das abzubildende System ist dabei das Firmengebäude. Als exogene Faktoren dienen dabei potentielle Kunden, die im Zeitverlauf immer wieder neue Projekte generieren, welche entweder an die simulierte Firma oder an einen der Mitbewerber vergeben werden. Dabei sind Auftragsumfang, Preis und Fertigstellungstermin fest vorgegebene Größen. Als dritte exogene Größe dient ein Markt mit Wissensquellen, der Weiterbildungsangebote und Bücher zu den abgebildeten Themen vorhält. Das Angebot kann im Umfang und Preis variiert werden. Auf eine Abbildung des Finanzmarktes kann nach Einschätzung des Autors verzichtet werden. Es wäre aber problemlos möglich, einen entsprechenden Mechanismus mit Verzinsung in das Modell zu integrieren.

Zur genaueren Beschreibung soll auf die Dimensionen der Umgebung zurückgegriffen werden, die von Ferber definiert wurden (Ferber 2001). Die simulierte Umgebung ist unzugänglich, d.h. jeder Mitarbeiter kann nur seine unmittelbare Umgebung direkt wahrnehmen und hat keinen Zugriff auf das Gesamtsystem. Das System ist weiterhin indeterministisch, da die Umweltzustände durch Agenten manipuliert werden können. Die Umgebung ist nicht episodisch, da die Aktionen und Reaktionen der Agenten von der gesamten Historie abhängen und nicht nur die Parameter der aktuellen Episode berücksichtigen. Die Umgebung ist statisch, da sie ausschließlich durch Systemaktivitäten

---

<sup>38</sup> <http://www.ifm-bonn.org/index.htm?dienste/definition.htm>, letzter Abruf 25.08.2007

(Aktionen der Agenten) verändert werden kann und sie ist diskret, da sie nur eine endliche Menge an Zuständen annehmen kann. Eine räumliche Abbildung der Umgebung ist erforderlich, die zweidimensionale Darstellung wird dabei als ausreichend erachtet.

***Laufzeitumgebung des Zielsystems:*** Das Simulationssystem soll auf einem handelsüblichen PC lauffähig sein und keinerlei kommerzielle Software zur Nutzung erfordern. Eine Verteilbarkeit des Systems auf mehrere Prozessorkerne, Prozessoren oder gar Maschinen zur Beschleunigung der Ausführung wäre zwar generell wünschenswert, stellt jedoch angesichts des Zusatzaufwands speziell bei verteilter Ausführung auf mehreren Rechnern und den oben dargestellten Performanceüberlegungen kein Musskriterium dar. Jeder Agent ist als eigenständiger Thread zu modellieren, um die Eigenständigkeit und Autonomie der Agenten zu gewährleisten. Die Ausführung erfolgt rundenbasiert. Kein Agent sollte bei der Zuteilung von Ausführungszeit bevorzugt berücksichtigt werden.

Die Simulationsläufe wurden auf einem PC mit AMD Athlon 64 3200+ (2.0 GHz) mit einem GB Hauptspeicher bzw. einem Notebook mit Intel Core Duo 2400 (1.83 GHz) und 2 GB Hauptspeicher durchgeführt. Als Betriebssystem kam in beiden Fällen Microsoft Windows XP mit Servicepack 2 zum Einsatz. Als Java Virtual Machine (JVM) wurde die Laufzeitumgebung Java Standard Edition von Sun in der Version 1.5.0\_09-b03 eingesetzt. Tests auf anderen Plattformen wurden nicht durchgeführt, es sollte aber problemlos möglich sein, das Simulationssystem auf Linux oder anderen Betriebssystemen laufen zu lassen, für die es eine Java Virtual Machine gibt.

### 5.3.3 Rollensicht

In der Rollensicht werden die verschiedenen Rollen dargestellt, in denen die Agenten agieren können. Das vorliegende Modell ist dabei insofern einfach gehalten, als die Rollenzuteilung statisch ist. Jeder Agent hat genau eine Rolle, die sich im Zeitverlauf nicht ändert. Es gibt drei Rollen, die jeweils von einem eigenen Agententypen ausgefüllt werden.

- **CEO:** Der Geschäftsführer des Unternehmens (Chief Executive Officer) ist zuständig für die Kommunikation mit den Kunden, Projekte anzunehmen oder abzulehnen und für abgeschlossene Projekte die Bezahlung entgegenzunehmen. Die Maximierung des Unternehmensgewinns ist sein Hauptziel.
- **PM:** Projektmanager sind für die reibungslose Abwicklung von Projekten verantwortlich. Sie stellen Projektteams zusammen, organisieren Teambe-

sprechungen und melden abgeschlossene Projekte an den CEO. Sie versuchen die ihnen zugewiesenen Projekte so schnell wie möglich zum Abschluss zu bringen.

- **Worker:** Projektmitarbeiter sind für das Abarbeiten der Projektarbeit zuständig. Sie haben das Ziel ihren Nutzen aus Einkommen und die Freizeit zu maximieren.

Die drei Rollen stehen in hierarchischer Beziehung zueinander

### 5.3.4 Interaktionssicht

In der Interaktionssicht werden die Abhängigkeiten zwischen den Agenten in den Rollen dargestellt. MASSIVE unterscheidet dabei die Zweckebene der Interaktion sowie die Protokoll- und die Transportebene.

Auf der Zweckebene ist festzustellen, dass es in erster Linie hierarchische Kommunikation gibt. Sie ist zur Abbildung des Projektablaufs nötig. Der CEO kommuniziert mit den Projektmanagern um Projekte zu verteilen und fertig gestellte Projekte wieder einzusammeln. Ähnlich ist die Kommunikation zwischen Projektmanagern und Projektmitarbeitern. Dort werden Arbeitspakete an Mitarbeiter verteilt und die Fertigstellung oder Probleme mit dem Arbeitspaket zurückgemeldet.

Darüber hinaus gibt es aber auch informelle Kommunikation, die zum Wissensaustausch dient. Diese findet in erster Linie zwischen Projektmitarbeitern auf der gleichen Ebene statt. Es können aber auch Projektmanager befragt werden, um einen geeigneten Kollegen ausfindig zu machen. Projektmanager fungieren damit als Vermittler und geben Metawissen weiter.

Agenten sind grundsätzlich egoistisch eingestellt, verhalten sich andererseits aber im Rahmen der Projektarbeit aus egoistischen Gründen kooperativ, um ihren Arbeitsplatz zu behalten und weiterhin Einkommen zu erwirtschaften.

Auf der Protokollebene wird die Kommunikation in Sprechakten umgesetzt. Diese sind nach der Sprechakttheorie gestaltet und orientieren sich an KQML (siehe Abschnitt 4.2.4). Der semantische Gehalt eines Sprechakts ergibt sich dabei aus dem Performative (z.B. *achieve, ask, ready*), dem übermittelten Inhalt (z.B. *work package, skill, agent*) und der Stellung des Sprechenden (CEO, PM oder Worker). Tabelle 5-2 listet die möglichen Sprechakte und ihre jeweilige Bedeutung.

Tabelle 5-2: Sprechakte in SimKnowledge

<b>Sprechakt</b>	<b>Semantische Bedeutung</b>
PM.offer(work package, team)	Ein Projektmanager bittet einen Worker ein Arbeitspaket zu übernehmen und Mitglied im Team zu werden.
PM.ask(work package)	Ein Projektmanager fragt einen Projektmitarbeiter nach dem Status des Arbeitspaktes.
ready(work package)	Ein Projektmitarbeiter teilt einem Projektmanager mit, dass er mit der Arbeit an seinem Arbeitspaket fertig ist.
deny(work package, skill)	Negative Antwort auf PM.achieve. Aus der Höhe des Skills lässt sich ablesen, ob wegen mangelndem Wissen oder belegter Kapazität abgelehnt wurde.
advertise(skill)	Ein Projektmitarbeiter erzählt einem Kollegen oder Projektmanager, dass er über eine gewisse Fähigkeit verfügt.
advertise(free time)	Ein Projektmitarbeiter erzählt, dass er schon seit einiger Zeit kein Arbeitspaket mehr hatte und an einem Projekt mitarbeiten will.
ask(skill)	Ein Projektmitarbeiter fragt nach Informationen in einem Wissensgebiet, um etwas dazu zu lernen.
ask(work package)	Ein Projektmitarbeiter fragt nach Informationen in dem Wissensgebiet, das er für sein aktuelles Arbeitspaket benötigt, um etwas dazu zu lernen.
recommend (skill, agent)	Ein Projektmanager oder –mitarbeiter empfiehlt einen anderen Agenten, der über eine bestimmte Fertigkeit verfügt.
sorry(skill) sorry(work package)	Negative Antwort auf die entsprechenden ask Sprechakte der Mitarbeiter
sorry(work package)	Ein Projektmitarbeiter teilt seinem Projektleiter mit, dass er mit dem Arbeitspaket überfordert ist und gibt es zurück.
achieve(work package)	Einen Projektmitarbeiter um Hilfe bei der Bearbeitung eines Arbeitspakets fragen und die positive Antwort auf diese Frage
tell(name, coordinates)	Ein Agent stellt sich vor.

Auf der Transportebene ist jeder Sprechakt als Java Nachrichtenobjekt abgebildet. Der Agent schickt den Sprechakt als Nachricht an die Umgebung. Die Umgebung dient als Transportkanal für Nachrichten und übermittelt sie an alle Empfänger, die in der Nähe des Senders sind. Dadurch bekommt sowohl der eigentliche Adressat die Nachricht als auch andere Agenten in der Umgebung. Somit wird es möglich, dass Nachrichten mitgehört werden. Dies ist ein Weg um informellen Wissensaustausch zu simulieren.

### 5.3.5 Gesellschaftssicht

In der Gesellschaftssicht werden die Agenten aus der Makroperspektive beschrieben. Es werden Gruppenbildungen und andere Makrophänomene dargestellt. Die entwickelte Simulationsplattform ist eine geschlossene Systemumgebung. Es können keine unbekanntenen Agenten von außen in das System eindringen.

Die Kommunikationsstruktur ist wie schon dargestellt zum einen Teil flach, da jeder Agent mit jedem anderen Agenten reden kann. Andererseits sind viele Interaktionen formalisiert und hierarchisch. Die Konsistenz der Gesellschaft ist heterogen. Es gibt analog zu den drei Rollen auch drei verschiedene Agententypen mit unterschiedlichen internen Architekturen.

Die Struktur ist bezogen auf die Hierarchieebenen statisch, es ist kein Aufstieg von Workern zu PMs oder von PMs zum CEO möglich. Andererseits werden Projektteams dynamisch zusammengestellt, so dass sich wechselnde Interaktionsstrukturen ergeben.

Durch die geschilderten Eigenschaften ergeben sich soziale Netzwerke, die sich v.a. durch räumliche Nähe und gemeinsame Projekte ergeben. Ein Beispiel so eines Netzwerkes ist in Abbildung 5-3<sup>39</sup> zu sehen.

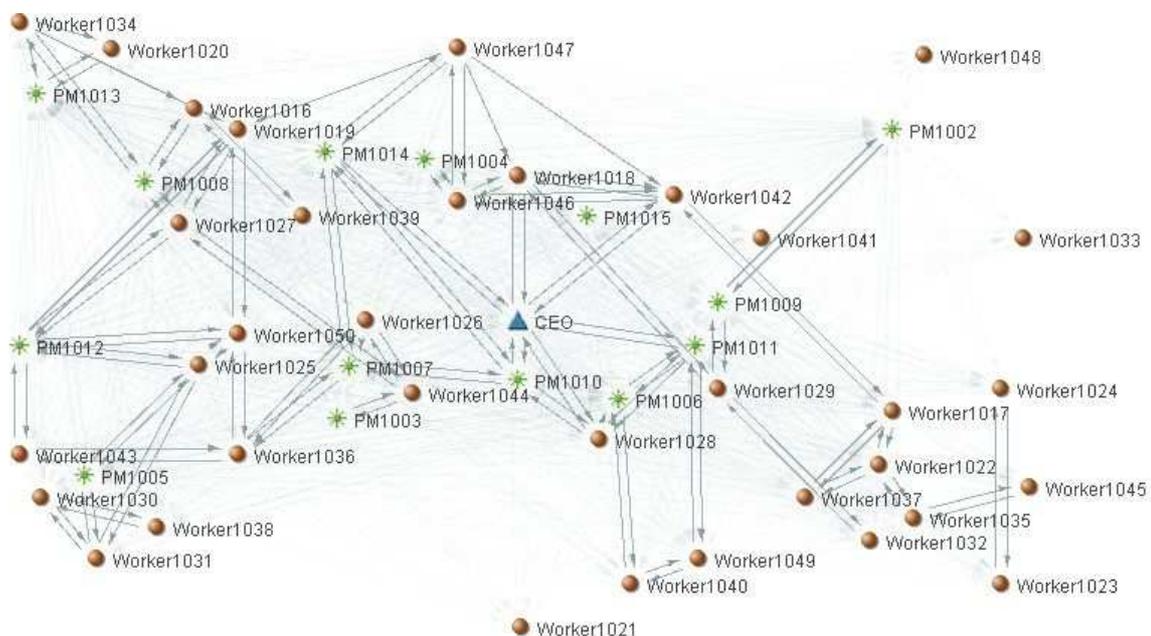


Abbildung 5-3: Soziales Netzwerk in SimKnowledge generiert mit AGNA (eigene Darstellung)

Die meisten Agenten stehen mit 1-4 anderen Agenten in häufigem Kontakt, symbolisiert durch eine dickere Verbindungslinie. Die meisten dieser Verbindungen sind symmetrisch,

<sup>39</sup> Das Netzwerk besteht aus 49 Knoten und 1411 Kanten. Die Netzwerkdichte beträgt 0,5759 (gewichtet 323,1657). Die Kohäsion beträgt 0,5493.

es besteht also Kontakt in beide Richtungen. Es gibt aber am Rande des simulierten Gebäudes auch einzelne Projektmitarbeiter (z.B. Nr. 1021, 1033 und 1048), die kaum in das Kommunikationsnetz eingebunden sind, was durch die dünnen grauen Linien symbolisiert wird.

### 5.3.6 Architektursicht

In der Architektursicht wird sowohl die Gesamtarchitektur des Simulationssystems als auch die Agentenarchitektur beschrieben.

**Gesamtarchitektur:** Laut MASSIVE Methode sind auf der Ebene der Gesamtarchitektur die folgenden Aspekte zu definieren.

- Entitäten im System
- Kontroll- und Informationsfluss zwischen den Entitäten
- Agentenmanagement (Scheduling, Remixing)
- Kommunikationsmodell
- Datenbanken
- Externe Komponenten

Die Entitäten im System sind entweder aktive und passive Entitäten. Die aktiven Entitäten sind die beschriebenen Agententypen. Die passiven Entitäten werden als Objekte bezeichnet und sind die simulierten Projekte und Arbeitspakete sowie die IT-Systeme, die Zugriff auf die Wissensmanagement-Maßnahmen bieten. Ursprünglich war es geplant, auch Bücher und Telefon als eigene Objekte zu realisieren. Dies wurde verworfen, da der PC beide Funktionen übernehmen kann und dies in der Praxis teilweise auch schon tut (z.B. E-Book, VoIP und Medienkonvergenz). Deswegen wurde sowohl der Zugriff auf dokumentiertes Wissen, wie auch die Kommunikation über elektronische Medien auf dieses eine Objekt beschränkt, um die Implementierung nicht unnötig komplex zu machen.

Der Informationsfluss wurde bereits weiter oben dargestellt. An dieser Stelle soll daher auf die Abläufe aus Sicht eines Projekts und aus Sicht eines Mitarbeiters eingegangen werden.

Der Ablauf aus Sicht des Projekts stellt sich wie folgt dar

1. Ein Auftrag wird von dem Kundenobjekt generiert und dem CEO der Firma angeboten.
2. Der CEO entscheidet über die Annahme.
3. Der Auftrag bestimmt Preis, Termin, Arbeitsvolumen und das benötigte Wissen der Projektmitarbeiter.

4. Der CEO bestimmt einen Projektmanager (PM) für die Durchführung.
5. Der PM sucht Mitarbeiter mit dem geforderten Wissen und freier Kapazität.
6. Die Mitarbeiter bearbeiten den Auftrag.
7. Reicht das Wissen nicht aus, muss der Mitarbeiter sich das fehlende Wissen beschaffen (Kollegen fragen, im Intranet nachschlagen, eine Schulung besuchen, ...).
8. Wissen und Engagement der Mitarbeiter sowie die benötigte Zeit für Weiterbildung bestimmen, wann der Auftrag fertig gestellt wird.
9. Nach Fertigstellung wird der vereinbarte Preis ausgezahlt (evtl. abzüglich Vertragsstrafe für Verspätung).
10. Einnahmen durch fertig gestellte Aufträge pro Jahr abzüglich der Personalkosten und Kosten für Weiterbildung ergeben den Unternehmensgewinn.  
(sonstige Aufwendungen werden nicht berücksichtigt)
11. Mitarbeiter versuchen entsprechend ihrer Skills und der Arbeitsmarktlage entlohnt zu werden.

Dieser Ablauf ist in Abbildung 5-4 auch noch einmal grafisch als UML-Ablaufdiagramm dargestellt. Das `customers` Objekt vom Typ `SalesMarket` bildet alle Kunden ab. Auf die Modellierung einzelner Kunden als Agenten wurde verzichtet, da der Fokus auf der Analyse der `Worker` liegt. Die dargestellten Methoden entsprechen nicht vollständig den internen Aktionen, da diese umfassender sind und Ausnahmen berücksichtigen müssen, wogegen der dargestellte Ablauf idealisiert und vereinfacht ist.

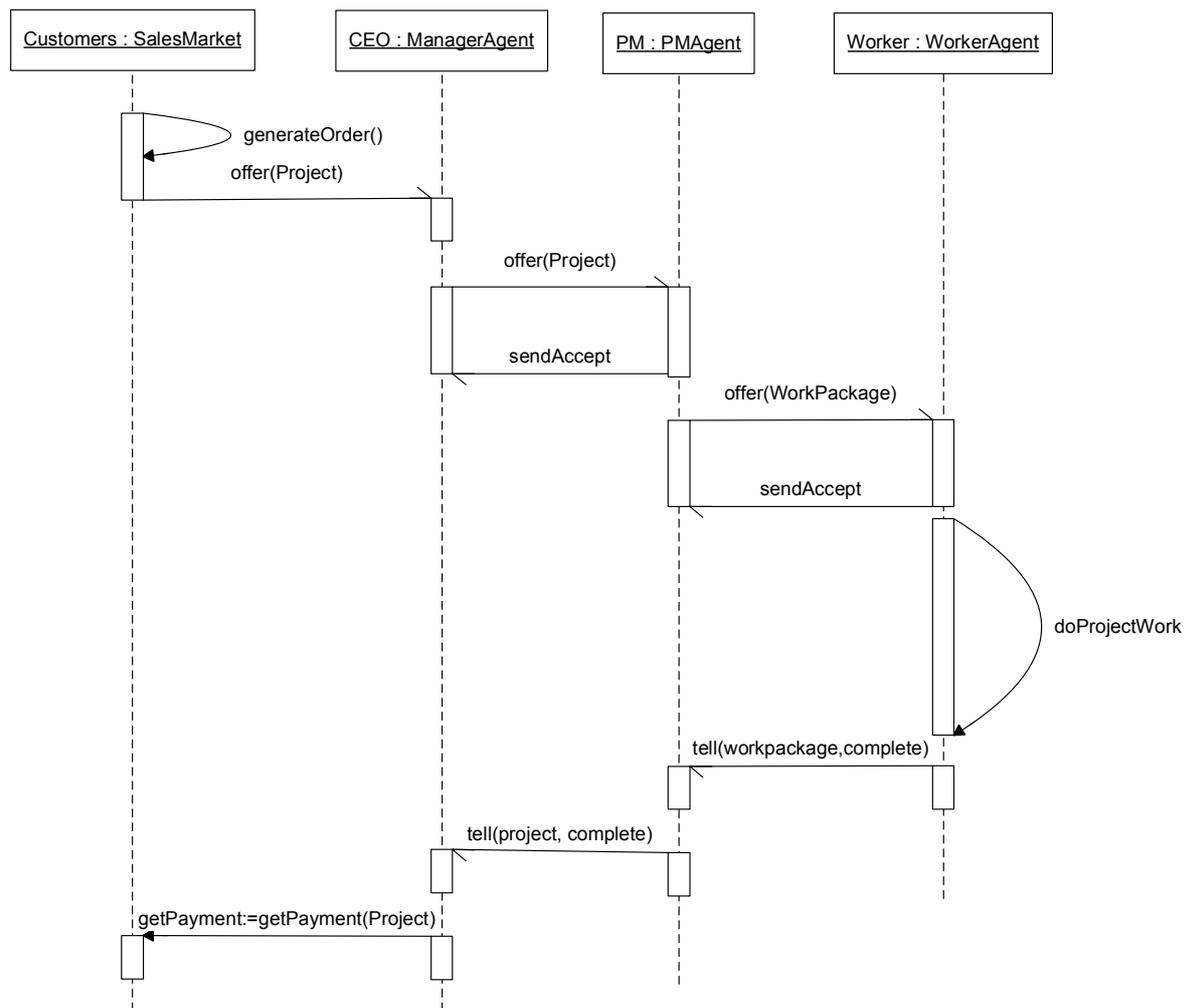


Abbildung 5-4: Ablauf aus Sicht eines Projekts (eigene Darstellung)

Aus Sicht eines Projektmitarbeiters ist der Tagesablauf dagegen folgendermaßen (siehe auch Abbildung 5-5).

1. Der Mitarbeiter kommt morgens in die Firma
2. Ist er in einem Projekt involviert, so arbeitet er vorrangig an seinem Arbeitspaket weiter
3. Reicht sein Wissen nicht aus, so versucht er sich weiter zu bilden
4. Scheitert er dauerhaft bei der Bearbeitung seines Arbeitspakets (8 Zyklen aus Projektarbeit und Lernen ohne Erfolg), so gibt er sein Arbeitspaket zur Bearbeitung durch einen anderen Kollegen zurück an den PM
5. Durch den Einsatz seines Wissens im Projekt oder Weiterbildungsmaßnahmen steigt das Wissen des Mitarbeiters, wird es nicht benutzt so vergisst er langsam wieder

6. Ist er in keinem Projekt oder nicht motiviert für die Projektarbeit, so kann er private Sachen erledigen (aus Sicht der Firma: `idle`), oder sich freiwillig in einem Thema weiterbilden (`learn from PC`)
7. Der Mitarbeiter versucht seinen Nutzen  $U = \text{Einkommen} + \text{Freizeit}$  zu maximieren.
8. Abends verlässt der Agent das Firmengebäude wieder und kann Freizeitaktivitäten nachgehen (derzeit nur `idle` und `sleep` implementiert).

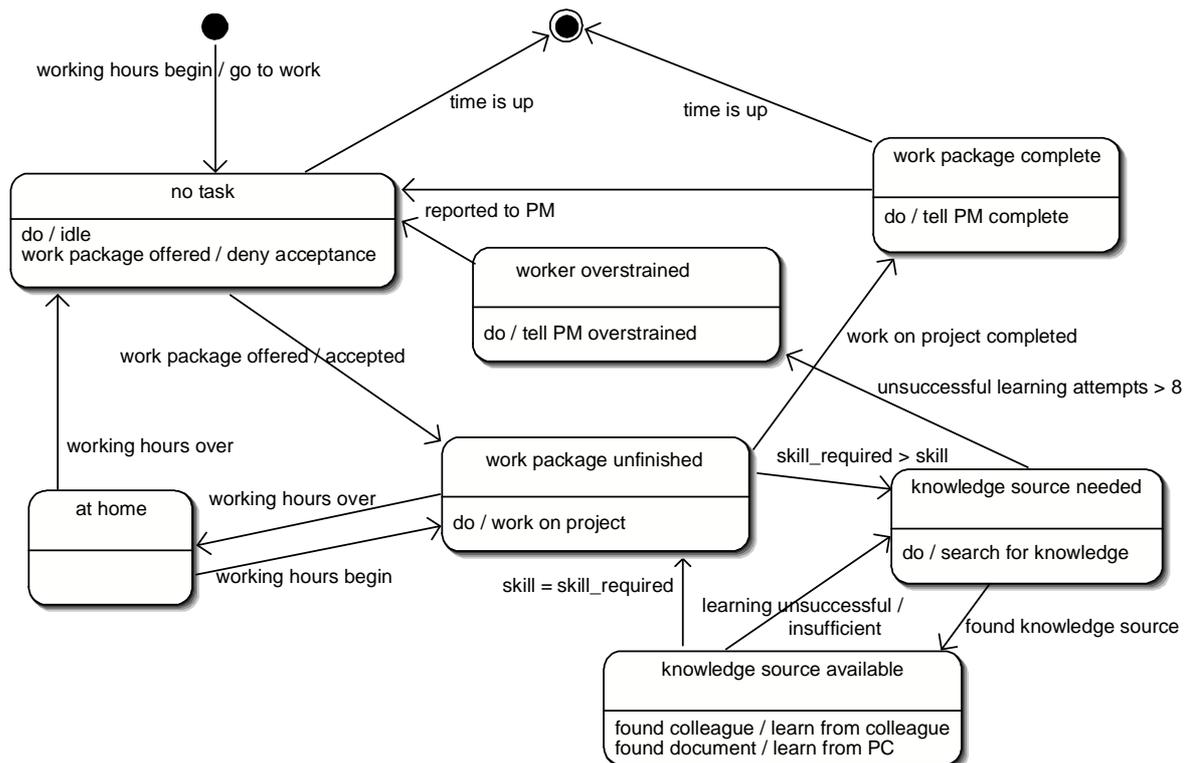


Abbildung 5-5: Zustandsübergangsdiagramm für WorkerAgent (eigene Darstellung)

Der Kontrollfluss ist nötig, um den rundenbasierten Ablauf zu gewährleisten. Dazu wird in jeder Runde jedem Agenten die Möglichkeit gegeben neue Eindrücke von der Umgebung wahrzunehmen (`sense`), mit diesen neuen Informationen die nächste Aktion zu planen (`reason`) und die geplante Aktion auszuführen (`act`). Die Aktionsplanung besteht dabei aus der Aktualisierung der eigenen Wissensbasis mit den neuen Umgebungsdaten (`processBeliefs`), der Situationserkennung (`recognizeSituation`) und der Planung im engeren Sinne (`doPlanning`). Um diesen Rundenablauf zu koordinieren, ohne den Agenten die Autonomie zu beschneiden werden Koordinationspunkte definiert. Jeder Agent gibt nach Ausführen einer Methode (`sense`, `reason`, `act`) die Kontrolle „freiwillig“ an die Simulationskontrollereinheit (`SimulationEngine`) zurück. Erst wenn alle Agenten sich zurück gemeldet haben, wird die nächste Phase der Runde begonnen.

Um Chancengleichheit zu gewährleisten, wird die Reihenfolge, in der die Agententhreads gestartet werden nach jeder Runde zufällig neu ermittelt.

Das Kommunikationsmodell wurde bereits an anderer Stelle beschrieben und soll hier aus Gründen der Redundanz nicht noch einmal dargestellt werden.

Als Datenbank soll ein leichtgewichtiges eingebettetes DBMS verwendet werden, für das ein JDBC Treiber existiert. Dabei besteht am Markt eine große Auswahl an Systemen. Das Repertoire reicht dabei von minimalistischen Systemen wie SQLite<sup>40</sup>, die nur die Grundfunktionen abdecken, bis hin zu für den Unternehmenseinsatz entwickelten Systemen wie dem von IBM entwickelten Cloudscape (mittlerweile Apache Derby<sup>41</sup>). Weitere Kandidaten sind HSQLDB<sup>42</sup>, das auch als Basis für die Datenbankkomponente von OpenOffice Verwendung findet, und Firebird<sup>43</sup>, welches auch als Datenbank Backend für Unternehmensanwendungen dienen kann. Passend zu dem in Java implementierten Simulationssystem wurde mit HSQLDB ein vollständig in Java geschriebenes DBMS ausgesucht. Dabei gaben zum einen der geringe Speicherverbrauch und zum anderen die Unterstützung in OpenOffice den Ausschlag. Dadurch wurde erwartet ein leicht zu bedienendes und komfortables Frontend zu haben, so dass z.B. die Parameter für die einzelnen Experimente über eine Maske einfacher eingegeben werden können. Dies hat allerdings nicht so gut wie erwartet funktioniert. Trotz zahlreicher Verbesserungen und Fehlerbereinigungen in den Versionen 2.0.1 bis 2.0.4 wurden erst in der Ende 2006 veröffentlichten Version 2.1 die größten Probleme des Base Moduls von OpenOffice gelöst<sup>44</sup>. Nichts desto trotz funktioniert HSQLDB als Datenbankbackend problemlos. Das mitgelieferte Werkzeug zur Verwaltung ist recht rudimentär, so dass z.B. sämtliche Interaktionen mit dem DBMS als SQL Befehle formuliert werden müssen. Als Ersatz wurde daher die ebenfalls freie grafische Oberfläche SquirrelSQL<sup>45</sup> verwendet, da sie wesentlich höheren Komfort bietet und auch direktes Editieren von einzelnen Werten erlaubt. HSQLDB kann sowohl im Server-Modus betrieben werden, wobei das DBMS als

---

<sup>40</sup> <http://www.sqlite.org/>, letzter Abruf am 26.12.2007

<sup>41</sup> <http://db.apache.org/derby/>, letzter Abruf am 26.12.2007

<sup>42</sup> <http://www.hsqldb.org/>, letzter Abruf am 26.12.2007

<sup>43</sup> <http://www.firebirdsql.org/>, letzter Abruf am 26.12.2007

<sup>44</sup> [http://development.openoffice.org/releases/OpenOffice\\_org\\_2\\_x.html](http://development.openoffice.org/releases/OpenOffice_org_2_x.html), letzter Abruf am 26.12.2007

<sup>45</sup> <http://sourceforge.net/projects/squirrel-sql/>, letzter Abruf am 26.12.2007

eigenständiger Prozess läuft und die Kommunikation über TCP/IP abgewickelt wird, als auch eingebettet im Prozess des aufrufenden Programms gestartet werden<sup>46</sup>.

Abbildung 5-6 zeigt das verwendete Datenmodell in Information Engineering Notation (siehe z.B. Halpin 2001, S. 327). Im linken Teil des Modells sind die Simulationsexperimente abgebildet, die zu Studien zusammengefasst werden, für die Parameter festgelegt und mehrere Simulationläufe durchgeführt werden können. Pro Experiment gibt es eine „Population“ an Agenten, die dann in einem Lauf instanziiert und mit Skills, Statusvariablen und ihrem Gedächtnis über andere Agenten gespeichert werden.

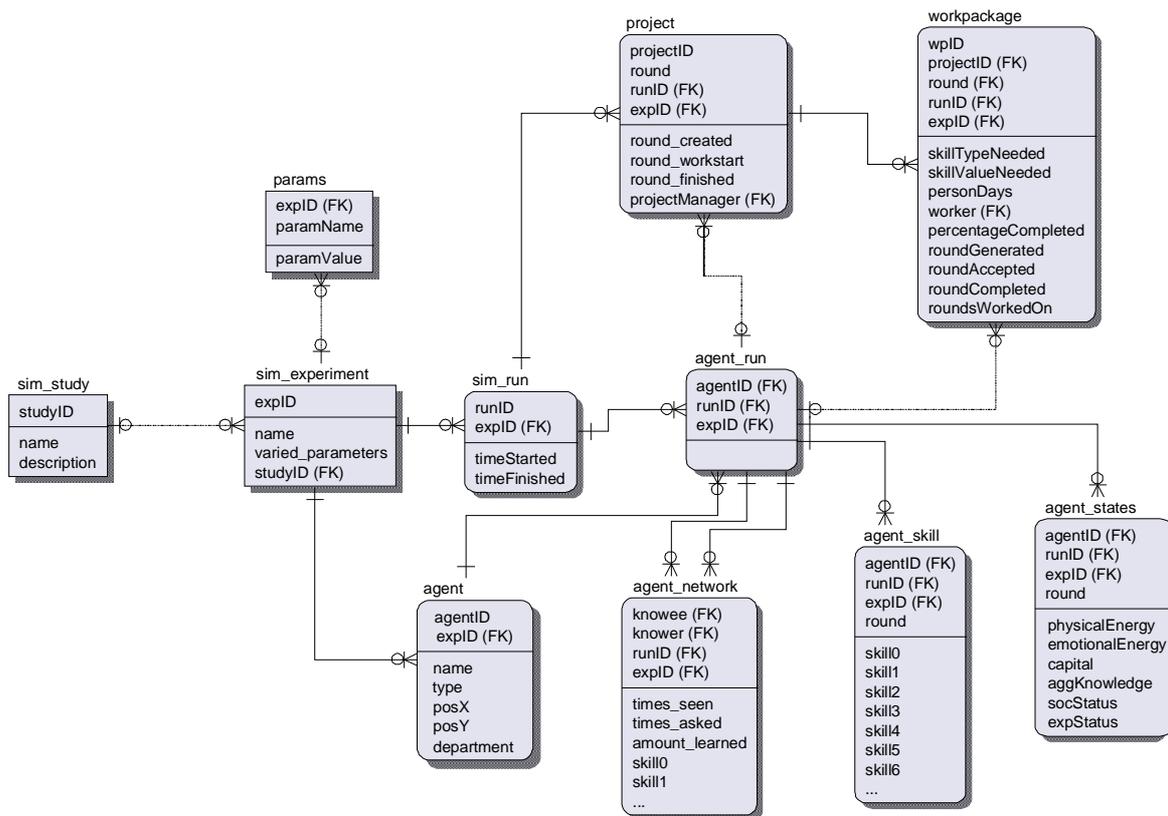


Abbildung 5-6: Datenmodell von SimKnowledge (eigene Darstellung)

Dabei wurde zugunsten einer einfacheren Auswertbarkeit und leichter Handhabung auf eine vollständige Normalisierung bis zur dritten Normalform verzichtet und stattdessen alle Skills eines Agenten zusammen in einem Datensatz der Tabelle `agent_skill` gespeichert. Der Nachteil dieser Lösung ist, dass maximal die vorgesehenen 16 Skills gespeichert werden können und im Falle von Experimenten mit weniger als 16 Skills eine Reihe von Nullwerten auftreten. Schließlich finden sich in der rechten oberen Ecke des Modells die Referenz auf die bearbeiteten Projekte und Arbeitspakete.

<sup>46</sup> <http://www.hsqldb.org/doc/guide/ch01.html>, letzter Abruf am 26.12.2007

Als externe Komponenten sind neben dem DBMS verschiedene Auswertungsprogramme zu nennen. Für die statistischen Analysen wurden Excel 2003 und SPSS 11 verwendet<sup>47</sup>. Für die soziale Netzwerkanalyse wurde auf das Open Source Werkzeug AGNA (Applied Graph & Network Analysis) in Version 2.1 zurückgegriffen<sup>48</sup>. Es bietet eine Reihe von textuellen Auswertungsmöglichkeiten, zusätzlich auch eine grafische Darstellung des Netzwerks und ein textbasiertes Dateiformat, das einfach zu erzeugen ist und neben den Daten über das soziale Netzwerk selbst auch Daten über die grafische Position und Darstellung enthalten kann.

**Agentenarchitektur:** Die entwickelte Agentenarchitektur setzt auf den bereits vorgestellten Architekturen InterRaP (Fischer et al. 1994) und PECS (Urban 2000) auf. InterRaP setzt wiederum auf das bekannte BDI-Modell auf (Rao, Georgeff 1991). Erweitert wird die Architektur um nutzenbasierte Aktionsauswahl wie das im Utilitarian Desire (Lang et al. 2000) vorgeschlagen wurde. Im Folgenden wird beschrieben, wie genau die Synthese dieser unterschiedlichen Architekturen vollzogen wurde.

Die SimKnowledge Agentenarchitektur orientiert sich vorrangig an InterRaP. Von dort übernimmt es die Schichtenarchitektur. Andererseits muss der Objektorientierung Rechnung getragen werden. Deshalb wurden die Schichten CPL (cooperative planning layer) und LPL (local planning layer) zu einem Objekt zusammengefasst, da auf der Ebene LPL bereits die eine Spezialisierung in die drei Agententypen Manager, Projektleiter und Worker notwendig ist (siehe Abbildung 5-7). Dies hätte zu einem unnötigen Overhead durch zusätzliche Klassen geführt.

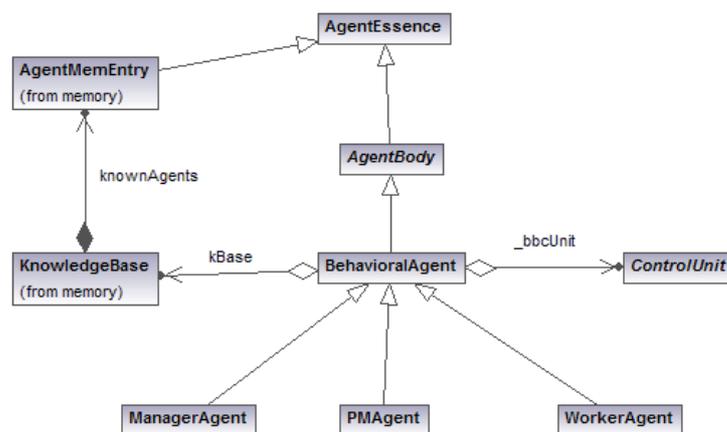


Abbildung 5-7: Klassenübersicht der Agentenklassenhierarchie (eigene Darstellung)

<sup>47</sup> <http://office.microsoft.com/de-de/excel> und <http://www.spss.com/de/spss/>, letzter Abruf am 26.12.2007

<sup>48</sup> <http://www.geocities.com/imbenta/agna/>, letzter Abruf am 26.12.2007

Eine zweite Änderung ist die zusätzliche Berücksichtigung der PECS Architektur. Die vier Bereiche lassen sich direkt den einzelnen Schichten zuordnen, also Physis dem AgentBody, Emotion dem Behavior-based Layer (BBL), Cognition dem LPL und Status dem CPL. Der Ablauf innerhalb eines Agenten ist ebenfalls stark an den Vorgaben von InterRaP orientiert (siehe Abbildung 5-8), bildet aber ebenso das Schema von PECS ab.

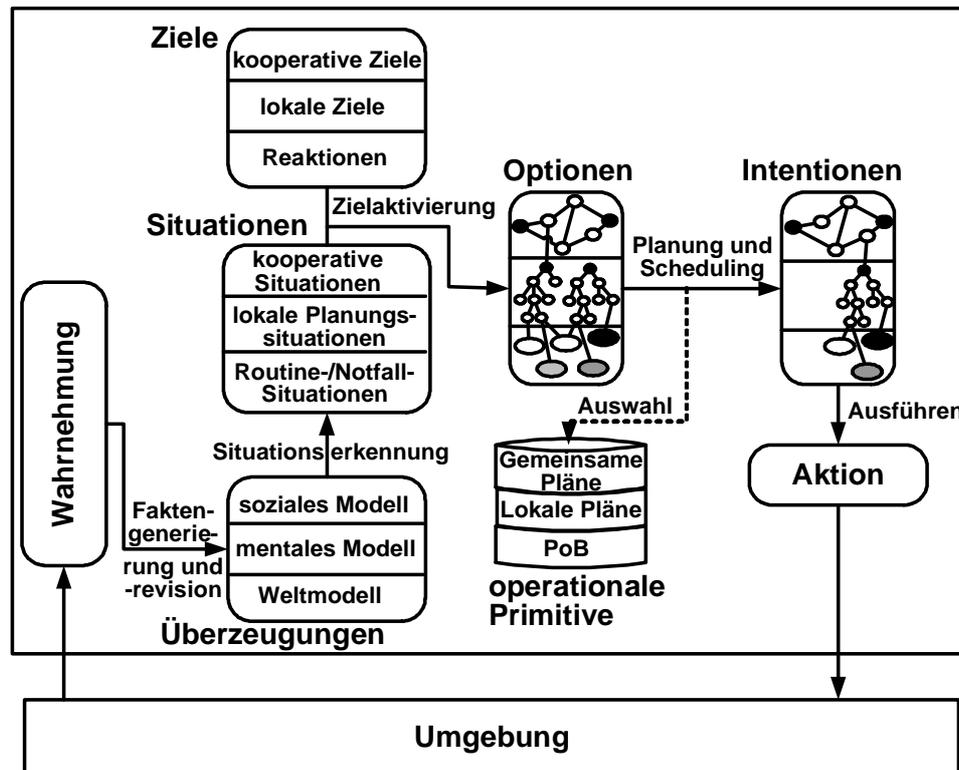


Abbildung 5-8: InterRaP Detailmodell (Fischer et al. 1994)

Die Wahrnehmungen gelangen über die Sensoren mittels der `perceive` Methode in den Agenten. Sie werden als visuelle und auditive Signale im Eingangspuffer abgelegt. Dort werden sie von der Methode `sense` erfasst, verarbeitet und führen gegebenenfalls zu einer Revision der Wissensbasis (Klasse `KnowledgeBase`), die bei SimKnowledge nicht so streng in Weltmodell, mentales Modell und soziales Modell getrennt ist, obgleich alle Teile vorhanden sind. Darauf aufbauend arbeitet die Situationserkennung für die Situation, in der sich der Agent gerade befindet. Als Ergebnis können eine oder mehrere Situationen erkannt worden sein. So könnte z.B. gleichzeitig ein Projekt als fertig gemeldet werden (`situation.completedProject`) und ein anderes vom CEO angeboten werden (`situation.newProject`). Dann muss über die Zielaktivierung priorisiert werden, wobei auch Pläne zum Umgang mit beiden Situationen gemacht werden können. Die Ziele entsprechen dabei dem Optimieren der Werte der Zustandsvariablen, also z.B. möglichst

hoch im Ansehen der Kollegen zu stehen, oder möglichst wenig erschöpft zu sein. Es handelt sich also um eine implizite Abbildung der Ziele. Für jede Situation existiert ein „Masterplan“ der aus Aktionen und Subplänen besteht. Jede Aktion und jeder Subplan kann entweder erfolgreich ausgeführt werden oder fehlschlagen und zudem kann es an manchen Stellen Alternativen für einen Subplan oder eine Aktion geben. Diese Alternativen werden als Optionen bezeichnet und anhand des zu erwartenden Nutzens mittels Nutzenfunktion ausgewählt. Die Auswahl anhand des Nutzens ist laut Wooldridge (2002, S. 37ff) zwar weniger fortgeschritten als echtes reasoning im Sinne der KI-Forschung, aber zulässig. In den Wirtschaftswissenschaften, v.a. der Entscheidungslehre und Mikroökonomie ist es ohnehin üblich den Menschen als (beschränkt) rationalen Entscheider zu betrachten, der eine Nutzenfunktion als Grundlage seiner Entscheidungen hat. (Lang et al. 2000) stellt fest, dass sich eine nutzenbasierte Entscheidungsfindung mit einer BDI-Architektur vereinbaren lässt. Daher wird auch in SimKnowledge eine Nutzenfunktion als zentrales Auswahlkriterium verwendet. Sind die `Options` evaluiert und eine Entscheidung getroffen, so wird die nächste auszuführende Aktion des Plans in den `Scheduler` zur Planung des Zeitpunktes der Durchführung gebracht. Dieser `Scheduler` kann in SimKnowledge neue Aktionen entweder am Beginn des `NextTasks` Vektors platzieren oder an deren Ende. Die dort befindlichen Aktionen werden dann der Reihe nach ausgeführt.

Während der CEO und die Projektmanager über eine einfache regelbasierte Architektur verfügen, implementieren Worker Agenten die erweiterte InterRaP Architektur mit nutzenbasierter Aktionsauswahl und PECS Zustandsstruktur wie gerade beschrieben. Die Planungsfähigkeit der Agenten ist dabei insgesamt recht einfach gehalten und verdient die Bezeichnung „Künstliche Intelligenz“ nicht. Ressourcenbeschränkungen liegen v.a. im Bezug auf beschränkte Arbeitszeit und beschränktes Wissen vor. Geld ist zwar ebenfalls ein knappes Gut in der Simulation und wird entsprechend von den Agenten nachgefragt, allerdings existiert keine Möglichkeit es innerhalb der Simulation sinnvoll auszugeben.

Der interne Kontrollfluss ist sehr eng an die InterRaP Architektur angelehnt. Umweltinformationen gelangen über die Sensoren in den Agenten. Dort werden Sie in die Wissensbasis integriert. Dazu werden die drei Schichten von unten nach oben durchlaufen. Müller spricht von bottom-up Aktivierung. Wurde eine Aktion ausgewählt oder ein Plan gefasst, so wird von oben nach unten das Signal zur Aktivierung der Effektoren durchgereicht (top-down Ausführung).

**Simulationsumgebung:** Die Simulationsumgebung ist zweidimensional als Netz von zellulären Automaten implementiert (siehe auch Abschnitt 4.1.2). Ein einzelner zellulärer Automat wird dabei durch eine Instanz der Klasse `FloorPiece` gebildet. Ein `FloorPiece` soll in etwa einen Quadratmeter Büroboden entsprechen. Die `FloorPieces` werden von einem `Floor` Objekt in einem zweidimensionalen Array verwaltet. Zusätzlich hält jedes `FloorPiece` eine Referenz auf alle seine Nachbarn. Um die Nachbarn in der zahlenorientierten Speicherstruktur trotzdem semantisch adressieren zu können, wurden Konstanten definiert, die den Indizes 0 bis 7 für die acht Nachbarn der Moore-Nachbarschaft mit  $n=1$  geografische Bezeichnungen von Nord über Südost bis Nordwest zuordnen. Dadurch entsteht ein eng vermaschtes Netz, in dem Signale in alle Richtungen weitergeleitet werden können und sich die Agenten von einer Zelle zur nächsten bewegen können. Die `FloorPieces` erben von der Java Klasse `java.awt.Component`, damit die Anzeige leichter zu bewerkstelligen ist. `Component` ist die Basisklasse des Abstract Windowing Toolkits (AWT), welches das erste Java GUI Framework darstellt und auch für Java Swing noch in vielen Bereichen die Grundlage darstellt (z.B. `java.awt.Dimension` zum Verwalten der X- und Y-Ausdehnung von allen Fensterkomponenten). Das `FloorPiece` zeichnet dabei nicht nur sich selbst, sondern auch alle zeichenbaren Komponenten (z.B. Agenten), die sich auf ihm befinden.

Die Objekte und Entitäten, die sich auf dem `FloorPiece` befinden werden in einem `Vector` Objekt verwaltet. Diese `objectsOn` können nicht nur dem Benutzer angezeigt werden, sondern über das Emittieren von `VisualSense` Objekten auch den Agenten „gezeigt“ werden. Dies wird einmal pro Runde angestoßen, so dass sich die Agenten jede Runde ein Bild ihrer näheren Umgebung machen können. Da jedes `FloorPiece` alle Sensordaten abschwächt, bevor sie weitergeleitet werden, und nur dann weiterleitet, wenn sie über einem einstellbaren Schwellenwert liegen, wird der Effekt der lokalen Kommunikation und Wahrnehmung erzeugt. Dies dient auch der Skalierbarkeit, da in großen Gebäuden andernfalls eine Überflutung mit Sensordaten drohen würde.

**Nutzenfunktion der Projektarbeiter:** Um eine operationalisierte Nutzenfunktion zu erhalten, anhand derer der Agent entscheiden kann, welche der potentiell sinnvollen Aktionen er als nächstes ausführen soll, muss die in Abschnitt 3.3.1 diskutierte Nutzenfunktion weiter verfeinert werden.

Betrachtet man z.B. die Arbeitszeit, als Bestandteil der Nutzenfunktion, so muss man feststellen, dass eine Betrachtung der Art:

(9)  $Arbeitszeit = Anwesenheitszeit$  im Büro

zu kurz greift. Da Freizeit als positiver Beitrag zum Nutzen betrachtet wird und Arbeitszeit als negativer, müssen positiv bewertete Anteile der Zeit im Büro, wie z.B. Pausen und Sozialisation von der Arbeitszeit abgezogen werden. Daraus ergibt sich für die Arbeitszeit

(10)  $Arbeitszeit_t = Projektarbeit_t + Lernen_t + Koordination_t$

Die Wertschätzung und damit Gewichtung der Freizeit in der Nutzenfunktion, insbesondere der Ruhephasen, steigt umso mehr je müder der Mitarbeiter ist. Wenn also die physische Energie ein Zustand des Mitarbeiters ist, so kann dessen Kehrwert multiplikativ als Wichtungsfaktor herangezogen werden. Dabei muss darauf geachtet werden, dass nach vielen gängigen Motivationstheorien die körperlichen Bedürfnisse (z.B. Schlaf, Hunger, Durst) intellektuelle Motive (z.B. Streben nach Geld) verdrängen (siehe Abschnitt 3.3.2). Eine Aktion, die also Erholung (physische Energie) bringt, muss durch die Gewichtung einen höheren erwarteten Nutzen bringen als Aktionen, die keine Erholung verschaffen.

Detailliert man das zukünftige Gehalt als eine Funktion, die in einem gewissen Leistungsbereich konstant ist (das aktuelle Gehalt), darunter auf Null sinkt (Entlassung) und darüber sich erst schneller, dann abflachend einem Maximalgehalt annähert (erwartete Gehaltserhöhung), so interessiert als nächstes, aus welchen Komponenten sich die Leistung zusammensetzt. Die Ursachen für Erfolg oder Misserfolg liegen laut Weiner in der Gegenüberstellung der Fähigkeiten und Motivation des Mitarbeiters mit der Schwierigkeit der Aufgabe und dem Glück in der Situation (nach, Gebert, Rosenstiel 2002). Man kann also sagen

(11)  $Leistung_t = f_1(Skills_t, Motivation_t, Schwierigkeit_t, Glück_t)$

Die Fähigkeiten in der Periode t ergeben sich aus den Fähigkeiten der Vorperiode und dem neu Erlernten bzw. gerade Vergessenen.

(12)  $Skill_t = Skill_{t-1} + Gelerntes_{t-1} - Vergessenes_{t-1}$

Die Motivation des Mitarbeiters ergibt sich durch die im Domänenmodell definierten Motive und kann folgendermaßen als Formel dargestellt werden:

(13)  $Motivation_t = f_2(Anreiz_{Geld}, Anreiz_{Status}, Anreiz_{Zugehörigkeit}, Anreiz_{Wissen})$

Die Schwierigkeit der Aufgabe ist unmittelbar durch den Kundenauftrag vorgegeben, verteilt sich aber auf das gesamte Projektteam, so dass ein einzelner Mitarbeiter nur die

Schwierigkeit einer Teilkomponente zu bewältigen hat. Ist also das restliche Team sehr leistungsfähig, so bleibt für den Mitarbeiter eine leichte Aufgabe übrig.

$$(14) \text{ Schwierigkeit}_t = f_3(\text{Schwierigkeit}_{\text{Projekt}}, \text{Leistung}_{\text{restl. Team}})$$

Das Glück kann als stochastische Variable mit Normalverteilung und Erwartungswert 1 multiplikativ in die Leistungsfunktion eingehen und die Chance die Aufgabe zu erfüllen geringfügig positiv oder negativ beeinflussen. Da Glück aber nicht die Auswahl der Aktion beeinflusst, hat es für die Nutzenfunktion keine Bedeutung.

Die Komponente Gelerntes in Periode t-1 schließlich setzt sich aus dem unmittelbar projektinduzierten Gelernten und dem selbstmotiviert Gelernten ohne Projektbezug zusammen.

$$(15) \text{ Gelerntes}_{t-1} = \text{Gelerntes}_{t-1}^{\text{Projekt}} + \text{Gelerntes}_{t-1}^{\text{selbstmotiviert}}$$

Eine Kalkulation aller in der Simulation abgebildeten Einflüsse auf den Nutzen wäre recht aufwändig, so dass für jede Aktion nur die erwartete Auswirkung auf Agent und Umwelt und die damit verbundene erwartete Änderung im Nutzen ( $\Delta U$ ) in Abhängigkeit von der Situation und dem Zustand des Mitarbeiters berechnet wird, um anschließend die Deltas aller relevanter Aktionen miteinander zu vergleichen.

Zustandsinformationen, die in die Nutzenberechnung einfließen sollen, sind demnach:

- Physis: physische Energie
- Emotion: emotionale Energie
- Cognition: Einkommen, Wissen
- Social Status: soziale Zugehörigkeit, Expertenstatus

Die soziale Zugehörigkeit kann über die Anzahl der sozialen Kontakte berechnet werden, wobei in der Simulation schon das sich gegenseitig Sehen als Kontakt gewertet wird. Kommt es zu einem Gespräch, so geht dieser Kontakt mit höherer Gewichtung in die Berechnung ein. Damit die Zustandsvariable im Laufe der Zeit nicht stetig wächst, wird dies im Verhältnis zur Arbeitszeit insgesamt gesehen.

Der Expertenstatus ist schwieriger zu berechnen, da es sich sozusagen um das Meta-Meta-Wissen des Agenten handelt, also die Annahmen des Agenten darüber widerspiegelt, was andere Agenten von seinem Wissen halten. Dabei muss zudem noch das Verhältnis des eigenen Wissens zum Wissen der Kollegen bedacht werden, da sich ein hoher Status auf hohem relativem Wissen (also mehr als die Kollegen) begründet und weniger auf

absolutem. Eine Formel, die das Ganze stark abstrahiert, wäre das Verhältnis von eigenem Wissen zu dem der Kollegen. Dies lässt zwar die doppelte Reflexion außen vor, kommt aber nach Meinung des Autors trotzdem nahe an die Realität heran, da auch in der Praxis Selbstbild und Fremdbild oft stark differieren.

Die Wissensrepräsentation ist in der vorliegenden Simulation in zwei Dimensionen zu betrachten. Zum einen muss das simulierte Wissen abgebildet werden. Dafür werden in der Literatur zwei grundsätzliche Möglichkeiten diskutiert. Es kann entweder als kontinuierliche oder als diskrete Variable gespeichert werden. Da zum Zeitpunkt der Entscheidung keine klaren Argumente für eine der beiden Alternativen sprachen, wurde die kontinuierliche Variable als vermeintlich einfacher zu implementierende Variante gewählt. Der Wertebereich der kontinuierlichen Variablen ist das Intervall  $[0; 5,0]$ . Die entsprechenden Ausprägungen können als Expertenstufen im Sinne von Dreyfus/Dreyfus interpretiert werden. Im Rückblick betrachtet scheint die diskrete Variante Vorteile zu besitzen (siehe dazu Abschnitt 6.7.5).

Die Entscheidungsfindung der Agenten folgt einem beschränkt rationalen Paradigma. Es wird dabei zuerst eine regelbasierte Vorauswahl getroffen, die erkennt, welche Situation vorliegt. Für jede Situation ist eine oder sind mehrere sinnvolle Aktionen hinterlegt. Sind mehrere Aktionen möglich, so erfolgt eine nutzenbasierte Auswahl der Aktion mit dem höchsten Nutzenbeitrag. Die Schichtenarchitektur von InterRaP sieht eine bottom-up Aktivierung vor, so dass Bedürfnisse auf den unteren Ebenen überwiegen, es wird also z.B. trotz nicht fertig gestelltem Arbeitspaket eine Pause gemacht, um den Wert der Zustandsvariable `physicalEnergy` zu erhöhen. Dies geschieht ohne weitere Nutzenabschätzung, da die beschränkt rationale Planung erst auf der zweiten Schicht beginnt.

### 5.3.7 Systemsicht

In der Systemsicht sollen die definierten Anforderungen aufgegriffen und in einen Systementwurf überführt werden. Weiterhin wird die Benutzerschnittstelle behandelt, sowie Ausnahmebehandlung und die Art der Auslieferung beschrieben.

Das System soll in Java implementiert werden, um eine plattformunabhängige Ausführbarkeit zu gewährleisten. Jeder Agent wird als eigenständiger Thread programmiert. Er kann über Sensoren seine unmittelbare Umgebung wahrnehmen. Dabei können auditive Informationen erfasst werden, um mit anderen Agenten zu kommunizieren und visuelle Informationen dienen zur Wahrnehmung der aktiven und passiven Entitäten sowie deren Aktivitäten. Beide Informationstypen werden über die Umgebung weiterge-

geben, in Analogie zur realen physikalischen Welt wo Geräusche über Schallwellen und Licht über elektromagnetische Wellen durch das Medium Luft übertragen werden. Die Wahrnehmbarkeit der Informationen nimmt dabei mit zunehmender Entfernung von der Quelle ab und wird weiterhin durch die Beschaffenheit der Umgebung beeinflusst. Über Effektoren können Agenten mit der Umgebung interagieren. Es sind dabei nur die grundlegenden operationale Primitive Bewegungen (`go`), Sprechen (`tell`) und Manipulieren (`work`) vorgesehen. Komplexere Aktionen ergeben sich durch Kombination mit Sprechakten und anderen Entitäten.

Die zentrale Einheit ist die Klasse `SimulationEngine`, welche gemäß dem Singleton Entwurfsmuster gestaltet wurde, damit immer nur eine Instanz gleichzeitig aktiv sein kann. Sie lädt zuerst Basisparameter aus einer XML-Konfigurationsdatei, in der die Verbindungsdaten zur Datenbank, sowie weitere technische Parameter wie das Verzeichnis für Logdateien und der Loglevel abgelegt sind. Alle Parameter, die unmittelbar das Simulationsszenario betreffen wie Anzahl Agenten und eingesetzte WM-Maßnahmen, werden im zweiten Schritt aus der Datenbank geladen. Dazu werden die gewünschte Studie und das Experiment entweder vom Benutzer interaktiv in einem Dialog ausgewählt oder als Kommandozeilenparameter übergeben. Anschließend werden die Agenten als Threads erzeugt und mit Zufallsvariablen oder den Datenbankinhalten initialisiert. Dann kann die eigentliche Simulation beginnen, die rundenbasiert ist. Jede Runde besteht aus den Teilen (a) Externe Entitäten generieren neue Informationen (b) Wahrnehmen der Veränderungen in der Umwelt (`perceive`), (c) Planen der nächsten Aktion (`reason`) und (d) Ausführen der Aktion (`act`). Vor jeder neuen Runde wird die Reihenfolge, in der die Agententhreads aufgerufen werden neu ermittelt, damit „Chancengleichheit“ für alle Agenten besteht.

Das System besitzt eine Java Swing basierte Benutzerschnittstelle. Dort wird das Firmengebäude als zweidimensionales Gitter dargestellt. Agenten werden als Kreise visualisiert. Unterschiedliche Farben stellen die unterschiedlichen Rollen der Agenten dar. Zudem wird über die Farbe auch zwischen Agenten mit und ohne Projekt, bzw. Arbeitspaket unterschieden, wie in Abbildung 5-9 zu sehen.

Der Benutzer kann die Simulation jederzeit pausieren und dann per Mausklick die wichtigsten Zustandsvariablen der Agenten inspizieren. Weiterhin sind auf der rechten Seite des Fensters Anzeigen für die aktuelle Runde, die aktuelle Uhrzeit und den aktuellen Tag. Mit dem „statistics“ Knopf können eine Reihe von statistischen Daten angezeigt werden, wie z.B. die Summe aller Wissensvariablen und die durchschnittlichen Maximal-,

Minimal- und Mittelwerte der Wissensbereiche. Der Knopf „move agent“ dient nur während des Tests zum Verschieben der Heimatkoordinaten eines Agenten. Mit dem „end sim“-Knopf kann die Simulation beendet werden. Weitergehende Interaktionsmöglichkeiten bestehen nicht. Die grafische Benutzeroberfläche kann allerdings auch deaktiviert werden. Die Ergebnisse der Simulation sind dann ausschließlich in der Datenbank und in der Logdatei zu finden.

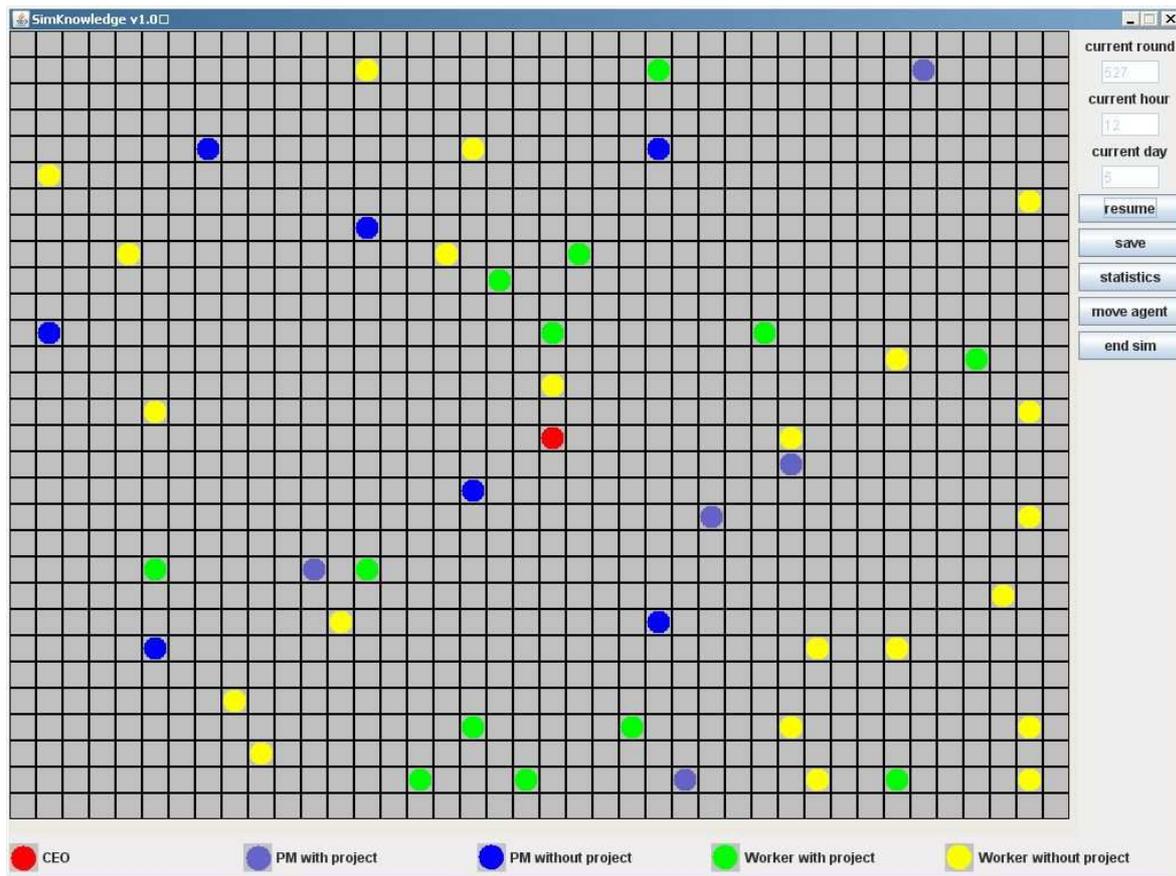


Abbildung 5-9: Screenshot der SimKnowledge Benutzerschnittstelle (eigene Darstellung)

Die Ausnahmebehandlung des Programms wird über die in Java üblichen try-catch-Blöcke abgewickelt. Die Installation des Programms ist ebenfalls wie für Java üblich. Es gibt keine eigene Installationsroutine. Die Binärdateien werden einfach auf die Festplatte kopiert oder aus der ZIP-Datei extrahiert. Ist eine Java Runtime Umgebung installiert, so reicht ein Aufruf der Form „java -cp . simknowledge/SimulationEngine“, um das Programm zu starten. Um die Datenbankfunktionalität zu nutzen, muss außerdem der Datenbankserver gestartet und der Zugriff darauf in der XML-Konfigurationsdatei settings.xml konfiguriert sein. Detaillierte Informationen dazu finden sich unter <http://sourceforge.net/projects/simknowledge>. Dort ist auch der vollständige Quellcode des Simulationsprogramms veröffentlicht.

## 5.4 Auswahl der Simulationssprache und des Frameworks

Für die Umsetzung des Modells in ein Softwaresystem muss zuerst eine geeignete Simulationssprache gewählt werden. Brooks und Robinson unterscheiden zwischen drei Klassen (Brooks, Robinson 2001, S. 55f).

- Programmiersprachen (z.B. Pascal, C++, Java)
- Simulationssprachen (z.B. Simula, SimPas, Silk)
- Visuelle, interaktive Modellierungssysteme (z.B. Arena, Sesam, Simile)

Programmiersprachen bieten den Vorteil, dass darüber eher Vorkenntnisse bestehen, als für spezielle Simulationssprachen, sie weit verbreitet und oft frei verfügbar sind und auch sehr komplexe Sachverhalte abgebildet werden können (vgl. *ibid.* S.56).

Simulationssprachen dagegen sind speziell auf die Anforderungen von Simulationen zugeschnitten und bieten eine Reihe von Funktionen, die die Erstellung von Simulationsprogrammen erleichtern, wie z.B. Ereignisbehandlung und Scheduling, grafische Darstellung der Simulationsobjekte, Datensammlungs- und Auswertungsfunktionen sowie Unterstützung bei der Verwaltung von Simulationsexperimenten (*ibid.*).

Visuelle und interaktive Modellierungssysteme (VIMS) schließlich erlauben es auch Nutzern ohne Programmierkenntnisse Simulationsprogramme zu erstellen. Die Entitäten können grafisch angelegt und entsprechend den Bedürfnissen parametrisiert werden. Gute VIMS sind zudem über eine eigene Skriptsprache oder eine konventionelle Programmiersprache erweiterbar, so dass auch komplexere Systeme umgesetzt werden können (*ibid.* S. 57). In Abbildung 5-10 sind die Vor- und Nachteile der einzelnen Systeme noch einmal zusammengefasst.

	Programmiersprachen	Simulationssprachen	VIMS
Anwendungsbereich	Breiter	→	Enger
Einfachheit der Modellentwicklung	Niedriger	→	Höher
Entwicklungszeit	Länger	→	Kürzer
Ausführungsgeschwindigkeit	Schneller	→	Langsamer
Werkzeugkosten	Niedriger	→	Höher

Abbildung 5-10: Vor- und Nachteile von Alternativen bei Simulationssoftware nach (Brooks, Robinson 2001, S. 57)

Auch Law und Kelton stellen die Vor- und Nachteile von Simulationspaketen mit Programmiersprachen gegenüber (Law, Kelton 2000, S. 203f). Sie identifizieren neben den bereits vorgestellten Kriterien noch bessere Wartbarkeit und Fehlersuche in Simulationspaketen und weiter verbreitetes Wissen über Programmiersprachen als relevante Faktoren. Die leichtere Fehlersuche in Simulationspaketen kann allerdings aus der Erfahrung des Autors und Diskussionen mit anderen Simulationsforschern nicht bestätigt werden, da moderne Entwicklungsumgebungen für Programmiersprachen oft über ausgezeichnete Debugging-Werkzeuge verfügen, was für Simulationspakete nicht gilt.

### **5.4.1 Anforderungen an MABS-Frameworks**

Für MABS bietet sich weiterhin die Möglichkeit an, Frameworks oder Plattformen zu benutzen, die in einer konventionellen Programmiersprache entwickelt wurden und Funktions- oder Klassenbibliotheken bereitstellen, die im eigenen Simulationsprojekt benutzt werden können. Dadurch kann man die Flexibilität von Programmiersprachen mit den Vorteilen von Simulationssprachen zumindest teilweise verbinden.

Im Folgenden sollen daher einige Alternativen gegenübergestellt werden. Brooks und Robinson schlagen vor, für die Auswahl folgende Kriterien zu berücksichtigen (Brooks, Robinson 2001, S. 60f).

1. Eignung für den Zweck
2. Einfachheit und Grenzen der Modellentwicklung
3. Unterstützung für Zufallsgrößen und -verteilungen
4. Grafische Fähigkeiten
5. Unterstützung für Modellprüfung und Verifikation
6. Unterstützung für Durchführung der Experimente
7. Auswertungsfunktionen
8. Hard- und Softwareanforderungen
9. Support der Entwickler
10. Verbreitung und Herstellerunterstützung
11. Kosten

Einige dieser Kriterien werden auch von Law und Kelton genannt (Law, Kelton 2000, S. 208). Noch detailliertere und v.a. technischere Kriterien stellen Marietto et al. (2002) vor. Frameworks sollten demnach folgende Fähigkeiten besitzen (S. 3ff)

1. Verschiedene Scheduling Mechanismen anbieten
2. Verschiedene Agententypen bereitstellen und starten können
3. Technische und logische Fehler behandeln
4. Vollständig kontrollierte aber auch etwas freiere Umgebungen für Agenten bereitstellen
5. Unterstützung für die Entwicklung von Agentenarchitekturen durch Bereitstellen von Vorlagen für reaktive und intentionale Agententypen
6. Kommunikationsmethoden bereitstellen, die einfache Übermittlungsmechanismen und Konformität zu KQML oder FIPA ACL kombinieren
7. Organisatorische Abstraktionen wie Rollen und Gruppen bereitstellen
8. Mehrere Gesellschaften verwalten, z.B. in hierarchischen Organisationsstrukturen
9. Verhalten und kognitive Aktivitäten der Agenten aufzeichnen
10. Graphische Datenanalyse
11. Auf verhaltensbasierte und kognitive Ereignisse reagieren

Die Evaluation von Tobias und Hofmann (2004) konzentriert sich auf freie Java Bibliotheken zur agentenbasierten Simulation. Die von ihnen erarbeiteten Kriterien sind daher z.T. etwas anderer Natur und stellen darüber hinaus sehr weitgehende Anforderungen dar (ibid. Absatz 3.5, 3.10, 3.15). In Klammern finden sich einige der möglichen Ausprägungen.

1. Lizenzmodell (Quellcode ganz oder teilweise offen gelegt)
2. Dokumentation (vollständig oder nicht, Einführung und API Dokumentation)
3. Support (Mailing Listen, direkter Kontakt zu Entwicklern)
4. Verbreitung (nur von Entwicklern genutzt oder auch von anderen Wissenschaftlern)
5. Aktive Weiterentwicklung (letzte Aktualisierung liegt lange zurück oder häufige Updates)
6. Modellierungsunterstützung (nur Klassenbibliothek oder grafisch editierbar)
7. Simulationskontrolle (dynamisches Eingreifen in die Simulation möglich)
8. Experimentierunterstützung (Simulationsserien und Parameteroptimierung)
9. Projektorganisation (Verwaltung von Simulationsläufen und Modellelementen)
10. Bedienbarkeit (Programmierkenntnisse nötig oder von Laien bedienbar)

11. Verbreitung eigener Modelle und Ergebnisse (Verfügbarmachen im Internet)
12. Installation (einfach oder komplex, nur mit Unterstützung möglich)
13. Große Anzahl komplexer Agenten möglich (nur einfache, nur wenige, viele)
14. Agentenkommunikation (Datenaustausch direkt möglich, auch komplexe Muster)
15. Verschachteln von Agenten möglich (eingeschränkt, beliebig)
16. Generieren von Agentenpopulationen (Datenimport, verschiedene Verteilungen)
17. Agentennetzwerke generieren (basierend auf Parametern sozialer Netzwerke)
18. Räumliche Verteilungen ermöglichen (Bewegung in Umgebung möglich)
19. Dynamische Änderungen am Modell möglich (während der Ausführung)

Aus diesen Kriterien wurden einige für die eigene Untersuchung ausgewählt (siehe Abschnitt 5.4.3), die besonders relevant erscheinen und zwischen den Kandidaten am besten diskriminieren. Die folgende Liste zeigt auf, wo die eigenen Schwerpunkte liegen.

1. Verwendung bestehender Standards wo immer sinnvoll
2. Komplexität nicht unnötig erhöhen (keine offene Plattform nötig, Agenten müssen nicht migrieren können, ...)
3. Benutzen einer weit verbreiteten, objektorientierten Programmiersprache, um Wiederverwendbarkeit und Verständlichkeit der Software zu erhöhen
4. Die Agentenarchitektur muss frei wählbar oder leicht anpassbar sein
5. Kommunikation sollte auf räumlich nahe Agenten einschränkbar sein
6. Die Abbildung einer räumlichen Umgebung für die Agenten muss möglich sein
7. Die grafische Benutzerschnittstelle muss die Visualisierung der Agenten in einer 2D-Umgebung ermöglichen
8. Die Datenanalysewerkzeuge sollten fortgeschrittene Analysen wie soziale Netzwerkanalyse unterstützen oder leicht dafür angepasst werden können.

Neben reinen Frameworks für MABS wurden auch einige solche in die Vorauswahl mit einbezogen, die aus der klassischen Simulation stammen (PECS/Simplex 3, Urban 2000), und solche, die für Multiagentensysteme (MAS) entwickelt wurden (JADE, Bellifemine et al. 2000).

Eine wesentliche Erkenntnis dabei war, dass die Anforderungen von MABS grundlegend andere sind als die von MAS. Bedingt durch die stark differierende Zielsetzung, effizientes verteiltes Problemlösen einerseits und möglichst realistische Simulation (sozialer) Prozesse

andererseits, ist es für MAS z.B. sinnvoll, dass Agenten jederzeit direkt kommunizieren können, sie brauchen keine räumliche Umgebung und arbeiten nicht rundenbasiert, sondern bekommen vom Betriebssystem je nach Aufgabe Rechenzeit zugeteilt.

Auch Drogoul et al. stellen fest, dass der Begriff Agenten in unterschiedlichen Arbeiten sehr verschieden benutzt wird und speziell zwischen MAS-Arbeiten und MABS-Arbeiten erhebliche Unterschiede in der Auffassung des Agentenbegriffs bestehen, die durch das konzeptionelle Vorgehen und die Rollen bei Simulationsprojekten entstehen (Drogoul et al. 2002). Daher werden im Folgenden nur reine MABS-Frameworks berücksichtigt.

#### **5.4.2 Überblick über MABS-Frameworks**

Bevor die Frameworks einander gegenübergestellt werden, sollen die Alternativen kurz allgemein vorgestellt werden.

(Tobias, Hofmann 2004) betrachten in ihrer Evaluation Quicksilver, RePast, Swarm und VSEit (Absatz 2.2). (Marietto et al. 2002) untersuchen Cormas, MadKit und Swarm. Weiterhin konnten als Kandidaten die Frameworks Ascape (Parker 2001), Breve (Klein 2002), JAS (Margarita, Sonnessa 2003), MASON (Luke et al. 2004), NetLogo (Blikstein et al. 2005), SeSAM (Klügl 2001) und Sim\_Agent (Sloman, Logan 1999) identifiziert werden.

Tobias und Hofmann kommen zu dem Ergebnis, dass RePast, welches mittlerweile in Version drei vorliegt (North et al. 2006), bessere Unterstützung für Simulationen bietet als Swarm, auf dem RePast aufbaut (Tobias, Hofmann 2004, Absatz 5.2f). Quicksilver und VSEit liegen weit hinter RePast. Auch nach einer Gewichtung der einzelnen Kriterien anhand ihrer Relevanz ändert sich daran nichts (ibid. Absatz 5.25ff).

(Marietto et al. 2002) fokussieren das Vorstellen der Anforderungen, wogegen der Vergleich selbst sehr knapp ausfällt und sie resümieren, dass alle Kandidaten in etwa gleichauf liegen (ibid. S. 11). Das stimmt aus Sicht des Autors für alle Anforderungen, bis auf die für die vorliegende Arbeit so wichtigen organisatorischen Abstraktionen. Diesbzgl. schneidet MadKit besser ab als Cormas und Swarm (ibid. S. 8). Nachdem MadKit darüber hinaus noch gepflegt wird (letzte Version 4.1.2 vom 22.11.2005) soll es zusammen mit RePast in das Testfeld aufgenommen werden.

Von den noch nicht verglichenen Kandidaten sind Ascape und Sim\_Agent schon seit langer Zeit nicht weiterentwickelt worden und werden daher nicht betrachtet, so dass als zu

vergleichende Alternativen noch Breve, JAS, MadKit, MASON, NetLogo, RePast und SeSAM bleiben, welche in die eigene Gegenüberstellung aufgenommen wurden.

**Breve:** Breve wird von Klein entwickelt und liegt zum Zeitpunkt des Vergleichs in Version 2.3 vom 31.08.2006 vor<sup>49</sup>. Von der nächsten Version 2.4 existieren aber schon mehrere Beta-Versionen, die letzte vom 24.04.2006. Auch die Aktivitäten im Diskussionsforum lassen auf eine aktive Entwickler- und Nutzergemeinde schließen. Die Besonderheit von Breve ist konsequente 3D-Visualisierung der Simulation und die Möglichkeit realistische physikalische Effekte zu modellieren. Eigene Modelle werden in der objektorientierten Programmiersprache *steve* entwickelt, die sich an Objective C und Perl anlehnt. Die 3D-Visualisierung basiert auf OpenGL und kann in Form von PNG-Schnappschüssen oder MPEG-Filmen festgehalten werden. Simulationszustände können als XML-Daten gespeichert und geladen werden. Simulationsobjekte können im Netzwerk verteilt werden und auch der Zugriff auf die Benutzeroberfläche über einen Webbrowser ist möglich. Erweiterungen können in C geschrieben und als Plug-In in Breve eingebunden werden. Ein Alleinstellungsmerkmal von Breve im Testfeld ist die Integration der Programmiersprache *Push*, mit der genetische bzw. evolutionäre Programmierung möglich wird. Dies ist besonders für Artificial Life Modelle wichtig, einem der Haupteinsatzgebiete von Breve.

**JAS:** Die Java Agent-Based Simulation Library (JAS) wird an der Universität von Turin in Italien unter Federführung von Sonnessa entwickelt und liegt zum Zeitpunkt des Vergleichs in der Version 1.2.1 vom 18.03.2006 vor<sup>50</sup>. Es lehnt sich wie RePast und MASON an Swarm an und zeichnet sich durch Unterstützung von XML-basiertem Datenimport und -export sowie Anbindung an das relationale Datenbankmanagementsystem (DBMS) HypersonicDB aus. JAS ist noch recht jung. Die Version 1.0 wurde im Mai 2004 vorgestellt. Eigene Modelle werden mit Hilfe der bereitgestellten Bibliothek in Java entwickelt. Neben dem JAS Desktop Paket gibt es auch eine Version namens Sim2Web, die mit Hilfe des Zope Application Servers eine Internetverfügbarkeit der Simulationsumgebung ermöglicht, so dass Benutzer über das Internet Simulationen starten können und Ergebnisse eigener Simulationen auch leicht im Internet verfügbar gemacht werden können. JAS bietet Pakete für genetische Algorithmen, neuronale Netze, statistische Auswertungen und graphische Analyse sozialer Netzwerke.

---

<sup>49</sup> <http://www.spiderland.org/breve/>, letzter Abruf am 31.12.2007

<sup>50</sup> <http://jaslibrary.sourceforge.net>, letzter Abruf am 31.12.2007

**MadKit:** MadKit wird am Montpellier Laboratory of Computer Science, Robotics and Microelectronics in Frankreich von Gutknecht, Ferber und Michel entwickelt und liegt zum Zeitpunkt des Vergleichs in der Version 4.1.2 vom 22.11.2005 vor<sup>51</sup>. Es ist eines der am längsten verfügbaren Frameworks und wurde bereits in (Ferber 1999) besprochen, auch wenn es erst 2002 als Version drei auf Sourceforge unter der GPL freigegeben wurde. MadKit basiert auf Java und hat als zentrales Konzept den Dreiklang Agent-Rolle-Gruppe. Es ist sehr modular aufgebaut. Die zentrale Applikation ist weitgehend funktionslos und dient nur als grafische Umgebung zum Laden der Plug-Ins, welche die eigentliche Funktionalität realisieren. Eine weitere Auffälligkeit ist die Möglichkeit eigene Agenten neben Java auch in den Skriptsprachen BeanShell und Scheme, mit Regeln die vom Regelinterpreter Jess interpretiert werden, oder in Python zu erstellen. Nicht nur die aktiven Entitäten der Simulation sind in MadKit als Agenten realisiert sondern auch alle administrativen Komponenten wie die Projektverwaltung, die Datensammlung und die grafische Ausgabe.

**NetLogo:** NetLogo wird an der Northwestern University in Evanston (USA) entwickelt und liegt zum Zeitpunkt des Vergleichs in der Version 3.1 vom 14.04.2006 vor<sup>52</sup>. Es basiert auf StarLogo<sup>53</sup> und kann als Nachfolger der StarLogo Erweiterung StarLogoT verstanden werden. Mit seinem Erscheinen im April 2002 und der Historie von StarLogo blickt Netlogo schon auf eine längere Entwicklungszeit zurück. Entwickelt wird zum einen mit einer Erweiterung der prozeduralen Programmiersprache Logo und zum anderen mit der grafischen Entwicklungsumgebung. NetLogo selbst ist aber Java-basiert. Dadurch können Modelle auch als Java-Applet in Webseiten eingebunden werden. Bemerkenswert an NetLogo sind v.a. die umfangreiche Modellbibliothek, die sowohl zum Erlernen der Sprache als auch als Basis für eigene Modelle genutzt werden kann und die ausführliche Dokumentation. Neben der hier besprochenen Basisversion existiert auch eine 3D-Preview Version vom März 2005, die hier nicht weiter betrachtet wird.

**MASON:** Das Akronym MASON steht laut der Internetseite des Frameworks für „Multi-Agent Simulator Of Neighborhoods oder Networks oder etwas ähnlichem“<sup>54</sup>. Es wird an

---

<sup>51</sup> <http://www.madkit.org/>, letzter Abruf am 31.12.2007

<sup>52</sup> <http://ccl.northwestern.edu/netlogo/>, letzter Abruf am 31.12.2007

<sup>53</sup> <http://education.mit.edu/starlogo/>, letzter Abruf am 31.12.2007

<sup>54</sup> <http://cs.gmu.edu/~eclab/projects/mason/>, letzter Abruf am 31.12.2007

der George Mason Universität in Fairfax, VA (USA) entwickelt und liegt zum Zeitpunkt des Vergleichs in der Version 11 vom 06.04.2006 vor. Die recht hohe Versionsnummer führt aber in die Irre und ist kein Hinweis auf eine lange Entwicklungszeit (Version 1 wurde 2003 vorgestellt) sondern liegt an der fehlenden Verwendung von Subversionen. Es ist in Java entwickelt und lehnt sich an Swarm an. Die Modelle sind unabhängig von der Visualisierung und können zu beliebigen Zeitpunkten gespeichert und auf andere Systeme übertragen werden. Als eines der wenigen Frameworks sind in MASON neben 2D-Visualisierungen auch gleichberechtigt 3D-Visualisierungen möglich. Von der grafischen Ausgabe können PNG-Schnappschüsse und Quicktime-Filme erstellt werden.

**RePast:** RePast steht für Recursive Porus Agent Simulation Toolkit, entstand an der Universität von Chicago und wird jetzt von der Non-Profit Organisation ROAD weiterentwickelt (RePast Organization for Architecture and Development). Es liegt zum Zeitpunkt des Vergleichs in der Version 3.1 vor<sup>55</sup> und blickt auf eine langjährige Entwicklung seit Januar 2000 zurück. Es gibt neben der Java Hauptversion auch eine vollwertige .NET Version und eine Version, die grafisch orientiert ist und mit einer an Python angelehnten Skriptsprache arbeitet. Aus den grafisch erstellten und mit Skripts detaillierten Agenten kann aber Java Quellcode erzeugt werden, so dass auch diese RePast Py Version alle Möglichkeiten der freien Programmierung bietet. RePast lehnt sich an Swarm an, bietet aber weiter reichende Funktionalität wie Regressionsanalyse der Simulationsdaten und genetische Algorithmen. Darüber hinaus sind die vielen verschiedenen mitgelieferten 2D-Umgebungen hervorzuheben, die mit entsprechenden Displays (grafische Entsprechungen von Agenten und Umgebungen) leicht zu visualisieren sind und mit denen es direkt möglich ist, Filmaufnahmen vom Ablauf einer Simulation anzufertigen. Sogar der Import von GIS-Daten (Geografisches Informationssystem) ist vorgesehen.

**SeSAM:** SeSAM steht für Shell for Simulated Agent Systems und wird von Klügl und ihren Kollegen an der Universität Würzburg entwickelt<sup>56</sup>. Zum Zeitpunkt des Vergleichs ist die Version 2, die bisher aber erst in einer Betaversion vom 08.03.2006 vorliegt. Die letzte stabile Version 1.9.2 wurde am 24.05.2005 veröffentlicht und steht am Ende einer Entwicklungsreihe mit einem guten Dutzend Versionen seit der Version 1 aus 2002. Es legt den Schwerpunkt auf die grafische Entwicklung von Agenten, bietet aber aufgrund der

---

<sup>55</sup> <http://repast.sourceforge.net/>, letzter Abruf 31.12.2007

<sup>56</sup> <http://www.simsesam.de/>, letzter Abruf 31.12.2007

zugrunde liegenden Sprachen SeSAM-Impl und SeSAM-UML vollständige Programmierfreiheit, da die Sprachen Turing-vollständig sind. SeSAM wurde für die Simulation von Staaten-Bildenden Tieren entwickelt und bietet im weiteren Sinne damit Möglichkeiten für soziale Simulationen. Weitere bemerkenswerte Funktionen sind die integrierte graphische Datenanalyse und die Möglichkeit zum Verteilen der Rechenlast der Simulation im Netzwerk. Plug-Ins zum Erweitern des Systems um neue Primitive zur grafischen Modellierung können in Java programmiert werden.

### 5.4.3 Vergleich ausgewählter MABS Frameworks

Im folgenden Abschnitt sind die Ergebnisse des Vergleichs der Frameworks zusammengefasst. Auf eine Punktbewertung wurde bewusst verzichtet, da nur ein geeigneter Kandidat für die eigene Simulation gefunden werden muss. In Tabelle 5-3 und Tabelle 5-4 sind die Bewertungen in der Übersicht dargestellt. Im Anschluss werden die Tabelleneinträge im Text näher erläutert.

*Tabelle 5-3: Zusammenfassung des Vergleichs der MABS-Frameworks #1*

<b>Kriterium</b>	<b>RePast</b>	<b>JAS</b>	<b>NetLogo</b>	<b>MASON</b>
Grafische Oberfläche	Ja	Ja	Ja	Ja
Zufallszahlen	COLT	COLT	Gleichverteilt	Gaußsche Vert.
Experimentieren	Grafisch, Datendatei	Grafisch, Datendatei	Grafisch, Datendatei	Grafisch, Datendatei
Generieren von Agenten	Textimport	Text, Excel, HSQLDB, grafisch	Textimport	Nicht unterstützt
Verhalten aufzeichnen	Beliebiges	Beliebiges	Beliebiges	Beliebiges
Datenexport	CSV	CSV, Excel, HSQLDB	CSV	Keiner
Datenanalyse	Zeitreihen, Balkendiagr.	Zeitreihen, Balkendiagr.	Zeitreihen	Zeitreihen
Soziale Netzwerkanalyse	Nur Anzeige	Nicht unterstützt	Netzwerkmodell ist Beta	Nicht unterstützt
Agententypen	nur einfache	nur einfache	nur einfache	nur einfache
Agentenkommunikation	Methodenaufruf	Methodenaufruf	Ask-Primitive	Methodenaufruf
Organisationsabstraktion	Keine	Keine	Keine	Keine
Räumliche Umgebung	2D int, double, hexagonal	2D int, double	2D int	2D und 3D, int, double

<b>Kriterium</b>	<b>RePast</b>	<b>JAS</b>	<b>NetLogo</b>	<b>MASON</b>
Dokumentation	Tutorial, API, Beispiele	Tutorial, API, Beispiele	umf. Handbuch, API, Beispiele	Tutorial, API, Beispiele
Freie Programmierung	Java, .Net	Java	Logo Dialekt	Java

Eine grafische Benutzeroberfläche zum Konfigurieren und Verwalten der Simulation vereinfacht für den Benutzer das Experimentieren. Die getesteten Frameworks bieten hier außer Breve alle gute Unterstützung. Bei den Zufallszahlen liegen RePast und JAS vor den Mitbewerbern, da sie durch Einbinden der freien COLT Bibliothek sechs verschiedene Verteilungen bereitstellen und zudem die Möglichkeit bieten, empirisch ermittelte Verteilungen nachzustellen. Unterstützung für das Experimentieren bieten die Swarm Ableger RePast, JAS und MASON gleichermaßen gut.

Auch NetLogo ermöglicht es Parameter im Vorfeld festzulegen und dann im Batch-Lauf die zugehörigen Simulationsläufe abzuarbeiten. Bei SeSAM können die Parameter grafisch festgelegt werden, Batch-Läufe scheinen hingegen nicht möglich zu sein, soweit es sich der lückenhaften Dokumentation entnehmen und beim Test feststellen lässt. Gar keine Unterstützung fürs Experimentieren bieten Breve und MadKit. Beim Generieren von Agenten bietet sich ein ähnliches Bild. Bei SeSAM lässt sich grafisch festlegen, welche Agententypen in welcher Anzahl mit welchen Parametern generiert werden sollen. Bei RePast, JAS und NetLogo kann man auch über Import von CSV-Dateien<sup>57</sup> das Generieren von Agenten steuern. JAS hebt sich durch zusätzliche Unterstützung von Excel-Dateien und Daten aus HSQLDB Datenbanken positiv von den anderen Kandidaten ab, während MASON hier ebenso wie Breve und MadKit keine Unterstützung bieten, so dass man die Agenten per Programmcode erstellen muss.

Das Aufzeichnen des Agentenverhaltens, womit sowohl beobachtbares Verhalten, wie kognitive Zustände gemeint sind, ist bei RePast, JAS, MASON und NetLogo gleichermaßen gut möglich. Der Anwender muss zwar selbst im Programmcode festlegen, welche Daten gesammelt werden, es stehen ihm dazu jedoch gut vorbereitete Mittel zur Verfügung, so dass das Datensammeln stark erleichtert wird. In MadKit sind die Möglichkeiten auf Beeinflussen der Nachrichten und Änderungen in der Organisation beschränkt. Kognitive Zustände der Agenten sind nicht durch vorgefertigte Mittel aufzeichnenbar. Bei SeSAM sind die Möglichkeiten zum Aufzeichnen des Verhaltens

---

<sup>57</sup> Comma Separated Values, Textdatei mit Werten, die durch Kommas getrennt sind

unbegrenzt, man muss die Variablen aber wie alles andere auch grafisch festlegen, was in diesem Fall nicht als Einschränkung empfunden wird. Breve bietet wiederum keine Unterstützung. Beim Datenexport bietet sich ein ähnliches Bild. Während MadKit, MASON und Breve hier keine Unterstützung bieten, ist bei allen anderen Frameworks CSV-Export möglich. SeSAM bietet mit einem Zusatzmodul auch eine MySQL Anbindung, während JAS hier wie beim Import Excel und HSQLDB Unterstützung.

*Tabelle 5-4: Zusammenfassung des Vergleichs der MABS-Frameworks #2*

<b>Kriterium</b>	<b>Breve</b>	<b>MadKit</b>	<b>SeSAM</b>
Grafische Oberfläche	Rudimentär	Ja	Ja
Zufallszahlen	Nicht unterstützt	Nicht unterstützt	Exp, Gamma, Normal, Poisson
Experimentieren	Nicht unterstützt	Nicht unterstützt	grafisch
Generieren von Agenten	Nicht unterstützt	Nur grafisch	CSV
Verhalten aufzeichnen	Nicht unterstützt	Nachrichten und Organisation	Beliebiges
Datenexport	Keiner	Keiner	MySQL, CSV
Datenanalyse	Keine	Sequenzdiagr.	Zeitreihen
Soziale Netzwerkanalyse	Nicht unterstützt	Nicht unterstützt	Nicht unterstützt
Agententypen	Nur einfache	nur einfache	Nur einfache
Agentenkommunikation	Über publish-subscribe	nach Adresse oder Rolle	Message Passing
Organisationsabstraktion	Keine	Gruppe, Rolle, Community	Keine
Räumliche Umgebung	3D, double	2D int mit TurtleKit	2D int
Dokumentation	Online Handbuch	Benutzer, Entwickler	Tutorial, Wiki, Beispiele
Freie Programmierung	Objektorientierte Sprache steve	Java, Scheme, Python, Jess, BeanShell	Grafisch, an UML angelehnt, Turing complete

Eine Form grafischer Datenanalyse ist bei allen Kandidaten außer Breve vorgesehen. MadKit bietet allerdings nur eine grafische Darstellung des Nachrichtenaustauschs in Form von Sequenzdiagrammen an. Alle übrigen können Zeitreihen erzeugen, RePast und JAS zusätzlich noch Balkendiagramme. Die Java-basierten Frameworks setzen dafür die freie

Bibliothek JFreeChart<sup>58</sup> ein. Soziale Netzwerkanalyse im engeren Sinne wird von keinem Framework unterstützt. RePast bietet aber schon die Möglichkeit, Kommunikationsbeziehungen grafisch anzuzeigen. Eine Analyse der Netzwerkparameter (z.B. Zentralität) ist aber nicht möglich. Auch NetLogo bietet mit der Klasse Link erste Vorbereitungen für Netzwerkanalysen, wobei diese Klasse noch im Betastadium ist. Bei Agententypen und Agentenkommunikation fehlen bei allen getesteten Kandidaten standardkonforme Implementierungen von deliberativen Architekturen und Kommunikationsprotokollen. Weder BDI- oder InterRaP-Agenten noch FIPA-ACL oder KQML Kommunikation wird unterstützt. Die Unterschiede in diesen beiden Punkten sind also nicht relevant. Organisatorische Abstraktionen werden ebenfalls kaum unterstützt. MadKit ist hier eine positive Ausnahme, welches mit Rollen, Gruppen und Communities die wichtigsten Konzepte bereitstellt. Bei den räumlichen Umgebungen ist die Unterstützung dagegen bei allen Kandidaten als gut zu bezeichnen. Zweidimensionale, begrenzte Umgebungen mit diskreten Raumschritten (`int`) sind überall möglich, mit Ausnahme von Breve das vollständig auf dreidimensionale Umgebungen setzt. Einige Frameworks bieten darüber hinaus noch kontinuierliche Umgebungen (`double`) und/oder unbegrenzte Umgebungen in Torus-Form. Bei der Dokumentation treten große Unterschiede zu Tage. Neben einer API-Dokumentation (oft Java-Doc) sind für den Benutzer ein Benutzerhandbuch und ein Entwicklerhandbuch hilfreich. Auch Tutorials, die schrittweise das Erstellen einer einfachen Beispielsimulation erläutern, sowie Beispielcode für einige Modelle helfen dem Einsteiger. Hervorzuheben ist bei diesem Kriterium NetLogo, das durch ein umfangreiches Handbuch hervorsteht und neben einer durchschnittlichen API-Dokumentation eine große Anzahl an Beispielen bereitstellt. Von diesem positiven Beispiel abgesehen, sind die Dokumentationen der übrigen Kandidaten alle mehr oder weniger ergänzungsbedürftig. Insbesondere fällt auf, dass die Dokumentation oft auf einem früheren Versionsstand als das Simulationsframework ist und nicht alle Klassen in gleichem Umfang beschrieben sind. Das Kriterium „Freie Programmierung“ listet schließlich auf, in welchen Programmiersprachen der Benutzer die Simulation erweitern kann oder ob nur eine grafische Anpassung möglich ist. Die Sprache Java ist dabei am meisten verbreitet. NetLogo und Breve bieten mit dem Logo-Dialekt und der objektorientierten Sprache `steve` wenig gängige Sprachen an. In SeSAM können zwar nur grafische Anpassungen

---

<sup>58</sup> <http://www.jfree.org/jfreechart>, letzter Abruf am 31.12.2007

vorgenommen werden, durch die Verwendung von UML-Diagrammen zur Spezifikation und der Turing-Vollständigkeit der SeSAM-Sprachen kann man aber trotzdem von freier Programmierbarkeit sprechen. Für Benutzer mit Programmiererfahrung erscheinen dem Autor jedoch Quelltexteditoren als produktivere Werkzeuge gegenüber der grafischen Variablen- und Funktionsspezifikation. Von RePast gibt es neben Java auch noch eine .NET Version, während MadKit die Spezifikation von Agenten in einer Reihe von Skriptsprachen erlaubt.

#### 5.4.4 Fazit

Wie der Vergleich zeigt, bieten einige der untersuchten Kandidaten zwar gute Ansätze, aber im Bereich der Agententypen und Agentenkommunikation, sowie den Organisationsabstraktionen und der sozialen Netzwerkanalyse ist kaum Unterstützung vorhanden. Neben dem Einarbeitungsaufwand in die Frameworks und der Unsicherheit bzgl. Fehlerkorrekturen seitens der Entwickler ist also ein hoher Anpassungsaufwand notwendig. Ein Teil der benötigten Funktionalität ist darüber hinaus durch Einbinden anderer freier Java-Bibliotheken wie COLT oder JFreeChart realisiert worden, was auch für eine von Grund auf selbst entwickelte Simulation möglich ist. Daher soll das Simulationsmodell in Java realisiert werden. Die Sprache Java ist zum einen sehr verbreitet (in der Wirtschaftsinformatik einerseits, v.a. aber auch in der Community der Agentenforscher, z.B. die Agentenplattformen Aglets, FIPA-OS, JACK, JADE, SAP, Tryllian und ZEUS), bietet eine große Klassenbibliothek mit wiederverwendbaren Klassen und ist leicht verständlich. Sie ist objektorientiert und unterstützt moderne Konzepte wie Interfaces, Fehlerbehandlung mit `Exceptions` und ab der Version 1.5 auch erweiterte Typsicherheit durch `Generics`. Eine automatische Speicherverwaltung erleichtert das Programmieren und mit der Java-Doc Funktionalität kann eine technische Dokumentation des Quellcodes schnell und einfach erzeugt werden. Java ist eine interpretierte Sprache, die durch die breite Verfügbarkeit der Laufzeitumgebung (Java Virtual Machine, JVM) auf nahezu allen gängigen Computerplattformen eine gute Wiederverwendbarkeit des Simulationsprogramms garantiert. Zudem sind eine Reihe von guten Entwicklungsumgebungen (z.B. Eclipse, Netbeans) frei verfügbar.

In die engere Wahl der zu verwendenden Programmiersprache wurde auch C# gezogen, da die Sprache außer der Plattformunabhängigkeit alle Vorzüge von Java aufweist und zudem das Konzept der `Delegates` kennt, eine Art typisierter Funktionszeiger, das für die Behandlung der Aktionen der Agenten gut eingesetzt werden könnte. Durch die

Entwicklung von Mono durch Novell ist mittlerweile auch die Verfügbarkeit auf nicht Windows-Plattformen zum großen Teil garantiert<sup>59</sup>. Der breite Einsatz von Java in der Agenten-Community und auch in der Wirtschaftsinformatik-Ausbildung sowie die bis vor kurzem kostenpflichtige Entwicklungsumgebung Visual Studio für C# hat jedoch zur Wahl von Java geführt.

Die Entscheidung zur Eigenentwicklung musste schon zu einem relativ frühen Zeitpunkt der Arbeit getroffen werden, da die Entwicklung viel Zeit in Anspruch nimmt. Der hier geschilderte Vergleich wurde Mitte 2006 noch einmal mit den zu diesem Zeitpunkt verfügbaren Versionen der Frameworks aktualisiert. Die Entscheidung für Java als Grundlage für die eigene Simulation wurde aber auf Basis der Anfang 2003 aktuellen und noch weniger weit entwickelten Versionen getroffen worden. In der Retrospektive und auf Grundlage der neuesten Versionen sowie den Erfahrungen mit der Eigenentwicklung würde bei einer erneuten Entscheidungssituation die Verwendung von JAS oder RePast<sup>60</sup> einer Eigenentwicklung vorgezogen worden wäre, da die Implementierung von nicht direkt agenten-bezogener Funktionalität wie Datenaufzeichnung und Auswertung mehr Zeit in Anspruch nahm als antizipiert.

### 5.5 Besonderheiten der Implementierung

Das Simulationsprogramm wurde in der objektorientierten Programmiersprache Java entwickelt. Es ist nicht zielführend sämtliche Details der Implementierung hier darzustellen. Im Folgenden werden daher nur einige Besonderheiten herausgegriffen, die nach Ansicht des Autors nicht ganz alltägliche Lösungen für die Herausforderungen der Entwicklung darstellen.

#### 5.5.1 Generics für das Gedächtnis

Zu Beginn der Entwicklung wurde die Java Version 1.4 verwendet. Nach Verfügbarkeit der Version 1.5 wurde auf diese gewechselt, da das neu eingeführte Konzept der Generics die Programmierung mit Containerobjekten wesentlich vereinfacht. Generics ist die Bezeichnung für einen Templating-Mechanismus, der in fast identischer Form schon in C++ mit der Standard Template Library (STL, z.B. Musser, Saini 1995) seit vielen Jahren existiert und auch in .NET seit Version 2.0 verfügbar ist (Kennedy, Syme 2001). Er

---

<sup>59</sup> <http://www.mono-project.com>, letzter Abruf 31.12.2007

<sup>60</sup> RePast ist im Dezember 2007 in einer neuen Version RePast Symphony erschienen

ermöglicht es, die von der Java Klassenbibliothek zur Verfügung gestellten Containerobjekte (z.B. `Vector`, `List`, `Collection`) bei der Definition und Instanzierung auf einen bestimmten Typ für die Inhalte festzulegen (Bracha 2004). Dadurch wird es möglich, schon zur Übersetzungszeit Typprüfungen durchzuführen und die Typsicherheit insgesamt deutlich zu erhöhen. Zusätzlich steigt auch der Komfort für den Entwickler, da nicht bei jedem Zugriff auf Objekte innerhalb des Containers Typprüfungen und Casts programmiert werden müssen. Diese Container, insbesondere der `Vector`, werden im Simulationsprogramm häufig eingesetzt und v.a. im Gedächtnis der Agenten, der `KnowledgeBase` verwendet. Aber auch an anderen Stellen werden typisierte Sammlungen von Objekten benötigt. So beinhaltet z.B. ein Projekt eine Sammlung von Arbeitspaketen und auch die Verwaltung der Agenten durch die `SimulationEngine` wird über die Klasse `AgentCollection` mit mehreren `Vector` und `Dictionary` Containern erledigt.

### **5.5.2 Agentenaktionen als innere Klassen**

Die Aktionen eines Agenten stellen eine besondere Herausforderung dar, da sie nicht als normale Methoden des Agenten definiert werden können, wenn eine Auswahl der Aktionen über einen Reasoning-Mechanismus ermöglicht werden soll. Es bietet sich also an, die Aktionen als eigene Klassen zu implementieren. Da jedoch zum Ausführen einer Aktion weitgehender Zugriff auf innere Zustände eines Agenten nötig ist und diese, um das Prinzip der Autonomie und „Selbstbestimmung“ eines Agenten nicht zu verletzen, auch nicht mit öffentlichem Zugriff versehen werden dürfen, muss ein anderer Mechanismus gefunden werden. Java kennt das Prinzip der inneren Klassen. Das sind Klassendefinitionen, die innerhalb einer anderen Klassendefinition stehen und dadurch Zugriff auf alle Membervariablen der umgebenden Klasse mit Sichtbarkeit `protected` erhalten (Gosling et al. 2005, S. 181-83). Noch genauer ausgedrückt handelt es sich bei den Aktionen um Member classes (Flanagan 2005, S. 142). Das bedeutet, dass die inneren Klassen mehrfach instanziiert werden können, im Gegensatz zu statischen inneren Klassen und lokalen Klassen. Agentenaktionen werden in der Planungseinheit des Agenten generiert und je nach Dringlichkeit nach dem FIFO Prinzip (First In First Out, Schlange) hinten an die schon früher geplanten Aktionen angehängt, oder nach dem LIFO Prinzip (Last In First Out, Stapel) vor die schon geplanten Aktionen eingereiht. Der Agent selbst führt dann nach Abschluss der aktuellen Aktion die jeweils nächste in der Reihe aus. Dadurch lässt sich das Problem der planbaren Aktionen recht elegant lösen.

### 5.5.3 Nutzenbasierte Aktionsauswahl mit Reflection

Die Auswahl der Aktionen selbst erfolgt in vielen Fällen über Wenn-Dann-Regeln, um den grundsätzlichen Arbeitsablauf abzubilden. Dies entspricht dem routinisiertem Verhalten ohne Vergleich von Alternativen (siehe Abschnitt 3.3.1). An einigen Stellen sind aber auch mehrere Aktionen gleichermaßen sinnvoll und es muss anhand des aktuellen Zustands des Agenten und des Nutzenbeitrags der Aktion entschieden werden, welche der möglichen Aktionen den größten Nutzen stiftet. Dazu wurde jeder definierten Situation eine Liste mit möglichen und sinnvollen Aktionen mitgegeben. Da eine Aktion aber eine innere Klasse ist und im Kontext der Agenteninstanz erzeugt werden muss, kann dies erst zur Laufzeit passieren. Zur Situation wird also eine Liste von Typen (Agentenaktionsklassen) gespeichert. Um diese Klassen zur Laufzeit instanzieren zu können, wird der Java Reflection Mechanismus genutzt. Mit Reflection können zur Laufzeit Informationen über die Eigenschaften von Objekten gesammelt werden (Sosnoski 2003). Darüber hinaus kann ohne Kenntnis des Klassennamens ein neues Objekt der Klasse erzeugt werden (Liang, Bracha 1998). Diese Art der Programmierung wird auch als Meta-Programmierung bezeichnet (Ullenboom 2005, S. 487).

```
AgentAction action = null;
Constructor[] con = possibleActions.get(0).getConstructors();
AgentAction bestAction = (AgentAction)
    con[0].newInstance(this.agent, this.agent);
for (int i=1; i<possibleActions.size();i++)
{
    Constructor[] con2 =
        possibleActions.get(i).getConstructors();
    action = (AgentAction) con2[0].newInstance(this.agent,
        this.agent);
    if (getUtility(bestAction.getExpectedContribution()) <
        (getUtility(action.getExpectedContribution())))
        bestAction = action;
}
return bestAction;
```

*Code-Beispiel 1: Nutzenbasierte Aktionsauswahl mit Reflection*

Wie im Code-Beispiel 1 zu sehen ist, wird zuerst der Constructor der Aktion ermittelt, dann mit diesem Constructor eine neue Instanz der Aktion erzeugt. Da jede Aktion einen Constructor definieren muss, der als einzigen Parameter den Agenten benötigt für den die Aktion ausgeführt werden soll, kann einfach der erste Constructor verwendet werden. Da es sich jedoch um innere Klassen handelt, ist der erste Parameter immer das Objekt, in

dessen Kontext die Klasse instanziiert werden soll. Mit der so erzeugten Aktionsinstanz kann dann der aktuelle Nutzensgewinn ermittelt werden, der aus dem Nutzenbeitrag der Aktion und der Gewichtung mit den aktuellen Zuständen des Agenten berechnet wird.

#### **5.5.4 Agenten als Thread**

Um die Forderung nach Autonomie des Agenten zu erfüllen, ist jeder Agent als eigener Thread implementiert. Dadurch können alle Agenten nebenläufig arbeiten. Um die rundenbasierte Simulation trotzdem synchronisieren zu können, werden mehrere logische Synchronisationspunkte definiert. Jeder Agent führt pro Runde einmal seine Situationserkennung, Aktionsplanung und anschließend die Aktion durch. Nach jedem Schritt wird gewartet, bis alle anderen Agenten ebenfalls fertig sind. Um außerdem keinen Agenten zu bevorzugen, wird nach jeder Runde die Reihenfolge, in der die Threads gestartet werden, nach dem Zufallsprinzip neu festgelegt.

Die Implementierung als Thread bringt aber für den Programmierer einige zusätzliche Schwierigkeiten mit sich, da er sich auch um die technische Synchronisierung kümmern muss. Das bedeutet, dass alle Methoden und Objekte, die von verschiedenen Threads gleichzeitig manipuliert werden können, mit dem `synchronized` Schlüsselwort versehen bzw. ihre Manipulation in `synchronized` Blöcke geschachtelt werden müssen.

Insgesamt ist Java aber gut geeignet nebenläufige Programme zu entwickeln. Auch die Verteilung auf mehrere Prozessoren oder CPU-Kerne wird in aktuellen Java-Versionen relativ gut gelöst. Im Test mit dem Dual Core Prozessor wurden beide CPU-Kerne gleichmäßig belastet. Allerdings wurde keine 100% Auslastung erreicht, sondern beide Kerne nur zu rund 65% ausgelastet, so dass sich nur geringe Performance-Vorteile im Gegensatz zum Einkernsystem gezeigt haben.

## **6 Simulationsstudien**

Im Folgenden werden die durchgeführten Simulationsstudien dokumentiert. Dazu werden zuerst die als Vorstudien bezeichneten Tests dargestellt, deren Ergebnisse maßgeblich für die implementierten Funktionen des Grundmodells sind. Anschließend werden die eigentlichen Studien durchgeführt. Dort wird ein WM-Instrument zur Umsetzung einer Personalisierungsstrategie, nämlich Skillmanagement, sowie eines zur Umsetzung einer Kodifizierungsstrategie, nämlich Erfahrungsdokumentation, untersucht (siehe Abschnitt 2.4.2). Abschließend werden die Ergebnisse der Studien ausgewertet und reflektiert.

Bevor die Simulationsstudien durchgeführt werden muss das entwickelte Simulationssystem zuerst entsprechend parametrisiert werden und das Modell auf Fehlerfreiheit und Plausibilität überprüft werden. Diese Punkte werden im Folgenden behandelt.

### **6.1 Parametrisierung**

Um die Aufgabe der Parametrisierung zu erleichtern, wurde in SimKnowledge versucht, wo immer möglich auf empirische Daten zurückzugreifen und die Parameter so zu gestalten, dass mit realen Größen gerechnet wird, statt mit abstrakten Werten. So wird z.B. das Gehalt der Mitarbeiter direkt in Euro angegeben, statt mit unspezifischen Geldeinheiten zu rechnen, so dass die Werte direkt aus Gehaltsstudien entnommen werden können. In den folgenden Abschnitten werden die empirischen Untersuchungen vorgestellt, aus denen sich die Parameter ableiten.

#### **6.1.1 Studien zu Skills**

Eine groß angelegte Studie in Nordamerika untersucht die im Studium vermittelten Wissensgebiete und deren Relevanz für die Praxis als IT-Professional (Lethbridge 1999). Die von Lethbridge abgefragten Wissensgebiete sind dabei sehr technisch orientiert und spiegeln die breite Ingenieurausbildung wider, die sich Studenten in den USA in einem Informatikstudium durch geschickte Kurskombinationen aneignen können, bzw. die breite der Abschlüsse, die bei Softwareentwicklern anzutreffen ist. So wirbt z.B. die bekannte Harvard Universität explizit damit, dass die Informatik eng mit den Elektrotechnik, Physik,

Chemie und Biologie Fakultäten verknüpft ist<sup>61</sup>. Die 75 definierten Wissensgebiete sind in die Bereiche Softwareentwicklungsprozess, Softwareentwurf, Softwareentwicklung, Softwareentwurf, Kernthemen, Software Subsystementwurf, Computersoftware und sonstige Softwarethemen, sowie theoretische Informatik, mathematische Grundlagen der Informatik, sonstige Mathematik, Computerhardware, sonstige Hardware, Basiswissenschaften, Betriebswirtschaft und Geisteswissenschaften mit Softskills eingeteilt. Zu den Themen zählen z.B. VLSI<sup>62</sup>, DSP<sup>63</sup> und Chemie, welche für die in SimKnowledge definierte Domäne keine Rolle spielen. Diese wurden aber auch in der von Lethbridge befragten Zielgruppe als unwichtig eingestuft. Befragt wurden Mitarbeiter von Firmen, die mit der Softwareentwicklung befasst waren. Die Firmen arbeiten zur knappen Hälfte in der Softwareindustrie und zum anderen Teil im Industriesektor sowie Bank, Handel und Telekommunikation. Ausgewertet wurden 212 vollständig ausgefüllte Fragebögen, was einem Rücklauf von 5% entspricht. Von den Teilnehmern besitzen rund die Hälfte einen Bachelor Abschluss, gut ein Drittel einen Master Abschluss und 6% einen PhD. Die Fachgebiete sind zu über der Hälfte Informatik (computer science), zu 17,5% Mathematik und zu rund 13% Softwaretechnik (software engineering) und weitere 15% auf andere Informatikstudiengänge (computer engineering, computing information systems). Die entwickelten Anwendungen umfassen MIS, Standardsoftware und eingebettete, bzw. Echtzeitsysteme. Gefragt wurde nach der Gewichtung der in der Ausbildung vermittelten Inhalte und der Praxisrelevanz. Die Wissensgebiete mit Wichtigkeit insgesamt  $\geq 3.0$  sind in Tabelle 6-1 zusammengefasst.

Die dargestellten Wissensgebiete erscheinen zum großen Teil gut für das geplante Szenario geeignet. Nur die Softskills Führungseigenschaften und professionelles Verhalten passen nicht zum Rest der Wissensgebiete, da sie kaum auf Faktenwissen aufbauen und daher wesentlich schlechter aus explizitem Wissen gelernt werden können. Auch Projektmanagement passt weniger in das SimKnowledge Szenario, da spezialisierte Projektmanager modelliert wurden, für die keine Skillbetrachtung erfolgt und die Projektmitarbeiter sich nicht um die Abläufe kümmern müssen.

---

<sup>61</sup> <http://www.seas.harvard.edu/research/computerscience.html>, für eine Übersicht der Informatik-Kurse siehe <http://webdocs.registrar.fas.harvard.edu/courses/ComputerScience.html>, letzter Abruf 28.12.2007

<sup>62</sup> VLSI: Very large-scale integration, hochintegrierte Schaltkreise auf einem Chip

<sup>63</sup> DSP: Digital signal processing, Digitale Signalverarbeitung

Tabelle 6-1: Wissensgebiete mit großer praktischer Bedeutung (nach Lethbridge 1999)

<b>Wissensgebiet</b>	<b>Bemerkung</b>
Betriebssysteme	Administrations- und Anwendungswissen. über das Betriebssystem, für das entwickelt wird
Datenbanken	SQL, Administration von Datenbanken, DB-Optimierung
Netzwerke	Netzwerkgrundlagenwissen zu den wichtigsten Protokollen (z.B. TCP/IP)
Programmiersprachen	Wissen über die von der Person am meisten genutzte Programmiersprache (z.B. C++)
Anforderungsanalyse	Wissen über die einzelnen Phasen des Softwareentwicklungsprozesses und Methoden zur Unterstützung einzelner Phasen.  Wissen über die einzelnen Phasen des Softwareentwicklungsprozesses und Methoden zur Unterstützung einzelner Phasen.
Analyse- und Entwurfsmethoden	
Softwareentwurf und -muster	
Test / Verifikation / Qualitätsmanagement	
Konfigurations- und Versionsmanagement	
Softwarearchitektur	Wissen über idealtypische Architekturen und deren Vor- und Nachteile
Datenstrukturen	Strukturierung von Daten sowohl im relationalen als auch objektorientierten Sinn.
Objektorientierte Konzepte	Grundlagen der Objektorientierung
Algorithmik	Grundlagen zum Entwurf neuer Algorithmen und Wissen über bestehende, oft verwendete Algorithmen
Benutzeroberflächen	Gestaltungsempfehlungen und Wissen über typisches Benutzerverhalten
Präsentationstechnik	In der Simulation z.B. für die Projektnachbesprechungen
technisches Schreiben	für Projektberichte, Handbücher, Spezifikationen, etc.
Projektmanagement	Verwaltung der Projektabläufe, etc.
Führungseigenschaften	Anleiten von Mitarbeitern, Softskill
Professionelles Verhalten	Kollegialer Umgang, Pünktlichkeit, etc. Softskill

Die Simulationsstudie (Hanakawa et al. 2002) verwendet als Wissensbereiche für Softwareentwickler Wissen über Programmiersprachen, Entwicklungsmethodik,



Für Skillmanagementsysteme in Organisationen werden oft weitaus detailliertere Wissensgebiete erfasst, damit bei Nachfragen oder der Teambesetzung direkt ein passender Mitarbeiter gefunden wird. So ist es in der betrieblichen Praxis nötig über die Tatsache hinaus, wie gut ein Mitarbeiter Programmieren kann auch noch zu wissen in welcher Programmiersprache, da sich zwar viele Konzepte übertragen lassen, die praktische Erfahrung in einer Sprache jedoch trotzdem einen großen Unterschied für die Produktivität und Fehlerfreiheit macht. Deswegen wird z.B. bei der Firma itCampus ein Skillbaum mit 3 Hierarchiestufen entwickelt und im Skillmanagementsystem eingesetzt, der auf der zweiten Stufe auf der selben Granularität ist wie die oben besprochenen Wissensbereiche (z.B. Programmiersprachen, Datenbanken, Betriebssysteme), welche dann jedoch noch fünf bis 15 konkrete Produkte oder Sprachen detailliert werden (siehe auch Anhang E).

Auch bei den c't Gehaltsstudien 2003-2005 (siehe Abschnitt 6.1.2) wurden die erhobenen Fertigkeiten eine Detaillierungsstufe feiner erhoben, als bei Lethbridge. Als Kategorien von Fertigkeiten werden dort Anwendungen, Betriebssysteme, Bibliotheken, Datenbanken, Hardware-Service, Netzwerke, Programmierung, Unternehmenssoftware und Web-Commerce genannt. Auf der feineren Ebene werden dann entweder konkrete Produkte (z.B. Solaris, Linux, Windows XP) oder Technologien genannt (z.B. XML/XSLT, TCP/IP, SQL).

Für das Simulationsmodell werden 13 Skills von Lethbridge (ohne Projektmanagement, Führungseigenschaften und professionelles Verhalten) übernommen.

### **6.1.2 Gehaltsstudien**

Bezüglich der Gehälter von IT-Mitarbeitern gibt es eine Reihe von Quellen, sowohl aus dem wissenschaftlichen, wie auch praxisnahen Bereich. Als wissenschaftliche Quelle ist z.B. die Studie von Scholz zu nennen (Scholz 1999). Diese wurde im Auftrag der Computerwoche durchgeführt, ist recht ausführlich beschrieben und daher gut nachvollziehbar. Im Praxisbereich führt z.B. die Computerzeitschrift c't regelmäßig zum Jahresende Online-Befragungen ihrer Leser durch, die dann nach der Auswertung im Folgejahr veröffentlicht werden (Apfelbaum 2006; Becher, Apfelbaum 2005; Meyer 2002; Meyer 2003; Meyer 2004). Ähnlich gut beschrieben wie die Scholz-Quelle sind die von der IG-Metall durchgeführten Befragungen (Krischer, Metall 2003; Krischer, Metall 2004). Vergleichbare Studien gibt es auch in den USA, die exemplarisch für den internationalen Vergleich herangezogen werden können (z.B. DICE 2003).

Interessant für die Simulation ist neben den absoluten Werten für die Parametrisierung v.a. auch die Langzeitentwicklung über die Jahre, da sie Anhaltspunkte für den Erfolg der oft jährlichen Gehaltsverhandlungen bzw. Leistungszulagen gibt. Die unterschiedlichen Studien setzen verschiedene Schwerpunkte. Die c't versucht z.B. neben einem Bundesländervergleich in den letzten Jahren auch eine Zuordnung von Gehältern auf Wissensbereiche zu erreichen. Die IG-Metall stellt traditionell auf die Eingruppierung in standardisierte Stellenbeschreibungen (z.B. Berater, Software-Ingenieur) und Erfahrungsstufen innerhalb der einzelnen Stellentypen ab (z.B. Junior-Berater, Berater, Senior-Berater, Chef-Berater). Scholz hingegen setzt seinen Schwerpunkt auf den Einfluss der Fachrichtung des Studiums auf den späteren Verdienst.

Die hier interessierenden Gehälter sind die der Berufsgruppe Softwareentwickler einerseits und der Projektmanager andererseits. Bei Scholz finden sich dafür Angaben von 47,9 T€ für Softwareentwickler (n=65), bzw. 61,6 T€ für Anwendungsentwickler (n=38). Projektmanager werden nicht getrennt ausgewiesen. Alle Angaben sind Bruttojahresgehälter inkl. aller monetären Zusatzleistungen in Tausend Euro bzw. Tausend Deutschen Mark vor der Euroeinführung. Bei der IG-Metall sind die Ergebnisse in Tabelle 6-2 aufgelistet.

Tabelle 6-2: Gehälter 2003 und 2004 nach IG-Metall

Stellentyp <sup>64</sup>	2003			2004		
	Min.	gew. Mittel	Max.	Min.	gew. Mittel	Max.
Software-Ingenieur I	33,4 T€	37,7 T€	45,3 T€	26,6 T€	40,5 T€	47,4 T€
Software-Ingenieur II	39,2 T€	51,7 T€	69,4 T€	35,4 T€	52,3 T€	59,8 T€
Software-Ingenieur III	43,4 T€	60,2 T€	78,0 T€	39,7 T€	61,0 T€	78,0 T€
Projekt-Ingenieur	48,1 T€	67,5 T€	86,4 T€	47,7 T€	173 T€	93,1 T€
Projekt-Leiter	60,6 T€	70,1 T€	87,5 T€	54,7 T€ <sup>65</sup>	67,7 T€	87,5 T€

Bei der Interpretation der Ergebnisse ist zu beachten, dass die Zuordnung zu den einzelnen Berufsgruppen, oder Stellen von den Befragten selbst vorgenommen wurde, so dass es dort zu Verfälschungen kommen kann. Insbesondere die Unterscheidung zwischen Projekt-

<sup>64</sup> Alle Angaben verstehen sich als geschlechtsneutrale Bezeichnung und gelten für Frauen wie Männer gleichermaßen.

<sup>65</sup> Hier wurde nur der fixe Anteil angegeben, da der Gesamtdurchschnitt mit 11.2 T€ falsch zu sein scheint.

Ingenieur (SW4) und Projekt-Leiter (SW5) scheint den Befragten nicht vollständig klar gewesen zu sein. Zudem scheint es in 2004 einen Fehler bei der Berechnung der SW5 Gehälter gegeben zu haben (siehe Fußnote).

Zusätzlich zu den detaillierten Werten finden sich in der Studie von 2004 auch Langzeitvergleiche. Der Vergleich für die Jobfamilie Software-Engineering ist in Abbildung 6-2 wiedergegeben. Man sieht, dass relativ starke Schwankungen von Jahr zu Jahr auftreten, insgesamt aber ein moderater Aufwärtstrend zu beobachten ist. Die in der Legende angegebenen Kürzel SW1 bis SW5 entsprechen den Stellentypen aus Tabelle 6-2.

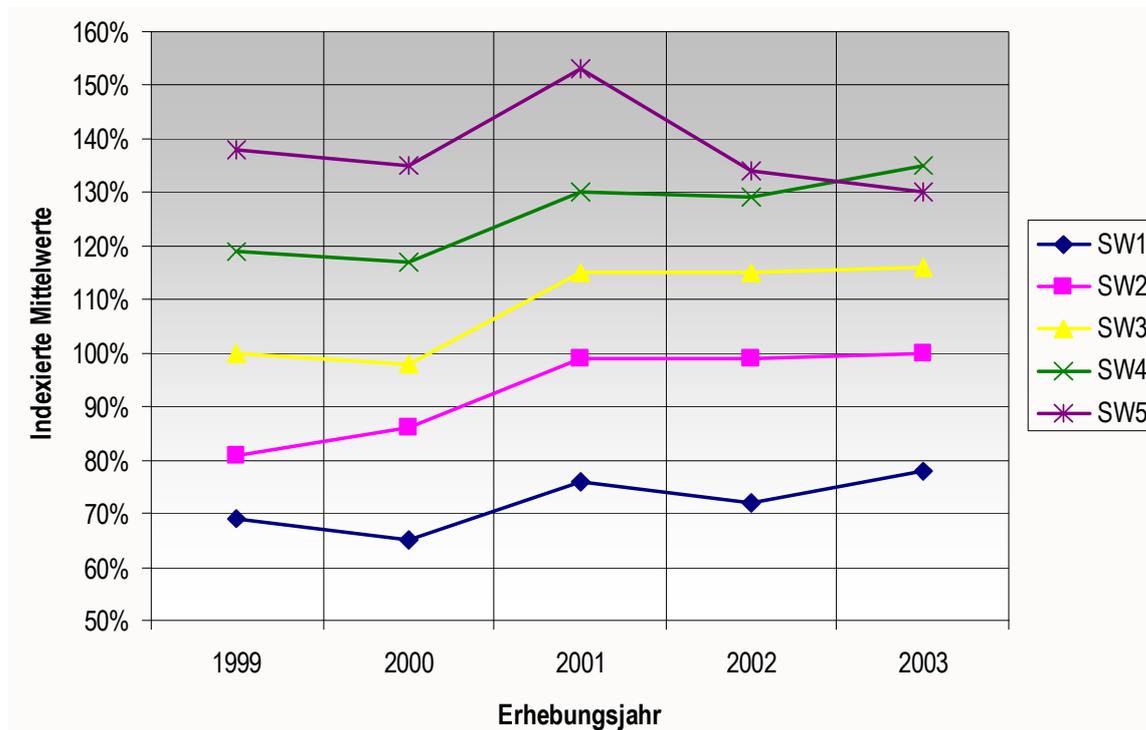


Abbildung 6-2: Langzeitvergleich der Gehälter im Software-Engineering nach IG-Metall

In Tabelle 6-3 und Tabelle 6-4 sind die Ergebnisse der c't Gehaltsstudien zusammengefasst. Leider wurde das Design nach der Studie 2002 überarbeitet, so dass die Ergebnisse nur bei Datenbank-Programmierer, Software-Programmierer und Projektleiter durchgängig vergleichbar sind. Im Vergleich zu den IG-Metall Studien fällt auf, dass die Gehälter insgesamt niedriger liegen. Dies könnte daran liegen, dass die von der IG-Metall Befragten vermutlich alle zumindest tariflich bezahlt werden und damit die Minimalgehälter höher sind. Ein anderer Grund könnte sein, dass in der c't Studie die Berufserfahrung nur in Bezug zur Ausbildung dargestellt wird, nicht aber nach Stelle, die ausgewiesenen Durchschnittsgehälter also einen Querschnitt aller Berufserfahrungen enthalten, während bei der IG-Metall die Berufserfahrung implizit in den

Stellenbezeichnungen (I, II, III) die Berufserfahrung mit enthalten ist. Dementsprechend ist das ermittelte Durchschnittsgehalt für Software-Ingenieur I etwas geringer als für die entsprechenden Stellentypen in den c't Studien, wogegen SW2 und SW3 wesentlich höhere Durchschnittsgehälter aufweisen als in den c't Studien.

Tabelle 6-3: Gehälter 2001 und 2002 nach c't

	2001	2002
DB-Programmierer	25,6 T€ 42,7 T€ 74,1 T€ (182)	19,9 T€ 42,0 T€ 80,0 T€ (121)
DB-Ingenieur	26,6 T€ 48,3 T€ 70,6 T€ (47)	–
Software- Programmierer	25,3 T€ 44,0 T€ 76,7 T€ (104)	19,5 T€ 42,0 T€ 76,7 T€ (687)
Software-Ingenieur	25,3 T€ 49,7 T€ 76,8 T€ (960)	36 T€ 49,8 T€ 80,0 T€ (478)
Projektleiter	26,6 T€ 53,9 T€ 76,7 T€ (261)	25,1 T€ 58,4 T€ 80,3 T€ (155)

In den c't Studien ab 2003 wird auch versucht einen Zusammenhang zwischen Fertigkeiten und Gehältern herzustellen. Dazu wird in den oben genannten Kategorien jeweils die Fertigkeit mit dem höchsten Durchschnittsgehalt mit den drei meistgenannten Fertigkeiten gegenübergestellt. Es ergeben sich Unterschiede von rund zehn Prozent zwischen Maximum und den meistgenannten Fertigkeiten, aber auch zwischen den meistgenannten Fertigkeiten liegen zum Teil mehr als zehn Prozent Gehaltsunterschied. Die Tendenz bleibt über die Jahre gleich, auch wenn die Abstände schwanken. Dennoch lassen sich aus den Statistiken keine Gesetzmäßigkeiten ableiten. Meist sind die Fertigkeiten mit dem höchsten Gehaltsdurchschnitt auch die mit den wenigsten Vertretern. Jedoch sind z.B. auch Mitarbeiter mit Oracle Datenbankkenntnissen wesentlich besser bezahlt als Mitarbeiter mit SQL Server Kenntnissen, obwohl Oracle Kenntnisse unter den Befragten auch am meisten vertreten sind. Der Vergleich mit den USA zeigt, dass dort in 2002 die Gehälter im Schnitt nur geringfügig höher lagen (DICE 2003). Dort lag das Einstiegsgehalt bei knapp 53 T\$ (50,3 T€)<sup>66</sup> und wuchs bis zum achten Berufsjahr auf 76 T\$ (72,2 T€). Noch weiter steigende Berufserfahrung führte zu einer Seitwärtsbewegung des Gehalts, allerdings mit beträchtlicher Schwankungsbreite (80,5 T\$ max und 74,8 T\$ min). Insgesamt ergibt sich für die 45.906 Befragten ein Durchschnittsgehalt von 71 T\$ (67,4 T€) bei einer mittleren Berufserfahrung von 4,7 Jahren.

<sup>66</sup> Laut <http://waehrungskurs.de> lag der Dollarkurs 2002 im Mittel bei 0,95 (1 € = 0,95 USD)

Tabelle 6-4: Gehälter 2003 bis 2005 nach c't

	<b>2003</b>	<b>2004</b>	<b>2005</b>
Datenbank-Programmierer	38,7 T€	41,1 T€	44,8 T€
Software-Programmierer	41,3 T€	42,4 T€	43,9 T€
Anwendungsentwickler	42,2 T€	42,8 T€	43,4 T€
Tester	41,7 T€	46,8 T€	50,2 T€
Systementwickler	49,0 T€	49,2 T€	50,5 T€
Software-Architekt	54,2 T€	53,4 T€	52,4 T€
Projektleiter	59,2 T€	56,7 T€	57,5 T€

Die neueren Daten aus 2006 zeigen ein geringfügig gestiegenes Durchschnittsgehalt in Landeswährung von 72,6 T\$ (58,1 T€)<sup>67</sup>, was sich aber aufgrund des rapide gefallenen Währungskurses in einer deutlichen Absenkung des Eurodurchschnitts auswirkt. Die Ergebnisse beziehen sich auf Antworten von 74419 Befragten mit einer mittleren Berufserfahrung von 5,13 Jahren. Auch die Einstiegsgehälter lagen mit 54 T\$ (43,2 T€) geringfügig höher als 2002, was aber prozentual nicht einmal 2% ausmacht und somit deutlich unterhalb der Inflationsrate liegt<sup>68</sup>.

Für die Simulation wurden die Monatsgehälter aus den Durchschnittsgehältern der Softwareingenieure II der IG Metall und den Projektleitern errechnet, wodurch sich rund 4400€ Brutto für die Projektmitarbeiter und knapp 6000€ Brutto für Projektleiter ergab.

### 6.1.3 Dauer-Aufwands-Relationen

Moore untersucht die Zusammenhänge zwischen Projektbesetzungsrate (MBR, Manpower Buildup Rate), den Fertigkeiten der Projektmitarbeiter und der Projektdauer (Moore 1999). Er baut dafür auf die Rayleigh Kurve auf, die besagt, dass ein Projekt im Zeitverlauf anfangs wenig, dann schnell mehr und gegen Ende wieder weniger Personal benötigt (ibid. S. 6\_5). Diese wurde zur Putnam Norden Rayleigh Formel weiter entwickelt, welche einen direkten Zusammenhang zwischen dem Projektaufwand und der vierten Potenz der Zeit postuliert.

<sup>67</sup> Laut <http://waehrungskurs.de> lag der Dollarkurs 2006 im Mittel bei 1,25 (1 € = 1,25 USD)

<sup>68</sup> Laut <http://www.bls.gov/cpi/home.htm> lag die kumulierte Inflationsrate von 2002 bis 2006 bei 12%

$$(16) \text{ Aufwand} = \left[ \frac{S * B^{\frac{1}{3}}}{p} \right]^3 * \left( \frac{1}{t^4} \right)$$

Dabei ist S die Projektgröße in Codezeilen (LoC, Lines of Code), B ein spezifischer Skillfaktor, der für verschiedene Klassen von Projektgrößen zwischen 0,16 (bis 15k LoC) und 0,39 (über 50k LoC) schwankt. Die Variable p steht für einen Produktivitätsparameter, der die Produktivität der Organisation insgesamt misst und nicht nur von Mitarbeitern, sondern auch eingesetzten Werkzeugen und Prozessen abhängt, wogegen t die Projektlaufzeit darstellt. Die Zusammenhänge zu den übrigen Variablen sind in Form eines Netzwerk-Modells in Abbildung 6-3 dargestellt.

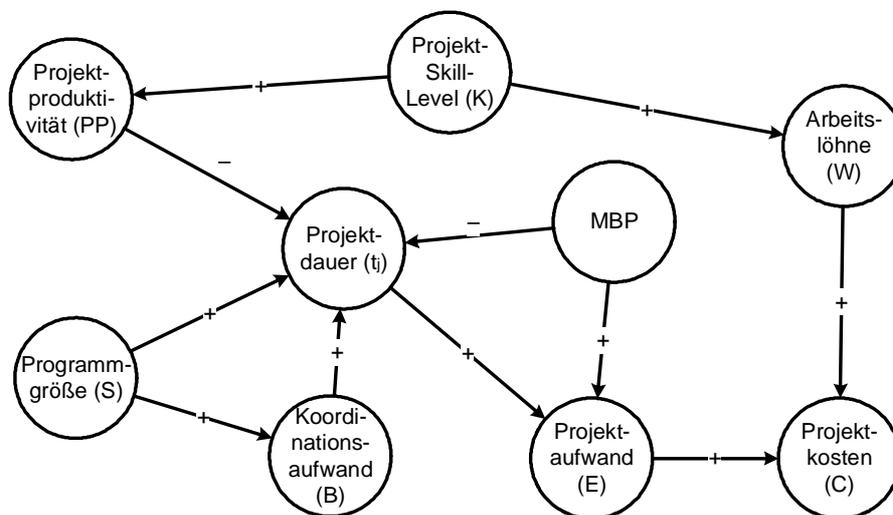


Abbildung 6-3: Zusammenhänge zwischen Skill und Projektdauer (Moore 1999, S. 6\_19)

Im Simulationsmodell ist die Produktivität darauf aufbauend folgendermaßen abgebildet. Jedes Arbeitspaket in einem Projekt definiert einen erforderlichen Skillwert, der zum Abarbeiten des Arbeitspakets notwendig ist. Hat ein Agent genau den erforderlichen Skillwert, so benötigt er zum Abarbeiten genau die vom Arbeitspaket definierte Zeit mit geringen Schwankungen durch stochastische Einflüsse. Ist sein Skillwert höchstens 0,5 (auf einer Skala von 0-5) geringer als erforderlich, so kann er trotzdem Fortschritte erzielen, ist aber langsamer als wenn er den erforderlichen Skillwert besitzt. Ist sein Skillwert höher als erforderlich, so kann er mit seiner Arbeit schneller fertig werden. Die Beschleunigung oder Verlangsamung verläuft dabei linear.

$$(17) \text{ Projektfortschritt}_{\text{Runde}} = (1 + \text{Skill}_{\text{actual}} - \text{Skill}_{\text{required}}) * \text{Projektfortschritt}_{\text{normal}} \\ \forall \text{Skill}_{\text{actual}} \in \{\text{Skill}_{\text{required}} - 0,5; \text{Skill}_{\text{required}} + 0,5\}$$

### 6.1.4 Fallstudie itCampus

Die itCampus Software- und Systemhaus GmbH (im Folgenden kurz itCampus) ist ein mittelständischer IT-Dienstleister mit rechnerisch 52 Vollzeitkräften<sup>69</sup> und Sitz in Leipzig. Die Firma hat im Geschäftsjahr 2003 2,2 Mio. € Umsatz mit Projektgeschäft gemacht, wovon der überwiegende Anteil Dienstleistungsvergütung war und nur ein kleiner Teil Softwarelizenzen und Hardware. Es wurden rund 50 Projekte abgewickelt, wovon fünf ein Volumen von über 100 T€ und 10 ein Volumen von 50-100 T€ hatten. Es handelte sich um Softwareentwicklungen in verschiedensten Branchen im deutschsprachigen Raum mit rund 40% Anteil in der Region Halle/Leipzig. Die Mitarbeiter sind zwischen 20 und 35 Jahren alt und haben zum großen Teil (57,8%) ein Hochschulstudium absolviert. Sie besitzen durchschnittlich 3,5 Jahre Berufserfahrung und sind seit 2,8 Jahren bei itCampus beschäftigt. Die Fachrichtungen der Mitarbeiter verteilen sich auf Wirtschaftsinformatik, Informatik, sowie Medien und Design. Sie haben in der Zeit 5,8 Projekte mit einer Laufzeit von 10 Monaten abgeschlossen. Das bedeutet, dass nur wenige Projekte parallel bearbeitet wurden (Überlappung zwischen Arbeitsmonaten pro Projekt und angegebener Projektlaufzeit: 1,36). Die Projekte werden überwiegend mit 2 bis 5 Mitarbeitern bearbeitet. Zwei große Projekte mit 10, bzw. 15 Mitarbeitern heben den Durchschnitt von 3,33 ( $\sigma$  1,07) auf 4,61 ( $\sigma$  3,25).

Zur Überprüfung einiger Ergebnisse von Lethbridge 1999 und Teilen der finnischen Adult Education Surveys wurden im Mai 2004 ein strukturiertes Interview mit dem Geschäftsleiter durchgeführt und anschließend im Juni 2004 ein Fragebogen an alle Mitarbeiter verteilt, die ins Projektgeschäft eingebunden sind. Dadurch konnten einige zusätzliche Daten zur Wissensweitergabe erhoben werden, die als Ausgangswerte für Parameter der Simulation dienen.

**Ergebnisse des Interviews:** Gezielt gefördert werden von der Unternehmensleitung Weiterbildung in den Programmiersprachen Java und C#, sowie sog. Softskills (z.B. Konflikt- oder Zeitmanagement). Andere Fertigkeiten werden projektgetrieben geschult. Als Weiterbildungsmaßnahmen werden Seminarteilnahmen angeboten und interne Schulungen durchgeführt, Bücher und Zeitschriften gekauft, sowie uneingeschränkter Internetzugang zur Verfügung gestellt.

---

<sup>69</sup> 36 festangestellte Mitarbeiter, 7 freiberufliche Mitarbeiter und 18 studentischen Hilfskräften

An gezielten WM-Maßnahmen wird in erster Linie Skillmanagement eingesetzt. Ein selbst entwickeltes SkMS wurde sechs Monate vor dem Interview eingeführt. Einmal jährlich werden Mitarbeitergespräche geführt, in denen auch Lernziele festgehalten werden. Es gibt allerdings keine Erfolgskontrolle. Informeller Wissensaustausch wird gefördert, indem im Essensraum gemeinsam zu Mittag gegessen wird. Es gibt gemeinsame Freizeitaktivitäten wie Grillen und Netzwerkspiele, einen 3-tägigen Firmenausflug pro Jahr und einen Entspannungsraum im Keller, mit Gelegenheit zum Flippern und Kickern. Büros sind mit zwei bis vier Personen besetzt. Es wird auch kodifiziertes Wissen in Form von Projektdokumentation auf Dateiservern hierarchisch nach Kunden gegliedert abgelegt.

Die Personalpolitik ist bisher nicht direkt auf Wissensmanagement ausgerichtet, da kaum Fluktuation besteht und neue Mitarbeiter meist auf Bedarf durch neue Projekte eingestellt und aus vorherigen studentischen Hilfskräften rekrutiert.

**Fragebogendesign:** Der Fragebogen orientiert sich an zwei Vorbildern: zum einen die schon erwähnte Studie von Lethbridge (siehe Abschnitt 6.1.1) und zum anderen eine Liste der Skills aus dem Skill Management-System von itCampus. Die Abfrage der Skills stellt auch den zentralen Teil des Fragebogens dar (Teile II bis IV). Dazu wurden die Skills von itCampus neu geordnet, um sie in etwa mit den von Lethbridge definierten Skills in Einklang zu bringen. Auch die Studie der c't (Meyer 2003), in der 2003 erstmalig auch Skills erhoben wurden, sowie die Arbeit von Scholz (2000) wurden bei der Neuordnung berücksichtigt. Es sollten dabei die 16 ausgewählten der 70 Skills von Lethbridge vertreten sein (v.a. Frage 30 – Software-Entwicklungsprozess), das Granularitätsniveau so niedrig liegen, dass konkrete Antworten und gute Einschätzungen möglich sind (z.B. bei Datenbanken und Betriebssystemen), sowie alle befragten Mitarbeiter von itCampus sich wieder finden. Dazu wurden zwei Bereiche (Fragen 27 und 34, „Grafik- und Videosoftware“ sowie „Design von grafischen Benutzeroberflächen“) detaillierter mit aufgenommen, als man das als neutraler Beobachter anhand der Wichtigkeit für die Softwareentwicklung einschätzen würde. Auch ein kleiner Teil für Projektleiter (Frage 38 – Projektmanagement) wurde mit aufgenommen, da es bei itCampus keine strikte Trennung zwischen Projektleitern und Projektmitarbeitern gibt, wie das im Simulationsmodell vereinfachend angenommen wird. Darüber hinaus wurde versucht die einzelnen Kategorien im Rahmen des beschränkten Platzes so vollständig wie möglich zu machen, um Verzerrungen zu vermeiden. Dafür wurden wiederum bestehende empirische Studien herangezogen, z.B. (Albrecht, Schweizerische Vereinigung für Datenverarbeitung 2000)

für die Projektmanagement Skills (S. 62-67 und 248), bzw. Produktinformationen von Herstellerwebseiten verwendet (z.B. Frage 37 – Module betrieblicher Standardsoftware). Die Fragen 26 und 37 zu betrieblicher Standardsoftware, sowie 25 und 33 (Infrastrukturserver und Internet Markup Sprachen) wurden wegen der gestiegenen Bedeutung in den letzten Jahren (vermehrter Einsatz von SAP R/3 auch im Mittelstand, Bährle 2006), sowie der speziellen Ausrichtung des befragten Unternehmens mit aufgenommen.

Der Teil I des Fragebogens fragt Daten zum Lernen am Arbeitsplatz und in der Freizeit, sowie zur Wissensweitergabe unter Kollegen ab, da diese Daten zentral für die Simulation sind und dem Autor keine empirischen Studien bekannt sind, die diesen Bereich abdecken. Im letzten Teil (Teil V) werden schließlich demographische Daten, sowie Daten zur aktuell ausgeübten Tätigkeit erhoben. Der Fragebogen wurde an die 36 festangestellten Mitarbeiter ausgegeben. Davon wurden 19 Fragebögen ausgefüllt (52,7%).

**Ergebnisse des Fragebogens:** Kenntnisse über den Softwareentwicklungsprozess und bestimmter Softwareprogramme wurden als besonders wichtig eingestuft (Mittelwert 4,32 auf einer Likert-Skala von 5 – sehr wichtig bis 1 – unwichtig). Programmierkenntnisse wurden immer noch überwiegend für wichtig oder sehr wichtig eingestuft (Mittelwert 4,21), während Fremdsprachen-, Branchen-, und Netzwerkkennnisse als weniger wichtig erachtet wurden (Mittelwerte 3,95, 3,53 und 3,26).

Als Wissensquelle zur Lösung von Problemen, die während der Projektarbeit auftreten wird am häufigsten das Internet genutzt, nämlich häufig (jede Woche). An zweiter Stelle stehen Kollegen, von denen auch mindestens 1-3x pro Monat oder sogar wöchentlich Ratschläge eingeholt werden. Fast genauso häufig werden Bücher zum Nachschlagen verwendet. Zeitschriften und Experimentieren werden nur rund einmal monatlich zur Weiterbildung verwendet, während das Intranet und interne sowie externe Schulungen selten genutzt wurden. Zur Weiterbildung außerhalb konkreter Projekte sieht die Verteilung sehr ähnlich aus, mit dem Unterschied, dass Zeitschriften Bücher als dritthäufigste Wissensquelle ablösen.

Zum erfolgreichen Abschließen eines Projekts verwenden je sieben Befragte bis 10% und 11-20% der Projektarbeitszeit auf Weiterbildung. Fünf Befragte geben sogar einen Bedarf von 21-30% der Projektarbeitszeit an. Zur Weiterbildung außerhalb konkreter Projektvorhaben werden im Monat rund acht Stunden Arbeitszeit ( $\mu$  7,86,  $\sigma$  4,89) und über 16 Stunden Freizeit aufgewendet ( $\mu$  16,91,  $\sigma$  14,9).

---

Als Motivation zur Weiterbildung in Arbeits- und Freizeit wird in erster Linie privates Interesse an den Inhalten angegeben (100% Zustimmung), sowie die Erwartung das nächste Projekt mit geringerem Aufwand erledigen zu können (88% Zustimmung). Respekt der Kollegen, Aufstiegschancen, oder finanzielle Gründe werden sehr unterschiedlich bewertet mit vielen Unentschiedenen, was auch darauf hindeuten kann, dass diese Motive nicht offen gelegt werden wollen.

Die Befragten fragen haben je zwei bis zehn Personen im letzten Monat um Rat gefragt ( $\mu$  5,52,  $\sigma$  2,62). Dies kann auch als Hinweis auf die Größe des sozialen Netzes gewertet werden. Es werden in der Regel nur Ideen des Kollegen benötigt, um das Problem lösen und die Projektarbeit fortsetzen zu können (14 Nennungen, 73,6%). Oft reicht sogar Reden über das Problem zur Problemlösung aus, ohne dass der Kollege einen direkten Lösungsbeitrag liefert (13 Nennungen, 69,4%, siehe auch „thinking along“ bei Berends in Abschnitt 3.1). Eine vollständige Erläuterung des Lösungswegs oder direktes Zeigen der Lösung am Produkt kommen dagegen nur selten vor (4 bzw. 2 Nennungen).

Gefragt werden in erster Linie Kollegen, manchmal auch Freunde, nur selten Kunden oder Mitarbeiter von Partnerunternehmen. Von den Kollegen werden v.a. räumlich nahe sitzende oder Teamkollegen gefragt (77,8% bzw. 100% Zustimmung), wogegen andere Kollegen aus der Abteilung, aus demselben Büro oder vom anderen Standort signifikant weniger gefragt werden (99,9% Signifikanz). Weiterhin werden frühere Hilfeleistungen der Kollegen sowie das Meta-Wissen, dass der Kollege über das benötigte Wissen verfügt stärker als Auswahlkriterium benutzt als eine Freundschaftsbeziehung (99% Signifikanz). Meta-Wissen wird in erster Linie aus gemeinsamer Projektarbeit bezogen und fast genauso häufig aus privaten Gesprächen. Empfehlungen von anderen Kollegen werden nur manchmal genutzt, am seltensten das Skillmanagementsystem.

Bei der Selbsteinschätzung zum Wissen in verschiedenen Gebieten zeigt sich deutlich eine Zweiteilung in eine Gruppe von Grafikdesignern (4 Personen) und die übrigen Mitarbeiter. In allen Gebieten außer Office-Anwendungen zeigen sich signifikante Unterschiede zwischen den beiden Gruppen (Signifikanzniveau 0,001). Während die Grafikdesigner die Wissensgebiete „Grafik- und Videosoftware“ und „Entwicklung grafischer Benutzeroberflächen“ gut beherrschen (gewichteter Mittelwert 3,68 bzw. 3,79) sind die anderen Mitarbeiter hauptsächlich in den Gebieten „Software-Entwicklungsprozess“, Programmiersprachen und „Internet Markup Sprachen“ gut geschult (gewichtete Mittelwerte 3,14, 3,37 und 3,76). Innerhalb der einzelnen Gebiete sind auch Produktwissen über Windows und

das DBMS MySQL stark ausgeprägt ( $\mu$  3,40 und 3,27). Innerhalb der Softwareentwicklung ist Wissen über „Datenbankdesign und ER-Modellierung“ besonders verbreitet ( $\mu$  3,33). Schließlich ist Wissen über das Netzwerkprotokoll http ebenfalls verbreitet bei den Befragten ( $\mu$  3,13). Wissen über Office-Anwendungen ist in beiden Gruppen mit Ausnahme der Desktopdatenbank ( $\mu$  1,50 bzw. 2,67) hoch eingeschätzt ( $\mu$  3,25 bis 3,80). Besonders wenig ausgeprägt ist dagegen Wissen über betriebliche Standardsoftware und ERP-Module ( $\mu$  0,83 und 1,13).

Insgesamt liegen die Einschätzungen bei der Gruppe der Grafikdesigner bei Werten von maximal 3,79, im Durchschnitt 1,72, und minimal 0,38. Bei den restlichen Mitarbeitern („Softwareentwicklern“) bei maximal 3,76, im Durchschnitt 2,42 und minimal 0,83.

### **6.2 Kalibrierung**

Um eine hohe Nachvollziehbarkeit der Ergebnisse der Simulation zu erreichen, soll das Simulationsmodell so einfach wie möglich sein. Das betrachtete Gebiet der Wissensweitergabe ist jedoch, wie in Kapitel 2 dargestellt, so komplex, dass eine so hohe Abstraktion, wie sie in vielen anderen Simulationsmodellen realisiert wird (z.B. Segregation Modell) nicht sinnvoll erscheint (siehe dazu auch die Diskussion zu rigor vs. relevance Benbasat, Zmud 1999; Davenport 1999). Um diese Aussage weiter zu belegen, werden im Folgenden Ergebnisse aus Studien mit Vorversionen des später präsentierten Grundmodells dokumentiert. Diese zeigen, dass sich mehrere grundlegende Probleme aus zu hoher Abstraktion ergeben, die dazu führen, dass nur sehr eingeschränkt Rückschlüsse auf die betriebliche Realität gezogen werden können. Dennoch lassen sich auch hier schon einige interessante Erkenntnisse ableiten, die sich jedoch mit in der Literatur dokumentierten Erfahrungen decken.

#### **6.2.1 Projektarbeit ohne Lernen**

Die Minimalversion des Simulationsprogramms beinhaltet nur die Projektbearbeitung, ohne jegliche Lernkomponente. Projektmitarbeiter können in diesem Fall nur Projektarbeit verrichten oder nichts tun (`IdleAction`), neben den obligatorischen Möglichkeiten zur Kommunikation mit dem Projektmanager, um neue Arbeitspakete anzunehmen oder abzulehnen und fertig gestellte Arbeitspakete zu melden. Die Skills werden zu Beginn der Simulation mit einem Zufallswert initialisiert und verändern sich nicht.

Dabei entsteht das Problem, dass Arbeitspakete, die wegen mangelnden Wissens nicht erfolgreich bearbeitet werden können, nie fertig werden und daher den Mitarbeiter auf

Dauer blockieren. Die noch gravierendere Auswirkung davon ist aber, dass auch der Projektmanager blockiert bleibt, da das gesamte Projekt erst dann fertig ist, wenn alle Arbeitspakete fertig gestellt sind. Dadurch kann dieser PM kein neues Projekt annehmen und es sind zwei weitere Mitarbeiter ohne Arbeitspaket, da sie mit ihrem vorherigen Arbeitspaket fertig sind, aber kein neues mehr bekommen.

### **6.2.2 Erweiterung um Lernen bei der Arbeit (learning by doing)**

Es muss also eine Lernkomponente eingeführt werden, damit Wissenslücken ausgeglichen werden können. Dafür wird Lernen bei der Projektarbeit vorgesehen. Das bedeutet, dass innerhalb gewisser Grenzen<sup>70</sup> durch Versuch und Irrtum trotz bestehender Wissenslücken Fortschritte bei der Projektarbeit erzielt werden können. Ein Fortschritt führt dabei auch zu einer Steigerung des Wissens. Auch wenn das vorhandene Wissen knapp über dem benötigten liegt wird unterstellt, dass durch Festigen des bestehenden Wissens kleine Wissenszuwächse erzielt werden können.

Durch diese Erweiterung wird das Problem der nicht abgeschlossenen Arbeitspakete leicht abgemildert. Jedoch steigt das Wissen der Mitarbeiter ungebremst, da es keine Gegenkraft gibt, die zum Sinken führen kann. Umgekehrt haben Wissensbereiche mit geringem Startwert keine Möglichkeit zur Steigerung, da Lernen bei der Projektarbeit nur innerhalb gewisser Schranken über und unterhalb des erforderlichen Wissensstands zugelassen wird.

Als wichtiger Einflussfaktor stellt sich die Strategie heraus, nach der Projektmitarbeiter ein Arbeitspaket annehmen oder ablehnen. Wird zu schnell angenommen, so führt dies tendenziell dazu, dass Mitarbeiter ihre Arbeitspakete nicht fertig stellen können. Wird nur bei ausreichend hohem Wissensstand angenommen, so brauchen Projektmanager lange, um alle erforderlichen Projektmitarbeiter zu rekrutieren. Zusätzlich bleiben Mitarbeiter mit geringem Startwissen lange Zeit oder sogar auf Dauer ohne Arbeitspaket. Als geeigneter Kompromiss hat sich die Strategie erwiesen, zuerst nur Arbeitspakete anzunehmen, für die ausreichendes Wissen vorliegt, nach einiger Zeit der Untätigkeit aber diese Schranke zu senken oder ganz aufzugeben. Diese Strategie ist in die Entscheidungsfunktion zur Annahme von Arbeitspaketen integriert und fließt nicht direkt in die Nutzenfunktion ein, die für die Aktionsauswahl verwendet wird. Sie kann jedoch mit der Nutzenfunktion erklärt werden, da bei längerer Untätigkeit des Agenten die antizipierte Gefahr der Entlassung steigt und damit der Grenznutzen der Projektarbeit zunimmt.

---

<sup>70</sup> Hier wurden als Grenzen 0,5 unterhalb und 0,3 oberhalb des benötigten Wissens festgelegt.

### **6.2.3 Erweiterung um Lernen am PC**

Um Mitarbeitern die Möglichkeit zu bieten das Wissen in solchen Bereichen zu verbessern, die sehr gering ausgeprägt sind wurde Lernen am PC eingeführt (Oliver et al. 2003, S. 129). Aus Gründen der Einfachheit und der immer weiter wachsenden Informationsmenge im Internet wurde dies stellvertretend für alle Arten von Lerninhalten die in textueller, grafischer und audiovisueller Form dargeboten werden, ohne große Interaktivität zu erlauben. Es wird also nicht zwischen Büchern in Papierform, E-Books, WebCasts und anderen elektronischen Inhalten (z.B. Foren, Hilfedateien) unterschieden. Es wird dabei unterstellt, dass bei hinreichend langer Beschäftigung mit einem Thema am PC auch ohne echte Praxiserfahrung ein gewisser Wissensstand erreicht werden kann<sup>71</sup>.

Das Einführen dieser neuen Lernmöglichkeit löst das Problem von dauerhaft niedrigem Wissen in einigen Wissensgebieten. Zugleich wird das Problem der kontinuierlich steigenden Wissens verschärft, da die vorher durch „Nichtstun“ belegte Arbeitszeit teilweise durch Lernen am PC ersetzt wird. Besonders auffällig ist, dass Mitarbeiter ohne Projektarbeit bis zu dem festgelegten Schwellenwert, der ohne Praxiserfahrung erreicht werden kann, mehr durch Lernen am PC dazu lernen, als durch Projektarbeit. Diese Beobachtung scheint zwar auf den ersten Blick den Erfahrungen in der Realität zu widersprechen, kann aber bei genauerer Analyse nicht von vorneherein als falsch ausgeschlossen werden. Bei der Projektarbeit wird nämlich ein gewisser Prozentsatz der Arbeitszeit auf Routinetätigkeiten verwendet, bei denen kaum dazu gelernt wird. Bei gezieltem Lernen am PC kann jedoch in mehreren Wissensgebieten relativ schnell ein gewisses Grundwissen erworben werden.

### **6.2.4 Erweiterung um Vergessen**

Um ein stetiges Ansteigen des Wissens zu vermeiden und den Stellenwert von Zeit als knappem Gut zu unterstreichen wurde Vergessen eingeführt. Wird Wissen in einem Gebiet über einen längeren Zeitraum nicht mehr benutzt (z.B. 25 Tage, siehe Abschnitt 6.3), so beginnt der Mitarbeiter zu vergessen (Nuthall 2000, S. 96). Laut Nuthall verlässt Wissen, welches noch nicht fest ins Langzeitgedächtnis integriert ist schon nach 2-3 Tagen das Langzeitarbeitsgedächtnis (siehe auch Abschnitt 3.3.5). Sein Wissensstand verschlechtert

---

<sup>71</sup> Ohne hierfür eigene Sensitivitätsanalysen anzustellen wurde als Obergrenze für das Lernen am PC ein Skillwert von 3,0 festgelegt.

sich also in diesem Wissensgebiet nach und nach, solange bis das Wissen erneut angewendet wird.

Das Einführen von Vergessen löst das Problem des einseitigen Steigens des Wissens. Die Wissensbilanz insgesamt wird dadurch wesentlich ausgeglichener gestaltet. Je nach Arbeitspaketen und Lernstrategie weisen die Wissensbilanzen der Mitarbeiter moderat gestiegene, ausgeglichene nur innerhalb der Wissensgebiete verschobene Werte, bis hin zu leicht sinkenden Wissensbilanzen auf.

Als wichtige Strategie kristallisiert sich die Lernstrategie heraus. Je nachdem, ob in Pausen zwischen zwei Projekten immer das Wissensgebiet mit dem geringsten Wissensstand oder das am längsten nicht genutzte Wissensgebiet aufge bessert wird und ob jeden Tag neu ausgewählt, oder ein einmal ausgewähltes Wissensgebiet solange beibehalten wird, bis kein Lernfortschritt mehr erzielt werden kann, sind unterschiedliche Effekte zu beobachten. In allen Fällen zeigt sich jedoch, dass die knappe Arbeitszeit nicht ausreicht, um alle Wissensgebiete gleichermaßen auszubauen, wenn die Auslastung hoch ist.

### **6.2.5 Erweiterung um Lernen von Kollegen**

Ein bis dahin ungelöstes Problem ist das zielgerichtete Lernen von Expertenwissen, das für die Fertigstellung eines Arbeitspakets benötigt wird. Es wird unterstellt, dass dafür in erster Linie Kollegen befragt werden, da das Internet in solchen Spezialgebieten i. d. R. keine direkten Antworten bereithält. Das Wissen eines erfahrenen Kollegen ist daher nach dieser Ansicht erforderlich. Dies wurde als Verbundplan (*joint plan*) realisiert (siehe Abschnitt 4.5.1).

Obwohl befragte Kollegen bei ungenügendem eigenem Wissen an andere Kollegen weiter verweisen können und bei ausreichendem Wissen immer helfen, wird trotzdem nur selten die Wissenslücke gefüllt. Dies liegt an zwei Dingen: mangelndes Metawissen über das Wissen der Kollegen und zu geringer Wissenszuwachs in der kurzen Zeit des Helfens. Projektmitarbeiter haben nur wenig Kontakt miteinander, da es im Simulationsmodell bisher keine Notwendigkeit zur Zusammenarbeit gibt. Daher gibt es kaum Metawissen über das Wissen anderer Mitarbeiter und im Falle der Hilfsbedürftigkeit wird nur selten ein geeigneter Kollege gefunden. Meist sind nur die Projektmanager bekannt sowie die Kollegen in unmittelbarer Nähe. Kommt es doch zu einem erfolgreichen Erfahrungsaustausch, so wird auch gelernt. Allerdings wurde der Wissensaustausch so gestaltet, dass er auf eine kurze Zeitspanne begrenzt ist. Man hilft einem Kollegen jederzeit, aber nur für maximal ein bis zwei Stunden, so die Unterstellung. In dieser

Zeitspanne wird nur wenig gelernt und damit nur dann ein echter Fortschritt erzielt, wenn die Wissenslücke ebenfalls gering war. Andernfalls wiederholt sich das Ganze für lange Zeit, bevor das Arbeitspaket tatsächlich fertig gestellt werden kann.

### **6.2.6 Strategie zur Auswahl der Projektmitarbeiter**

Das Problem nicht fertig gestellter Arbeitspakete wirkt sich weniger auf der Ebene der Projektmitarbeiter als auf der Ebene der Projektmanager aus. Im Laufe der Zeit sind immer mehr Projekte zu  $\frac{2}{3}$  fertig gestellt (bei drei Arbeitspaketen pro Projekt), ohne ganz fertig zu werden. Dadurch wird der Projektmanager blockiert und nimmt keine neuen Projekte mehr an. Da es weniger Projektmanager als Projektmitarbeiter gibt, wirkt sich dies gravierender aus, da dadurch die Auslastung der übrigen Mitarbeiter sinkt, weil nicht nur der eine Projektmitarbeiter mit unzureichendem Wissen keine Fortschritte erzielt, sondern auch zwei weitere Projektmitarbeiter ohne neues Arbeitspaket zur Unproduktivität verurteilt sind.

Eine wichtige Strategie in diesem Zusammenhang ist die Vergabe der Arbeitspakete durch den Projektmanager. Die einfachste Strategie ist die Vergabe der Reihe nach, so wie sie im Projekt definiert ist. Eine weitere Strategie ist die Vergabe des schwierigsten Arbeitspakets zuerst. Dadurch könnte die Chance steigen, dass Mitarbeiter mit ausreichend hohem Wissen gefunden werden, die bei einer „einfachstes Arbeitspaket zuerst“ Auswahl vielleicht schon mit einem anderen Arbeitspaket beschäftigt sind, die ihr Wissen nur zum Teil fordert. Ein weiterer Teil der Vergabestrategie besteht in der Reihenfolge, in der Mitarbeiter gefragt werden. Es kann der erste Mitarbeiter gefragt werden, der über ausreichendes Wissen verfügt, oder derjenige, dessen Wissen am nächsten an den Anforderungen liegt. Auch eine „Optimierung“ der Unterschiede zwischen Wissen der Mitarbeiter und den Anforderungen aller Arbeitspakete eines Projekts wäre denkbar. Dabei muss allerdings immer berücksichtigt werden, dass die Umwelt des Projektmanagers nicht statisch ist und er zumindest bzgl. der Auslastung der Projektmitarbeiter, ohne SkMS auch in Bezug auf deren Skills über unvollkommene Information verfügt. Deshalb können die ihm bekannten Mitarbeiter schon mit Arbeitspaketen anderer Projekte beschäftigt sein, oder sich ihr Wissen in der Zwischenzeit positiv oder negativ verändert haben. Insofern erscheint eine solche Optimierung als nicht sinnvoll.

Tabelle 6-5 zeigt die Ergebnisse von vier Simulationsläufen mit einem Monat Laufzeit im Modell mit Lernen von Kollegen. Man kann erkennen, dass sehr wenig oder gar nicht von Kollegen gelernt wird. Die Auslastung schwankt zwischen 57% und 62% und determiniert

den Unternehmenserfolg maßgeblich. Die Wissensbilanz hängt v.a. davon ab, wie viel vergessen und von PC gelernt wurde.

*Tabelle 6-5: Simulationsergebnisse nach 31 Tagen im Modell mit Lernen von Kollegen*

:	Lauf 1	Lauf 2	Lauf 3	Lauf 4
<b>Gesamtes Firmenkapital</b>	-91.946 €	24.403 €	36.649 €	-4.938 €
<b>Lohnkosten</b>	476.000 €	476.000 €	476.000 €	476.000 €
<b>Umsatz</b>	334.054 €	450.403 €	391.351 €	421.062 €
<b>Offene Projekte</b>	16	17	16	15
<b>Abgeschlossener Projekte</b>	8	10	9	10
<b>Angebote Projekte</b>	31	31	31	31
<b>Σ Gelernt vom PC</b>	25,698	16,683	17,315	27,902
<b>Σ Gelernt von Kollegen</b>	0,108	0,335	0,000	0,000
<b>Σ Gelernt bei Projektarbeit</b>	10,106	9,145	5,933	7,579
<b>Σ Vergessen</b>	14,804	16,89	17,398	31,527
<b>Ø Gelernt vom PC</b>	0,734	0,477	0,495	0,797
<b>Ø Gelernt von Kollegen</b>	0,003	0,010	0,000	0,000
<b>Ø Gelernt bei Projektarbeit</b>	0,289	0,261	0,170	0,217
<b>Ø Vergessen</b>	0,423	0,483	0,497	0,901
<b>Ø Max. Skill eines Agenten</b>	3,458	3,465	3,677	3,524
<b>Ø Skill eines Agenten</b>	2,206	2,192	2,311	2,307
<b>Ø Min. Skill eines Agenten</b>	1,623	1,387	1,479	1,723
<b>Wissensbilanz</b>	308,834	306,819	323,598	322,944
<b>Ø Auslastung der Agenten</b>	0,588	0,622	0,620	0,571

### 6.2.7 Erweiterung um Kenntnis der Teamkollegen

Um das Problem des fehlenden Metawissens zu mindern wurde das Modell so erweitert, dass die Projektmitarbeiter eines Projekts sich gegenseitig kennen und aufgrund der Anforderungen der Arbeitspakete auf das Wissen der Kollegen schließen. Eine direkte Kenntnis des realen Wissensstandes der Kollegen erscheint nicht sinnvoll. Darüber hinaus wird im Falle mangelnden Metawissens der Projektleiter befragt, welcher damit als Informationbroker fungiert.

## 6 Simulationsstudien

Durch die geschilderten Erweiterungen wird das soziale Netzwerk<sup>72</sup> wesentlich realitätsnäher gestaltet (siehe Abbildung 6-4). Vorher war die räumliche Verteilung das Hauptkriterium für Kontakte unter den Projektmitarbeitern. Nur die Projektmanager stellten den Kontakt zwischen den sonst meist isolierten lokalen Gruppen her und fungierten als Broker und „Brückenbauer“ zwischen den Gruppen (boundary spanner). Nun kommen die Projektzugehörigkeit und in geringerem Umfang auch nachgefragtes Wissen als zusätzliche Determinanten hinzu. Was weiterhin fehlt, in der Simulation aber zu diesem Zeitpunkt nicht als Problem auffällt, sind soziale Kontakte zwischen Mitarbeitern mit ähnlichem Wissen, also eine Art thematisch orientierte Community. Da es sich bei Communities aber um ein WM-Instrument handelt soll dies im Grundmodell auch nicht realisiert werden.

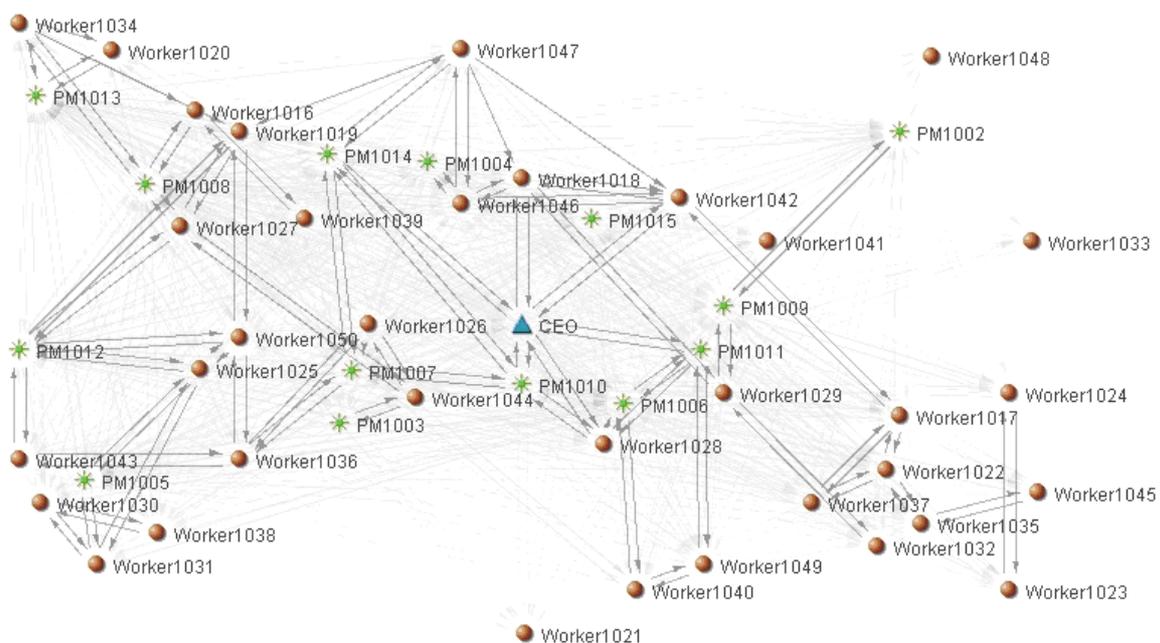


Abbildung 6-4: soziales Netzwerk nach 61 Tagen mit Kenntnis der Teamkollegen (eigene Darstellung)

Ein wesentlich dringenderes Problem besteht zu diesem Zeitpunkt der Entwicklung darin, dass die Simulation nicht über einen längeren Zeitraum stabil ist. Während in den ersten paar Monaten die hereinkommenden Projekte gut abgearbeitet werden, nimmt mit längerer Laufzeit die Anzahl an Arbeitspaketen, die aufgrund eines zu großen Skillgaps nicht abgearbeitet werden können, immer mehr zu. Dadurch entsteht eine Art „Teufelskreis“, der

<sup>72</sup> Siehe Abbildung 6-4, Netzwerkanalyse: 49 Knoten, 1140 Kanten, gewichtet und unsymmetrisch, Durchmesser 2,0, Dichte 0,465, gewichtete Dichte 104,7, Kohäsion: 0,444

dazu führt, dass die Simulation nach rund 6 Monaten aus dem Gleichgewicht gerät und immer weniger Projekte abgearbeitet werden können und dadurch aus den moderaten Gewinnen zu Beginn schnell große Verluste werden. Die Wirkungskette stellt sich folgendermaßen dar. Ein nicht fertig gestelltes Arbeitspaket bedeutet, dass auch das Projekt insgesamt nicht fertig wird. Dadurch wird der Projektmanager blockiert, da er immer nur ein Projekt gleichzeitig betreut. Deswegen werden nicht mehr alle Mitarbeiter regelmäßig mit neuen Arbeitspaketen versorgt. Ohne Projektarbeit wird aber auch mehr vergessen, da die Fähigkeiten nicht genutzt werden. Dadurch sinkt der allgemeine Wissensstand. Je länger aber keine Projektarbeit mehr verrichtet wurde, desto größer die Bereitschaft neue Arbeitspakete anzunehmen, auch wenn das eigene Wissen nicht ausreichend dafür ist. Dadurch erhöht sich die Zahl der Mitarbeiter, die ihr Arbeitspaket aufgrund zu geringen Wissens nicht fertig stellen können und damit ist der Kreis geschlossen.

### **6.2.8 Erweiterung um Rückgabe eines Arbeitspakets an den Projektmanager**

Um das Problem mit den nicht fertig gestellten Arbeitspaketen zu beseitigen, wurde die Möglichkeit eingeführt, dass ein Mitarbeiter nach einiger Zeit seine Situation erkennt und statt sehr lange zu versuchen genügend dazuzulernen, sein Arbeitspaket an den Projektleiter zurück gibt, welcher es dann neu an andere Mitarbeiter vergibt. Dadurch wird eine lange Blockade des Projektmanagers verhindert und die Auslastung der Mitarbeiter verbessert.

### **6.2.9 Zusammenfassung der Designentscheidungen**

Zur besseren Übersicht sind die Erweiterungen in Abbildung 6-5 noch einmal grafisch aufbereitet dargestellt.

(1) Anfangs ist nur Projektarbeit implementiert. (2) Lernen bei der Arbeit ist die erste Erweiterung, die durch (3) Lernen von PC ergänzt wurde. (4) Vergessen dient als Ausgleich zu den Lernmöglichkeiten. (5) Lernen von Kollegen (Wissensweitergabe im engeren Sinne) wird durch Pfeil fünf symbolisiert. (6) Metawissen über das Wissen der Kollegen ist eine wichtige Voraussetzung für Wissensweitergabe. (7) Rückgabe von Arbeitspaketen hilft, wenn überhaupt keine Fortschritte bei der Arbeit erzielt werden.

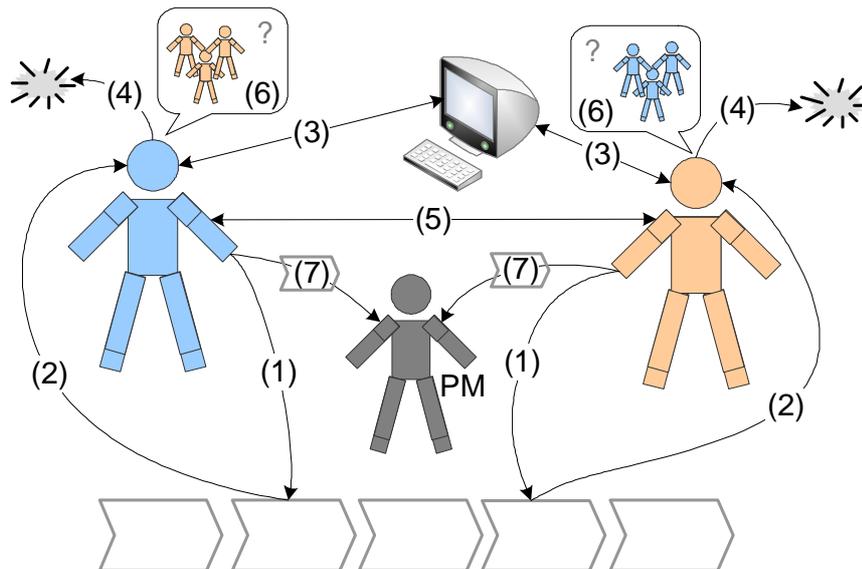


Abbildung 6-5: Überblick über die Modellerweiterungen (eigene Darstellung)

Im Laufe der Entwicklung sind eine Reihe weiterer Designentscheidungen notwendig geworden, die nur zum Teil aus der Literatur abgeleitet werden konnten. Diese sind im folgenden Überblick noch einmal zusammengefasst.

- Lernen bei der Projektarbeit
  - Bei Skillwert von weniger als 0,5 unter oder 0,3 überhalb des benötigten Werts kann durch Projektarbeit gelernt werden.
- Lernen am PC
  - Bis zu einem Maximalwert von 3,0 kann durch Lernen am PC Wissenszuwachs erzielt werden.
- Vergessen
  - Startet nach 25 Tagen, ohne dass der Wissensbereich benutzt wurde.
  - Es wird ein fünfundzwanzigtausendstel einer Skillstufe pro Runde vergessen (nur während der Arbeitszeit).
- Lernen durch Fragen eines Kollegen
  - Wurde nach acht Runden kein Fortschritt am Arbeitspaket erzielt, wird ein Kollege gefragt oder vom PC gelernt.
  - Bei ausreichendem Wissen wird immer geholfen.
  - Bei nicht ausreichendem Wissen kann auf einen Kollegen verwiesen werden.
- Strategie zur Mitarbeiterauswahl

- Es wird die Closest-first-Strategie eingesetzt, also der Mitarbeiter, bei dem der Unterschied zwischen benötigtem und vorhandenem Wissen am geringsten ist, wird als erstes gefragt.
- Die Arbeitspakete werden der Reihenfolge der Definition nach vergeben.
- Kenntnis der Teamkollegen
  - Die Mitarbeiter bekommen bei Zuweisung des Arbeitspakets auch die Information über die übrigen Teamkollegen
- Rückgabe der Arbeitspakete
  - Wurde nach vier Zyklen mit Projektarbeit und Lernen immer noch kein Fortschritt erzielt, so wird das Arbeitspaket zurückgegeben.

Erst diese sieben Funktionen zusammen haben zu einem stabilen und plausiblen Verhalten des Gesamtsystems geführt. Das so entstandene Modell dient als Grundlage für die folgenden Simulationsstudien.

### **6.3 Sensitivitätsanalysen**

Sensitivitätsanalysen dienen dazu festzustellen, welche Ausprägung eines Parameters am besten für die Simulation geeignet ist, und was mit der Simulation passiert, wenn die Ausprägung einer Variablen auf der einen oder anderen Seite aus dem Korridor herauswandert, der den semantisch sinnvollen Wertebereich markiert. Der Komplexität der hier entwickelten Simulation entsprechend gibt es eine große Anzahl an Parametern, die entsprechend justiert werden müssen. Die Justierung hat zum Ziel, dass die Simulation über einen langen Zeitraum inhaltlich stabil läuft, ohne zu kollabieren oder bzgl. einzelner Messgrößen in Extremwerte zu laufen. Ein Kollaps erfolgt z.B. wenn Projektmanager über lange Zeiträume blockiert sind, dadurch die Auslastung der Mitarbeiter sinkt und der Unternehmenserfolg entgegen der vorherigen Entwicklung stark negativ wird. Dabei gilt es jedoch immer zu beachten, dass die Parameter innerhalb vernünftiger Grenzen liegen, d.h. weder dem gesunden Menschenverstand noch wissenschaftlichen Erkenntnissen widersprechen dürfen. Wenn z.B. die Lernrate justiert wird, so muss darauf geachtet werden, dass der Mitarbeiter nicht innerhalb von ein paar Tagen vom Novizen zum Experten werden kann, auch wenn er ständig lernt. Andererseits muss beachtet werden, dass die in der Simulation behandelten Wissensgebiete enger gefasst sind als in der klassischen Expertiseforschung. Insofern kann davon ausgegangen werden, dass es keiner 5-10 Jahre, wie in der Expertiseforschung diskutiert, um z.B. im Bereich Datenbanken, Programmiersprachen oder Betriebssysteme ein Experte zu werden, wenn man jeweils nur

von den verallgemeinerbaren Grundlagen und der Anwendung in einem konkreten Produkt spricht (also z.B. Oracle 9i, C#, Windows 2003). Dafür gibt es Kurse, bei denen man innerhalb von wenigen Wochen auf Stufe drei (Kompetenter nach Dreyfus&Dreyfus) kommen kann. Kann man noch innerhalb einiger realer Projekte praktische Erfahrung sammeln, so sollte es möglich sein innerhalb von einigen Monaten auf das Niveau von Stufe vier kommen (Erfahrener / Proficient nach Dreyfus&Dreyfus). Die Expertenstufe könnte entsprechend schon nach ein bis zwei Jahren ausschließlicher Beschäftigung mit diesem Thema erreicht werden.

Im Folgenden werden die einzelnen Parameter und Wertebereiche vorgestellt, die Ergebnisse von einigen Simulationsläufen mit unterschiedlichen Ausprägungen innerhalb der Wertebereiche gezeigt und kurz diskutiert. Zum Schluss wird auf die Interdependenzen zwischen den Parametern eingegangen und die für das erweiterte Grundmodell verwendeten Ausprägungen festgehalten.

- Atomares Zeitintervall (Zeit pro Simulationsrunde)
- Anzahl der Wissensgebiete
- Anzahl benötigter Wissensgebiete pro Arbeitspaket
- Lernrate pro Zeiteinheit für Lernen vom PC, Learning by doing, Lernen von Kollegen
- Zeitraum bevor vergessen wird und Vergessensrate pro Zeiteinheit

Die simulierte Zeit, die einer Simulationsrunde entspricht, ist einer der Basisparameter für die Simulation und die Festlegung eines Zeitintervalls begrenzt auch unmittelbar die Granularität der möglichen Aktionen. Theoretisch kann das Zeitintervall im Bereich einige Sekunden bis hin zu einem Monat gewählt werden. Da jedoch schon mehrfach angesprochen wurde, dass es sich um eine Mikrosimulation mit recht hohem Detaillierungsgrad handeln soll ist klar, dass eine Runde im Höchstfall im Bereich von ca. einer Stunde liegen darf.

Die Anzahl der Wissensgebiete ist theoretisch im Wertebereich  $[1; \infty]$  wählbar. Tatsächlich ist es jedoch sinnvoll eine Obergrenze festzulegen. Diese wurde aus der Literatur abgeleitet (siehe Abschnitt 6.1.1) und als 16 für die Softwareentwicklung relevante Wissensgebiete definiert. In der Simulation hat sich jedoch herausgestellt, dass auch diese 16 Gebiete noch zu hoch gegriffen sind, da das Vergessen pro Wissensgebiet auftritt und bei begrenzter Zeit als limitierendem Faktor nur wenige Wissensgebiete aktuell gehalten werden können. Für

die Sensitivitätsanalyse wurden zwei, vier und acht Wissensgebiete getestet. Vier Wissensgebiete haben sich dabei als gut geeignet erwiesen, wobei bemerkt werden muss, dass eine starke Korrelation zu der Anzahl benötigter Wissensgebiete pro Arbeitspaket besteht. Je mehr Wissensgebiete pro Arbeitspaket benötigt werden, desto mehr Wissensgebiete sind möglich und sinnvoll, ohne dass zu viel vergessen wird.

Für die Übertragung dieser Erkenntnisse auf die Realität muss beachtet werden, dass dort oft nicht nur ein Arbeitspaket auf einmal bearbeitet wird, sondern zwei bis drei Arbeitspakete in einem Zeitraum von einigen Wochen parallel bearbeitet werden, wobei der Mitarbeiter von den Zeitvorgaben für die Fertigstellung abgesehen eine relativ freie Zeiteinteilung für die Abarbeitung hat. Nichts desto trotz ist auch in der Realität die Zeit begrenzt und aufgrund der kurzen Produktlebenszyklen in der Softwarebranche ist es schwierig in allen relevanten Wissensgebieten seinen einmal erworbenen Wissensstand zu halten oder gar aktuell zu halten.

Die Anzahl der benötigten Wissensgebiete pro Arbeitspaket kann theoretisch zwischen eins und der Anzahl an Wissensgebieten insgesamt variieren. Betrachtet man noch einmal die 16 definierten Wissensgebiete (siehe Abschnitt 3.4.1), so ergeben sich durchaus Anknüpfungspunkte zwischen ihnen. So ist z.B. das Gebiet objektorientierte Konzepte verwandt mit dem Gebiet Programmiersprachen, wenn es sich bei letzteren um objektorientierte Sprachen handelt. Ähnliches gilt auch für die Gebiete Datenstrukturen und Datenbanken, oder Softwareanalyse- und Designmethoden einerseits und Anforderungsanalyse, bzw. Softwareentwurf und -methoden andererseits. Insofern erscheinen Parameterwerte von eins bis drei sinnvoll. Diese wurden in der Sensitivitätsanalyse auch getestet und alle Werte erscheinen gleichermaßen möglich. Um das Modell einfach zu halten wurde der Wert eins ausgewählt.

Für die Lernrate kann prinzipiell jede positive Gleitkommazahl festgelegt werden. Anhaltspunkte für praktikable Werte sind aus der Expertiseforschung zu entnehmen. Dort geht man wie in Abschnitt 3.3.3 geschildert von bis zu zehn Jahren bis zum Erreichen des Expertenstatus aus. Wie ebenfalls schon diskutiert ist dabei jedoch zu berücksichtigen, wie groß die Domäne des Expertentums definiert wird. Geht man von 10 Jahren bis zum Erreichen des Maximalwert von 5,0 aus, mit durchschnittlich 2-4 Stunden Beschäftigung mit dem Thema pro Tag bei 220 Arbeitstagen pro Jahr und 4 Runden pro Stunde so ergibt sich ein Wert von  $5 / 10 \cdot 3 \cdot 4 \cdot 220 = 5 / 26400 = 0,00018$ . Als geringster sinnvoller Wert ergibt sich daher gerundet 0,0002. Als größter Wert für die Sensitivitätsanalyse wird 0,001

verwendet. Es wird dabei davon ausgegangen, dass Lernen von Kollegen bei gegebenem Wissen der Quelle am effizientesten ist, Lernen durch Projektarbeit am wenigsten effizient, da neue Erkenntnisse nur durch Versuch und Irrtum zustande kommen.

Für den Zeitraum bis zum Beginn des Vergessens und die Vergessensrate konnte der Autor kaum empirische Belege finden. (Nuthall 2000) findet in empirischen Untersuchungen Hinweise dafür, dass neu Gelerntes vergessen wird, wenn es nicht durch zusätzliche Übung innerhalb von zwei bis drei Tagen den Übergang von Langzeitarbeits- in das Langzeitgedächtnis schafft (siehe auch Abschnitt 3.3.5). (Pavlik, Anderson 2005) untersuchen die Vergessensrate von englischsprachigen Studenten, die Japanisch lernen. Sie lernen Englisch-Japanische Wortpaare mit unterschiedlichen Wiederholungen und müssen am selben Tag, am Tag darauf und sieben Tage später zeigen, wie viel sie behalten haben. In diesem Experiment zeigt sich, dass schon nach einem Tag ein Teil wieder vergessen wurde. (Markovitch, Scott 1988) argumentieren dagegen, dass Vergessen von unwichtigen Fakten zu einer Verbesserung der Erinnerungsleistung führt. Dies kann aber in der Simulation nicht berücksichtigt werden, da nur relevantes Wissen simuliert wird. (Tynjälä 1999) untersucht die Abhängigkeit von Lernen und Vergessen von der Art des Lernens und kommt zu dem Schluss, dass langsamer bzw. später vergessen wird, wenn das Wissen in einen Kontext eingebettet ist. Konkrete Daten oder Zeiträume werden aber nicht genannt. Theoretisch sind hier wiederum alle positiven Gleitkommazahlen valide Werte für die Vergessensrate. Für den Zeitraum, bevor vergessen wird, sind dagegen positive Ganzzahlen möglich, die die Anzahl an Runden festlegen, bevor vergessen wird. Die Sensitivitätsanalyse zeigt, dass Werte größer oder gleich der Lernrate dazu führen, dass zu schnell vergessen wird. Als minimale sinnvolle Einstellung haben sich 25 Tage und eine Vergessensrate von 0,0004 erwiesen, da ansonsten immer mehr vergessen als gelernt wird (siehe Spalte Skillentwicklung in Tabelle 6-6). Der im Vergleich zu den oben genannten Werten lange Zeitraum bevor vergessen wird ist nötig, da nicht zwischen verschiedenen Gedächtnisbereichen unterschieden wird und somit auch solches Wissen, das schon im Langzeitgedächtnis verankert wäre in der Simulation wieder vergessen werden kann.

Die Ergebnisse der Sensitivitätsanalysen sind in Tabelle 6-6 zusammengefasst.

Tabelle 6-6: Ergebnisse der Sensitivitätsanalysen<sup>73</sup>

Parameter	Wert	Abg. Projekte	Projekte pro Worker	Umsatz in Mio	Umsatz pro Mitarbeiter	Deckungsbeitrag in Mio	Auslastung in %	Skillentwicklung	Skill max	Skill avg	Skill min
Anzahl Agenten	49	437	12,14	20,083	0,56	3,191	75,4	0,33	3,66	2,27	1,41
	77	600	10,53	27,612	0,48	1,475	70,9	2,40	3,59	2,83	2,22
	77 mit 40 Runden	682	11,96	31,851	0,56	5,715	73,1	2,49	3,64	2,77	2,14
	101	630	11,05	28,967	0,51	-5,463	62,3	3,79	3,76	3,35	3,14
	101 mit 40 Runden	801	14,05	37,34	0,66	2,909	69,7	3,21	3,7	3,31	3,07
	101 mit 20 Runden	1013	17,77	46,757	0,82	12,327	76,9	0,16	3,75	2,45	1,28
Laufzeit	1 J	145	4,03	6,653	0,18	1,056	71,4	-2,58	3,75	1,77	0,96
	2 J	283	7,86	13,091	0,36	1,846	73,8	-1,00	3,69	2,17	1,38
	3 J	437	12,14	20,083	0,56	3,191	75,4	0,33	3,66	2,27	1,41
	4 J	596	16,56	27,622	0,77	5,659	75,9	1,78	3,72	2,82	2,13
	5 J	753	20,92	34,94	0,97	7,474	76,5	2,58	3,65	3,02	2,62
Absatzmarkt	alle 40 Runden	458	12,72	21,233	0,59	4,773	76,8	0,00	3,69	2,34	1,45
	50 Runden	449	12,47	20,607	0,57	4,148	75,5	0,31	3,67	2,5	1,89
	60 Runden	437	12,14	20,083	0,56	3,191	75,4	0,33	3,66	2,27	1,41
	70 Runden	415	11,53	19,017	0,53	2,557	72,8	2,31	3,67	2,48	1,57
	80 Runden	402	11,17	18,493	0,51	2,033	72,8	2,47	3,67	2,54	1,90
Schwierigkeit	2,0	452	12,56	20,952	0,58	4,492	74,0	1,06	3,24	2,65	2,19
	2,5	437	12,14	20,083	0,56	3,191	75,4	0,33	3,66	2,27	1,41
	3,0	428	11,89	19,771	0,55	3,321	76,3	0,03	4,23	2,44	1,49
Arbeitspaketgröße	10 Tage	632	17,56	14,151	0,39	-2,301	60,7	5,03	3,93	3,72	3,70
	20 Tage	437	12,14	20,083	0,56	3,191	75,4	0,33	3,66	2,27	1,41
	30 Tage	307	8,53	21,411	0,59	4,952	79,9	-2,72	3,68	1,68	1,04
Vergessen nach	15 Tage	441	12,25	20,181	0,56	3,722	76,2	-0,56	3,66	2,01	1,29
	25 Tage	437	12,14	20,083	0,56	3,191	75,4	0,33	3,66	2,27	1,41
	35 Tage	438	12,17	20,195	0,56	3,76	74,8	1,19	3,73	2,30	1,18
Vergessensrate	0,0004	460	12,78	21,155	0,59	4,695	73,4	3,42	3,71	3,27	3,10
	0,0006	436	12,11	19,837	0,55	3,378	74,1	2,50	3,57	3,01	2,87
	0,0008	431	11,97	19,72	0,55	3,26	74,4	1,92	3,51	2,90	2,44
	0,0010	429	11,92	19,492	0,54	3,033	74,9	1,39	3,46	2,76	2,15
	Adaptiv 15000	442	12,28	20,416	0,57	3,956	75,2	1,03	3,66	2,64	1,66
	Adaptiv 25000	447	12,42	20,699	0,57	4,228	74,2	2,00	3,76	2,48	1,57
	Adaptiv 35000	455	12,64	20,751	0,58	4,292	74,2	2,89	3,77	3,11	2,70
Anzahl Skills	3	438	12,17	19,821	0,55	3,361	74,0	1,81	3,65	3,09	3,05
	4	437	12,14	20,083	0,56	3,191	75,4	0,33	3,66	2,27	1,41
	5	433	12,03	19,933	0,55	3,474	75,7	-2,14	3,77	1,76	0,92
	6	434	12,06	19,941	0,55	3,481	77,5	-3,81	3,75	1,76	0,86
Lernrate pro Runde	0,005	454	12,61	20,806	0,58	4,346	76,1	-1,14	3,82	1,80	1,21
	0,010	443	12,31	20,245	0,56	3,785	75,7	0,17	3,70	2,42	1,63
	0,015	430	11,94	19,67	0,55	3,211	74,6	1,31	3,51	2,80	2,23
	0,020	422	11,72	19,194	0,53	2,734	74,8	1,08	3,48	2,47	2,00
Obere Lerngrenze	0,1	411	11,42	18,884	0,52	2,425	75,6	-0,03	3,52	2,36	1,69
	0,3	435	12,08	19,877	0,55	3,417	74,9	0,92	3,61	2,62	1,86
	0,5	461	12,81	21,133	0,59	4,674	75,0	1,08	3,74	2,40	1,63
Untere Lerngrenze	0,3	426	11,83	19,523	0,54	3,063	76,8	-1,00	3,41	2,11	1,37
	0,5	436	12,11	19,765	0,55	3,305	75,8	0,03	3,60	2,37	1,67
	0,7	458	12,72	21,02	0,58	4,561	74,2	1,72	3,90	2,85	2,07
	0,9	466	12,94	21,453	0,60	4,993	73,1	2,83	4,11	3,15	2,15

<sup>73</sup> Basis für die Sensitivitätsanalysen sind 49 Agenten, 3 Jahre Laufzeit, neue Projekte im Schnitt alle 60 Runden, Schwierigkeit 2.5, Arbeitspaketgröße 20 Tage, Vergessen nach 25 Tagen, Anzahl Skills 4, Lernrate 0.10, obere Lerngrenze 0.3, untere Lerngrenze 0.5

## 6.4 Verifikation und Validierung

Verifikation ist die Überprüfung eines Computerprogramms oder Simulationsmodells auf fehlerfreie Implementierung (Moss et al. 1997). In der objektorientierten Softwareentwicklung werden dazu neben Codereviews v.a. Unit-Tests eingesetzt. Ein Unit-Test besteht im systematischen Test einer in sich abgeschlossenen Einheit mit einer Reihe verschiedenen Eingabedaten. Es wird jede einzelne Methode getestet und die Outputs, egal ob explizit als Rückgabewert, oder implizit als interne Veränderung des Objekts selbst, mit den erwarteten Outputs verglichen. Dazu gibt es für die verschiedenen Programmiersprachen entsprechende Test-Frameworks. Die Java-Klassenbibliothek selbst bietet keine Unterstützung für Tests. Für die Java-Entwicklung ist JUnit das älteste, bekannteste und meist benutzte Framework (Lee 2005). In den letzten Jahren wird allerdings TestNG wegen seiner Vorteile gegenüber JUnit als Alternative empfohlen (Diotalevi 2005). Zu diesen Vorteilen zählt die Nutzung der Java 1.5 Annotationen zum Markieren der Tests, während es bei JUnit nötig ist, dass die Testklassen von der JUnit Klasse TestCase erbt. JUnit wird jedoch kontinuierlich weiterentwickelt und bietet seit der Version 4.0 (seit März 2006, aktuell im November 2006 ist Version 4.2) ebenfalls annotationsbasierte Tests an. Dies ist für Unit-Tests in der agentenorientierten Programmierung unabdingbare Voraussetzung, da ein Agent i. d. R. keine öffentlichen Methoden enthält, die Rückgabewerte als Ergebnis ihrer Ausführung zurückgeben. Es müssen also zwangsläufig die inneren Zustände des Agenten inspiziert werden, um über eine fehlerfreie Ausführung zu entscheiden. Muss die Testklasse von einer Klasse des Testframeworks erben, dann ist dies nicht möglich, ohne die entsprechenden Zustände mit öffentlichen Accessoren zu deklarieren. Dies steht aber nicht im Einklang mit den Grundprinzipien der Agentenorientierung, da dann auch Agenten die inneren Zustände anderer Agenten inspizieren könnten. Es erscheint daher wesentlich sinnvoller, die Testklassen von den zu testenden Klassen erben zu lassen, um Zugriff auf innere Zustände mit der Sichtbarkeit `protected` zu erhalten.

Dies ist jedoch nicht das einzige Problem bei Unit-Tests von Agenten. Da nicht nur die Daten gekapselt werden, wie bei Objekten, sondern auch das Verhalten gekapselt ist (Odell 2002), muss ein Agent erst mit einer Vielzahl an Daten initialisiert werden, die in der Simulation im Laufe der Lebenszeit des Agenten gesammelt werden, bevor eine bestimmte Aktion des Agenten getestet werden kann. Zudem ist ein Agent meist nicht ohne die benötigte Simulationsumgebung und wegen des sozialen Verhaltens auch nicht ohne

andere Agenten testbar. Dies alles macht die Verifikation von Agenten extrem schwierig. Von einigen wenig praktikablen Ansätzen wie formalen Beschreibungssprachen mit integrierter Verifikationsfunktion abgesehen, haben sich aber noch keine besseren Verfahren etablieren können, so dass für die vorliegende Arbeit trotz des hohen Aufwands Unit-Tests für die drei Agentenklassen ManagerAgent, PMAgent und WorkerAgent entwickelt wurden. Die Testklassen sind auf Basis von JUnit 4 entwickelt worden und als Kindklassen der zu testenden Klassen realisiert.

Die Validierung eines Modells soll sicherstellen, dass es das untersuchte Phänomen auch adäquat abbildet (Canessa, Riolo 2003, S. 160f). Dies sollte im Allgemeinen mindest die folgenden drei Schritte umfassen (Sargent 1988):

- Beurteilung der Plausibilität: dieser Schritt umfasst das Befragen von Experten auf dem Gebiet des untersuchten Phänomens, ob das Modell und sein Verhalten als vernünftig und plausibel eingeschätzt wird.
- Beurteilung der zugrunde liegenden Annahmen: dazu werden die verwendeten Theorien und Annahmen auf Widerspruchsfreiheit und Kompatibilität hin untersucht. Auch Hinweise in der Literatur auf Fehler in den verwendeten Theorien müssen beachtet werden.
- Beurteilung der Übereinstimmung der Ergebnisse mit der Realität: hier müssen die gesammelten Daten aus dem Simulationssystem mit empirisch ermittelten Daten gegenübergestellt werden.

Eine Validierung auf Basis eigener wissenschaftlich fundierter empirischer Untersuchungen in Unternehmen würde den Rahmen der Arbeit sprengen. Ohne Validierung kann andererseits keine Aussage über die Übertragbarkeit der Simulationsergebnisse auf Unternehmenssituationen gemacht werden. Es soll daher versucht werden Teilaspekte des Modells auf Basis vorhandener empirischer Untersuchungen aus der Literatur und eigenen Untersuchungen mit Studenten in Lernsituationen und IT-Mitarbeitern bei itCampus zu validieren.

Eine weitere Möglichkeit der Validierung ist die Replikation von Simulationsexperimenten anderer Forscher (Axelrod 1997, S. 31; Edmonds, Hales 2003).

Aus der Literatur bietet sich zunächst die Studie von Lethbridge an, der Softwareentwickler für eine Reihe von Skills nach ihrem Wissensstand nach der Ausbildung und nach einer Reihe von Jahren Praxiserfahrung befragt hat (Lethbridge 1999). Wenn diese

Wissensentwicklung in der Simulation nachgestellt werden kann, so wäre ein erster Schritt in Richtung Validierung erfolgt.

Weiterhin sollen empirische Studien zur Bildung von sozialen Netzen in Kleingruppen herangezogen werden und mit den Kommunikationsstrukturen in der Simulation verglichen werden, ähnlich wie in (Müller 2002a).

Bzgl. der Ausprägungen der Skillwerte stellt sich bei durchschnittlichen und maximalen Skillwerten eine hohe Übereinstimmung der Simulationsergebnisse mit den bei itCampus ermittelten Werten und auch den von Lethbridge erhobenen Werten ein (siehe Tabelle 6-7). Die niedrigeren Minimalwerte können sich dadurch ergeben, dass bei itCampus 15 und bei Lethbridge sogar 57 Skillkategorien abgefragt wurden während in der Simulation nur 4 Skillkategorien für die Simulation verwendet wurden. Für die nur halb so hohen Standardabweichungen ist dem Autor keine Ursache bekannt.

*Tabelle 6-7: Vergleich der Skillwerte von itCampus mit den Simulationsergebnissen*

	<b>Skill max</b>	<b>Skill avg</b>	<b>Skill min</b>	<b><math>\sigma</math></b>
Simulation ohne SkMS	3,66	2,38	1,29	0,19
Simulation mit SkMS f. PMs	3,72	2,33	1,17	0,30
Simulation mit SkMS f. alle	3,71	2,35	1,19	0,26
itCampus Softwareentwickler	3,76	2,42	0,83	0,79
Lethbridge Erhebung 1997	3,87	2,32	0,89	0,66

Davon abgesehen sind die Ergebnisse der Simulation nach drei Jahren mit 49 Agenten gut vergleichbar mit den itCampus Werten (43 festangestellte Mitarbeiter, durchschnittlich 3,55 Jahre Berufserfahrung und 2,85 Jahre Betriebszugehörigkeit, siehe Abschnitt 6.1.4). Die Projektteams sind mit drei Projektmitarbeitern und einem Projektmanager ebenfalls mit den Daten von itCampus vergleichbar (3,3 bzw. 4,6 Mitarbeiter pro Projekt). Allerdings werden in der Simulation wesentlich mehr Projekte abgearbeitet (37 pro Mitarbeiter in drei Jahren gegenüber rund 6 Projekten in knapp 3 Jahren bei itCampus). Das führt dazu, dass auch der Umsatz in der Simulation wesentlich höher ist als bei itCampus (6,6 Mio. € gegenüber 2,2 Mio. €). Die durchschnittliche Projektdauer in der Simulation liegt bei 75 Tagen (Max 243, Min 9 Tage). Gründe dafür sind die günstige Auftragslage und der Verzicht auf Wochenenden in der Simulation. Rechnet man diese Vereinfachung aus den Simulationsergebnissen heraus (220 statt 360 Arbeitstage pro Jahr), so verringern sich die Simulationsergebnisse auf 22,6 Projekte und 4,0 Mio. € Umsatz.

Tabelle 6-8 zeigt die Daten über die Kontakte bei itCampus und in der Simulation. In der Simulation hat ein Agent durchschnittlich 2,7 Kontakte, während bei itCampus im Schnitt mit 5,5 Mitarbeitern, also fast doppelt so vielen Kollegen wie in der Simulation regelmäßiger Kontakt besteht.

*Tabelle 6-8: Vergleich der Skillwerte von itCampus mit den Simulationsergebnissen*

	<b>Kontakte max</b>	<b>Kontakte avg</b>	<b>Kontakte min</b>
<b>Simulation ohne SkMS</b>	15	2,69	1
<b>itCampus Softwareentwickler</b>	10	5,52	2

Auch die Maximal- und Minimalanzahl an Kontakten weicht deutlich ab. Wie Tabelle 6-9 zeigt, sind die Maximalwerte eher als Ausreißer einzustufen, da alle Werte über elf nur 0,17% der Beobachtungen ausmachen. Beim Minimalwert ist jedoch eine hohe relative Häufigkeit von 17,34% zu verzeichnen. Legt man eine Normalverteilungshypothese zu Grunde, so sieht man, dass die Glockenkurve deutlich linksschief ist und das Maximum der Häufigkeit bei drei statt bei fünf oder sechs liegt, wie es sein müsste, wenn sie den Ergebnissen bei itCampus entsprechen würden.

*Tabelle 6-9: Häufigkeitsverteilung der Kontakte in der Simulation*

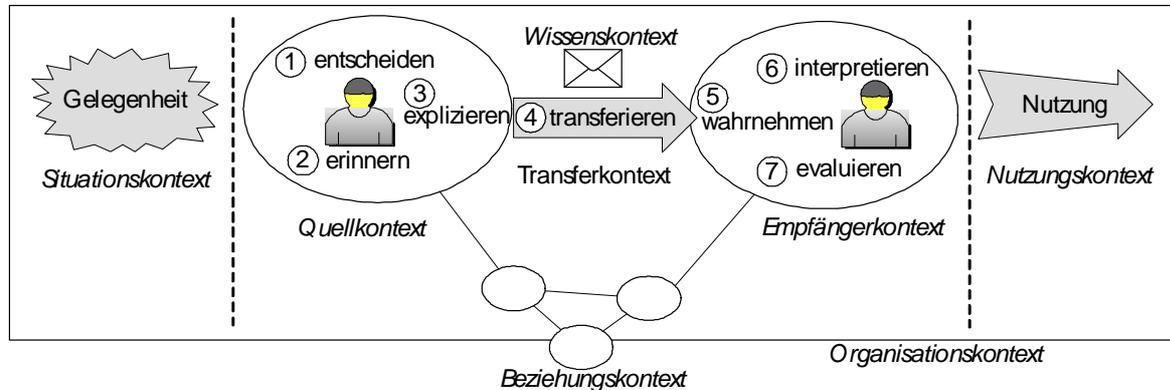
Anzahl Kontakte	1	2	3	4	5	6	7	8	9	10	11
Häufigkeit	998	1659	2187	555	122	97	51	22	19	18	16
relative H.	17,34%	28,83%	38,01%	9,65%	2,12%	1,69%	0,89%	0,38%	0,33%	0,31%	0,28%
kumulierte H.	17,34%	46,18%	84,18%	93,83%	95,95%	97,64%	98,52%	98,91%	99,24%	99,55%	99,83%

Diese Unterschiede bringen zum Ausdruck, dass die komplexe Sozialstruktur, die sich in einer Organisation ausbildet, in der Simulation bisher nur ansatzweise abgebildet wird. Insgesamt sind die Unterschiede aber insbesondere bei den Skillwerten sehr ermutigend und lassen auf valide Ergebnisse hoffen.

## 6.5 Studie 1: Skillmanagement

Die zentrale Forschungsfrage in der Simulationsstudie Skillmanagement lautet: „Wie wirkt sich die unterschiedliche Auswahl von Mitarbeitern für Teams auf die Wissensweitergabe und Verteilung von Wissen aus?“. Die Teamzusammenstellung wirkt nur implizit auf die Wissensweitergabe. Neben der oben angesprochenen Strategie der Auswahl ist vor allem die Kenntnis der Mitarbeiter, also das Metawissen relevant. Beides wirkt auf die Gelegenheiten für die Wissensweitergabe, also den Situationskontext und dort speziell auf Pull-Gelegenheiten (siehe Abschnitt 3.2.1). Die Teamzusammenstellung ist dabei

entscheidend dafür, wie groß die durchschnittlichen Unterschiede zwischen benötigtem und vorhandenem Wissen ist und damit auch auf die Notwendigkeit Wissen nachzufragen. Das Metawissen wirkt bei Projektmanagern auf die Zusammenstellung der Teams, bei Projektmitarbeitern auf die Auswahl des Kollegen, der um Hilfe gebeten wird.



Die Vergleichgrundlage bildet Experiment 1.1, welches ohne Skillmanagement arbeitet und eine firmenweite Zuordnung der Arbeitspakete auf beliebige freie Mitarbeiter zulässt. In Experiment 1.2 wird ein Skillmanagementsystem eingeführt, das dem Projektmanagern und in Experiment 1.3 zusätzlich auch den Projektmitarbeitern zum Identifizieren von Mitarbeitern mit bestimmten Skills zur Verfügung steht (siehe Abschnitt 2.4.2 zum WM-Instrument Kompetenzmanagement).

### 6.5.1 Experiment 1.1: Zuordnung der Arbeitspakete nach Kenntnis des PM

Projekte werden von jeweils neu gebildeten Teams abgewickelt, die pro Projekt anhand der Qualifikation und der freien Kapazität der Mitarbeiter vom Projektleiter ausgewählt werden. Der Projektleiter wählt anhand von informellen Bewerbungen der Mitarbeiter und persönlicher Kenntnis der Qualifikationen aus. Dies ist das Basisszenario und dient als Vergleichsgrundlage.

### 6.5.2 Experiment 1.2: Zuordnung mit Skillmanagement

Der Aufbau ist wie in Experiment 1, allerdings steht dem Projektleiter ein laufend aktualisiertes Skillmanagementsystem zur Verfügung, so dass die Auswahl der Mitarbeiter anhand der Skills erfolgt.

**Hypothese 1.1:** Mit Skillmanagement ist die durchschnittliche Bearbeitungszeit von Arbeitspaketen im Vergleich zum Basisszenario geringer.

Durch das größere Metawissen über die Wissensgebiete der Mitarbeiter ist der Projektmanager wesentlich „besser“ bei der Vergabe der Arbeitspakete wodurch es

seltener vorkommt, dass Mitarbeiter über zu wenig Wissen verfügen, um Fortschritte bei der Abarbeitung der Arbeitspakete zu machen. Das führt dazu, dass die durchschnittliche Bearbeitungsgeschwindigkeit steigt. Die Auswahlstrategie ist wie auch in den anderen Experimenten `ClosestFirst`, d.h. der Projektmanager erinnert alle Mitarbeiter mit einem Skill, der knapp unter (untere Lerngrenze) oder über dem benötigten Skill liegt und fragt zuerst denjenigen, dessen Abweichung zwischen tatsächlichem und benötigtem Skill am geringsten ist.

$$(18) \text{Strategy „ClosestFirst“: } \min \text{ Skillgap} = | \text{Skill}_{\text{present}} - \text{Skill}_{\text{required}} |$$

Dadurch werden zwar einige Arbeitspakete langsamer bearbeitet, weil es auch seltener vorkommt, dass Mitarbeiter wesentlich mehr Wissen haben als benötigt wird. Im Durchschnitt sollte die Bearbeitungsgeschwindigkeit aber sinken. Interpretiert man die vorgesehene Bearbeitungszeit bei  $\text{Wissen}_{\text{benötigt}} = \text{Wissen}_{\text{vorhanden}}$  als Aufwandsschätzung, entspricht dies der Annahme, dass die tatsächliche und geschätzte Bearbeitungszeit mit Skillmanagement weniger voneinander abweichen.

**Hypothese 1.2:** Mit Skillmanagement ist die Menge an Wissen, das durch Lernen bei der Arbeit (learning by doing) erworben wird, höher.

Da nicht die Mitarbeiter mit dem meisten Wissen sondern diejenigen mit dem geringsten Unterschied zum geforderten Wissen die Arbeitspakete bearbeiten müssen, werden Lerneffekte möglich, die bei Routinetätigkeiten (vorhandene Fähigkeiten liegen weit über den benötigten) nicht auftreten.

**Hypothese 1.3:** Mit Skillmanagement sinkt die Wissensweitergabe zwischen Mitarbeitern.

Durch die „bessere“ Zuteilung der Arbeitspakete treten seltener Situationen auf, die es erfordern andere Mitarbeiter um Rat zu fragen. Daher werden auch die Anzahl an Wissensweitergabeaktivitäten und die Menge des weitergegebenen Wissens sinken.

**Hypothese 1.4:** Im Zeitverlauf werden die oben beschriebenen Effekte geringer.

Dadurch, dass im Laufe der Zeit immer mehr Metawissen bei den Agenten dazu gewonnen wird, ist der Unterschied zwischen den Szenarien mit und ohne Skillmanagement zu Beginn am größten. Weil das Skillmanagement in der Simulation aber zu jedem Zeitpunkt mit aktuellen Werten gefüllt ist, während das Metawissen der Agenten veraltet, bleiben auch im Zeitverlauf beträchtliche Unterschiede bestehen.

**Hypothese 1.5:** Je höher die Anzahl an Mitarbeitern, desto größer ist der in Hypothese 1.2 beschriebene Effekt.

Da jeder Agent nur mit einer begrenzten Anzahl an Kollegen Kontakt haben kann, ist auch sein Metawissen über die Kollegen begrenzt. Mit steigender Mitarbeiteranzahl bleibt das Metawissen der Agenten relativ konstant, so dass die Unterschiede zwischen den Szenarien mit und ohne Skillmanagement umso höher sind, je mehr Mitarbeiter es gibt.

### **6.5.3 Experiment 1.3: Skillmanagement auch für Projektmitarbeiter**

Der Aufbau ist wie in Experiment 1.2, allerdings steht das laufend aktualisierte Skillmanagementsystem auch den Projektmitarbeitern zur Verfügung und kann für die Auswahl von Kollegen genutzt werden, die um Hilfe gebeten werden.

**Hypothese 1.6:** Mit Skillmanagement ist der Erfolg der Wissensweitergabeaktionen höher. Wie in Hypothese 1.4 vorhergesagt werden aufgrund guter Übereinstimmung von gefordertem und bestehendem Wissen nur wenige Wissensweitergabeaktivitäten (pull Gelegenheiten) entstehen. Tritt so eine Gelegenheit aber auf, so kann im Unterschied zu anderen Experimenten immer ein kompetenter Mitarbeiter gefragt werden, so dass der Erfolg der Wissensweitergabeaktivitäten erheblich steigt (siehe Abschnitt 2.5.3).

### **6.5.4 Ergebnisse**

Alle Experimente der Studie wurden in je 50 Simulationsläufen getestet. Die Ergebnisse der Simulationsläufe sind als Mittelwerte (fett gedruckt) und Standardabweichungen (normal gedruckt) in Tabelle 6-10 und Tabelle 6-11 zusammengefasst. Die Anzahl abgearbeiteter Projekte und der damit generierte Umsatz sind nahezu linear von der Anzahl der Agenten abhängig. Die Umsätze ergeben sich aus dem geschätzten Aufwand bei  $\text{Skill}_{\text{erforderlich}} = \text{Skill}_{\text{vorhanden}}$  multipliziert mit einem Tagessatz, der gleichverteilt zwischen 500€ und 1100€ schwankt (Mittelwert 800€). Der Deckungsbeitrag (DB) ergibt sich aus dem Umsatz abzüglich der Personalkosten. Diese ergeben sich aus Bruttomonatslöhnen von 4400€ für die Projektmitarbeiter und 6000€ für die Projektmanager (siehe dazu auch die Gehaltsstudien in Abschnitt 6.1.2) und von Lohnnebenkosten in selber Höhe ausgegangen. Da drei Arbeitspakete pro Projekt vorkommen, sind auch Projektmanager und Projektmitarbeiter im Verhältnis 1 zu 3 gewählt wurden. Daraus ergibt sich die ungerade Anzahl an Agenten (z.B. 1 CEO + 12 PMs + 36 Projektmitarbeiter = 49 Agenten<sub>gesamt</sub>).

Tabelle 6-10: Ergebnisse der Skillmanagement-Studie Teil 1

Agenten	WM-Maßnahme	Gewinn	Umsatz	Abg. Projekte	Abg. WPs	Skill max	Skill min	Skill avg	Skill total
49	keines	<b>4,300 Mio €</b>	<b>20,889 Mio €</b>	<b>450,90</b>	<b>1360,24</b>	<b>3,66</b>	<b>1,29</b>	<b>2,38</b>	<b>428,86</b>
		0,365 Mio €	0,365 Mio €	7,81	23,45	0,04	0,17	0,09	16,35
	SkMgmt f. PMs	<b>4,210 Mio €</b>	<b>20,806 Mio €</b>	<b>450,14</b>	<b>1358,20</b>	<b>3,72</b>	<b>1,17</b>	<b>2,33</b>	<b>419,97</b>
		0,331 Mio €	0,331 Mio €	7,10	20,70	0,03	0,12	0,07	13,15
	SkMgmt f. Alle	<b>4,109 Mio €</b>	<b>20,698 Mio €</b>	<b>448,85</b>	<b>1355,40</b>	<b>3,71</b>	<b>1,19</b>	<b>2,35</b>	<b>423,46</b>
		0,254 Mio €	0,254 Mio €	6,01	17,75	0,03	0,13	0,07	13,08
77	keines	<b>5,231 Mio €</b>	<b>31,497 Mio €</b>	<b>676,75</b>	<b>2038,50</b>	<b>3,69</b>	<b>1,56</b>	<b>2,63</b>	<b>749,75</b>
		0,318 Mio €	0,318 Mio €	6,59	19,71	0,01	0,12	0,06	15,91
	SkMgmt f. PMs	<b>4,607 Mio €</b>	<b>30,873 Mio €</b>	<b>663,70</b>	<b>1999,90</b>	<b>3,64</b>	<b>1,51</b>	<b>2,57</b>	<b>732,32</b>
		0,238 Mio €	0,238 Mio €	4,54	12,84	0,02	0,10	0,05	14,19
	SkMgmt f. Alle	<b>4,476 Mio €</b>	<b>30,742 Mio €</b>	<b>660,40</b>	<b>1989,00</b>	<b>3,63</b>	<b>1,52</b>	<b>2,57</b>	<b>731,96</b>
		0,330 Mio €	0,330 Mio €	7,66	21,98	0,03	0,20	0,08	21,89
101	keines	<b>9,445 Mio €</b>	<b>44,005 Mio €</b>	<b>944,22</b>	<b>2852,00</b>	<b>3,72</b>	<b>1,15</b>	<b>2,36</b>	<b>884,37</b>
		0,409 Mio €	0,409 Mio €	10,23	30,60	0,02	0,13	0,07	27,37
	SkMgmt f. PMs	<b>9,138 Mio €</b>	<b>43,698 Mio €</b>	<b>935,78</b>	<b>2826,44</b>	<b>3,65</b>	<b>1,10</b>	<b>2,29</b>	<b>859,42</b>
		0,574 Mio €	0,574 Mio €	12,50	36,34	0,03	0,07	0,06	23,60
	SkMgmt f. Alle	<b>8,640 Mio €</b>	<b>43,470 Mio €</b>	<b>932,07</b>	<b>2814,60</b>	<b>3,66</b>	<b>1,13</b>	<b>2,30</b>	<b>862,63</b>
		0,567 Mio €	0,567 Mio €	12,40	37,68	0,03	0,09	0,06	20,95

Bezüglich der Skillwerte ergibt sich kein einheitliches Bild. Während die Maximalwerte bei 49 Agenten keine signifikanten Unterschiede aufweisen, liegen sie bei 77 und 101 Agenten sogar geringfügig unter dem Basisszenario ( $\alpha=0,001$ ,  $t=0,00012$  bei 77 und  $\alpha=0,05$ ,  $t=0,01105$  bei 101 Agenten). Bei den Minimalwerten ist die Situation genau umgekehrt. Bei 49 Agenten liegt  $Skill_{min}$  signifikant unter dem Basisszenario ( $\alpha=0,001$ ,  $t=0,00000$ ), während bei 77 und 101 Agenten keine signifikanten Unterschiede auftreten. Bei den Durchschnittswerten sind schließlich bei 49 und 77 Agenten signifikant niedrigere Werte zu beobachten ( $\alpha=0,001$ ,  $t=0,00012$  bei 49 und  $\alpha=0,05$ ,  $t=0,01255$  bei 77 Agenten). Tabelle 6-11 zeigt den Grund für diese unerwarteten Ergebnisse. Dadurch, dass der PM einen Arbeiter mit möglichst geringer Abweichung zwischen benötigtem und vorhandenem Skillwert auswählt, steigt die durchschnittliche Bearbeitungsdauer der Arbeitspakete (angegeben in Tagen,  $\alpha=0,01$ ,  $t=0,00095$  bis  $0,00235$ ), es wird aber während der Projektarbeit mehr gelernt ( $\alpha=0,001$ ,  $t=0,0000$  bis  $0,00057$ ) und auch das Lernen von Kollegen steigt ( $\alpha=0,001$ ,  $t=0,00035$  bis  $0,00065$ ), weil öfter Probleme auftreten. Dies ist aber mit den verwendeten Parametern weniger effektiv als zwischen zwei Projekten durch Selbststudium (Lernen vom PC) zu lernen, so dass dort die Werte mit Skillmanagement signifikant ( $\alpha=0,01$ ,  $t=0,00000$  bis  $0,00371$ ) niedriger liegen als ohne und die Skillwerte insgesamt sinken. Durch steigende Bearbeitungsdauer der Arbeitspakete bei gleichbleibender Anzahl Projekte steigt auch die Auslastung der Agenten signifikant ( $\alpha=0,05$ ,  $t=0,00000$  bei 49 Agenten bis  $t=0,02061$  bei 101 Agenten).

Tabelle 6-11: Ergebnisse der Skillmanagement-Studie Teil 2

Agenten	WM-Maßnahme	Lernen PC	Lernen Kollege	Lernen Arbeit	Vergessen	Auslastung	WP Dauer	$\Delta$ WP Dauer
49	keines	<b>451,33</b>	<b>37,96</b>	<b>176,83</b>	<b>628,40</b>	<b>75,78%</b>	<b>17,66</b>	<b>1,44</b>
		17,42	2,47	8,31	14,02	0,40%	0,27	0,28
	SkMgmt f. PMs	<b>438,27</b>	<b>40,49</b>	<b>187,33</b>	<b>645,24</b>	<b>76,20%</b>	<b>18,18</b>	<b>1,29</b>
		17,54	2,37	7,47	14,16	0,34%	0,24	0,24
77	keines	<b>435,35</b>	<b>43,23</b>	<b>187,99</b>	<b>642,23</b>	<b>76,16%</b>	<b>18,26</b>	<b>1,22</b>
		15,01	2,37	8,43	12,05	0,32%	0,19	0,18
	SkMgmt f. PMs	<b>804,05</b>	<b>59,65</b>	<b>285,78</b>	<b>1051,29</b>	<b>74,57%</b>	<b>18,35</b>	<b>0,27</b>
		20,17	4,39	9,41	23,95	0,24%	0,20	0,52
101	keines	<b>779,74</b>	<b>66,71</b>	<b>296,68</b>	<b>1062,37</b>	<b>74,93%</b>	<b>18,88</b>	<b>0,18</b>
		20,72	3,65	10,61	17,38	0,25%	0,10	0,31
	SkMgmt f. PMs	<b>787,81</b>	<b>69,06</b>	<b>299,39</b>	<b>1075,86</b>	<b>74,81%</b>	<b>19,01</b>	<b>0,57</b>
		20,55	4,26	8,75	26,43	0,34%	0,23	0,00
101	keines	<b>924,34</b>	<b>84,67</b>	<b>367,26</b>	<b>1357,45</b>	<b>76,12%</b>	<b>17,99</b>	<b>1,55</b>
		22,15	5,30	7,65	16,15	0,32%	0,21	0,21
	SkMgmt f. PMs	<b>826,63</b>	<b>94,72</b>	<b>418,40</b>	<b>1345,89</b>	<b>76,56%</b>	<b>18,28</b>	<b>1,24</b>
		31,50	4,04	12,91	22,99	0,24%	0,18	0,18
101	keines	<b>839,48</b>	<b>97,01</b>	<b>416,22</b>	<b>1355,64</b>	<b>76,51%</b>	<b>18,35</b>	<b>1,18</b>
		31,47	4,62	12,46	27,16	0,22%	0,18	0,18
	SkMgmt f. PMs							
	SkMgmt f. Alle							

Nachdem die Ergebnisse allgemein dargestellt wurden, werden nun die Hypothesen im Einzelnen diskutiert.

**Hypothese 1.1:** Mit Skillmanagement sinkt die durchschnittliche Bearbeitungszeit von Arbeitspaketen im Vergleich zum Basisszenario.

Hypothese 1.1 muss abgelehnt werden. Die Bearbeitungszeit steigt im Schnitt statt zu sinken. Dies scheint jedoch der gewählten Strategie des PM geschuldet. Wählt man stattdessen eine Strategie, bei der immer derjenige Projektmitarbeiter mit dem höchsten Skillwert ausgewählt wird, so könnte die Hypothese dennoch zutreffen. Dies wurde jedoch in der vorliegenden Arbeit nicht mehr überprüft.

**Hypothese 1.2:** Mit Skillmanagement steigt die Menge an Wissen, dass durch Lernen bei der Arbeit (learning by doing) erworben wird.

Hypothese 1.2 muss angenommen werden. Das Wissen, das durch Lernen bei der Arbeit erworben wurde, ist signifikant ( $\alpha=0,001$ ,  $t=0,0000$  bis  $0,00057$ ) höher als ohne Skillmanagement.

**Hypothese 1.3:** Mit Skillmanagement sinkt die Wissensweitergabe zwischen Mitarbeitern.

Hypothese 1.3 muss abgelehnt werden. Die Wissensweitergabe zwischen Mitarbeitern steigt durch Skillmanagement signifikant an ( $\alpha=0,001$ ,  $t=0,00035$  bis  $0,00065$ ).

**Hypothese 1.4:** Im Zeitverlauf wird der in Hypothese 1.2 beschriebene Effekt geringer.

Die Unterschiede bzgl. Lernen von Kollegen und Lernen bei der Arbeit steigen beide im Zeitverlauf signifikant ( $\alpha=0,001$ ,  $t=0,0000$ ), wie aus Tabelle 6-12 abzulesen ist. Relevant

sind dabei v.a. die letzten beiden Spalten mit den jährlichen Veränderungen gegenüber dem Basisszenario. Daher muss Hypothese 1.4 abgelehnt werden.

Tabelle 6-12: Veränderung der Lernergebnisse im Zeitverlauf

Agenten	WM-Maßnahme	Jahr	Lernen Kollege	Lernen Arbeit	$\delta$ Kollege	$\delta$ Arbeit	$\delta$ Kollege p.a.	$\delta$ Arbeit p.a.
49	Keines	1	14,66	65,05				
		2	28,36	120,12				
		3	38,69	180,53				
	SkMgmt f. PMs	1	14,16	65,64	-0,50	0,60	-0,50	0,60
		2	28,73	124,10	0,37	3,98	0,18	1,99
		3	40,40	187,47	1,72	6,94	0,57	2,31
	SkMgmt f. Alle	1	15,40	63,35	0,74	-1,70	0,74	-1,70
		2	30,41	123,39	2,05	3,27	1,02	1,63
		3	43,23	187,99	4,54	7,46	1,51	2,49
77	Keines	1	24,37	101,80				
		2	46,40	186,55				
		3	59,78	285,01				
	SkMgmt f. PMs	1	25,44	96,43	1,07	-5,37	1,07	-5,37
		2	50,00	192,10	3,60	5,55	1,80	2,77
		3	66,71	296,68	6,93	11,67	2,31	3,89
	SkMgmt f. Alle	1	27,83	96,90	3,46	-4,90	3,46	-4,90
		2	53,60	196,04	7,20	9,49	3,60	4,74
		3	69,06	299,39	9,28	14,38	3,09	4,79
101	Keines	1	28,68	133,76				
		2	61,50	248,52				
		3	85,47	371,50				
	SkMgmt f. PMs	1	32,27	145,10	3,58	11,34	3,58	11,34
		2	67,40	285,76	5,89	37,24	2,95	18,62
		3	95,01	420,10	9,54	48,59	3,18	16,20
	SkMgmt f. Alle	1	33,10	137,66	4,42	3,91	4,42	3,91
		2	72,27	270,62	10,76	22,10	5,38	11,05
		3	100,49	412,65	15,02	41,15	5,01	13,72

**Hypothese 1.5:** Je höher die Anzahl an Mitarbeitern, desto größer ist der in Hypothese 1.2 beschriebene Effekt.

Rechnet man die Werte aus Tabelle 6-11, welche den gesamten Lernerfolg enthält, auf die Anzahl der Projektmitarbeiter, so erhält man die Werte aus Tabelle 6-13. Die Ergebnisse sind auch hier nicht eindeutig. Während der Zuwachs beim Lernen bei der Arbeit im Vergleich zum Basisszenario von 0,292 bei 49 Agenten auf 0,205 bei 77 Agenten sinkt, steigt er bei 101 Agenten auf 0,714. Dies entspricht einer signifikanten Steigerung von 0,0081 auf 0,0198 Lernzuwachs pro Agent vom Szenario mit 49 Agenten zum Szenario mit 101 Agenten ( $\alpha=0,001$ ,  $t=0,0000$ ). Eine eindeutige Aussage lässt sich also nicht treffen. Da keine eindeutige Bestätigung erkennbar ist, muss die Hypothese abgelehnt werden.

Tabelle 6-13: Vergleich der Wissensweitergabeerfolge bei verschiedener Anzahl Agenten

Agenten	WM-Maßnahme	Lernen Kollege	$\Delta$ Kollege	$\Delta$ Kollege p. Agent	Lernen Arbeit	$\Delta$ Arbeit	$\Delta$ Arbeit p. Agent
49	keines	1,0545			4,9118		
	SkMgmt f. PMs	1,1248	0,0703	0,0020	5,2037	0,2919	0,0081
	SkMgmt f. Alle	1,2007	0,1462	0,0041	5,2218	0,3100	0,0086
77	keines	1,0488			5,0002		
	SkMgmt f. PMs	1,1704	0,1216	0,0034	5,2049	0,2047	0,0057
	SkMgmt f. Alle	1,2116	0,1628	0,0045	5,2524	0,2522	0,0070
101	keines	1,1396			4,9534		
	SkMgmt f. PMs	1,2541	0,1145	0,0032	5,6674	0,7140	0,0198
	SkMgmt f. Alle	1,2803	0,1407	0,0039	5,5574	0,6041	0,0168

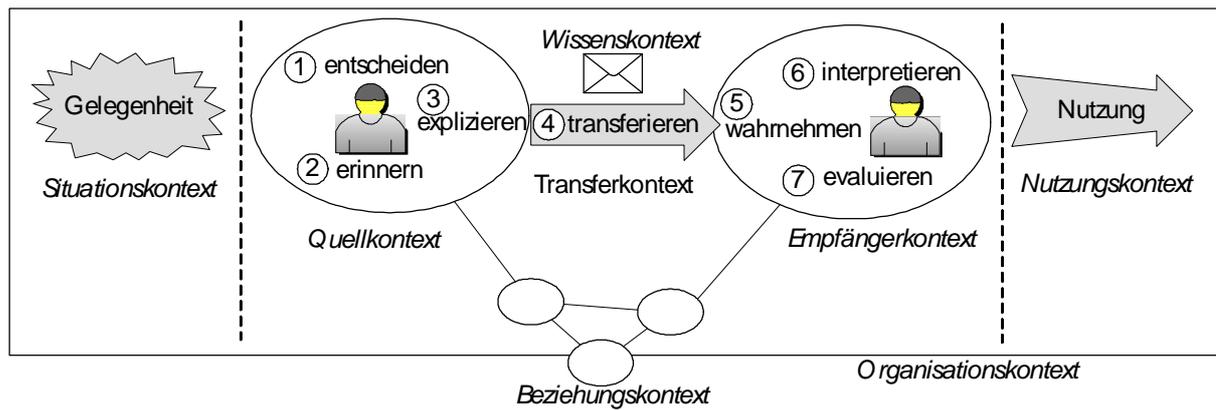
**Hypothese 1.6:** Mit Skillmanagement ist der Erfolg der Wissensweitergabeaktionen höher.

Misst man den Erfolg wie in Abschnitt 2.5.3 beschrieben, so bleibt für das vereinfachte Simulationsmodell die Gleichung  $\text{Erfolg} = \text{Menge an weitergegebenem Wissen} / \text{Anzahl an Wissensweitergabeaktionen}$ . Dabei steigt der Erfolg signifikant ( $\alpha=0,01$ ,  $t=0,00873$ ) von 0,0058 auf 0,0089<sup>74</sup>. Die Hypothese muss daher angenommen werden. Die Steigerung ist in erster Linie auf die Reduktion von Weitergabeaktionen zurückzuführen, die ohne weitergegebenes Wissen enden.

## 6.6 Studie 2: Dokumentation von Lernerfahrungen

Für die Simulationsstudie „Dokumentation von Lernerfahrungen“ ist die zentrale Forschungsfrage: „Wie wirken sich gezielt eingesetzte dokumenten-orientierte WM-Maßnahmen auf die Wissensweitergabe aus?“ (siehe Abschnitt 2.4.2). Auf das Wissensweitergabemodell bezogen stellt das Dokumentieren von Lernerfahrungen die erste Hälfte des Wissensweitergabeprozesses dar (Schritte 1-3 plus einen Teil von Schritt 4, siehe Abbildung), wenn ein einzelner Mitarbeiter sein Wissen dokumentiert. Werden Erfahrungen jedoch im Team diskutiert und bewertet, so stellt dies nach der in der Arbeit verwendeten Definition (siehe Abschnitt 2.2.3) den Rahmen für eine Reihe von einzelnen Wissensweitergabeaktivitäten dar. Idealtypisch werden dabei sowohl Push- als auch Pull-Weitergaben stattfinden (siehe Abschnitt 3.2.1). Push-Weitergaben liegen vor, wenn ein Mitarbeiter von sich aus Erfahrungen äußert, die für andere Mitarbeiter neue Informationen darstellen. Dadurch ergeben sich wieder Gelegenheiten für Nachfragen nach zusätzlichen Informationen, also Pull-Weitergaben.

<sup>74</sup> in Skilleinheiten pro Runde



In den folgenden Experimenten werden dazu die Auswirkungen der Dokumentation von Lernerfahrungen auf persönlicher Ebene in Experiment 2.2 und auf Teamebene in Experiment 2.3 untersucht und den Ergebnissen ohne dokumentenorientierte WM-Maßnahme in Experiment 2.1 gegenübergestellt.

### 6.6.1 Experiment 2.1: Keine dokumenten-orientierten WM-Maßnahmen

Dieses Experiment dient als Vergleichsgrundlage und ist identisch mit Experiment 1.1, bis auf die Tatsache, dass Lernen vom PC nur bis zu einem Maximalwert von 2,5 funktioniert. Die zugrunde liegende Annahme ist, dass in den öffentlich zugänglichen Bereichen des Internets nur Basiswissen vermittelt wird, mit dem alleine ein Mitarbeiter die Ebene des Kompetenten nicht erreichen kann. Die Absenkung des Startwertes von 3,0 in Experiment 1.1 auf 2,5 in Experiment 2.1 erfolgte, um den Effekt durch die Dokumentation von Lernerfahrungen deutlicher zu machen. Da dieser Wert in Studie 1 statisch ist, während er in Studie 2 durch die Dokumentation von Lernerfahrungen steigen kann, erscheint diese Maßnahme sinnvoll.

### 6.6.2 Experiment 2.2: persönliche Lernerfahrungen (Wissensdokumentation)

Jeder Mitarbeiter soll nach einer Lernerfahrung selbst entscheiden, ob er das neue erworbene Wissen dokumentieren und in ein zentrales Informationssystem einstellen will. Dieses dokumentierte Wissen kann von anderen Mitarbeitern dazu genutzt werden um beim Lernen vom PC den Skillwert über 2,5 hinaus zu steigern. Jeder Beitrag, der von einem Mitarbeiter mit höherem Skillwert als dem im ECM verfügbaren geschrieben wird, erhöht den Maximalwert, der durch Lernen am PC in diesem Wissensbereich erreicht werden kann. Der Anreiz für die Mitarbeiter zum Dokumentieren ist ein steigender Expertenstatus und eine steigende Wahrscheinlichkeit für Gehaltserhöhungen, also zwei Komponenten, die in die Nutzenfunktion eingehen. Dies ist dem WM-Instrument „persönliches Erfahrungsmanagement“ zuzuordnen (siehe Abschnitt 2.4.2).

**Hypothese 2.1:** Die Mitarbeiter dokumentieren regelmäßig ihr neu erworbenes Wissen.

Da die Auslastung der Mitarbeiter im Basisszenario relativ gering ist (im Bereich von 70%), ist genügend Zeit zur Dokumentation. Die Mitarbeiter werden daher die Zeit nutzen, um ihr Wissen zu dokumentieren, wenn der Nutzenbeitrag bei gängigen Zustandswerten höher ausfällt, als jener der `Idle` Aktion.

**Hypothese 2.2:** Die Wissensweitergabe über dokumentiertes Wissen ist höher.

Da die Mitarbeiter ihr Wissen häufig dokumentieren, steht mehr dokumentiertes Wissen zur Verfügung. Dadurch steigt der maximale Skillwert, der durch Lernen vom PC erworben werden kann. Dies wiederum erhöht die durch Dokumente medierte Wissensweitergabe.

**Hypothese 2.3:** Die Quantität sozialer Kontakte zwischen Mitarbeitern ist niedriger.

Da mehr Wissen in dokumentierter, expliziter Form vorliegt, besteht weniger Notwendigkeit für direkte persönliche Wissensweitergabe und damit auch für soziale Kontakte.

**Hypothese 2.4:** Das durchschnittliche Wissen der Mitarbeiter ist höher.

Da das Wissen durch Lernen von PC auf mehr als 2,5 gesteigert werden kann, wenn entsprechende Projekterfahrungen hinterlegt wurden, wird das durchschnittliche Wissen der Mitarbeiter steigen.

**Hypothese 2.5:** Im Zeitverlauf wird die Steigerung des im Intranet abgelegten Wissens immer geringer.

Da die maximal erreichten Skillwerte der Agenten nicht so stark steigen wie das im Intranet verfügbare Wissen, können immer weniger Agenten mit neuen Projekterfahrungen die Qualität der Wissensbasis im Intranet steigern.

### **6.6.3 Experiment 2.3: Lernerfahrungen im Team (Projektreview)**

Nach jedem abgeschlossenen Projekt dokumentieren die Beteiligten das neu erworbene Wissen und erarbeiten in ausführlichen Projektnachbesprechungen zusammen die gemachten Erfahrungen. Neben dem Effekt der Dokumentation des Wissens können in diesem Experiment alle Teammitglieder direkt von den Kollegen lernen wenn die Projekterfahrungen dokumentiert werden. Dies ist dem WM-Instrument „Erfahrungsmanagement im Team“ zuzuordnen (siehe Abschnitt 2.4.2).

**Hypothese 2.6:** Die Wissensweitergabe durch dokumentiertes Wissen ist höher als in Experiment 2.1.

Durch die Projektreviews wird zusätzliches Wissen im Intranet verfügbar gemacht, so dass der Skillwert durch Lernen von PC über den im Experiment 2.1 möglichen Wert von 2,5 gesteigert werden kann. Daher ist die Wissensweitergabe durch dokumentiertes Wissen höher.

**Hypothese 2.7:** Die Wissensweitergabe durch dokumentiertes Wissen ist geringer als in Experiment 2.2.

Da verhältnismäßig wenige Projekte bearbeitet werden und daher wenige Dokumentationsgelegenheiten auftreten wird vergleichsweise wenig Wissen dokumentiert. Daher wird auch nur wenig mehr Wissen aus Dokumenten gelernt als im Basisszenario.

**Hypothese 2.8:** Die Wissensweitergabe durch Lernen von Kollegen ist höher.

Die Projektnachbesprechungen dienen nicht nur zur Dokumentation von Wissen, sondern ermöglichen auch die Wissensweitergabe zwischen den Teammitgliedern. Da bei dieser Gelegenheit mehrere Mitarbeiter voneinander lernen, wird innerhalb kurzer Zeit relativ viel Wissen weitergegeben. Da die Menge an Wissen, die durch direkte Wissensweitergabe erworben wird, insgesamt auf niedrigem Niveau ist, ergibt sich ein hoher prozentualer Anstieg.

**Hypothese 2.9:** Die Quantität sozialer Kontakte zwischen Mitarbeitern ist durch die Projektnachbesprechungen höher.

Die Projektnachbesprechungen sind eine Gelegenheit, soziale Kontakte zu vertiefen. Durch diese zusätzliche Gelegenheit steigt die Kontakthäufigkeit.

**Hypothese 2.10:** Es wird weniger vergessen, da in den Projektnachbesprechungen viele Wissensbereiche genutzt werden.

Wissen wird vergessen, wenn es längere Zeit nicht benötigt wird. Da in jedem Arbeitspaket nur ein Wissensbereich benötigt wird, werden die übrigen Bereiche nicht genutzt und es kann zu Vergessen kommen. In den Projektnachbesprechungen wird über alle Arbeitspakete und damit potentiell verschiedene Wissensgebiete gesprochen. Dadurch werden mehr Wissensgebiete aktiviert als ohne Projektnachbesprechungen und es wird weniger vergessen.

**Hypothese 2.11:** Je mehr Mitarbeiter das Unternehmen hat, umso mehr Wissen steht im Intranet zur Verfügung.

Je mehr Mitarbeiter das Unternehmen hat, desto mehr Projekte werden abgewickelt und desto mehr Erfahrungen können gemacht werden. Da das dokumentierte Wissen im Intranet das kumulierte Wissen darstellt, steigt es mit zunehmender Mitarbeiteranzahl.

#### 6.6.4 Ergebnisse

Beide Formen der Dokumentation von Lernerfahrungen führen zu einer Steigerung der Leistungsfähigkeit der Organisation, da durch eine höhere Anzahl abgearbeiteter Projekte sowohl der Umsatz ( $\alpha=0,001$ ,  $t=0,00002$ , bzw.  $0,00020$  für 49 Agenten), als auch das kumulierte Wissen aller Mitarbeiter signifikant steigen ( $\alpha=0,01$ ,  $t=0,00937$  für Wissensdokumentation,  $\alpha=0,001$ ,  $t=0,00000$  für Projektdokumentation, siehe Tabelle 6-14).

Tabelle 6-14: Ergebnisse der Studie Wissensdokumentation Teil 1

Agenten	WM-Maßnahme	DB	Umsatz	Abg. Projekte	Abg. WPs	Skill max	Skill min	Skill avg	Skill total
49	keines	4,110 Mio €	20,569 Mio €	445	1342	3,73	1,19	2,27	408,07
		0,272 Mio €	0,272 Mio €	6,23	16,97	0,05	0,10	0,07	13,09
	Wissensdokumentation	5,066 Mio €	21,525 Mio €	466	1409	4,76	1,02	2,36	424,42
		0,395 Mio €	0,395 Mio €	9,51	28,29	0,11	0,08	0,06	11,56
	Projektreviews	4,667 Mio €	21,127 Mio €	456	1379	4,38	1,31	2,81	505,57
		0,258 Mio €	0,258 Mio €	3,81	12,45	0,07	0,07	0,08	15,17
77	keines	4,500 Mio €	30,636 Mio €	659	1985	3,64	1,40	2,43	692,64
		0,511 Mio €	0,511 Mio €	11,12	32,65	0,05	0,10	0,05	13,30
	Wissensdokumentation	6,066 Mio €	32,202 Mio €	686	2068	4,89	1,09	2,58	735,89
		0,483 Mio €	0,483 Mio €	9,33	27,88	0,06	0,10	0,06	17,19
	Projektreviews	5,045 Mio €	31,181 Mio €	678	2045	4,57	1,67	3,28	935,99
		0,487 Mio €	0,487 Mio €	10,54	31,49	0,08	0,18	0,10	28,53
101	keines	8,064 Mio €	42,823 Mio €	913	2758	3,67	1,17	2,31	865,30
		0,775 Mio €	0,775 Mio €	15	46	0,03	0,09	0,06	21,02
	Wissensdokumentation	10,558 Mio €	44,989 Mio €	967	2922	4,68	0,97	2,29	859,01
		0,768 Mio €	0,768 Mio €	17	48	0,05	0,13	0,04	16,49
	Projektreviews	10,271 Mio €	44,299 Mio €	959	2898	4,55	1,14	2,87	1078,00
		0,548 Mio €	0,548 Mio €	9	29	0,04	0,08	0,06	23,21

Das Dokumentieren von persönlichen Lernerfahrungen scheint dabei auf den ersten Blick effektiver zu sein, da die Deckungsbeiträge in diesem Experiment stärker steigen, als bei der Dokumentation auf Teamebene in Projektreviews. Bei genauerer Analyse wird jedoch deutlich, dass bei Dokumentation auf persönlicher Ebene Spezialistentum gefördert wird ( $\text{Skill}_{\text{avg}}$  steigt nur geringfügig,  $\Delta=0,09$ ,  $\alpha=0,01$ ,  $t=0,00940$ ,  $\text{Skill}_{\text{max}}$  dagegen stark,  $\Delta=1,03$ ,  $\alpha=0,001$ ,  $t=0,00000$  für 49 Agenten), während durch Projektreviews die Mitarbeiter zu Generalisten ausgebildet werden ( $\text{Skill}_{\text{avg}}$  steigt im Vergleich zum Basisszenario stärker als bei Wissensdokumentation  $\Delta=0,54$ ,  $\alpha=0,001$ ,  $t=0,00000$ ,  $\text{Skill}_{\text{max}}$  weniger stark,  $\Delta=0,65$ ,  $\alpha=0,001$ ,  $t=0,00000$  für 49 Agenten). Mit Projektreviews steigt das kumulierte Wissen der Mitarbeiter signifikant höher an als bei Wissensdokumentation ( $\Delta=81,15$ ,  $\alpha=0,001$ ,  $t=0,00000$  bei 49 Agenten).

Tabelle 6-15: Ergebnisse der Studie Wissensdokumentation Teil 2

Agenten	WM-Maßnahme	Lernen PC	Lernen Kollege	Lernen Arbeit	Vergessen	Auslastung	WP Dauer	$\Delta$ WP	Dauer
49	keines	419,66	43,03	186,41	626,99	76,35%	18,22	1,29	
		10,06	2,44	6,44	11,59	0,35%	0,18	0,17	
	Wissensdokumentation	413,96	53,31	144,42	573,23	76,01%	16,35	3,13	
		30,60	2,79	7,55	18,74	0,35%	0,30	0,31	
	Projektreviews	504,87	66,82	139,72	591,81	74,24%	17,21	2,29	
		20,88	3,75	10,59	11,68	0,39%	0,22	0,21	
77	keines	718,87	70,97	317,95	1059,83	75,26%	19,01	0,57	
		19,96	2,47	10,16	18,32	0,19%	0,34	0,33	
	Wissensdokumentation	752,45	98,22	233,07	1006,52	74,35%	16,75	2,83	
		31,51	4,29	13,44	29,12	0,35%	0,21	0,21	
	Projektreviews	932,39	98,82	222,80	962,70	72,38%	17,72	1,85	
		9,08	2,07	3,49	23,29	0,30%	0,34	0,33	
101	keines	822,80	99,59	441,10	1321,78	76,34%	18,30	1,22	
		21,43	4,22	15,20	10,03	0,30%	0,19	0,26	
	Wissensdokumentation	761,54	120,93	317,60	1164,65	76,38%	16,22	2,99	
		39,90	2,31	13,76	28,87	0,29%	0,27	0,00	
	Projektreviews	1030,45	133,84	325,15	1235,04	74,45%	17,02	2,50	
		13,68	5,47	14,95	18,23	0,22%	0,21	0,21	

Bemerkenswert ist auch, dass bei Wissensdokumentation keine signifikante Steigerung des Lernens von PC zu beobachten ist, während das Lernen von Kollegen signifikant steigt ( $\alpha=0,001$ ,  $t=0,00000$  bis  $0,00011$ ) und das Lernen bei der Arbeit signifikant sinkt ( $\alpha=0,001$ ,  $t=0,00000$  bis  $0,00003$ ). Die Steigerung beim Wissen wird in erster Linie über geringeres Vergessen realisiert ( $\alpha=0,001$ ,  $t=0,00002$ , bei 77 Agenten  $\alpha=0,05$ ,  $t=0,01746$ ). Nach dieser Übersicht über die Ergebnisse folgt die Analyse der Hypothesen.

**Hypothese 2.1:** Die Mitarbeiter dokumentieren regelmäßig ihr neu erworbenes Wissen.

Wie Tabelle 6-16 zeigt, ist das Dokumentieren von Lernerfahrungen eine häufige Tätigkeit, die rund 15% der Arbeitszeit in Anspruch nimmt. Dies entspricht der Annahme, dass die Dokumentation von Lernerfahrungen erhebliche Arbeitszeit kostet und nicht „nebenbei“ erledigt werden kann. Die Hypothese muss also angenommen werden.

Tabelle 6-16: Verteilung der Arbeitszeit auf einzelne Tätigkeiten in 1080 Tagen Arbeitszeit

Agenten	Tage Projektarbeit	Tage Lernen von Kollege	Tage Dokumentieren
49	704,75	14,50	178,94
77	682,91	15,59	157,56
101	679,03	17,00	163,00

**Hypothese 2.2:** Die Wissensweitergabe über dokumentiertes Wissen ist höher.

Diese Hypothese muss abgelehnt werden. Der Zuwachs an Wissen wird nicht über eine Erhöhung der Wissensweitergabe über dokumentiertes Wissen realisiert. Es ist keine signifikante Veränderung gegenüber dem Basisszenario festzustellen. Eine mögliche Erklärung dafür ist, dass die Mitarbeiter einen Teil der Arbeitszeit, die nicht für Projektarbeit aufgewendet wird, für die Dokumentation von Wissen verwenden und damit

weniger Zeit zum Lernen zur Verfügung haben. Lernen von Kollegen steht insofern nicht in unmittelbarer Konkurrenz, als es vorwiegend gewählt wird, wenn bei der Projektarbeit das Wissen nicht ausreicht, während Lernen von PC hauptsächlich zwischen zwei Projekten als Alternative zur `Idle` Aktion gewählt wird.

**Hypothese 2.3:** Die Quantität sozialer Kontakte zwischen Mitarbeitern ist niedriger.

Bei 49 Agenten sinkt die Kontakthäufigkeit tatsächlich wie prognostiziert ( $\alpha=0,01$ ,  $t=0,00472$ ). Bei 77 und 101 Agenten steigen die Anzahl der sozialen Kontakte allerdings signifikant an ( $\alpha=0,05$ ,  $t=0,03308$  bei 77,  $\alpha=0,001$ ,  $t=0,00078$  bei 101 Agenten, siehe Tabelle 6-17). Diese Entwicklung wird auch durch das gestiegene von Kollegen erworbene Wissen untermauert ( $\alpha=0,001$ ,  $t=0,0000$  bis  $0,00011$ , siehe Tabelle 6-15). Daher sollte die Hypothese abgelehnt werden.

Tabelle 6-17: Kontakthäufigkeit in den verschiedenen Szenarien

WM-Maßnahme	49 Agenten	77 Agenten	101 Agenten
keines	4402,82	2705,18	2340,80
Wissensdokumentation	3554,90	2752,00	2474,25
Projektreviews	4478,10	2771,00	2438,00

**Hypothese 2.4:** Das durchschnittliche Wissen der Mitarbeiter ist höher.

Wie in Tabelle 6-14 zu sehen ist, kann diese Hypothese angenommen werden. Das durchschnittliche Wissen der Mitarbeiter steigt bei 49 und 77 Agenten zwar weniger als erwartet, aber signifikant ( $\alpha=0,01$ ,  $t=0,00940$ , bzw.  $0,00400$ ). Bei 101 Agenten stellt sich dagegen keine signifikante Änderung ein. Das Wissen im dem Wissensgebiet mit dem höchsten persönlichen Wissen ( $Skill_{max}$ ) steigt allerdings weitaus deutlicher.

**Hypothese 2.5:** Im Zeitverlauf wird die Steigerung des im Intranet abgelegten Wissens immer geringer.

Tabelle 6-18 zeigt die Entwicklung des dokumentierten Wissens im Zeitverlauf. Man sieht eine signifikante Abnahme des neu hinzugekommenen Wissens vom ersten auf das zweite Jahr ( $\alpha=0,001$ ,  $t=0,00000$ ). Vom zweiten auf das dritte Jahr ist die Tendenz nicht mehr eindeutig. Bei 49 Agenten ist keine signifikante Abnahme des dokumentierten Wissens vom zweiten auf das dritte Jahr zu beobachten. Bei 77 und 101 Agenten ist die Abnahme geringer als vom ersten auf das zweite Jahr, jedoch noch signifikant ( $\alpha=0,05$ ,  $t=0,0372$ , bzw.  $0,04691$ ) zu beobachten. Insgesamt ist die Tendenz jedoch sinkend und die Hypothese muss daher angenommen werden.

Tabelle 6-18: Entwicklung des dokumentierten Wissens bei Wissensdokumentation im Zeitverlauf

Agenten	Intranetskill Jahr 1	Delta zu J0	Intranetskill Jahr 2	Delta zu J1	Intranetskill Jahr 3	Delta zu J2
49	2,887	0,387	3,076	0,190	3,301	0,224
77	2,922	0,422	3,207	0,285	3,439	0,231
101	2,929	0,429	3,157	0,228	3,335	0,178

**Hypothese 2.6:** Die Wissensweitergabe durch dokumentiertes Wissen ist höher als in Experiment 2.1.

Die Hypothese muss angenommen werden. Die Wissensweitergabe über dokumentiertes Wissen erhöht sich sogar stärker als vermutet und die Steigerung ist signifikant ( $\alpha=0,001$ ,  $t=0,00000$ , siehe Tabelle 6-15).

**Hypothese 2.7:** Die Wissensweitergabe durch dokumentiertes Wissen ist geringer als in Experiment 2.2.

Die Hypothese muss abgelehnt werden. Die Wissensweitergabe über dokumentiertes Wissen erhöht sich bei Projektreviews wesentlich stärker als bei Wissensdokumentation ( $\alpha=0,001$ ,  $t=0,00000$ , siehe Tabelle 6-15).

**Hypothese 2.8:** Die Wissensweitergabe durch Lernen von Kollegen ist höher.

Die Hypothese muss angenommen werden. Gelernt werden dabei insbesondere jene Wissensgebiete, die vorher noch weniger stark ausgeprägt waren ( $\alpha=0,001$ ,  $t=0,00000$ , siehe Tabelle 6-15).

**Hypothese 2.9:** Die Quantität sozialer Kontakte zwischen Mitarbeitern ist durch die Projektnachbesprechungen höher.

Die in Tabelle 6-17 abzulesende Steigerung ist für 49 Agenten nicht signifikant, für 77 und 101 Agenten jedoch signifikant ( $\alpha=0,01$ ,  $t=0,00333$  bzw.  $0,00689$ ). Dies kann erklärt werden, da bei 49 Agenten das Niveau der Kontakthäufigkeit schon sehr hoch ist, so dass sozusagen „jeder jeden kennt“ und aufgrund des vergleichsweise kleinen Gebäudes auch regelmäßig trifft. Im Vergleich dazu sind die Projektnachbesprechungen kein ausschlaggebender Faktor.

**Hypothese 2.10:** Es wird weniger vergessen, da in den Projektnachbesprechungen viele Wissensbereiche genutzt werden.

Die Hypothese muss angenommen werden (siehe Tabelle 6-15). Es wird signifikant weniger vergessen ( $\alpha=0,001$ ,  $t=0,00000$  bis  $0,00028$ ).

**Hypothese 2.11:** Je mehr Mitarbeiter das Unternehmen hat, umso mehr Wissen steht im Intranet zur Verfügung.

Diese Hypothese kann nicht bestätigt werden. Die Menge des dokumentierten Wissens steigt zwar den Zahlen nach geringfügig von 3,099 über 3,162 auf 3,191. Die Steigerung ist jedoch nur für den Übergang von 49 auf 77 Agenten signifikant ( $\alpha=0,01$ ), nicht jedoch für die Steigerung von 77 auf 101 Agenten (siehe Tabelle 6-19). Eine Erklärung dafür ist, dass der durchschnittliche Skillwert bei 101 Agenten nur bei 2,87 liegt, so dass in vielen Fällen durch die Dokumentation der Lernerfahrungen der im Intranet verfügbare Skillwert nicht steigt.

*Tabelle 6-19: Durchschnittlicher Skillwert des dokumentierten Wissen bei Projektreviews*

Agenten	Intranetskill	$\sigma$	t
49	3,099	0,019	
77	3,162	0,025	0,00418
101	3,191	0,015	0,09019

## 6.7 Diskussion der Ergebnisse

Im Folgenden werden die beschriebenen Ergebnisse reflektiert und die Erfahrungen bei der Entwicklung der Simulation mit einbezogen.

### 6.7.1 Erfahrungen mit der Entwicklung der Simulation

In der Retrospektive war die Entwicklung des eigenen Simulationssystems eine wertvolle Erfahrung, die nicht nur persönlich bereichernd war, sondern auch aus wissenschaftlicher Sicht als sinnvoll eingestuft werden kann. Diese Einschätzung wird im Folgenden anhand der Aspekte Agentenarchitektur, technische und methodische Defizite sowie inhaltliche Erfahrungen dargestellt.

### 6.7.2 Agentenarchitektur zur Beherrschung der Komplexität

Die Auswahl und Implementierung einer Agentenarchitektur führen zu Einsichten, die anders kaum gewonnen werden können. Zu Beginn wurde noch – etwas blauäugig – angenommen, dass auch ohne Architektur aus dem KI-Lehrbuch ein gut funktionierendes Agentensystem entwickelt werden kann. Dies ist im Nachhinein auch richtig, solange das System sehr überschaubar ist. Es hat sich jedoch gezeigt, dass die Komplexität schon bald so stark zunimmt, dass der Zusatzaufwand für die Implementierung einer Agentenarchitektur sich schnell bezahlt macht.

Ähnlich wie in herkömmlichen Systemen eine Schichtenarchitektur ab einer gewissen Komplexität sinnvoll ist, so ist auch bei Agentensystemen die Architektur ab einer gewissen Komplexität nötig, um den Überblick zu behalten und Fehlermöglichkeiten zu

begrenzen. Ähnlich wie mit der Schichtenarchitektur an sich, hat sich im Laufe der evolutionären Entwicklung des Systems auch mit einzelnen Details der Architektur gezeigt, dass sie zwar zu Beginn als unnötiger Overhead erscheinen, sich aber ab einem gewissen Stand der Entwicklung dann doch als sinnvoll und nötig erweisen. So wurde erst sehr spät erkannt, dass die Konzeption und Implementierung von Plänen durchaus lohnenswert ist und eine wichtige Aggregation von Aktionen darstellt, welche die Komplexität in den Planungskomponenten erst beherrschbar macht.

Es muss jedoch erwähnt werden, dass ein evolutionäres Vorgehen trotzdem richtig erscheint, da bei der Versuch von Beginn an die volle Komplexität zu verstehen und im Entwurf des Systems zu berücksichtigen die kognitiven Fähigkeiten der meisten Entwickler ohne Erfahrung mit MAS übersteigt (siehe dazu auch Balzert 2001, insb. Abschnitt 20.12 zum Spiralmodell). Eine Entwicklung der gesamten Basisklassen und -mechanismen für das MAS, ohne den Anwendungsfall mit zu entwickeln und schon frühzeitig sichtbare Ergebnisse erzielen zu können ist zwar möglich und beherrschbar, jedoch auch sehr trocken und damit aus Motivationsaspekten nachteilig.

### **6.7.3 Technische und methodische Defizite**

Durch die Implementierung von low-level Funktionen wie Persistierung von Daten in der Datenbank, Sammeln der Daten für die statistischen Auswertungen und Entwerfen der grafischen Oberflächen musste viel Zeit verwendet werden, die für die Entwicklung von inhaltlichen Aspekten der Simulation besser hätte investiert werden können. Insbesondere die Aufwände für die GUI-Entwicklung erscheinen unnötig hoch, da dies mit anderen Programmiersprachen und v.a. Entwicklungsumgebungen sehr einfach und schnell grafisch gelöst werden kann. Für Java sind aber immer noch keine gut funktionierenden GUI-Builder verfügbar. Im Laufe der Entwicklungszeit wurden sowohl der in NetBeans integrierte als auch diverse Plug-Ins für Eclipse getestet, anschließend jedoch immer auf die manuelle Implementierung zurückgegriffen. Aus dieser Perspektive betrachtet scheint es trotz der geschilderten Erfahrungen vorteilhaft zu sein, auf eine der getesteten Agentenframeworks zurückzugreifen, auch angesichts der Tatsache, dass diese permanent weiter entwickelt werden und einige der angesprochenen Defizite schon bald behoben sein können.

Weiterhin muss festgestellt werden, dass es aus Softwareentwicklungssicht sowohl bzgl. der Methodik, als noch mehr auf die Softwareunterstützung bezogen Defizite gibt. Es wurden einige der bekannteren Entwicklungsmethoden ausprobiert (z.B. Gaia, MASSIVE,

siehe Abschnitt 5.1), aber keine brachte Vorteile, die über eine Checklistenfunktion zu berücksichtigender Aspekte und die Dokumentation der Entscheidungen hinaus geht. Auch hier würde eine Softwareunterstützung vielleicht weiterhelfen, so wie bei der UML- und Datenmodellierung durch die Verfügbarkeit guter Werkzeuge mit Forward-Engineering, also Code-, bzw. Datenbankgenerierung aus den Modellen die Akzeptanz von Modellierungsmethoden noch einmal gestiegen ist. Doch davon scheint die agentenorientierte Softwareentwicklung noch Jahre entfernt.

Der nächste Aspekt betrifft das Debugging von nebenläufigen, rundenbasierten Anwendungen. Nachdem Nebenläufigkeit die Komplexität erhöht und dadurch zusätzliche Fehler provoziert, ist es besonders wichtig gute Hilfestellungen bei der Fehlersuche zu haben. Dies ist bei klassischen Entwicklungsumgebungen derzeit nicht der Fall. Der Entwickler benötigt die Möglichkeit zum Setzen von bedingten Haltepunkten, die trotz des abwechselnden Ausführens vieler Agenten des gleichen Typs und der Zerstückelung semantisch zusammenhängender Programmteilen in mehrere Methoden, eine systematische Fehlersuche innerhalb eines Agenten zulässt. Eclipse bietet zwar die Möglichkeit zur Filterung von Unterbrechungen nach dem Thread, jedoch muss man dazu erst die abstrakte Threadnummer dem gewünschten Agenten zuordnen können. Eine bessere Variante erscheint die direkte Angabe einer Bedingung der Form `id=<ID des gewünschten Agenten>`. Auch dies ermöglicht Eclipse, allerdings gab es dabei auch in der aktuellen Eclipse Version (3.2.1, Stand Dezember 2006) häufig Ausnahme-Fehler<sup>75</sup>, so dass die Funktion kaum benutzbar war. Auch die guten Logging-Funktionen des `java.util.logging` Pakets helfen in diesem Fall nur bedingt weiter, da es aufwändig ist in einer umfangreichen Logdatei die interessanten Passagen herauszufiltern. Unit-Tests sind zwar eine Hilfe, um Basisfehler zu vermeiden, sind jedoch wie der Name schon sagt für Integrationstests nicht geeignet, welche bei Multiagentensystemen einen noch höheren Stellenwert einnehmen als bei nicht-agentenorientierten Systemen. Zudem treten Fehler oft erst im Zeitverlauf auf, wenn sich bestimmte Konstellationen von Zuständen ergeben, deren Möglichkeit und Problematik nicht vorhergesehen wurden.

### 6.7.4 Inhaltliche Erfahrungen

Grundsätzlich erscheint die Multiagentensimulation auch rückblickend ein geeignetes Mittel zur Untersuchung der Wissensweitergabe – trotz der geschilderten technischen

---

<sup>75</sup> Conditional breakpoint encountered runtime exception. Reason: com.sun.jdi.ObjectCollectionException

Probleme. Es werden auf der Mikroebene viele Aspekte des Phänomens deutlich, die bei anderen Untersuchungen nur am Rande auftauchen oder unbeachtet bleiben, z.T. vielleicht weil sie selbstverständlich erscheinen. So wurde z.B. das Metawissen über die Fähigkeiten der Kollegen als eine wesentliche Voraussetzung für erfolgreiche Wissensweitergabe erkannt. Dies an sich ist noch nicht weiter überraschend und wird in vielen Studien wiedergegeben. Dem Autor sind jedoch aus der Literatur keine Untersuchungen bekannt, die systematisch die Arbeitssituationen analysiert, in denen Metawissen erworben wird. Es wird zwar oft die Bedeutung der viel zitierten Kaffeeküchen betont, es muss jedoch über diese sozialen Gelegenheiten hinaus noch weitere geben, die zu Metawissen führen, da sonst in der Praxis genau wie in der Simulation das Metawissen auf einem sehr geringen Niveau bleiben würde. Von der mangelnden Kenntnis dieser Situationen abgesehen ist es auch schwierig diese in der Simulation zu implementieren, da in der Praxis auch nicht die echten Wissensstände der Kollegen erkannt werden, sondern aus den Gesprächen, Zuschauen beim Verrichten einer Tätigkeit und gemeinsamer Arbeit an einem Projekt auf den Wissensstand geschlossen wird. Dieser Schlussfolgerungsprozess wird aber sowohl vom eigenen Wissensstand, als auch von Persönlichkeitsmerkmalen des Mitarbeiters (z.B. überheblich, oder wenig selbstbewusst) beeinflusst. Auch die übrigen Fähigkeiten des Kollegen (z.B. Rhetorik) können zu einer Abweichung von tatsächlichem und geschlussfolgertem Wissensstand führen (siehe Abschnitt 3.3.3). Die Schlussfolgerungsmechanismen, welche in der KI diskutiert werden, beruhen jedoch auf einer strengen Logik und Fakten. Zwar werden mittlerweile auch Folgerungen auf Grundlage von unsicheren Annahmen unterstützt (z.B. Benferhat et al. 1993), jedoch wird, soweit dem Autor bekannt ist, noch keine Selbstreflexion zur Berücksichtigung der Verhältnisse zwischen externer Beobachtung und eigenen Zuständen in der Schlussfolgerungskomponente berücksichtigt, so dass eine so unscharfe Abbildung, wie sie in der Praxis auftritt, kaum simuliert werden kann.

### **6.7.5 Limitationen des Modells und mögliche Weiterentwicklungen**

Im Folgenden werden einige der Einschränkungen vorgestellt, die das Simulationssystem besitzt, sowie Ideen für mögliche Weiterentwicklungen präsentiert, welche die Simulation realistischer gestalten.

Die Designentscheidung, Wissen als kontinuierliche Variable zu modellieren, hat sich zwar nicht direkt als Fehlentscheidung erwiesen, aber es entsteht doch der Eindruck, dass eine Modellierung als diskrete Variable auch Vorteile gehabt hätte. Insbesondere das aus der

Praxis bekannte Phänomen, dass einem nur eine kleine Information fehlt, um die aktuelle Aufgabe abzuschließen, könnte durch das Fehlen eines einzelnen Wissensbausteins sehr gut abgebildet werden. Das würde es zulassen, dass auch Kompetente oder Erfahrene noch Wissenslücken in den „unteren“ Bereichen haben könnten, die sie zum Nachfragen bei Kollegen zwingt. Ein Bitstring, in dem gesetzte Bits vorhandene Wissensbausteine simulieren und nicht gesetzte Bits für Wissenslücken stehen, erscheint gut zur Abbildung geeignet. Er könnte aus 5 Abschnitten á 32 Bit bestehen (Moss 2000, S. 252; Carley 2002a, S. 212; Carley, Schreiber 2004, S. 157). Bei der Weitergabe des Wissens würde der Abschnitt des benötigten Wissens weitergegeben, also einmal 8 Bit. Diese könnten mit dem darunter liegenden Abschnitt und einen Abschnitt auf gleicher Höhe eines verwandten Wissensgebietes kontextualisiert werden. Bei der Aufnahme der kontextualisierten Informationen entscheidet die Übereinstimmung, wie gut von der Quelle gelernt werden kann. Man könnte trotzdem einen Dezimalwert errechnen, der als Skillwert gilt. Dieser wäre nicht die direkte Dezimalrepräsentation der Binärzahl, sondern könnte nach Anzahl gesetzter Bits und dessen grober Position auf den Wertebereich [0; 5,0] abgebildet werden. Diese diskrete Modellierung des Wissens würde sich auch anbieten, wenn neben den projektübergreifenden Wissensgebieten auch Wissen simuliert wird, das im Projekt selbst generiert wird und somit projektspezifisch ist. So wird z.B. das Wissen über die Softwarearchitektur der Lösung, die in einem Arbeitspaket erarbeitet werden könnte, auch in anderen Arbeitspaketen benötigt. Ein weiteres Beispiel ist das Datenmodell, welches auch von den Programmierern benötigt wird, um den Datenzugriff und die –speicherung zu realisieren. Dadurch wird auch eine gewisse Abhängigkeit zwischen den Arbeitspaketen generiert, die zu mehr sozialen Kontakten und potentiell auch mehr Wissensweitergabe führt. Solche Abhängigkeiten werden z.B. bei (Kunz et al. 1998) modelliert. Es ergeben sich aber auch neue Schwierigkeiten bei der Modellierung, weil die Abstimmung zwischen den Mitarbeitern abgebildet werden muss. Auch die Kalibrierung wird erschwert, weil zusätzliche Zeit durch Abstimmungen zwischen den Mitarbeitern verbraucht wird und Wartezeiten auf Ergebnisse aus anderen Arbeitspaketen entstehen. Dadurch wird die Auslastung der Mitarbeiter sinken, wenn nicht mehrere Arbeitspakete gleichzeitig bearbeitet werden können. Dies ist für die Programmierung und die Tests sehr komplex, würde die Realitätsnähe aber vermutlich erhöhen. Es stellt sich außerdem die Frage, ob dieses projektspezifische Wissen zusammen mit dem projektunabhängigen Fachwissen in die Wissensbilanz eingehen darf, oder getrennt ausgewiesen werden muss.

Wissen entwickelt sich innerhalb eines Wissensgebiets während der Simulationszeit nicht weiter. Das führt dazu, dass weder neues Wissen hinzukommt, was von den Mitarbeitern auch mit hohem Skillwert neu gelernt werden muss, noch bestehendes Wissen obsolet wird. Dies ist jedoch gerade in der Softwareentwicklung eine entscheidende Einschränkung, da dort relativ häufig neue Technologien entwickelt werden, mit denen sich die Entwickler auseinandersetzen müssen. Das Fallenlassen dieser Einschränkung lässt sich nur sehr schwer in das Simulationssystem integrieren, wenn daran festgehalten werden soll, dass die Skillwerte der Simulation direkt den Expertisestufen von Dreyfus und Dreyfus entsprechen soll. Andernfalls ließe sich die Weiterentwicklung relativ leicht als im Zeitverlauf steigender Maximalwert implementieren. Dementsprechend müsste auch die durchschnittliche Schwierigkeit der Arbeitspakete steigen, um die steigenden Anforderungen der Kunden nach Integration neuer Technologien widerzuspiegeln. Es scheint, dass auch hier die Abbildung des Wissens als diskrete Variable vorteilhaft ist. In diesem Fall könnte die Weiterentwicklung des Wissens als steigende Anzahl Bits pro Skillstufe implementiert werden. Obsolet gewordenen Wissen kann dadurch simuliert werden, dass die entsprechenden Bausteine für das Abarbeiten neuer Arbeitspaketen nicht mehr benötigt werden. Somit lässt sich die Entsprechung zu den Expertisestufen von Dreyfus und Dreyfus aufrechterhalten, ohne auf die Weiterentwicklung des Wissens zu verzichten.

In der Praxis werden die Skillwerte einmalig zur Einführung des Skillmanagement erhoben und anschließend keine planmäßigen Aktualisierungen mehr vorgenommen, wie das bei itCampus der Fall war, oder aber es ist eine Aktualisierung im Jahresrhythmus im Rahmen der Jahresbeurteilung vorgesehen, wie das nach Kenntnis des Autors bei anderen Unternehmen der Fall ist. Im simulierten Skillmanagementsystem sind dagegen immer aktuelle Werte enthalten, die zudem nicht auf einer Selbsteinschätzung oder Beurteilung durch den Vorgesetzten beruhen, sondern den objektiv richtigen Werten entsprechen. Diese Einschränkung ließe sich zumindest zu dem Teil aufheben, dass nicht auf die aktuellen Werte, sondern auf eine im Jahresrhythmus aktualisierte Kopie zugegriffen wird. Dies würde zwar den Speicherverbrauch erhöhen und eine neue Speicherstruktur erfordern, ist jedoch trotzdem mit überschaubarem Aufwand in das Simulationssystem zu integrieren. Für eine subjektive Einschätzung der Skillwerte trifft wieder die schon oben für Metawissen diskutierte Problematik zu.

Der relationale Kontext mit den dort diskutierten Einflussfaktoren wird im Simulationsmodell derzeit noch nicht berücksichtigt. Da in der derzeitigen vereinfachten Abbildung seltener Gelegenheiten für Wissensweitergabe auftreten als erwartet, sollten die so initiierten Wissensweitergabeaktivitäten nicht noch durch soziale Barrieren behindert werden. Andererseits ist zu erwarten, dass ein konsequentes Einbeziehen sozialer Aspekte wie z.B. Treffen in Pausen und die dort stattfindenden Gespräche über Freizeit und Arbeit auch neue Gelegenheiten zur Wissensweitergabe schaffen und damit andere Barrieren wieder ausgleichen. Dies ließe sich durch eine neue Aktion implementieren, in dem einer von potentiell mehreren „Pausenräumen“ (z.B. Kaffeeküche, Rauchereck, Cafeteria, Kantine) aufgesucht wird und dort eine oder zwei Spielrunden verbracht werden. Allerdings entsteht hier das Problem, dass für die Abbildung von kurzen Pausen (z.B. Kaffeepause von 15 oder 20 Minuten) die Granularität der Simulation mit 15 min. pro Runde noch zu hoch ist. Damit würden schon das Aufsuchen des Pausenraums und das Zurückkehren zum Arbeitsplatz 30 min. dauern. Ein echtes Gespräch (Sprechakt mit mindestens einer Antwort) würde weitere 30 min. in Anspruch nehmen. Mehr als eine Mittagspause lässt sich damit kaum sinnvoll simulieren.

Das Fehlschlagen eines Entwicklungsprojekts in Form vorzeitigen Einstellens der Entwicklungsarbeit, sowie radikal geänderter oder geschrumpfter Anforderungen wird in der Simulation nicht berücksichtigt, obwohl die Literatur von einer Fehlerquote zwischen 33% und 75% ausgeht (Warne 2003, S. 106), abhängig davon wie ein fehlgeschlagenes Projekt definiert wird. Die vorliegende Studie geht davon aus, dass ein Projekt in jedem Fall fertig gestellt werden kann. Der schlechteste Fall ist ein deutliches Überziehen der avisierten Projektlaufzeit. Fehlende Akzeptanz auf Anwenderseite oder geänderte organisatorische Bedingungen, die eine bestellte Software überflüssig machen, werden nicht betrachtet, obwohl dies in der Praxis einen großen Anteil an gescheiterten Projekten ausmacht. Eine erste Annäherung an dieses Phänomen ließe sich in der Simulation dadurch erreichen, dass zufallsgesteuert ein einzelnes Arbeitspaket eines Projekts unabhängig von seinem Fertigstellungsgrad obsolet wird und damit nicht weiter bearbeitet werden muss. Gleichzeitig könnte ein neues Arbeitspaket dazu kommen, das neue Anforderungen der Kunden widerspiegelt.

Eine weitere Modifikation, die Vorteile mit sich bringen könnte, ist das Abweichen von den linearen Funktionen für Lernen und Vergessen. So könnte man z.B. mit einer konkaven Funktion wie der Wurzelfunktion für das Lernen arbeiten (z.B.  $y = 0,01 x^{1/2}$ ) und

eine konvexe Funktion wie die Quadratfunktion für das Vergessen verwenden (z.B.  $y = 0,01 x^2$ ). Auch eine Unterscheidung zwischen Lernen von Wissen, welches man sich ganz neu erarbeiten muss, und Wissen, welches man schon einmal besessen, aber wieder vergessen hat, erscheint plausibel. Eng damit verbunden ist das Gedächtnismodell, welches in der Simulation bisher sehr einfach ist. Eine Unterscheidung in verschiedene Bereiche wie Kurzzeit-, Langzeitarbeits-, und Langzeitgedächtnis mit entsprechend differenzierten Übergangsfunktionen könnte z.B. im Bereich des Vergessens höheren Realismus erzielen.

Weitere Möglichkeiten des Wissenserwerbs, wie z.B. das Besuchen einer Schulung, um in kurzer Zeit eine größere Wissenslücke schließen zu können, wären ebenfalls interessant zu untersuchen. Interne Schulungen für mehrere Mitarbeiter wären dabei leichter zu integrieren als externe, da die Mitarbeiter bei externen Schulungen einige Tage nicht im Unternehmensgebäude verbringen würden. Da das Unternehmensgebäude aber die Simulationsumgebung darstellt und keine räumliche Umgebung außerhalb davon vorgesehen ist, müsste eine größere Erweiterung dafür implementiert werden. Die interne Schulung könnte als Joint Plan umgesetzt werden, ähnlich den Projektnachbesprechungen. Im Unterschied dazu würde aber nicht der Projektmanager, sondern ein externer Schulungsleiter das Treffen koordinieren und sein Wissen weitergeben. Dieser müsste als neuer Agententyp modelliert werden, mit nur einer einzigen Aktion „Schulung abhalten“. In dieser Aktion würden jede Runde Sprechakte gesendet werden, die Wissen enthalten und von den Schulungsteilnehmern rezipiert werden können. Auf diese Art und Weise kann sehr feingranular auch eine Halbtageschulung simuliert werden.

Wie im Abschnitt 6.2 unter dem Punkt Sensitivitätsanalyse bereits diskutiert wurde, existieren durchaus Abhängigkeiten zwischen den Wissensgebieten. Neues Wissen kann immer dann schnell in die Wissensbasis integriert werden, wenn es genügend Anknüpfungspunkte an bestehendes Wissen gibt (Reinhardt 2002). Diese Anknüpfungspunkte können innerhalb desselben Wissensgebiets bestehen, oder auch in benachbarten. Im vorliegenden Simulationsmodell wurden solche „Nachbarschaften“ oder „Verwandschaftsverhältnisse“ zwischen einzelnen Wissensgebieten bisher nicht berücksichtigt. Auch unterschiedliche Lerngeschwindigkeiten abhängig von der Wissensquelle sind bisher nicht integriert worden, da dafür keine empirischen Daten als Beleg gefunden wurden.

## **7 Zusammenfassung und Ausblick**

Im Folgenden werden die im Laufe der Arbeit gesammelten Erfahrungen zusammengefasst, ein Ausblick auf mögliche Weiterentwicklungen des Simulationssystems gegeben und kurz diskutiert, inwiefern die Ergebnisse auf die betriebliche Praxis übertragbar sind.

### **7.1 Zusammenfassung**

Die vorliegende Arbeit leistet mehrere Beiträge, von denen die Ergebnisse der Simulationsstudien nur einen Teil darstellen.

Im Kapitel zwei wurden in der Literatur diskutierte Wissensprozesse gesammelt und in der gebotenen Kürze eine Erläuterung zu jedem Prozess gegeben. Der Wissensprozess Wissensweitergabe wurde genauer analysiert und die unterschiedlichen Foki der identifizierten Synonyme beschrieben. Aufbauend auf bestehenden Definitionen wurde eine umfassende Prozessdefinition der Wissensweitergabe erarbeitet und zu einem Modell ausgebaut, das die relevanten Kontextbereiche für Einflussfaktoren aufzeigt (Abschnitt 2.2.4). Die Kontextbereiche wurden gegeneinander abgegrenzt und damit die Grundlage für die Literaturanalyse in Kapitel drei gelegt. Schließlich wurden WM-Instrumente zusammengetragen, die Organisationen für Interventionen zur Verfügung stehen und Ansätze vorgestellt, wie der Erfolg solcher Interventionen gemessen werden kann. Sowohl bei den WM-Instrumenten als auch bei der Erfolgsmessung wurden eigene Vorschläge erarbeitet.

Im Kapitel drei wurden aus empirischen Studien und vorhandenen Literaturanalysen Einflussfaktoren für die Wissensweitergabe gesammelt, verdichtet und den Kontextbereichen des Wissensweitergabemodells zugeordnet. In jedem der Kontextbereiche wurden empirische Ergebnisse, Konzepte und Theorien zur Erklärung der Einflussfaktoren herangezogen. Dabei wurden in jedem Kontextbereich einzelne Theorien ausgewählt, die kompatibel zu den Theorien anderer Kontextbereiche sind und sich somit zu einem Domänenmodell der Wissensweitergabe verbinden lassen. Im Beziehungskontext wurde die Relation Model Theory (Fiske 1991) verwendet, um Wissensweitergabe als soziale Austauschbeziehung zu erklären. Im Quell- und Empfängerkontext wurden mehrere Entscheidungstheorien über die Aktivitätentheorie integriert und als unterschiedliche Aspekte des Entscheidens dargestellt, die in verschiedenen Entscheidungssituationen auftreten. Der „Prozess Gedächtnis“ mit den Prozessschritten Wahrnehmen, Lernen,

Wissen im Gedächtnis Repräsentieren, Erinnern und Explizieren wurde in Abhängigkeit von der Art des Wissens erläutert und der Fokus auf explizierbares Wissen im Softwareentwicklungsprozess begründet. Daraus ergibt sich eine Präferenz für symbolische Wissensrepräsentation gegenüber neueren Ansätzen der Situated Cognition. Bzgl. des Lernens wurde herausgearbeitet, dass ähnlich den Entscheidungstheorien keine einheitliche Theorie alle Aspekte erklären kann sondern differenziert nach der Expertisestufe des Lernenden unterschiedliche Theorien bessere Erklärungen liefern. Der Wissenskontext liefert demnach keinen eigenen Erklärungsbeitrag, sondern dient zur Einordnung der Situation der Quelle und des Empfängers, die determiniert, welche Erklärungsansätze aus den anderen Kontextbereichen herangezogen werden müssen. Dasselbe gilt für den Organisationskontext, der mit der Hypertextorganisation (Nonaka 1994) mit ihrer Einteilung in Primär-, Sekundär- und Tertiärorganisation und dem Softwareentwicklungsprozess gut beschrieben werden kann. Im Transferkontext kann aufbauend auf der Sprechakttheorie (Searle 1969) die Verbindung zum Beziehungskontext hergestellt werden, indem hierarchisch eine Ansammlung von Sprechakten als Episode und mehrere Episoden zusammen als soziale Begegnung aufgefasst werden (Fincham, Rhodes 2003). Weiterhin ist im Transferkontext die Medienunterstützung angesiedelt, deren Eignung über die Media Richness Theory (Daft, Lengel 1986) und darauf aufbauende Theorien erklärt werden kann. Die Wahl des geeignetsten Mediums für die Kommunikation ist eine der großen Herausforderungen in modernen Organisationen, da eine Vielzahl von Medien mit spezifischen Vor- und Nachteilen zur Auswahl steht. Im Situations- und Nutzungskontext schließlich konnten keine geeigneten Theorien gefunden werden, so dass auf einzelne empirische Ergebnisse zurückgegriffen werden musste. Wichtige Ergebnisse sind die Einteilung in formelle, organisatorisch verankerte und informelle, soziale Gelegenheiten zur Wissensweitergabe sowie in Push- und Pull-Gelegenheiten.

In Kapitel vier wurden zunächst die Grundlagen der Computersimulation und der Agententechnologie dargestellt. Anschließend wurden bestehende Agentenarchitekturen analysiert und verglichen, um eine geeignete Architektur für die Simulation von Menschen bei wissensintensiven Tätigkeiten zu ermitteln. Da sich keine der untersuchten Architekturen alleine als geeignet erwiesen hat, wurde eine eigene Agentenarchitektur entwickelt, die auf den Architekturen InterRaP (Müller, Pischel 1993) und PECS (Urban 2000) aufbaut, diese kombiniert und konkretisiert. Um die Architektur weiter zu verfeinern, wurde eine große Anzahl verwandter Arbeiten identifiziert und analysiert. Ausgewählte Aspekte davon

wurden in das Konzeptmodell übernommen, so dass ein operationalisiertes Domänenmodell entstanden ist, welches auf Konzepten der Multiagentensimulation aufbaut. Dieses Konzeptmodell ist für eine Implementierung allerdings zu umfangreich, da es immer noch eine Vielzahl an Aspekten berücksichtigt und somit schwer validier- und überschaubar wird.

Daher wurde in Kapitel fünf eine weitere Komplexitätsreduktion vorgenommen. Dazu wurde auf die Forschungsergebnisse aus der agentenorientierten Softwareentwicklung aufgebaut und dem Vorgehensmodell MASSIVE folgend iterativ zusätzliche Funktionen in ein Minimalmodell integriert, bis eine stabile Simulation entstanden ist, die alle benötigten Aspekte des Untersuchungsgegenstands enthält. Als Grundlage für die Auswahl der Simulations- bzw. Programmiersprache wurde ein Vergleich von Simulationsframeworks durchgeführt. Die Simulation wurde in Java implementiert und für die Auswertungen Schnittstellen zum relationalen Datenbankmanagementsystem HSQLDB, zum Tabellenkalkulationsprogramm Excel und zur sozialen Netzwerkanalysesoftware AGNA geschaffen. Um den besonderen Herausforderungen im Zusammenhang mit Multiagentensimulationen zu begegnen, wurden einige neuere Konzepte von Java verwendet und innovative Lösungen für Aktionsauswahl und Agentenaktionen entwickelt.

Im sechsten Kapitel wurde die Parametrisierung für die Simulation aus empirischen Studien abgeleitet, wobei eine Studie bei einem mittelständischen IT-Dienstleister selbst durchgeführt wurde. Zur Kalibrierung wurden Tests mit verschiedenen Ausprägungen der Parameter durchgeführt und plausible Werte ermittelt, die zu stabilen Simulationsergebnissen führen. In Sensitivitätsanalysen wurde die Robustheit des Simulationssystems verifiziert. Mit Unit-Tests wurde die Korrektheit des Simulationsprogramms sichergestellt und durch den Vergleich mit empirischen Ergebnissen wurde eine Teilvalidierung des Modells erreicht. Schließlich wurden Simulationsstudien durchgeführt, um die Wirkung des Einsatzes eines Skillmanagementsystems und von Wissensdokumentationsmaßnahmen zu überprüfen. Dabei wurde gezeigt, dass Multiagentensimulation ein probates Mittel zur Simulation von Wissensweitergabe in Organisationen ist.

Die Ergebnisse der Simulationsexperimente dienen aus Sicht des Autors in erster Linie dazu, die grundsätzliche Eignung des Simulationssystems zu demonstrieren. Um fundamental neue Erkenntnisse über die Wissensweitergabe zu generieren, müssten die Experimente wesentlich stärker mit empirischen Daten unterlegt und v.a. die erzielten

Ergebnisse in der Empirie validiert werden. Dies war im Rahmen der vorliegenden Arbeit nicht möglich, v.a. weil nicht auf ein bestehendes Simulationssystem, welches für die eigenen Studien nur noch entsprechend parametrisiert werden müsste, zurückgegriffen werden konnte sondern ein eigenes Simulationssystem von Grund auf neu entwickelt werden musste.

Trotz dieser Schwierigkeiten sind einige der Ergebnisse bemerkenswert. Für Skillmanagementsysteme hat sich ergeben, dass deren Einführung nicht automatisch zu positiven Ergebnissen führt, sondern der Erfolg stark von der Vergabestrategie der Arbeitspakete abhängt. Werden die Arbeitspakete in zufälliger Reihenfolge an den verfügbaren Mitarbeiter vergeben, bei dem der verfügbare Skill am wenigsten von dem erforderlichen abweicht, so steigt zwar der Unternehmenserfolg nicht signifikant, aber sowohl von Kollegen als auch durch Ausprobieren während der Projektarbeit wird signifikant mehr gelernt.

Durch Dokumentation von Lernerfahrungen steigen Unternehmenserfolg und auch das gesamte Wissen aller Mitarbeiter signifikant. Interessant dabei ist insbesondere, dass Dokumentation auf persönlicher Ebene dazu führt, dass Mitarbeiter eher zu Spezialisten werden, also ein einzelnes Wissensgebiet vertiefen, während bei Dokumentation auf Teamebene das Generalistentum gefördert wird. Dies rührt daher, dass nicht in erster Linie aus dem dokumentierten Wissen, sondern bei der Projektnachbesprechung selbst gelernt wird, was zu breiterem Wissen und geringerem Vergessen führt, da viele Skills aktiviert werden.

## **7.2 Ausblick**

In der Arbeit wurde auch weiterer Forschungsbedarf identifiziert. So wurden z.B. bei der Diskussion der WM-Instrumente einige Defizite angesprochen (siehe Abschnitt 2.4). Insbesondere die uneinheitliche Granularität von in der Literatur beschriebenen Instrumenten, sowie das Fehlen von Instrumenten für die Unterstützung von Wissen aus strukturierten Daten, organisationsexternem Wissen und den Wissensprozess des Beobachtens sind aufgefallen. Es wurden einige eigene Vorschläge für WM-Instrumente gemacht (siehe Abschnitt 2.4.3), die aber nicht als abschließend anzusehen sind und aus Sicht des Autors sowohl um weitere Instrumente ergänzt als auch inhaltlich noch genauer definiert und überprüft werden müssen.

Die Ergebnisse der Simulationsexperimente zeigen, dass auch im Bereich Wissensweitergabe noch Forschungsbedarf vorhanden ist. Insbesondere Gelegenheiten für Wissensweitergabe und Mechanismen zum Lernen von Meta-Wissen sind noch unzureichend untersucht worden. Die in der Simulation beobachteten Phänomene, dass Skillmanagement auch in größeren Organisationen nur unter bestimmten Bedingungen signifikante Vorteile bietet und dokumenten-orientierte Maßnahmen effektiver sind, kann nicht ohne weiteres auf die betriebliche Realität übertragen werden. So liegt der Simulation die Vereinfachung zu Grunde, dass von dokumentiertem Wissen genauso gut gelernt werden kann, wie direkt von einer menschlichen Wissensquelle. Ebenso werden unterschiedliche Lernertypen nicht berücksichtigt. Dazu müssten detaillierte empirische Überprüfungen vorgenommen werden, um auf Grundlage empirischer Daten die Simulation mit verfeinerten Parametern neu laufen zu lassen.

Eine mögliche Weiterentwicklung ist die Modellierung von Abhängigkeiten zwischen Arbeitspaketen. Dies würde zu mehr sozialen Kontakten zwischen den Projektmitarbeitern und potentiell auch zu mehr Wissensweitergabe führen. Die dafür nötige Abstimmung zwischen den Mitarbeitern wäre aber schwierig zu implementieren und die Mitarbeiter müssten möglichst oft an mehreren Arbeitspaketen arbeiten, um die Wartezeiten ausgleichen zu können, die durch Abhängigkeiten zwangsläufig entstehen.

Eine weitere Verbesserung könnte die Modellierung von projekt-, kunden- oder branchenspezifischem Wissen bringen. Dadurch würde die Notwendigkeit zur Wissensweitergabe erhöht. Auch die Weiterentwicklung von Wissen eines Wissensgebiets wäre insbesondere in der Domäne Softwareentwicklung eine wünschenswerte Funktion der Simulation. Dies könnte am besten dadurch abgebildet werden, dass Wissen nicht als kontinuierliche sondern als diskrete Variable abgebildet wird. Somit könnten zufallsverteilt einzelne Wissensbausteine als obsolet gekennzeichnet werden, während auf jeder Skillstufe neue Bausteine hinzukommen könnten. Alternativ dazu könnte auch der maximale Skillwert im Laufe der Zeit steigen, was steigende Anforderungen in den Arbeitspaketen bedingt.

Eine relativ leicht zu implementierende Weiterentwicklung wäre das Skillmanagementsystem von stets aktuellen Werten auf periodisch aktualisierte Werte (z.B. jährlich oder halbjährlich) umzustellen. Würde das Aktualisieren selbst als Agentenaktion abgebildet, so könnte untersucht werden, welches Aktualisierungsintervall die besten Ergebnisse erzielt.

Eine weitere sinnvolle Erweiterung wäre das Einbeziehen von Fehlern bei der Codierung. Diese könnten dann in einer Qualitätssicherungsaktion entdeckt werden und müssten

nachgebessert werden, bevor das Projekt vollständig abgeschlossen ist. Dies hätte zwar im ersten Schritt keinen direkten Einfluss auf die Wissensweitergabe, würde jedoch eine realistischere Bewertung des Unternehmenserfolgs erlauben, da nicht durch langwieriges Ausprobieren ein qualitativ gleichwertiges Projektergebnisse erzielt werden könnte wie durch Einsatz eines erfahrenen Entwicklers mit hohem Skillwert.

### **7.3 Fazit**

Reflektiert man die zu Beginn der Arbeit definierten Ziele, so können diese als erreicht eingestuft werden. Multiagentensimulation als Forschungsansatz hat sich in der Arbeit bewährt und eignet sich nach Einschätzung des Autors sehr gut, um die definierten Fragestellungen zu untersuchen. Es wurde wie oben beschrieben ein umfassendes Domänenmodell entwickelt und aufgrund notwendiger Komplexitätsreduktion weiter abstrahiert und detailliert zu einem handhabbaren Konzeptmodell. Dieses wurde in ein funktionsfähiges und valides Simulationssystem überführt, welches von Grund auf neu entwickelt wurde, da kein geeignetes System vorhanden war. Es wurden zwei Simulationsstudien mit je drei Experimenten definiert, die personenorientierte und dokumentenorientierte WM-Maßnahmen abbilden. Die Ergebnisse zeigen, dass ein Skillmanagementsystem nicht zu mehr Wissensweitergabe, höherem kumuliertem Wissen oder gar höheren Unternehmenserträgen führt, wogegen beide untersuchten Formen der Dokumentation von Lernerfahrungen sowohl zu mehr kumuliertem Wissen als auch zu signifikant steigenden Unternehmenserträgen führen. Unterlegt man die Simulationsexperimente noch stärker mit empirischen Daten eines Unternehmens, so können mit dem entwickelten Simulationssystem nach Einschätzung des Autors Ergebnisse erzielt werden, die diesem Unternehmen ex-ante konkrete Anhaltspunkte für die Effektivität einer einzuführenden WM-Maßnahme liefern können.

## Literatur

- Abecker, A., Decker, S., Wolf, T. (1999): Unterstützung des Wissensmanagements durch Informations- und Kommunikationstechnologie, Electronic Business Engineering - 4. Internationale Tagung Wirtschaftsinformatik Saarbrücken.
- Abraham, M. (2001): Rational Choice-Theorie und Organisationsanalyse, Tagung der Arbeitsgruppe "Organisationssoziologie", Universität Bielefeld.
- Acuña, S. T., Juristo, N. (2003): Modelling Human Competencies in the Software Process, 4th International Workshop on Software Process Simulation and Modeling (ProSim 2003), Portland, OR.
- Alavi, M., Leidner, D. E. (2001): Review: Knowledge Management and Knowledge Management Systems - Conceptual Foundations and Research Issues., in: Management Information Systems Quarterly - MISQ, 25(1), S. 107-136.
- Albrecht, P., Schweizerische Vereinigung für Datenverarbeitung (2000): Berufe der Informatik - mit Funktionen in den Bereichen Wirtschaftsinformatik, technische Informatik, Organisation, 5., stark überarb. und erw. Aufl Auflage, vdf, Zürich.
- Allweyer, T. (1998a): Knowledge Process Redesign - Modellierung und Gestaltung der Wissensverarbeitung im Unternehmen, IAO-Forum Innovatives Wissensmanagement.
- Allweyer, T. (1998b): Modellbasiertes Wissensmanagement, in: Information Management, 13(1), S. 37-45.
- Alonso, G., Casati, F., Kuno, H., Machiraju, V. (2004): Web Services - Concepts, Architectures and Applications, Springer, Berlin.
- Althoff, K.-D., Bomarius, F., Tautz, C. (2000): Knowledge Management for Building Learning Software Organizations, in: Information Systems Frontiers, 2(3/4), S. 349-367.
- Anderson, J. R. (1976): Language, Memory and Thought, Hillsdale, NJ.
- Anderson, J. R. (1977): Memory for information about individuals, in: Memory & Cognition, 5(4), S. 430-442.
- Anderson, J. R. (1987): Methodologies for studying human knowledge, in: Behavioral and brain sciences, 10, S. 467-505.
- Anderson, J. R. (1996): ACT - A simple theory of complex cognition, in: American Psychologist, 51, S. 355-365.
- Anderson, J. R., Bothell, D., Byrne, M. D., Douglas, S., Lebiere, C., Quin, Y. (2004): An Integrated Theory of Mind, in: Psychological Review, 111(4), S. 1036-1060.
- Anderson, J. R., Graf, R., Grabowski, J. (2001): Kognitive Psychologie, 3. Aufl Auflage, Spektrum, Heidelberg [u.a.].
- Anderson, J. R., Matessa, M., Lebiere, C. (1997): ACT-R - A Theory of Higher Level Cognition and its Relation to Visual Attention, in: Human-Computer Interaction, 12(4), S. 439-462.
- Anderson, J. R., Schunn, C. D. (2000): Implications of the ACT-R Learning Theory - No Magic Bullets, in: R. Glaser (Hrsg.): Advances in instructional psychology, Erlbaum, Mahwah, NJ.
- Angermeier, W. F. (1994): Operantes Lernen : Methoden, Ergebnisse, Anwendung ; ein Handbuch, Reinhardt, München et al.
- Anjewierden, A., Shostak, I., de Hoog, R. (2002): KMsim - A Meta-Modelling Approach and Environment for Creating Process-Oriented Knowledge Management

- 
- Simulations, 13th European Conference on Knowledge Acquisition, Management and Modelling (EKAW), Sigüenza, S. 65-79.
- Apfelbaum, D. (2006): Wer verdient wie viel? Ergebnisse der c't-Gehaltsumfrage 2005, in: c't magazin für computer und technik, 06(6), S. 130.
- Apostolou, D., Mentzas, G. (2003): Experiences from knowledge management implementations in companies of the software sector, in: Business Process Management Journal, 9(3), S. 354-381.
- APQC (1998):
- Argote, L., Ingram, P. (2000): Knowledge Transfer - A Basis for Competitive Advantage in Firms, in: Organizational Behavior and Human Decision Processes, 82(1), S. 150-169.
- Argote, L., Ingram, P., Levine, J. M., Moreland, R. L. (2000): Knowledge Transfer in Organizations - Learning from the Experience of Others, in: Organizational Behavior and Human Decision Processes, 82(1), S. 1-8.
- Arora, R. (2002): Implementing KM – a balanced score card approach, in: Journal of Knowledge Management, 6(3), S. 240-249.
- Asprey, L., Middleton, M. (2003): Integrative Document and Content Management - Strategies for Exploiting Enterprise Knowledge, Idea Group Publishing, Hershey et al.
- Auer, K., Norris, T. (2001): "ArrierosAlife" a Multi-Agent Approach Simulating the Evolution of a Social System: Modeling the Emergence of Social Networks with "Ascape", in: Journal of Artificial Societies and Social Simulation, 4(1), verfügbar unter: <http://jasss.soc.surrey.ac.uk/4/1/6.html>.
- Austin, J. L. (1962): How to Do Things with Words, Harvard University Press.
- Axelrod, R. (1997): Advancing the Art of Simulation in the Social Sciences, in: R. Conte, R. Hegselmann, P. Terna (Hrsg.): Simulating Social Phenomena, Springer, Berlin, S. 21-40.
- Bach, J. (2003): The MicroPsi Agent Architecture, 5th International Conference on Cognitive Modeling (ICCM-5), Bamberg, Germany, S. 15-20.
- Bach, N., Homp, C. (1998): Objekte und Instrumente des Wissensmanagements, in: Zeitschrift für Führung und Organisation, 67(3), S. 139-146.
- Bach, V. (1999): Business Knowledge Management - von der Vision zur Wirklichkeit, in: V. Bach, P. Vogler, H. Österle (Hrsg.): Business Knowledge Management, Springer Berlin, S. 37-84.
- Badiru, A. B., Sieger, D. B. (1998): Neural network as a simulation metamodel in economic analysis of risky projects, in: European Journal of Operational Research, 105(1), S. 130-142.
- Bährle, C. (2006):
- Bailey, J. W., Christensen, K., Krcmar, H., Lassez, J.-L., Shen, V. Y., Woodfield, S. N. (1982): Evaluating the Effect of Inter-Programmer Communication - Levels of Productivity by Varying Programming Team Structure, in: Software Engineering Notes, 7(1), S. 10-21.
- Ballantine, J., Bonner, M., Levy, M., Martin, A., Munro, I., Powell, P. L. (1998): Developing a 3-D Model of Information Systems Success, in: E. J. Garrity, G. L. Sanders (Hrsg.): Information Systems Success Measurement, London, S. 46-59.
- Balzert, H. (2001): Lehrbuch der Software-Technik, 2 Auflage, Spektrum Akad. Verl., Heidelberg.
- Banks, J. (1998): Handbook of simulation - principles, methodology, advances, applications, and practice, Wiley, New York, NY [u.a.].
-

- Bansler, J. P., Havn, E. C. (2001): Sharing Best Practices - An Empirical Study of IT-Support for Knowledge Sharing, 9th European Conference on Information Systems (ECIS 2001), Bled, Slovenia, S. 653-664.
- Barreto, C., D'Eredita, M. (2004): Unraveling Expertise and Knowledge Creation - Towards an Instance-Based Model of Knowledge (IMoK), 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Bates, J. (1994): The role of emotions in believable agents, in: Communications of the ACM, 37(7): (7), S. 122-125.
- Baumgartner, R., Flesca, S., Gottlob, G. (2001): Visual Web Information Extraction with Lixto, VLDB Conference.
- Becher, C., Apfelbaum, D. (2005): Wer verdient wie viel? Ergebnisse der c't-Gehaltsumfrage 2004, in: c't magazin für computer und technik, 05(6), S. 102.
- Beck, S. (2003): Skill and Competence Management as a Base of an Integrated Personnel Development - A Pilot Project in the Putzmeister, Inc./Germany, in: Journal of Universal Computer Science, 9(12), S. 1381-1387.
- Becker, M. (2002): Personalentwicklung als Kompetenzentwicklung, Hampp, München [u.a.].
- Becker, M. C., Knudsen, M. P. (2003 ): Barriers and Managerial Challenges to Knowledge Transfer Processes, DRUID Summer Conference on Creating, Sharing and Transferring Knowledge, Copenhagen, Sweden.
- Bellifemine, F., Poggi, A., Rimassa, G., Turci, P. (2000): An Object-Oriented Framework to Realize Agent Systems, WOA 2000 Workshop, Parma, Italy, S. 52-57.
- Ben Said, L., Drogoul, A., Bouron, T. (2001): Multi-Agent Based Simulation of Consumer Behaviour: Towards a New Marketing Approach, International Congress On Modelling and Simulation, Canberra.
- Benbasat, I., Zmud, R. W. (1999): Empirical Research in Information Systems: The Practice of Relevance, in: MIS Quarterly, 23(1), S. 3-16.
- Bender, S., Fish, A. (2000): The transfer of knowledge and the retention of expertise - the continuing need for global assignments, in: Journal of Knowledge Management, 4(2), S. 125-137.
- Benferhat, S., Dubois, D., Prade, H. (1993): Argumentative inference in uncertain and inconsistent knowledge bases, 9th Annual Conference on Uncertainty in Artificial Intelligence, Washington, D.C.
- Benjamins, V. R., al., e. (2002): Skills Management in Knowledge-Intensive Organizations, Intelligent Software Components, S.A., Madrid, Spain.
- Berends, H. (2002): Knowledge sharing and distributed cognition in industrial research, 3rd European Conference on Organizational Knowledge, Learning and Capabilities (OKLC 2002), Athens, Greece.
- Berends, H., van der Bij, H., Debackere, K., Weggeman, M. (2004): Knowledge sharing mechanisms in industrial research, Working Paper 04.04, Eindhoven Centre for Innovation Studies, Eindhoven.
- Beydeda, S., Book, M., Gruhn, V., Hrsg. (2005): Model-Driven Software Development, Springer, Berlin.
- Birkhölzer, T., Dantas, L., Dickmann, C., Vaupel2, J. (2004): Interactive Simulation of a Software Producing Organisation's Operations based on Concepts of CMMI and Balanced Scorecards, 5th International Workshop on Software Process Simulation and Modeling (ProSim 2004), Edinburgh, Scotland.
- Blackler, F. (1995): Knowledge, Knowledge Work and Organizations: An Overview and Interpretation, in: Organization Studies, 16(6), S. 1021-1046.

- 
- Blaschke, S. (2004): Knowledge Creation in Organizations - An Agent-Based Demographic Perspective, 5th European Conference on Organizational Knowledge, Learning and Capabilities (OKLC 2004), Innsbruck, Austria, S. Session G-2.
- Blikstein, P., Dor Abrahamson, Wilensky, U. (2005): NetLogo: Where We Are, Where We're Going, Interaction Design & Children, Boulder, Colorado.
- Blöink, R. (2006): Die Struktur der Intelligenz im Hamburg-Wechsler-Intelligenztest für Erwachsene - HAWIE-III: ein Beitrag zur Konstruktvalidität, Kovac, Hamburg.
- Boer, N.-I., Berends, H. (2003): The relational dimension of knowledge sharing - an empirical study of an industrial research group, 4th European Conference on Organizational Knowledge, Learning and Capabilities (OKLC 2003), Barcelona, Spain.
- Boer, N.-I., van Baalen, P. J., Kumar, K. (2002a): An Activity Theory Approach for Studying the Situatedness of Knowledge Sharing, 35th Hawaii International Conference on System Sciences (HICSS 2002), Big Island, Hawaii.
- Boer, N.-I., van Baalen, P. J., Kumar, K. (2002b): The importance of sociality for understanding knowledge sharing processes in organizational contexts, ERIM Report Series ERS-2002-05-LIS, ERIM, Rotterdam, verfügbar unter: <http://www.eur.nl/WebDOC/doc/erim/erimrs20020308162214.pdf>.
- Bots, P. W. G., de Bruijn, H. (2002): Effective Knowledge Management in Professional Organizations - Going by the rules, 35th Hawaii International Conference on System Sciences (HICSS 2002), Big Island, Hawaii.
- Boud, D., Middleton, H. (2003): Learning from others at work - communities of practice and informal learning, in: Journal of Workplace Learning, 15(5), S. 192-202.
- Bracha, G. (2004): Generics in the Java Programming Language, verfügbar unter: <http://java.sun.com/j2se/1.5/pdf/generics-tutorial.pdf>, letzter Abruf: 25.08.2007.
- Braf, E. (2001): Knowledge or Information - What makes the difference?, IFIP WG8.1 Working Conference, Organisational Semiotics - evolving a science of information systems, Montreal, Canada.
- Bratman, M. E. (1984): Two Faces of Intention, in: Philosophical Review, 93(3), S. 375-405.
- Bratman, M. E. (1987): Intentions, Plans, and Practical Reasoning, Harvard University Press, Cambridge, MA.
- Bratman, M. E., Israel, D. J., Pollack, M. E. (1988): Plans and resource-bounded practical reasoning, in: Computational Intelligence, 4, S. 349-355.
- Bredl, K. (2005): Kompetenz von Beratern - Analyse des Kompetenzerwerbs bei Unternehmensberatern im Kontext der Expertiseforschung, PhD, Universität Regensburg, Regensburg
- Brooks, R. A. (1991): Intelligence without representation, in: Artificial Intelligence, 47(1), S. 139-159.
- Brooks, R. J., Robinson, S. (2001): Simulation, Palgrave, Basingstoke et al.
- Brown, P., Levinson, S. C. (1987): Politeness - Some universals in language usage, Cambridge University Press, Cambridge, MA.
- Bruderer, E., Maiers, M. (1997): From Margin to the Mainstream - An Agenda for Computer Simulations in the Social Sciences, in: R. Conte, R. Hegselmann, P. Terna (Hrsg.): Simulating Social Phenomena, Springer, Berlin, S. 89-102.
- Bueren, A., Schierholz, R., Kolbe, L., Brenner, W. (2004): Customer Knowledge Management - Improving Performance of Customer Relationship Management with Knowledge Management, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
-

- Bullinger, H.-J., Wörner, K., Prieto, J. (1997): Wissensmanagement heute, Fraunhofer Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart.
- Burt, R. S. (1992): Structural holes - The social structure of competition, Harvard Univ. Press, Cambridge.
- Bush, A. A., Tiwana, A. (2005): Designing Sticky Knowledge Networks, in: Communications of the ACM, 48(5), S. 67-71.
- Byrne, D. (1998): Complexity Theory and the Social Sciences - An Introduction, Routledge, London.
- Cain, J. W., McCrindle, R. J. (2002): An Investigation into the Effects of Code Coupling on Team Dynamics and Productivity, 26th International Computer Software and Applications Conference (COMPSAC 2002), Oxford, England, S. 907-913.
- Canals, A., Boisot, M., MacMillan, I. (2004): Evolution of Knowledge Management Strategies in Organizational Populations : A Simulation Model, 5th European Conference on Organizational Knowledge, Learning and Capabilities (OKLC 2004), Graz, Austria.
- Canessa, E., Riolo, R. L. (2003): The Effect of Organizational Communication Media on Organizational Culture and Performance - An Agent-Based Simulation Model, in: Computational & Mathematical Organization Theory, 9, S. 147-176.
- Carley, K. M. (1992): Organizational learning and personnel turnover, in: Organization Science, 3(1), S. 20-46.
- Carley, K. M. (1994): Computational organization theory, Erlbaum, Hillsdale, NJ [u.a.].
- Carley, K. M. (1996): A comparison of artificial and human organizations.
- Carley, K. M. (2000a): Computational Analysis of Social and Organizational Systems, in: Organizational Science, 34(2).
- Carley, K. M. (2000b): Intraorganizational complexity and computation, in: J. A. C. Baum (Hrsg.): The Blackwell Companion to Organizations, Blackwell, Oxford, UK, S. 208-232.
- Carley, K. M. (2001): Computational Organization Science - A New Frontier, in: C. A. Irvine (Hrsg.): Arthur M. Sackler Colloquium Series on Adaptive Agents, Intelligence, and Emergent Human Organization: Capturing Complexity through Agent-Based Modeling, National Academy of Sciences Press, S. 7257-7262.
- Carley, K. M. (2002a): Information Technology and Knowledge Distribution in C3I teams, Command and Control Research and Technology Symposium, Naval Postgraduate School, Monterey, CA.
- Carley, K. M. (2002b): Smart Agents and Organizations of the Future, in: L. Lievrouw, S. Livingstone (Hrsg.): The Handbook of New Media - 2002, Sage, Thousand Oaks, CA, S. 206-220.
- Carley, K. M., Gasser, L. (1999): Computational Organization Theory, in: G. Weiss (Hrsg.): Distributed Artificial Intelligence, MIT Press, Cambridge, MA.
- Carley, K. M., Kjaer-Hansen, J., Newell, A. F., Prietula, M. J. (1992): Plural-SOAR - capabilities and coordination of multiple agents, in: M. Masuch, M. Warglien (Hrsg.): Artificial Intelligence in Organization and Management Theory - Models of Distributed Activity, North-Holland, Amsterdam, S. 87-118.
- Carley, K. M., Prietula, M. J. (1993): Plural-Soar - Towards the Development of a Cognitively Motivated Theory of Organizations, Symposium conducted for the National Science Foundation.
- Carley, K. M., Prietula, M. J. (1994): ACTS Theory - Extending the Model of Bounded Rationality, in: K. M. Carley, M. J. Prietula (Hrsg.): Computational Organization Theory, Lawrence Earlbaum Associates, Hillsdale, NJ, S. 55-87.

- 
- Carley, K. M., Schreiber, C. (2002): Information Technology and Knowledge Distribution in C3I teams, Proceedings of the 2002 Command and Control Research and Technology Symposium, Naval Postgraduate School, Monterey, CA.
- Carley, K. M., Schreiber, C. (2004): Going Beyond the Data: Empirical Validation Leading to Grounded Theory, in: Computational & Mathematical Organization Theory, 10(2), S. 155-164.
- Carley, K. M., Tsvetovat, M. (2004): Modeling Complex Socio-technical Systems using Multi-Agent Simulation Methods, in: KI - Künstliche Intelligenz, 04(2), S. 23-28.
- Carlo, J. L., Yoo, Y. (2003): Language Games in Computer-Mediated Communication, in: Sprouts: Working Papers on Information Environments, Systems and Organizations, 3(Winter), verfügbar unter: <http://weatherhead.cwru.edu/sprouts/2003/030103.pdf>.
- Carmel, E. (1999): Global software teams: Collaborating across borders and time zones, Prentice Hall, Upper Saddle River, NJ.
- Castelfranchi, C., Conte, R., Paolucci, M. (1998): Normative reputation and the costs of compliance, in: Journal of Artificial Societies and Social Simulation, 1(3), verfügbar unter: <http://www.soc.surrey.ac.uk/JASSS/1/3/3.html>.
- Cataldo, M., Carley, K. M., Argote, L. (2000): The Effect of Personnel Selection Schemes on Knowledge Transfer, 9th International Conference on Computational Analysis of Social and Organizational Systems, Pittsburgh, PA.
- Cau, A., Concas, G., Melis, M., Turnu, I. (2005): A Discrete Event Simulation Model of Extreme Programming Process, 6th International Workshop on Software Process Simulation and Modeling (ProSim 2005), St. Louis, MO.
- Chowdhury, S. (2002): Role of Trust in Knowledge Search and Transfer, Association for Small Business and Entrepreneurship 2002 Annual Conference, St. Louis, Missouri, S. 70.
- Chun, K. J., Park, H. K. (1998): Examining the conflicting results of GDSS research, in: Information & Management, 33, S. 313-325.
- Cios, K. J., Kurgan, L. (2005): Trends in Data Mining and Knowledge Discovery, in: N. R. Pal, L. C. Jain, N. Teoderesku (Hrsg.): Knowledge Discovery in Advanced Information Systems, Springer, Berlin, S. 1-26.
- Clancey, W. J. (1997): Situated Cognition - On Human Knowledge and Computer Representation, Cambridge University Press, Cambridge, MA.
- Clancey, W. J. (2002): Simulating Activities: Relating Motives, Deliberation, and Attentive Coordination, in: Cognitive Systems Research, 3(3), S. 471-499.
- Clancey, W. J., Sachs, P., Sierhuis, M., van Hoof, R. (1998): Brahms: simulating practice for work systems design, in: International Journal of Human-Computer Studies, 49, S. 831-865.
- Clancey, W. J., Sierhuis, M., Damer, B., Brodsky, B. (2004): Cognitive modeling of social behaviors, in: R. Sun (Hrsg.): Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation, Cambridge University Press.
- Cockburn, A., Williams, L. (2001): The Costs and Benefits of Pair Programming, in: G. Succi, M. Marchesi (Hrsg.): Extreme programming examined, Addison Wesley, S. 223-243.
- Cohen, D. (1998): Toward a Knowledge Context - Report on the First Annual U.C. Berkeley Forum on Knowledge and the Firm, in: California Management Review, 40(3), S. 22-39.
- Cohen, G. P. (1992): The Virtual Design Team: An Object-Oriented Model of Information Sharing in Project Teams, Doctoral, Stanford University, Stanford, CA
-

- Cohen, P. R., Levesque, H. J. (1990): Performatives in a Rationally Based Speech Act Theory, 28th Association for Computational Linguistics Conference, Pittsburgh, Pennsylvania, USA, S. 79-88.
- Conradi, R., Jaccheri, L., Torchiano, M. (2003): Using software process modeling to analyze the COTS based development process, 4th International Workshop on Software Process Simulation and Modeling (ProSim 2003), Portland, OR.
- Conte, R., Castelfranchi, C., Dignum, F. (1999): Autonomous Norm Acceptance, 5th International Workshop on Intelligent Agents: Agent Theories, Architectures, and Languages (ATAL'98), Paris, France.
- Conte, R., Edmonds, B., Moss, S., Sawyer, R. K. (2001): Sociology and Social Theory in Agent Based Social Simulation - A Symposium, in: Computational & Mathematical Organization Theory, 7(3), S. 183 - 205.
- Conte, R., Hegselmann, R., Terna, P., Hrsg. (1997): Simulating social phenomena, Lecture notes in economics and mathematical systems, Springer, Berlin.
- Conway, S. (2003): Valuing Knowledge Management Behaviors - Linking KM Behaviors to Strategic Performance Measures, in: C. W. Holsapple (Hrsg.): Handbook on Knowledge Management, Springer, Berlin et al., S. 461-475.
- Cowan, N. (1988): Evolving Conceptions of Memory Storage, Selective Attention, and Their Mutual Constraints Within the Human Information-Processing System, in: Psychological Bulletin, 104(2), S. 163-191.
- Cowan, R., Jonard, N. (2004): Network structure and the diffusion of knowledge, in: Journal of Economic Dynamics & Control, 28(8), S. 1557 - 1575.
- Cowan, R., Jonard, N. (2005): Knowledge Creation and Flows in Science, in: D. Fornahl, C. Zellner, D. B. Audretsch (Hrsg.): The Role of Labour Mobility and Informal Networks for Knowledge Transfer, Springer, New York.
- Creplet, F., Dupouet, O., Kerna, F., Mehmanpazir, B., Munier, F. (2001): Consultants and experts in management consulting firms, in: Research Policy, 30, S. 1517-1535.
- Cross, R., Parker, A., Prusak, L. (2000): Knowing What We Know - Supporting Knowledge Creation and Sharing in Social Networks, in: Organizational Dynamics, 3(2), S. 100-120.
- Csikszentmihalyi, M. (2004): Flow im Beruf - Das Geheimnis des Glücks am Arbeitsplatz, 2 Auflage, Klett-Cotta, Stuttgart.
- Cubert, R. M., Fishwick, P. A. (1997): MOOSE - an object-oriented multimodeling and simulation application framework, Technical Report 97-012, Department of Computer & Information Science and Engineering, University of Florida, Gainesville, FL.
- Cummings, J. L. (2001): Work groups and knowledge sharing in a global organization, Summary of PhD thesis online: <http://www.scienceofcollaboratories.org/WorkshopStuff/June2001/pdfs/jnc-dis.pdf>, last access: 2005-10-28, Carnegie Mellon University, Pittsburgh, PA.
- Cummings, J. L., Teng, B.-S. (2003): Transferring R&D knowledge - the key factors affecting knowledge transfer success, in: Journal of Engineering and Technology Management, 20(1), S. 39-68.
- Cyert, R. M., March, J. G. (1963): A Behavioral Theory of the Firm, Prentice-Hall, Eaglewood Cliffs, NJ.
- Cyert, R. M., March, J. G. (2001): A behavioral theory of the firm, 2. Auflage, Blackwell, Malden, MS.
- Daft, R. L., Lengel, R. H. (1986): Organizational information requirements, media richness and structural design, in: Management Science, 32(5), S. 554-571.

- 
- Dal Forno, A., Merlone, U. (2002): A multi-agent simulation platform for modeling perfectly rational and bounded-rational agents in organizations, in: *Journal of Artificial Societies and Social Simulation*, 5(2), S. <http://jasss.soc.surrey.ac.uk/5/2/3.html>, verfügbar unter: <http://jasss.soc.surrey.ac.uk/5/2/3.html>.
- Davenport, T. H. (1999): Rigor vs. relevance revisited: response to Benbasat and Zmud, in: *MIS Quarterly*, 23(1), S. 19.
- Davenport, T. H., Jarvenpaa, S. L., Beers, M. C. (1996): Improving Knowledge Work Processes, in: *Sloan Management Review*, 37(4), S. 53-65.
- Davenport, T. H., Prusak, L. (1998): *Working knowledge - how organizations manage what they know*, Harvard Business School Press, Boston.
- Davidsson, P. (2002):
- De Cenzo, D. A., Robbins, S. P. (1994): *Human Resource Management - Concepts & Practices*, 4 Auflage, John Wiley & Sons, Inc., New York et al.
- Dehnbostel, P., Dybowski, G. (2000): *Lernen, Wissensmanagement und berufliche Bildung*, Bertelsmann, Bielefeld.
- Deiters, W., Lucas, R., Weber, T. (1999): *Skill-Management - Ein Baustein für das Management flexibler Teams* 50, Fraunhofer Institut, Dortmund, verfügbar unter: <http://www.do.isst.fgh.de/wm/veroeffentlichungen/>
- DeLone, W. H., McLean, E. R. (1992): Information Systems Success - The Quest for the Dependent Variable, in: *Information Systems Research*, 3(1), S. 60-95.
- Delphi (1997): *Delphi on Knowledge Management - Research & Perspectives on Today's Knowledge Landscape*, Delphi Consulting Group, Boston, MA.
- Dennis, A. R. (1996): Information Exchange and Use in Group Decision Making: You Can Lead a Group to Information, but You Can't Make It Think, in: *MIS Quarterly*, 20(4), S. 433-456.
- Dennis, A. R., Kinney, S. T. (1998): Testing Media Richness Theory in the New Media - The Effects of Cues, Feedback, and Task Equivocality, in: *Information Systems Research*, 9(3), S. 256-274.
- Dennis, R., Valacich, J. (1999): Rethinking media richness - Towards a theory of media synchronicity  
32nd Hawaii International Conference on Systems Sciences (HICSS 1999), Big Island, Hawaii.
- Dervisoglou, G., Berberb, A. (2004): Knowledge Flow During the Product Development Process and Role of the Mediator - A Model Presentation 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Despres, C., Chauvel, D. (1999): Mastering Information Management: Part Six - Knowledge Management, in: *Financial Times*, 1999(8. March), S. 4-6.
- DICE (2003):
- Dingsøy, T., Conradi, R. (2002): A Survey of Case Studies of the Use of Knowledge Management in Software Engineering, in: *International Journal of Software Engineering and Knowledge Engineering*, 12(4), S. 391-414.
- Diotalevi, F. (2005): TestNG makes Java unit testing a breeze - Try this testing framework for its advances over JUnit, verfügbar unter: <http://www-128.ibm.com/developerworks/java/library/j-testng/>, letzter Abruf: 31.08.2007.
- Disterer, G. (2002): Management of project knowledge and experiences, in: *Journal of Knowledge Management*, 6(5), S. 512-520.
- Dörner, D. (1999): *Bauplan für eine Seele*, Reinbeck, Rowohlt.
- Dornhaus, A., Klügl, F., Puppe, F., Tautz, J. (1998): *Task Selection in Honeybees - Experiments Using Multi-Agent Simulation*, GWAL, Bochum, Germany.
-

- Dreyfus, H. (2002): A Phenomenology of Skill Acquisition as the Basis for a Merleau-Pontian Non-Representationalist Cognitive Science, International Conference "Foundations and the ontological quest. Prospects for the new millennium", Rome, Pontifical Lateran University.
- Dreyfus, H. L., Dreyfus, S. E. (1986): Mind over machine - the power of human intuition and expertise in the era of the computer, Free Press, New York.
- Drogoul, A., Ferber, J. (1992): Multi-Agent Simulation as a Tool for Modeling Societies - Application to Social Differentiation in Ant Colonies, MAAMAW, Viterbo.
- Drogoul, A., Vanbergue, D., Meurisse, T. (2002): Multi-Agent Based Simulation - Where are the Agents?, Multi-Agent Based Simulation (MABS'02), Bologna, Italy, S. 1-15.
- Drucker, P. F. (1959): Landmarks of Tomorrow, Harper, New York, NY.
- Drumm, H. J. (2000): Personalwirtschaft, 4. Auflage, Springer, Berlin.
- Dunnette, M. D. (1976): Attitudes, abilities and skills, in: M. D. Dunnette, L. M. Hough (Hrsg.): Handbook of Industrial and Organizational Psychology, Rand McNally College Publ., Chicago, S. 473-520.
- Edelmann, W. (1996): Lernpsychologie, 5. Auflage, Psychologie Verlags Union, Weinheim.
- Edmonds, B., Hales, D. (2003): Replication, Replication and Replication - Some Hard Lessons from Model Alignment, in: Journal of Artificial Societies and Social Simulation, 6(4), S. <http://jasss.soc.surrey.ac.uk/6/4/11.html>.
- Edwards, J. S. (2003): Managing Software Engineers and Their Knowledge, in: A. Aurum, R. Jeffery, C. Wohlin, M. Handzic (Hrsg.): Managing Software Engineering Knowledge, Springer, Berlin, S. 5-27.
- Efimova, L. (2004): Discovering the iceberg of knowledge work - A weblog case, OKLC 2004, Vienna, Austria.
- Elliman, T., Eatock, J., Spencer, N. (2005): Modelling knowledge worker behaviour in business process studies, in: The Journal of Enterprise Information Management, 18(1), S. 79-94.
- Elster, J. (1989): Social Norms and Economic Theory, in: Journal of Economic Perspectives, 3(4), S. 99-117.
- Emerson, R. M. (1976): Social exchange theory, in: Annual Review of Sociology, 2, S. 335-362.
- Engel, A., Möhring, M. (1995): Der Beitrag der sozialwissenschaftlichen Informatik zur sozialwissenschaftlichen Modellbildung und Simulation, in: M. Gsänger, J. Klawitter (Hrsg.): Modellbildung und Simulation in den Sozialwissenschaften, Röhl, Dettelbach, S. 39-59.
- Engeström, Y. (1987): Learning by Expanding - An Activity-theoretical Approach to Developmental Research, Oriental-Konsultit, Helsinki.
- Engeström, Y. (2000): Activity Theory and the Social Construction of Knowledge - A Story of Four Umpires, in: Organization, 7(2), S. 301-310.
- Enkel, E. (2002): Knowledge networks - an integrated approach to manage explicit and implicit knowledge, OKLC 2002.
- Ensign, P. C., Hébert, L. (2004): Knowledge Sharing among R&D Scientists, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Eppler, M. J. (2001): Making Knowledge Visible Through Intranet Knowledge Maps: Concepts, Elements, Cases, 34th Hawaii International Conference in System Sciences (HICSS 2001), Big Island, Hawaii, S. 1-10.

- 
- Eppler, M. J. (2003): Making Knowledge Visible through Knowledge Maps - Concepts, Elements, Cases, in: C. W. Holsapple (Hrsg.): Handbook on Knowledge Management, Springer, Berlin et al., S. 189-205.
- Eppler, M. J., Burkhard, R. A. (2005): Knowledge Visualization - Towards a New Discipline and its Fields of Application, in: D. Schwartz (Hrsg.): Encyclopedia of Knowledge Management, Idea Group, Hershey, PA, S. 317-325.
- Eppler, M. J., Mengis, J. (2004): Wissensdialoge - Dimensionen, Prinzipien und Probleme der gesprächsbasierten Wissenskommunikation, in: M. J. Eppler, R. Reinhardt (Hrsg.): Wissenskommunikation in Organisationen - Methoden, Instrumente, Theorien, Springer, Berlin, S. 88-107.
- Eppler, M. J., Reinhardt, R. (2004): Zur Einführung - Das Konzept der Wissenskommunikation, in: M. J. Eppler, R. Reinhardt (Hrsg.): Wissenskommunikation in Organisationen - Methoden, Instrumente, Theorien, Springer, Berlin, S. 1-12.
- Eppler, M. J., Schindler, M., Schmid, B. F., Seifried, P. (1999): Das Enterprise Knowledge Media-Referenzmodell, Arbeitsbericht HSG/MCM/Nr. 6, Kompetenzzentrum EKM der Universität St. Gallen, St. Gallen.
- Eppler, M. J., Sukowski, O. (2000): Managing Team Knowledge - Core Processes, Tools and Enabling Factors, in: European Management Journal, 18(3), S. 334-341.
- Erickson, T., Kellogg, W. A. (2003): Knowledge Communities - Online Environments for Supporting Knowledge Management and its Social Context, in: M. S. Ackerman, V. Pipek, V. Wulf (Hrsg.): Sharing Expertise - Beyond Knowledge Management, MIT Press, Cambridge, MA, S. 299-326.
- Ericsson, K. A., Krampe, R. T., Tesch-Romer, C. (1993): The role of deliberate practice in the acquisition of expert performance, in: Psychological Review, 100(3), S. 363-406.
- Esser, H. (1990): "Habits", "Frames" und "Rational Choice" - Die Reichweite von Theorien der rationalen Wahl, in: Zeitschrift für Soziologie, 19, S. 231-247.
- Faraj, S., Sproull, L. (2000): Coordinating Expertise in Software Development Teams in: Management Science, 46(12), S. 1554-1568.
- Fayyad, U., Uthurusamy, R. (1996): Data Mining and Knowledge Discovery in Databases, in: Communications of the ACM, 39(11), S. 24-26.
- Fehr, E., Fischbacher, U. (2004): Social norms and human cooperation, in: Trends in Cognitive Sciences, 8(4), S. 185-190.
- Fensel, D. (2004): Ontologies - A Silver Bullet for Knowledge Management and Electronic Commerce, 2 Auflage, Springer, Berlin.
- Ferber, J. (1999): Multi-Agent Systems - An introduction to distributed artificial intelligence, Addison-Wesley, London et al.
- Ferber, J. (2001): Multiagentensysteme - eine Einführung in die Verteilte Künstliche Intelligenz, Addison-Wesley, Munich et al.
- Ferrán-Urdaneta, C. (1999): Teams or Communities? Organizational Structures for Knowledge, SIGCPR, New Orleans (LA) USA.
- Fincham, R., Rhodes, P. (2003): Principles of Organizational Behaviour, 3. Auflage, Oxford University Press, Oxford, MA.
- Finin, T., Fritzson, R., McKay, D., McEntire, R. (1994): KQML as an Agent Communication Language, 3rd International Conference on Information and Knowledge Management (CIKM'94), Gaithersburg, Maryland.
- Finke, R., Muck, P. M., Schuler, H. (2001): Mitarbeitergespräche, in: H. Schuler (Hrsg.): Lehrbuch der Personalpsychologie, Hogrefe, Göttingen, S. 433-480.
-

- FIPA (2002): FIPA Communicative Act Library Specification, verfügbar unter: <http://www.fipa.org/specs/fipa00037/SC00037J.html>, letzter Abruf: 23.08.2007.
- Fischer, K., Müller, J. P., Pischel, M. (1994): Unifying Control in a Layered Agent Architecture, Technical Report TM-94-05, DFKI, Saarbrücken.
- Fiske, A. P. (1991): Structures of social life - the four elementary forms of human relations, Free Pr., New York.
- Fiske, A. P., Haslam, N. (2005): The four basic social bonds - Structure for coordinating interaction, in: M. W. Baldwin (Hrsg.): Interpersonal Cognition, Guilford Press, New York, S. 267-298.
- Flanagan, D. (2005): Java in a Nutshell, 3. Auflage, O'Reilly, Beijing [u.a.].
- Flap, H., Bulder, B., Völker, B. (1998): Intra-organizational Networks and Performance - A Review, in: Computational & Mathematical Organization Theory, 4(2), S. 109-147.
- Flores-Méndez, R. A. (2002): Modelling Agent Conversations for Action, Dissertation, UNIVERSITY OF CALGARY, CALGARY, ALBERTA
- Frank, U. (2006): Towards a pluralistic conception of research methods in information systems research, ICB-Research Report 7, Universität Duisburg-Essen, Essen.
- Galbraith, J. R. (1974): Organization Design: An Information Processing View, in: Interfaces, 1974(May), S. 28-36.
- Gandon, F. (2002): Distributed Artificial Intelligence and Knowledge Management - Ontologies and Multi-agent Systems for a corporate Semantic Web, PhD, INRIA and University of Nice, Nice
- Gebert, D. (1992): Kommunikation, in: E. Frese (Hrsg.): Enzyklopädie der Betriebswirtschaftslehre, Stuttgart, S. 1110-1121.
- Gebert, D., Rosenstiel, L. v. (2002): Organisationspsychologie - Person und Organisation, 5., aktualisierte und erw. Aufl Auflage, Kohlhammer, Stuttgart [u.a.].
- Geißler, H. (1999): Standardisierung und Entstandardisierung von Wissen als Aufgabe von Wissensmanagement, in: P. W. Beratung (Hrsg.): Organisationslernen durch Wissensmanagement, Lang, Frankfurt am Main [u.a.], S. 39-63.
- Gershenson, C. (2002): Philosophical Ideas on the Simulation of Social Behaviour, in: Journal of Artificial Societies and Social Simulation, 5(3), S. <http://jasss.soc.surrey.ac.uk/5/3/8.html>.
- Gilbert, N., Pyka, A., Ahrweiler, P. (2001): Innovation Networks - A Simulation Approach, in: Journal of Artificial Societies and Social Simulation, 4(3), verfügbar unter: <http://www.soc.surrey.ac.uk/JASSS/4/3/8.html>.
- Gobet, F., Wood, D. (1999): Expertise, models of learning and computer-based tutoring, in: Computers & Education, 33 S. 189-207.
- Goh, S. C. (2002): Managing effective knowledge transfer - an integrative framework and some practice implications, in: Journal of Knowledge Management, 6(1), S. 23-30.
- Golder, S. A., Huberman, B. A. (2006): Usage patterns of collaborative tagging systems in: Journal of Information Science, 32(2), S. 198-208.
- Goldspink, C. (2000): Modelling social systems as complex - Towards a social simulation meta-model, in: Journal of Artificial Societies and Social Simulation, 3(2), S. <http://www.soc.surrey.ac.uk/JASSS/3/2/1.html>.
- Gómez-Pérez, A., Fernández-López, M., Corcho, O. (2004): Ontological Engineering, Springer, London.
- Gosling, J., Joy, B., Steele, G., Bracha, G. (2005): The Java Language Specification, 3. Auflage, Prentice Hall, Englewood Cliffs, New Jersey.

- 
- Graefe, G. (2003): Incredible Information on the Internet - Biased Information Provision and a Lack of Credibility as a Cause of Insufficient Information Quality, 8th International Conference on Information Quality (ICIQ 2003), Cambridge, MA.
- Gray, P. H. (2001): The impact of knowledge repositories on power and control in the workplace, in: *Information Technology & People*, 14(4), S. 368-384.
- Grothe, M., Gentsch, P. (2000): *Business Intelligence. Aus Informationen Wettbewerbsvorteile gewinnen*, München et al.
- Gruenfeld, D. H., Martorana, P. V., Fan, E. t. (2000): What Do Groups Learn from Their Worldliest Members? Direct and Indirect Influence in Dynamic Teams, in: *Organizational Behavior and Human Decision Processes*, 82(1), S. 49-59.
- Guanawardena, C. N. (1995): Social Presence Theory and Implications for Interaction and Collaborative Learning in Computer Conferences, in: *International Journal of Educational Telecommunications*, 1(2/3), S. 147-166.
- Guay, F., Vallerand, R. J., Blanchard, C. (2000): On the Assessment of Situational Intrinsic and Extrinsic Motivation: The Situational Motivation Scale (SIMS), in: *Motivation and Emotion*, 24(3), S. 175-213.
- Guest, D. (1997): Human resource management and performance - a review and research agenda, in: *International Journal of Human Resource Management*, 8(3), S. 263-276.
- Güldenbergl, S. (2001): *Wissensmanagement und Wissenscontrolling in lernenden Organisationen - Ein systemtheoretischer Ansatz*, 4. Auflage, Deutscher Universitätsverlag, Wiesbaden.
- Günther, H.-O., Lehmann, M., Raffel, W.-U., Wagner, G. (2001):
- Gutwin, C., Greenberg, S. (1998): Effects of awareness support on groupware usability, *Proceedings of the SIGCHI conference on Human factors in computing systems*, Los Angeles, California, S. 511 - 518.
- Haak, L. (2002): Konzeption zur Integration eines Data Warehouse mit Wissensmanagementsystemen, in: E. v. Maur, R. Winter (Hrsg.): *Data Warehousing 2002 - Vom Data Warehouse zum Corporate Knowledge Center*, Physica-Verlag, Heidelberg, S. 301-317.
- Haberfellner, R., Nagel, P., Becker, M., Büchel, A., von Massow, H. (1999): *Systems Engineering - Methoden und Praxis*, 10. Auflage, Verlag Industrielle Organisation, Zürich.
- Habermas, J. (1981): *Theorie des Kommunikativen Handelns*, Suhrkamp, Frankfurt a.M.
- Hädrich, T., Maier, R. (2004): Modeling Knowledge Work, *Multikonferenz Wirtschaftsinformatik (MKWI 2004)*, University Duisburg-Essen, S. 189-203.
- Hakanson, L., Nobel, R. (1998): Technology characteristics and reverse technology transfer, in: *Management International Review*, 40(1, Special Issue), S. 29-48.
- Halpin, T. A. (2001): *Information Modeling and Relational Databases - from conceptual analysis to logical design*, Morgan Kaufmann Publishers Inc., San Francisco, CA.
- Hanakawa, N., Matsumoto, K.-I., Torii, K. (2002): A Knowledge-Based Software Process Simulation Model, in: *Annals of Software Engineering*, 14, S. 383-406.
- Hanakawa, N., Morisaki, S., Matsumoto, K.-I. (1998): A Learning Curve Based Simulation Model for Software Development, *20th International Conference on Software Engineering*, Kyoto, Japan, S. 350-359.
- Handzic, M., Lazaro, O., Van Toorn, C. (2004): *Enabling Knowledge Sharing - Culture versus Technology*  
5th European Conference on Organizational Knowledge, Learning and Capabilities (OKLC 2004), Linz, Austria.
-

- Hanley, S., Malafsky, G. (2003): A Guide for Measuring the Value of KM Investments, in: C. W. Holsapple (Hrsg.): Handbook on Knowledge Management - Vol. 2, Springer, Berlin et al., S. 369-390.
- Hanne, T., Neu, H. (2004): Simulating Human Resources in Software Development Processes, Bericht 64, Fraunhofer-Institut für Techno- und Wirtschaftsmathematik.
- Hannig, U. (2002): Knowledge Management und Business Intelligence, Berlin et al.
- Hannula, M., Okkonen, J., Virtanen, P. (2004): Using Performance Measurement in Knowledge Work – Empirical Findings and Prospective Challenges, 4th European Academy of Management Annual Conference (EURAM 2004), University of St Andrews, Scotland.
- Hansen, M. T. (1999): The search-transfer problem - the role of weak ties in sharing knowledge across organization subunits, in: Administrative Science Quarterly, 44(1), S. 82-111.
- Hansen, M. T., Nohria, N., Tierney, T. (1999): What's Your Strategy for Managing Knowledge?, in: Harvard Business Review, 77(3-4), S. 106-116.
- Heap, S. H., Hollis, M., Lyons, B., Sugden, R., Weale, A. (1992): The Theory of Choice - A Critical Guide, Blackwell Publishers, Oxford, UK.
- Heckhausen, H. (1989): Motivation und Handeln, 2., völlig überarb. und erg. Aufl Auflage, Springer, Berlin [u.a.].
- Heideloff, F., Baitsch, C. (1998): Wenn Wissen Wissen generiert - Erläuterungen rund um ein Fallbeispiel.: Wissensmanagement : Erfahrungen und Perspektiven, Gabler, Wiesbaden, S. 67-83.
- Heider, F. (1958): The psychology of interpersonal relations, 2. Aufl Auflage, Wiley, London.
- Heiss, S. F. (2004): Personale und interpersonale Faktoren für die Wissenskommunikation in Communities of Practice, in: M. J. Eppler, R. Reinhardt (Hrsg.): Wissenskommunikation in Organisationen - Methoden, Instrumente, Theorien, Springer, Berlin, S. 157-176.
- Henderson-Sellers, B., Giorgine, P., Hrsg. (2006): Agent-Oriented Methodologies, IDEA Group, London.
- Hendriks, P. (1999): Why Share Knowledge? The Influence of ICT on the Motivation for Knowledge Sharing, in: Knowledge and Process Management, 6(2), S. 91-100.
- Hendriks, P. (2004): Assessing the role of culture in knowledge sharing, OKLC 2004, Graz.
- Henninger, S. (1997): Capturing and Formalizing Best Practices in a Software Development Organization, 9th International Conference on Software Engineering and Knowledge Engineering (SEKE 1997), Madrid, Spain, S. 24-31.
- Herring, S. C., Scheidt, L. A., Bonus, S., Wright, E. (2004): Bridging the Gap - A Genre Analysis of Weblogs, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Herrmann, T., Kienle, A. (2004): Kontextberücksichtigung als Kernaufgabe der Wissenskommunikation, in: M. J. Eppler, R. Reinhardt (Hrsg.): Wissenskommunikation in Organisationen - Methoden, Instrumente, Theorien, Springer, Berlin, S. 50-68.
- Herschel, R. T., Jones, N. E. (2005): Knowledge management and business intelligence: the importance of integration, in: Journal of Knowledge Management 9(4), S. 45-55.
- Hiermann, W., Höfferer, M. (2005): Skill Management: Searching Highly Skilled Employees for Teambuilding and Project Management Tasks, 5th International Conference on Knowledge Management (I-Know'05), Graz, Austria.

- 
- Hindelang, G. (1983): Einführung in die Sprechakttheorie, Niemeyer, Tübingen.
- Hinds, P. J., Pfeffer, J. (2003): Why Organizations Don't "Know What They Know" - Cognitive and Motivational Factors Affecting the Transfer of Expertise, in: M. S. Ackerman, V. Pipek, V. Wulf (Hrsg.): Sharing Expertise - Beyond Knowledge Management, The MIT Press, Cambridge, Massachusetts, S. 3-26.
- Hmelo-Silver, C. E., Pfeffer, M. G. (2004): Comparing expert and novice understanding of a complex system from the perspective of structures, behaviors, and functions, in: Cognitive Science, 28 S. 127-138.
- Ho, T. B., PAKDD (2005): Advances in knowledge discovery and data mining : 9th Pacific-Asia conference, PAKDD 2005, Hanoi, Vietnam, May 18 - 20, 2005 ; proceedings, Springer, Berlin u.a.
- Hochstein, L., Carver, J., Shull, F., Asgari, S., Basili, V., Hollingsworth, J. K., Zelkowitz, M. V. (2005): Parallel Programmer Productivity - A Case Study of Novice Parallel Programmers, The 2005 ACM / IEEE Conference on Supercomputing, Seattle, WA, S. 35-43.
- Hoffmann, M., Goesmann, T., Kienle, A. (2003): Analyse und Unterstützung von Wissensprozessen als Voraussetzung für erfolgreiches Wissensmanagement, in: A. Abecker, K. Hinkelmann, H. Maus, H. J. Müller (Hrsg.): Geschäftsprozessorientiertes Wissensmanagement, Springer, Berlin, S. 159-184.
- Holsapple, C. W., Hrsg. (2003): Handbook on Knowledge Management. Vol. 1+2, Berlin.
- Holsapple, C. W., Joshi, K. D. (2002): Knowledge manipulation activities - results of a Delphi study, in: Information & Management, 39, S. 477-490.
- Homburg, V., Meijer, A. (2001): Why Would Anyone Like To Share His Knowledge?, 34th Hawaii International Conference on System Sciences (HICSS 2001), Big Island, Hawaii.
- Horling, B., Lesser, V. E., Vincent, R. (2000): Multi-Agent System Simulation Framework, 16th IMACS World Congress 2000 on Scientific Computation, Applied Mathematics and Simulation, Lausanne, Switzerland.
- Huotari, M.-L., Iivonen, M. (2004): Managing Knowledge-Based Organization Through Trust, in: M.-L. Huotari, M. Iivonen (Hrsg.): Trust in Knowledge Management and Systems in Organizations, Idea Group Publ., Hershey, Pa., S. 1-29.
- Huysman, M., de Wit, D. (2003): A Critical Evaluation of Knowledge Management Practices, in: M. S. Ackerman, V. Pipek, V. Wulf (Hrsg.): Sharing Expertise - Beyond Knowledge Management, MIT press, Cambridge, MA, S. 27-56.
- Hyman, J., Summers, J. (2004): Lacking balance? Work-life employment practices in the modern economy, in: Personnel Review, 33(4), S. 418-429.
- Inuzuka, A. (2003): How to Share Knowledge Effectively?, in: T. Terano, H. Deguchi, K. Takadama (Hrsg.): Meeting the Challenge of Social Problems via Agent-Based Simulation, Springer, Tokio et al., S. 123-137.
- Inuzuka, A., Nakamori, Y. (2002a): An Agent-based Approach for an Inquiry of Knowledge Management, Basic Theory and Application for Artificial Society/Organization/Economy, Tokyo, S. 29-36.
- Inuzuka, A., Nakamori, Y. (2002b): Technical Problems for Knowledge Sharing - through a simulation study, Spring National Conference 2002, Tokyo.
- Inuzuka, A., Nakamori, Y. (2004): A recommendation for IT-driven knowledge sharing, in: Systems and Computers in Japan, 35(3), S. 1-11.
- Issing, L. J. (1988): Wissensvermittlung mit Medien, in: H. Mandl (Hrsg.): Wissenspsychologie, Psychologie-Verlags Union, München-Weinheim, S. Kapitel 22.
-

- Jacobson, M. J. (2001): Problem solving, cognition, and complex systems: Differences between experts and novices, in: *Complexity*, 6(2), S. 1-9.
- Jennex, M. E., Olfman, L. (2003): A Knowledge Management Success Model - An Extension of DeLone and McLean's IS Success Model, 9th Americas Conference on Information Systems.
- Jennex, M. E., Olfman, L. (2004): Assessing Knowledge Management Success/Effectiveness Models, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Jennings, N. R., Sycara, K., Wooldridge, M. J. (1998): A Roadmap of Agent Research and Development, in: *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1), S. 275-306.
- Jennings, N. R., Wooldridge, M. J. (1998): Applications of Intelligent Agents, in: N. R. Jennings, M. J. Wooldridge (Hrsg.): *Agent Technology: Foundations, Applications and Markets*, Springer-Verlag, Heidelberg, Germany, S. 3-28.
- Jiang, Y., Chun, M. M. (2001): Selective attention modulates implicit learning, in: *The Quarterly Journal of experimental Psychology*, 54A(4), S. 1105-1124.
- Jimes, C., Lucardie, L. (2003): Reconsidering the tacit-explicit distinction - A move toward functional (tacit) knowledge management, in: *Electronic Journal of Knowledge Management*, 1(1), S. 23-32.
- Johnston, R. B. (2001): Situated Action, Structuration and Actor-Network Theory - An Integrative Theoretical Perspective, *Global Co-Operation in the New Millennium - 9th European Conference on Information Systems*, Bled, Slovenia., S. 232-242.
- Joshi, K. D., Sarker, S., Sarker, S. (2004): Knowledge Transfer Among Face-to-Face Information Systems Development Team Members - Examining the Role of Knowledge, Source, and Relational Context, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Joshi, K. D., Sarker, S., Sarker, S. (2005): The Impact of Knowledge, Source, Situational and Relational Context on Knowledge Transfer During ISD Process, 38th Hawaii International Conference on System Sciences (HICSS 2005), Big Island, Hawaii.
- Jost, P.-J. (2000): *Organisation und Motivation - eine ökonomisch-psychologische Einführung*, 1. Aufl Auflage, Gabler, Wiesbaden.
- Jost, P.-J., Hrsg. (2001): *Die Prinzipal-Agenten-Theorie in der Betriebswirtschaftslehre*, Schäffer-Poeschel, Stuttgart.
- Kang, M. (2001): Team-soar: A computational model for multilevel decision making, in: *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 31(6), S. 708-714.
- Kang, M., Waisel, L. B., Wallace, W. A. (1998): *Team soar - a model for team decision making: Simulating organizations - computational models of institutions and groups*, MIT Press, Cambridge, MA, S. 23-45
- Kankanhalli, A., Tan, B. C. Y. (2004): A Review of Metrics for Knowledge Management Systems and Knowledge Management Initiatives, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Kaplan, R. S., Norton, D. P. (1996): Using the Balanced Scorecard as a Strategic Management System, in: *Harvard Business Review*, 74(1-2), S. 75-85.
- Kautz, K., Thaysen, K. (2001): Knowledge, learning and IT support in a small software company, in: *Journal of Knowledge Management*, 5(4), S. 349-357.
- Kellner, M. I., Madachy, R., Raffo, D. M. (1999): Software Process Modeling and Simulation - Why, What, How, in: *Journal of Systems and Software*, 46(2/3).
- Kelloway, E. K., Barling, J. (2000): Knowledge work as organizational behavior, in: *International Journal of Management Reviews*, 2(3), S. 287-304.

- 
- Kendall, E., Murali Krishna, P. V., Suresh, C. B., Pathak, C. V. (2000): An Application Framework for Intelligent and Mobile Agents, in: *ACM Computing Surveys*, 32(1, Article 20).
- Kennedy, A., Syme, D. (2001): The Design and Implementation of Generics for the .NET Common Language Runtime, 2001 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), Snowbird, Utah, USA.
- Kienle, A. (2003): *Integration von Wissensmanagement und kollaborativem Lernen durch technisch unterstützte Kommunikationsprozesse*, EUL Verlag, Köln.
- Kilduff, M., & Krackhardt, D. (1994): Bringing the individual back in - A structural analysis of the internal market for reputation in organizations, in: *Academy of Management Journal*, 37, S. 87-108.
- Kilduff, M., Tsai, W. (2003): *Social Networks and Organizations*, Sage, London.
- Kim, L., Nelson, R. R. (2000): *Technology, Learning, and Innovation: Experiences of Newly Industrializing Economies*, Cambridge University Press, Cambridge, UK.
- Kirk, D., Tempero, E. (2004): *A Flexible Software Process Model*, Research Report UoA-SE-2004-3, University of Auckland, Department for Software Engineering, Auckland, New Zealand.
- Kirn, S. (2002): Kooperierende intelligente Softwareagenten, in: *Wirtschaftsinformatik*, 44(1), S. 53-63.
- Klein, G. (1999): *Sources of power - How people make decisions*, MIT press, Cambridge, MA.
- Klein, J. (2002): Breve - A 3D simulation environment for the simulation of decentralized systems and artificial life, 8th International Conference on the Simulation and Synthesis of Living Systems (Artificial Life VIII).
- Klügl, F. (2001): *Multiagentensimulation - Konzepte, Werkzeuge, Anwendungen*, Addison-Wesley, München et al.
- Klügl, F., Oechslein, C., Puppe, F., Dornhaus, A. (2002): Multi-Agent Modelling in Comparison to Standard Modelling, in: F. J. Barros, N. Giambiasi (Hrsg.): *AIS'2002 (Artificial Intelligence, Simulation and Planning in High Autonomy Systems*, SCS Publishing House, S. 105-110.
- Ko, D.-G., Dennis, A. R. (2004): Who Profits from Knowledge Management? A Case of Experience versus Expertise, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Koch, S., Mandl, H. (1999): *Wissensmanagement - Anwendungsfelder und Instrumente für die Praxis*, Forschungsbericht 103, Ludwig-Maximilians-Universität, Lehrstuhl für Empirische Pädagogik und Pädagogische Psychologie, München.
- Kostova, T. (1999): Transnational transfer of strategic organizational practices: a contextual perspective, in: *Academy of Management Review*, 24(2), S. 308-324.
- Krcmar, H. (2000): *Informationsmanagement*, 2 Auflage, Springer, Berlin.
- Krischer, T., Metall, I. (2003): *Entgelt in der IT-Branche5. Erhebung*, IG-Metall, Frankfurt a.M., Germany.
- Krischer, T., Metall, I. (2004): *Entgelt in der IT-Branche6. Erhebung*, IG Metall, Frankfurt a.M.
- Krishnan, M. S. (1998): The role of team factors in software cost and quality - An empirical analysis, in: *Information Technology & People*, 11(1), S. 20-35.
- Kunz, J. C., Levitt, R. E., Jin, Y. (1998): The Virtual Design Team: A Computational Simulation Model of Project Organizations, in: *Communications of the ACM*, 41(11), S. 84-92.
-

- Kuutti, K. (1996): Activity Theory as a Potential Framework for Human-Computer Interaction Research, in: B. A. Nardi (Hrsg.): Context and Consciousness - Activity Theory and Human-Computer Interaction, MIT Press, Cambridge, MA, S. 17-44.
- Labrou, Y. (2001): Standardizing Agent Communication, in: M. Luck, V. Marik, O. Stepankova, R. Trappl (Hrsg.): Multi-Agent Systems & Applications, Advanced Course on Artificial Intelligence (ACAI 2001), Springer Verlag, Prague, Czech Republic.
- Labrou, Y., Finin, T. (1996): Semantics for an Agent Communication Language, in: M. P. Singh, A. Rao, M. J. Wooldridge (Hrsg.): Intelligent Agents IV - Agent Theories, Architectures, and Languages, Springer, Heidelberg, Germany, S. 209-214.
- Labrou, Y., Finin, T. (1997): A Proposal for a new KQML Specification, Technical Report CS-97-03, Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, Baltimore.
- Ladd, A., Ward, M. A. (2002): An Investigation of Environmental Factors Influencing Knowledge Transfer  
in: Journal of Knowledge Management Practice, 3(August).
- Laird, J. E., Newell, A. F., Rosenbloom, P. S. (1986): Chunking in Soar - The Anatomy of a General Learning Mechanism, in: Machine Learning, 1(1), S. 11-46.
- Laird, J. E., Newell, A. F., Rosenbloom, P. S. (1987): SOAR - An architecture for general intelligence, Elsevier Science Publishers Ltd.
- Laird, J. E., Rosenbloom, P. S. (1996): The Evolution of the Soar Cognitive Architecture, in: A. F. Newell, D. Steier, T. Mitchell (Hrsg.): Mind Matters: A Tribute to Allen Newell, Erlbaum, Lawrence, S. 59.
- Lang, J., Torre, L. v. d., Weydert, E. (2000): Utilitarian Desires, in: Autonomous Agents and Multi-Agent Systems, 5(3), S. 329-363.
- Lang, J., van der Torre, L., Weydert, E. (2001): Two Kinds of Conflicts Between Desires (and how to resolve them), AAAI Spring Symposium G3: Game theoretic and decision theoretic agents, Stanford, USA.
- Lauf, T., Sure, Y. (2002): Introducing Ontology-based Skills Management at a large Insurance Company Modellierung 2002.
- Laugero, G., Globe, A. (2002): Enterprise Content Services - Connecting Information and Profitability, Addison-Wesley, Boston (MA) et al.
- Law, A. M., Kelton, W. D. (2000): Simulation Modeling and Analysis, 3. Auflage, McGraw Hill, Boston.
- Law, L.-C. (1998a): Bridging the gap between knowledge and action - A situated cognition view, Research Report 92, Lehrstuhl für Empirische Pädagogik und Pädagogische Psychologie der LMU, München.
- Law, L.-C. (1998b): A situated action view of the role of plans and planning in program design and program debugging, Forschungsbericht 86, Institut für Pädagogische Psychologie und Empirische Pädagogik, LMU München, München.
- Lawler, E. J., Thye, S. R. (1999): Bringing Emotions into Social Exchange Theory, in: Annual Review of Sociology, 25, S. 217-244.
- Lebiere, C., Anderson, J. R. (1993): A Connectionist Implementation of the ACT-R Production System, 15th Annual Conference of the Cognitive Science Society, Hillsdale, NJ, S. 635-640.
- Lee, J. (2005): Test Framework Comparison, verfügbar unter: <http://www.theserverside.com/tt/articles/article.tss?l=TestFrameworkComparison>, letzter Abruf: 31.08.2007.

- 
- Lehman, J. F., Laird, J. E., Rosenbloom, P. S. (1996): A gentle introduction to Soar, an architecture for human cognition, in: S. Sternberg, D. Scarborough (Hrsg.): Invitation to Cognitive Science, MIT Press, Cambridge, MA.
- Lehner, F., Hildebrandt, K., Maier, R., Hrsg. (1995): Wirtschaftsinformatik - Theoretische Grundlagen, Hanser, Munich.
- Lesser, V. E. (1999): Cooperative Multiagent Systems - A Personal View of the State of the Art, in: IEEE Transactions on Knowledge and Data Engineering, 11(1).
- Lethbridge, T. C. (1999): The Relevance of Education to Software Practitioners - Data from the 1998 Survey, Technical Report TR-99-06, School of Information Technology and Engineering, University of Ottawa.
- Leuf, B., Cunningham, W. (2001): The Wiki Way - Quick Collaboration on the Web, Addison-Wesley, Boston.
- Levitt, R. E., Thomsen, J., Christiansen, T. R., Kunz, J. C., Jin, Y., Nass, C. (1999): Simulating Project Work Processes and Organizations: Toward a Micro-Contingency Theory of Organizational Design, in: Management Science, 45(11), S. 1479-1495.
- Li, E. Y. (1997): Perceived Importance of Information Systems Success Factors - A Meta Analysis of Group Differences, in: Information & Management, 32, S. 15-28.
- Liang, S., Bracha, G. (1998): Dynamic class loading in the Java virtual machine, in: ACM SIGPLAN Notices, 33(10), S. 36-44.
- Liebl, F. (1995): Simulation - problemorientierte Einführung, 2. Auflage, Oldenbourg, München et al.
- Lin, N. (2001): Building a Network Theory of Social Capital, in: N. Lin, K. Cook, R. S. Burt (Hrsg.): Social capital - theory and research, Aldine de Gruyter, New York, NY, S. 3-29.
- Lindvall, M., Rus, I. (2003): Knowledge Mangement for Software Organizations, in: A. Aurum, R. Jeffery, C. Wohlin, M. Handzic (Hrsg.): Managing Software Engineering Knowledge, Springer, Berlin, S. 73-94.
- Litke, H.-D. (1995): Projektmanagement - Methoden, Techniken, Verhaltensweisen, Carl Hanser, München, Wien.
- Logan, G. D. (1988): Toward an instance theory of automatization. , in: Psychological Review, 95, S. 492-527.
- Lombard, M., Ditton, T. (1997): At the Heart of It All - The Concept of Presence, in: Journal of Computer-Mediated Communication, 3(2), verfügbar unter: <http://jcmc.indiana.edu/vol3/issue2/lombard.html>.
- Looß, M. (2001):
- Loyall, A. B. (1997): Believable Agents - Building Interactive Personalities, PhD, Carnegie Mellon University, Pittsburgh, PA
- Luck, M., d'Inverno, M. (2001): A conceptual framework for agent definition and development, in: The Computer Journal, 44(1), S. 1-20.
- Luger, G. F. (2001): Künstliche Intelligenz - Strategien zur Lösung komplexer Probleme, 4 Auflage, Addison-Wesley, München.
- Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K. (2004): MASON: A New Multi-Agent Simulation Toolkit, SwarmFest Workshop 2004, Ann Arbor, Michigan USA.
- Luna-Reyes, L. F., Cresswell, A. M., Richardson, G. P. (2004): Knowledge and the Development of Interpersonal Trust: a Dynamic Model, HICSS 2004, Big Island, Hawaii.
- Madachy, R. (2005): People Applications in Software Process Modeling and Simulation, 6th International Workshop on Software Process Simulation and Modeling (ProSim 2005), St. Louis, MO.
-

- Madauss, B. J. (2000): Projektmanagement, 6. Auflage, Schäffer-Poeschel, Stuttgart.
- Maier, R. (2002): Knowledge Management Systems - Information and Communication Technologies for Knowledge Management, Springer, Berlin et al.
- Maier, R. (2004): Knowledge Management Systems - Information and Communication Technologies for Knowledge Management, 2 Auflage, Springer, Berlin.
- Maier, R., Hädrich, T. (2001): Modell für die Erfolgsmessung von Wissensmanagementsystemen, in: Wirtschaftsinformatik, 43(5), S. 497-510.
- Maier, R., Hädrich, T., Peinl, R. (2005): Enterprise Knowledge Infrastructures, Springer, Berlin.
- Maier, R., Lehner, F. (1995): Daten, Informationen, Wissen, in: F. Lehner, K. Hildebrandt, R. Maier (Hrsg.): Wirtschaftsinformatik - Theoretische Grundlagen, Hanser, München, S. 165-272.
- Maier, R., Peinl, R. (2005a): Semantische Dokumentbeschreibung in Enterprise Knowledge Infrastructures, in: HMD - Praxis der Wirtschaftsinformatik(246).
- Maier, R., Peinl, R. (2005b): What Is a Knowledge Management Instrument? A Practical Definition and an Approach to Evaluation, Forschungsbericht 3, Martin-Luther-Universität Halle-Wittenberg, Halle / Saale.
- Maier, R., Peinl, R., Holzhammer, U. (2003): Datenmanagement: das organisatorische Fundament für Business Intelligence - Eine empirisch fundierte Neugestaltung der Organisation des Datenmanagements, Arbeitsbericht 1, Martin Luther Universität Halle-Wittenberg, Lehrstuhl für Wirtschaftsinformatik, Halle.
- Maier, R., Remus, U. (2002): Defining Process-oriented Knowledge Management Strategies, in: Knowledge and Process Management, 9(2), S. 103-118.
- Maier, R., Remus, U. (2003): Implementing Process-oriented Knowledge Management Strategies, in: Journal of Knowledge Management, 7(4), S. 62-74.
- Maier, R., Sametinger, J. (2002): Infotop - An Information and Communication Infrastructure for Knowledge Work, 3rd European Conference on Knowledge Management, Trinity College Dublin, Ireland.
- Maier, R., Schmidt, A. (2007): Characterizing Knowledge Maturing - A Conceptual Process Model for Integrating E-Learning and Knowledge Management, 4th Conference Professional Knowledge Management: Experiences and Visions (WM 07), Potsdam.
- Mäki, E., Järvenpää, E., Ziegler, K. (2004): Communication and knowledge sharing in a decentralized organization, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Mandl, H. (2003): Implementation von E-Learning und Wissensmanagement - ein mitarbeiterorientierter Ansatz, Wirtschaftsinformatik 2003, Medien - Märkte - Mobilität, Dresden, S. 35-48.
- Mandl, H., Felix, F., Hron, A. (1988): Theoretische Ansätze zum Wissenserwerb, in: H. Mandl (Hrsg.): Wissenspsychologie, Psychologie-Verlags Union, München-Weinheim, S. Kapitel 6.
- Mangold, R. (1999): Inter-, Intra- und Extranet als Instrumente für das Wissensmanagement, in: C. H. Antoni, T. Sommerlatte (Hrsg.): Spezialreport Wissensmanagement., Symposion Publishing, Düsseldorf, S. 92-96.
- March, J. G., Olsen, J. P. (1975): The Uncertainty of the Past, in: European Journal of Political Research, 3, S. 147-171.
- March, J. G., Simon, H. A. (1958): Organizations, Wiley, New York.
- Margarita, S., Sonnessa, M. (2003): Sim2Web: an Open Source system for web-enabling economic and financial simulations, in: Journal of Artificial Societies and Social Simulation, 6(4), verfügbar unter: <http://jasss.soc.surrey.ac.uk/6/4/12.html>.

- 
- Marietto, M. B., David, N., Sichman, J. S., Coelho, H. (2002): Requirements Analysis of Agent-Based Simulation Platforms: State of the Art and New Prospects, MABS, Bologna, Italy.
- Markovitch, S., Scott, P. D. (1988): The Role of Forgetting in Learning, 5th International Conference on Machine Learning, Ann Arbor, MI.
- Martinez-Miranda, J., Aldea, A. (2002): A Social Agent Model to Simulate Human Behaviour in Teamwork, Workshop 2002: Agent-Based Simulation 3, Passau, S. 18-23.
- McAteer, E., Marsden, S. (2004): Frameworks for the Representation and Analysis of Networked Learning Activity, Networked Learning Conference (NLC2004), Lancaster University, England.
- McEvily, B., Zaheer, A. (1999): Bridging ties - A source of firm heterogeneity in competitive capabilities, in: Strategic Management Journal, 20, S. 1133-1156.
- McKeen, J. D., Staples, D. S. (2003): Knowledge Managers - Who are they and what do they do?, in: C. W. Holsapple (Hrsg.): Handbook on Knowledge Management, Springer, Berlin, S. 21-42.
- McKelvey, B. (1999): Complexity Theory in Organization Science - Seizing the Promise or Becoming a Fad?, in: Emergence, 1(1), S. 5-32, verfügbar unter: [http://scholar.google.com/scholar?hl=en&lr=&q=cache:PohQt6vBVswJ:www.emergence.org/Emergence/Archive/Issue1\\_1/Issue1\\_1\\_1.pdf+McKelvey+%22complexity+theory%22](http://scholar.google.com/scholar?hl=en&lr=&q=cache:PohQt6vBVswJ:www.emergence.org/Emergence/Archive/Issue1_1/Issue1_1_1.pdf+McKelvey+%22complexity+theory%22).
- McKenna, C. J. (1985): Uncertainty and the Labor Market: Recent Developments in Job-Search Theory, Brighton.
- Mendling, J., Neumann, G., Nüttgens, M. (2004): A Comparison of XML Interchange Formats for Business Process Modelling, EMISA 2004 - Information Systems in E-Business and E-Government.
- Meredith, R., Burstein, F. (2000): Getting the Message across with Communicative Knowledge Management, Australian Conference on Knowledge Management and Intelligent Decision Support (ACKMIDS 2000), Monash University, Australia.
- Mertins, K., Heisig, P., Vorbeck, J., Hrsg. (2003): Knowledge management - concepts and best practices, Springer, Berlin et al.
- Meyer, A. (2002): Wer verdient wie viel? Ergebnisse der c't-Gehaltsumfrage, in: c't magazin für computer und technik(06), S. 110-119.
- Meyer, A. (2003): Wer verdient wie viel? Ergebnisse der c't-Gehaltsumfrage, in: c't magazin für computer und technik(06), S. 118-127.
- Meyer, A. (2004): Wer verdient wie viel? Ergebnisse der c't-Gehaltsumfrage 2003, in: c't magazin für computer und technik, 04(6), S. 106-113.
- Meyer, B., Sugiyama, K. (2007): The concept of knowledge in KM - a dimensional model, in: Journal of Knowledge Management, 11(1), S. 17-35.
- Mi, P., Scacchi, W. (1990): A Knowledge-Based Environment for Modeling and Simulating Software Engineering Processes, in: IEEE Transactions on Knowledge and Data Engineering 2(3), S. 283 - 294.
- Mickeler, F., Größler, A., Milling, P. M. (2000): Simulation verschiedener Formen organisatorischer Lernprozesse, Technical Report 2000-1, Universität Mannheim, Mannheim.
- Mieg, H. A. (2001): The social psychology of expertise - case studies in research, professional domains and expert roles, Erlbaum, Mahwah, NJ [u.a.].
- Mishra, A. K. (1996): Organizational responses to crisis - the centrality of trust, in: R. M. Kramer, T. M. Tyler (Hrsg.): Trust in Organizations: Frontiers of Theory and Researc, Sage, Thousand Oaks, CA, S. 261-287.
-

- Misic, V., Gevaert, H., Rennie, M. (2004): Extreme Dynamics - Modeling the Extreme Programming Software Development Process, ProSim 2004.
- Moore, C. R. (1999): Performance Measures for Knowledge Management, in: J. Liebowitz (Hrsg.): Knowledge Management Handbook, CRC Press, Boca Raton et al., S. 6-1 - 6-29.
- Moreño, A., López, C., Sabater, R. (2004): KM strategy and instruments alignment - helping SMEs to choose, OKLC 2004, Innsbruck, Austria.
- Moreno, A., Valls, A., Marín, M. (2003): Multi-agent simulation of work teams, 3rd International/Central and Eastern European Conference on Multi-Agent Systems, CEEMAS, Prague, Czech Republic, S. 281-291.
- Mortensen, D. T. (1988): Matching: Finding a Partner for Life or Otherwise, in: American Journal of Sociology, 94, S. 215-240.
- Moss, S. (2000): Canonical Tasks, Environments and Models for Social Simulation, in: Computational & Mathematical Organization Theory, 6(3), S. 249-275.
- Moss, S., B., E., S., W. (1997): Validation and Verification of Computational Models with Multiple Cognitive Agents, CPM Report 97-25, verfügbar unter: <http://cfpm.org/pub/papers/crit1c.pdf>, letzter Zugriff: 31.08.2007.
- Mui, L. (2003): Computational Models of Trust and Reputation - Agents, Evolutionary Games, and Social Networks, PhD, Massachusetts Institute of Technology, Massachusetts, MA
- Mui, L., Mohtashemi, M., Halberstadt, A. (2002): A Computational Model of Trust and Reputation, HICSS 2002, Big Island, Hawaii.
- Müller, C. (2002a): Simulation sozialer Netzwerke mit neuronalen Netzen.
- Müller, J., Pischel, M. (1993): The Agent Architecture InteRRaP - Concept and Application., Technical Report RR-93-26, DFKI, Saarbrücken, Germany.
- Müller, J. P. (1996): The design of intelligent agents - a layered approach, Springer, Berlin.
- Müller, W. (2002b): Wissensmanagement in der SW-Entwicklung, in: U. Hanning (Hrsg.): Knowledge Management und Business Intelligence, Springer, Berlin, S. 191-201.
- Musselman, K. J. (1998): Guidelines for Success, in: J. Banks (Hrsg.): Handbook of Simulation, Wiley, New York, S. 721-743.
- Musser, D. R., Saini, A. (1995): The STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library, Addison Wesley Longman Publishing Inc., Redwood City, CA.
- Myers, B. L., Kappelman, L. A., Prybutok, V. R. (1998): A Comprehensive Model for Assessing the Quality and Productivity of the Information Systems Function: Toward a Theory for Information Systems Assessment, in: Garrity E. J., Sanders G. L. (Hrsg.): Information Systems Success Measurement, London, S. 94-121.
- Nahm, U., Mooney, R. (2002): Text mining with information extraction, AAAI 2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases, Stanford, CA, S. 60-68.
- Nedeß C., Jacob U. (2000): Das Knowledge Warehouse vor der Gefahr der Komplexitätsfalle, in: Krallmann H. (Hrsg.): Wettbewerbsvorteile durch Wissensmanagement. Methodik und Anwendungen des Knowledge Management, Stuttgart, S. 91-116.
- Neisser, U., Boodoo, G., Bouchard Jr., T. J., Boykin, A. W., Brody, N., Ceci, S. J., Halpern, D. F., Loehlin, J. C., Perloff, R., Sternberg, R. J., et al. (1996): Intelligence: Knowns and unknowns, in: American Psychologist, 51(2), S. 77-101.
- Nerdinger, F. W. (2001): Motivierung, in: H. Schuler (Hrsg.): Lehrbuch der Personalpsychologie, Hogrefe, Göttingen, S. 349-371.

- 
- Neumann, R., Stingl, A., Grillitsch, W. (2002): Best Practices and Lessons Learned in Knowledge Management Projects, 3rd European Conference on Knowledge Management, Dublin, Ireland.
- Neuweg, G. H. (1999): Auf dem Weg zu Experten? Die Phänomenologie des Fertigkeitserwerbs nach Dreyfus/Dreyfus, in: G. H. Neuweg (Hrsg.): Könnerschaft und implizites Wissen, Waxmann, Münster et al., S. 296-316.
- Newell, A. F. (1990): Unified theories of cognition, Harvard University Press.
- Newell, A. F., Simon, H. A. (1972): Human problem solving, Prentice Hall.
- Nilsson, N. (1998): Artificial Intelligence - A New Synthesis, Morgan Kaufmann, San Francisco.
- Nissen, M. E. (1999): Knowledge-Based Knowledge Management in the Reengineering Domain, in: Decision Support Systems 27(1-2), S. 47-65.
- Nissen, M. E. (2002): An extended model of knowledge-flow dynamics, in: Communications of the Association for Information Systems, 8(online), S. 251-266.
- Nissen, M. E., Kamel, M., Sengupta, K. (2000): Integrated Analysis and Design of Knowledge Systems and Processes, in: Information Resources Management Journal, 13(1).
- Nissen, M. E., Levitt, R. E. (2004): Agent-Based Modeling of Knowledge Flows: Illustration from the Domain of Information Systems Design, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Noack, J., Schienmann, B. (1999): Objektorientierte Vorgehensmodelle im Vergleich, in: Informatik-Spektrum, 22(3), S. 166-180.
- Nohr, H. (2000): Wissen und Wissensprozesse visualisieren, in: H. Nohr (Hrsg.): Wissensmanagement, Business Village eBook, Göttingen.
- Nonaka, I. (1994): A Dynamic Theory of Organizational Knowledge Creation, in: Organization Science, 5(1), S. 14-37.
- Nonaka, I., Konno, N. (1998): The Concept of "Ba": Building a Foundation for Knowledge Creation, in: California Management Review, 40(3), S. 40-54, verfügbar unter: <http://www.haas.berkeley.edu/cmvr/>.
- Nonaka, I., Takeuchi, H. (1995): The knowledge-creating company: how Japanese companies create the dynamics of innovation, Oxford University Press, New York.
- Nooteboom, B. (2002): Trust : forms, foundations, functions, failures and figures, Elgar, Cheltenham.
- Norling, E., Sonenberg, L., Rönquist, R. (2001): Enhancing Multi-Agent Based Simulations with Human-Like Decision Making Strategies, in: S. Moss, P. Davidsson (Hrsg.): Multi-Agent-Based Simulation, Second International Workshop, MABS 2000, Springer, Boston, MA, USA, S. 214-228.
- North, M. J., Collier, N. T., Vos, J. R. (2006): Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit, in: ACM Transactions on Modeling and Computer Simulation, 16(1), S. 1-25.
- Nuthall, G. (2000): The Role of Memory in the Acquisition and Retention of Knowledge, in: Cognition and Instruction, 18(1), S. 83-139.
- Nwana, H. S. (1996): Software Agents - An Overview, in: Knowledge Engineering Review, 11(2), S. 205-244.
- O'Dell, C., al., e. (2003): Successful KM Implementations: A Study of Best-Practice Organizations, in: C. W. Holsapple (Hrsg.): Handbook on Knowledge Management, Springer, Berlin, S. 411-441.
-

- O'Dell, C., Grayson, C. J. (1998): If Only We Knew What We Know - Identification and Transfer of Internal Best Practices, in: California Management Review, 40(3), S. 154-174.
- o.V. (2005): ACT-R 6 version 1.0 [r145] - Tutorial.
- o.V. (2006): European ICT Skills Meta-Framework - State-of-the-Art review, clarification of the realities, and recommendations for next steps, European Committee For Standardization, Brussels, verfügbar unter: <ftp://ftp.cenorm.be/PUBLIC/CWAs/e-Europe/ICT-Skill/CWA15515-00-2006-Feb.pdf>.
- Odell, J. J. (2002): Objects and Agents Compared, in: Journal of Object Technology 1(1), S. 41-53.
- Odell, J. J., Parunak, H. V. D., Bauer, B. (2000): Representing Agent Interaction Protocols in UML, First International Workshop in Agent-Oriented Software Engineering - AOSE 2000, Limerick, Ireland, S. 121-140.
- Oliver, G. R., D'Ambra, J., Van Toorn, C. (2003): Evaluating an Approach to Sharing Software Engineering Knowledge to Facilitate Learning, in: A. Aurum, R. Jeffery, C. Wohlin, M. Handzic (Hrsg.): Managing Software Engineering Knowledge, Springer, Berlin, S. 119-134.
- Ong, C.-S., Lai, J.-Y. (2004): Developing an instrument for measuring user satisfaction with knowledge management systems, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Opwis, K., Plötzner, R. (1996): Kognitive Psychologie mit dem Computer : ein Einführungskurs zur Simulation geistiger Leistungen mit Prolog, Spektrum, Heidelberg [u.a.].
- Paauwe, J. (2004): HRM and Performance - Achieving Long Term Viability, Oxford University Press, New York.
- Parker, M. T. (2001): What is Ascape and Why Should You Care?, in: Journal of Artificial Societies and Social Simulation, 4(1), verfügbar unter: <http://www.soc.surrey.ac.uk/JASSS/4/1/5.html>.
- Patel, V. L., Groen, G. J. (1991): The general and specific nature of medical expertise - A critical look, in: K. A. Ericsson, J. Smith (Hrsg.): Toward a general theory of expertise - Prospects and limits, Cambridge University Press, Cambridge, S. 93-125.
- Pavlik, P. I. J., Anderson, J. R. (2005): Practice and Forgetting Effects on Vocabulary Memory - An Activation-Based Model of the Spacing Effect, in: Cognitive Science, 29(4), S. 559-586.
- Pawlowski, P. (1999): Wpzu Wissensmanagement?, in: K. Götz (Hrsg.): Wissensmanagement - Zwischen Wissen und Nichtwissen, Rainer Hampp, München, S. 113-129.
- Peinl, R. (2006): A Knowledge Sharing Model illustrated with the Software Development Industry, Multikonferenz Wirtschaftsinformatik (MKWI 2006), Passau, Germany, S. 389-401.
- Petre, M., Blackwell, A. F. (1997): A Glimpse of expert Programmers' Mental Imagery, 7th Workshop on Empirical studies of programmers, Alexandria, VA, S. 109 - 123.
- Pfahl, D., Laitenberger, O., Dorsch, J., Ruhe, G. (2003): An Externally Replicated Experiment for Evaluating the Learning Effectiveness of Using Simulations in Software Project Management Education, in: Empirical Software Engineering, 8(4), S. 367-395.
- Piaget, J. (1971): Science of education and the psychology of the child, Viking Press, New York.

- 
- Pidd, M. (2004): Computer simulation in Management Science, 5. Auflage, Wiley, Chichester, England.
- Pitt, L. F., Watson, R. T., Kavan, C. B. (1995): Service Quality - A Measure of Information Systems Effectiveness, in: MIS Quarterly, 19(2 (June)), S. 173-187.
- Polanyi, M. (1966): The tacit dimension, Routledge & Kegan Paul Ltd., London.
- Probst, G. J. B., Raub, S., Romhardt, K. (1998): Wissen managen - Wie Unternehmen ihre wertvollste Ressource optimal nutzen., 2. Auflage, Gabler, Wiesbaden.
- Ramesh, B., Tiwana, A. (1999): Supporting Collaborative Process Knowledge Management in New Product Development Teams, in: Decision Support Systems, 27(1), S. 213-235.
- Rao, A. S., Georgeff, M. P. (1991): Modelling Rational Agents within a BDI-Architecture Technical Note 14, Australian Artificial Intelligence Institute.
- Rao, A. S., Georgeff, M. P. (1995): BDI Agents: From Theory to Practice Technical Note 56, Australian Artificial Intelligence Institute.
- Rational (1999): Rational Unified Process - Best Practices for Software Development Teams, White Paper online [http://www.augustana.ab.ca/~mohrj/courses/2000.winter/csc220/papers/rup\\_best\\_practices/rup\\_bestpractices.pdf](http://www.augustana.ab.ca/~mohrj/courses/2000.winter/csc220/papers/rup_best_practices/rup_bestpractices.pdf), A Rational Software Corporation.
- Reichwald, R. (1993): Kommunikation und Kommunikationsmodelle, in: W. Wittmann, W. Kern (Hrsg.): Enzyklopädie der Betriebswirtschaftslehre - Bd. 1: Handwörterbuch der Betriebswirtschaft, Stuttgart, S. 2174-2188.
- Reiff, R. (1974): The Control of Knowledge - The Power of the Helping Professions, in: The Journal of Applied Behavioral Science, 10(3), S. 451-461.
- Reilly, W. S. (1996): Believable Social and Emotional Agents, PhD, Carnegie Mellon University, Pittsburgh, PA
- Reinhardt, R. (2002): Wissen als Ressource - theoretische Grundlagen, Methoden und Instrumente zur Erfassung von Wissen, Lang, Frankfurt a. M. et al.
- Reinhardt, R. (2004): Improving organizational performance by a knowledge related measurement- and monitoring-system, 5th European Conference on Organizational Knowledge, Learning and Capabilities (OKLC 2004), Innsbruck, Austria.
- Reinmann, G., Mandl, H. (2004): Psychologie des Wissensmanagements : Perspektiven, Theorien und Methoden, Hogrefe, Göttingen.
- Remus, U. (2002): Prozessorientiertes Wissensmanagement - Konzepte und Modellierung, Dissertation, Universität Regensburg, Regensburg
- Reuter, C. (2004): Instrumente des Wissensmanagement - Definition, Klassifikation und Einsatz im Unternehmen anhand einer Fallstudienanalyse, Diplomarbeit, Martin-Luther-Universität Halle-Wittenberg, Halle
- Rich, E. H. (2002): Modeling The Dynamics Of Organizational Knowledge, PhD, University at Albany, State University of New York, New York
- Rich, E. H., Duchessi, P. (2004): Modeling the Sustainability of Knowledge Management Programs, Modeling the Sustainability of Knowledge Management Programs (HICSS 2004), Big Island, Hawaii.
- Rieger, B., Kleber, A., Maur, E. v. (2000): Metadata-Based Integration of Qualitative and Quantitative Information Resources, 8th European Conference on Information Systems (ECIS 2000), Vienna, Austria, S. 372-378.
- Riempp, G. (2003): Eine Architektur für integriertes Wissensmanagement, Wirtschaftsinformatik 2003, Medien - Märkte - Mobilität, Dresden, Germany, S. 255-276.
- Riempp, G. (2005): Integriertes Wissensmanagement - Strategie, Prozesse und Systeme wirkungsvoll verbinden, in: HMD - Praxis der Wirtschaftsinformatik(246), S. 6-19.
-

- Ripperger, T. (2003): *Ökonomik des Vertrauens*, 2. Auflage, Mohr Siebeck, Tübingen.
- Roberts, C. A., Dessouky, Y. M. (1998): An Overview of Object-Oriented Simulation, in: *SIMULATION*, 70(6), S. 359-368.
- Robillard, P. N. (1999): The Role of Knowledge in Software Development, in: *Communications of the ACM*, 42(1), S. 87-92.
- Roehl, H. (2000): *Instrumente der Wissensorganisation - Perspektiven für eine differenzierende Interventionspraxis*, Wiesbaden.
- Romhardt, K. (1996): Interventionen in die organisatorische Wissensbasis zwischen Theorie und Praxis online <http://know.unige.ch/publications/system.pdf>, Universität Genua, Genua, verfügbar unter: <http://know.unige.ch/publications/system.pdf>.
- Ross, S. M. (2002): *Simulation*, 3. Auflage, Academic Press, San Diego, CA.
- Rubinstein, A. (2002): *Modeling bounded rationality*, 3. print Auflage, MIT Press, Cambridge, Mass. [u.a.].
- Russell, S. J., Norvig, P. (2003): *Artificial intelligence - a modern approach*, 2. ed., internat. ed Auflage, Prentice Hall, Upper Saddle River, NJ [u.a.].
- Sargent, R. (1988): A Tutorial on Validation and Verification of Simulation Models, 1988 Winter Simulation Conference, S. 33-39.
- Sarker, S., Sarker, S., Nicholson, D., Joshi, K. D. (2003): Knowledge Transfer in Virtual Information Systems Development Teams: an Empirical Examination of Key Enablers, 36th Hawaii International Conference on System Sciences (HICSS 2003), Big Island, Hawaii.
- Scheutz, M., Sloman, A., Logan, B. (2000): Emotional States and Realistic Agent Behaviour, 1st International Conference on Intelligent Games and Simulation (GAME-ON 2000), London, UK.
- Schindler, M. (2001): *Wissensmanagement in der Projektabwicklung*, 2. Auflage, Josef Eul, Köln.
- Schnauffer, H.-G., Stieler-Lorenz, B., Peters, S., Hrsg. (2004): *Wissen vernetzen - Wissensmanagement in der Produktentwicklung*, Springer, Berlin.
- Schoder, D., Eymann, T. (2000): The Real Challenges of Mobile Agents, in: *Communications of the ACM* 43(6), S. 111-112.
- Scholz, C. (1999): IT-Gehälter für Absolventen der BWL, Informatik und Wirtschaftsinformatik – Ergebnisse einer explorativen Studie<sup>71</sup>, Universität des Saarlandes, Lehrstuhl für BWL, insb. Organisation, Personal- und Informationsmanagement, Saarbrücken.
- Scholz, C. (2000): Personalarbeit im IT-Bereich - Erfolgskritische Aktionsfelder, in: *Wirtschaftsinformatik*, 42(Sonderheft), S. 14-23.
- Schultze, U. (2003): On Knowledge Work, in: C. W. Holsapple (Hrsg.): *Handbook on Knowledge Management*, Springer, Berlin, S. 43-58.
- Schüppel, J. (1997): *Wissensmanagement - organisatorisches Lernen im Spannungsfeld von Wissens- und Lernbarrieren*, Nachdr. der 1. Aufl. 1996 Auflage, Dt. Univ.-Verl., Wiesbaden.
- Schwaiger, A., Stahmer, B. (2003): SimMarket - Multiagent-Based Customer Simulation and Decision, 1st German Conference on Multiagent System Technologies (MATES), Erfurt, Germany, S. 74-84.
- Schwarz, N. (2000): Emotion, cognition, and decision making, in: *Cognition and Emotion*, 14(4), S. 433-440.
- Searle, J. R. (1969): *Speech acts - an essay in the philosophy of language*, Cambridge University Press, Cambridge et al.
- Searle, J. R. (1992): *Sprechakte - Ein sprachphilosophischer Essay*, 5. Auflage Auflage, Cambridge Univ. Press, Cambridge.

- 
- Seddon, P. B. (1997): A Respecification and Extension of the DeLone and McLean Model of IS Success, in: *Information Systems Research*, 8(3), S. 240-253, verfügbar unter: <http://isr.commerce.ubc.ca/>.
- Seiwert, L. J. (1992): Kommunikation im Betrieb, in: E. Gaugler, W. Weber (Hrsg.): *Enzyklopädie der Betriebswirtschaftslehre*, Stuttgart, S. 1126-1139.
- Shannon, C. E., Weaver, W. (1969): *The mathematical theory of communication*, Univ. of Illinois Press, Urbana.
- Short, J., Williams, E., & Christie, B. (1976): *The social psychology of telecommunications*, Wiley, London.
- Shostak, I., Anjewierden, A., de Hoog, R. (2002): Modelling and Simulating Process-Oriented Knowledge Management, 3rd European Conference on Knowledge Management (ECKM), Dublin, Ireland, S. 634-648.
- Sierhuis, M. (2001): Modeling and simulating work practice - BRAHMS.
- Sierhuis, M., Clancey, W. J. (1997): Knowledge, Practice, Activities and People, AAAI Spring Symposium on Artificial Intelligence in Knowledge Management, Stanford, S. 142-148.
- Sierhuis, M., Clancey, W. J. (2002): Modeling and Simulating Work Practice: A Method for Work Systems Design, in: *IEEE Intelligent Systems*, Special issue on Human-Centered Computing at NASA, 17(5), S. 32-41.
- Sierhuis, M., Clancey, W. J., van der Hoof, R. (2002): Brahms - A multiagent modeling environment for simulating work practice in organizations., in: *Journal for Simulation Modelling Practice and Theory*, 10(5-7, Special Issue on "Organisational Processes"), S. 253-269.
- Sierhuis, M., Clancey, W. J., Van Hoof, R., De Hoog, R. (2000): Modeling and Simulating Human Activity, AAAI Fall Symposium on Simulating Human Agents, North Falmouth, MA.
- Simon, H. A. (1947): *Administrative Behaviour*, Macmillan, New York.
- Simon, H. A. (1955): A Behavioral Model of Rational Choice, in: *The Quarterly Journal of Economics*, 69(1), S. 99-118.
- Simon, H. A. (1997a): *Administrative Behaviour*, Free Press.
- Simon, H. A. (1997b): *Models of bounded rationality - Vol 3*, MIT Press, Cambridge, Mass. [u.a.].
- Singh, M. P. (1998): A Semantics for Speech Acts: *Annals of Mathematics and Artificial Intelligence*, S. 47-71.
- Singley, M. K., Anderson, J. R. (1989): *The Transfer of Cognitive Skill*, Harvard University Press, Cambridge, MA.
- Skyrme, D., Amidon, D. (1997): The Knowledge Agenda, in: *The Journal of Knowledge Management*, 1(1), S. 27-37, verfügbar unter: <http://www.mcb.co.uk/jkm.htm/>.
- Sloman, A. (2000): Models of Models of Mind, *The DAM Symposium - How to Design a Functioning Mind*, Birmingham, UK.
- Sloman, A. (2001): Beyond Shallow Models of Emotion, in: *Cognitive Processing*, 2(1), S. 177-198.
- Sloman, A., Beaudoin, L., Wright, I. (1994): *Computational Modelling of Motive-Management Processes*, ISRE Conference, Cambridge, MA.
- Sloman, A., Chrisley, R., Scheutz, M. (2004): The Architectural Basis of Affective States and Processes, in: Fellous, Arbib (Hrsg.): *Who Needs Emotions? The Brain Meets the Machine*, Oxford University Press, Oxford, UK.
- Sloman, A., Logan, B. (1999): Building cognitively rich agents using the SIM\_AGENT toolkit, in: *Communications of the Association of Computing Machinery*, 43(2), S. 71-77.
-

- Sloman, A., Logan, B. (2000): Evolvable Architectures for Human-like Minds, 13th Toyota Conference, on "Affective Minds", Nagoya, Japan.
- Sloman, A., Scheutz, M. (2002): A Framework for Comparing Agent Architectures, UK Workshop on Computational Intelligence (UKCI02), Birmingham UK.
- Smits, M., de Moor, A. (2004): Measuring Knowledge Management Effectiveness in Communities of Practice, 37th Hawaii International Conference on System Science (HICSS 2004), Big Island, Hawaii.
- Song, M., van der Bij, H., Weggeman, M. (2003): An empirical investigation into the antecedents of knowledge dissemination at the strategic business unit level, in: Journal of Product Innovation Management, 20(2), S. 163-179.
- Sonne, A., Harmsen, H. (2005): Knowledge sharing in international product development teams, 18th Scandinavian Academy of Management meeting, Aarhus, Denmark.
- Sonntag, K., Schaper, N. (2001): Wissensorientierte Verfahren der Personalentwicklung, in: H. Schuler (Hrsg.): Lehrbuch der Personalpsychologie, Hogrefe, Göttingen, S. 241-263.
- Sosnoski, D. (2003): Java programming dynamics, Part 2: Introducing reflection, verfügbar unter: <http://www.ibm.com/developerworks/library/j-dyn0603/>, letzter Abruf: 25.08.2007.
- Staab, S., Studer, R., Sure, Y. (2003): Knowledge Processes and Meta Processes in Ontology-based Knowledge Management, in: C. W. Holsapple (Hrsg.): Handbook on Knowledge Management, Springer, S. 47-68.
- Stachle, W. H. (1999): Management - eine verhaltenswissenschaftliche Perspektive, 8 Auflage, Vahlen Verlag, München.
- Staller, A., Petta, P. (2001): Introducing Emotions into the Computational Study of Social Norms - A First Evaluation, in: Journal of Artificial Societies and Social Simulation, 4(1), verfügbar unter: <http://www.soc.surrey.ac.uk/JASSS/4/1/2.html>.
- Stasser, G., Vaughan, S. I., Stewart, D. D. (2000): Pooling unshared information - The benefits of knowing how access to information is distributed among group members, in: Organizational Behavior and Human Decision Processes, 82(1), S. 102-116.
- Staw, B. (1975): Attribution of the "Causes" of Performance - A General Alternative Interpretation of Cross-Sectional Research on Organizations, in: Organizational Behaviour and Human Performance, 13, S. 414-32.
- Steels, L. (2005): The Emergence and Evolution of Linguistic Structure: From Lexical to Grammatical Communication Systems, in: Connection Science, 17, S. 213-230.
- Stenmark, D. (2002): Information vs. Knowledge - The Role of Intranets in Knowledge Management, 35th Hawaii International Conference on System Sciences (HICSS 2002), Big Island, Hawaii.
- Stewart, T. A. (1997): Intellectual capital - the new wealth of organizations, 1st Auflage, Doubleday / Currency, New York.
- Stieler-Lorenz, B., Paarmann, Y. (2004): Wissenskommunikation und Lernen in Organisationen, in: M. J. Eppler, R. Reinhardt (Hrsg.): Wissenskommunikation in Organisationen - Methoden, Instrumente, Theorien, Springer, Berlin, S. 177-197.
- Strehlitz, M. (2005): Contentvereinigung überfordert Anwender, in: Computer Zeitung, 35(41), S. 9, verfügbar unter: [http://www.netigator.de/netigator/live/fachartikelarchiv/ha\\_artikel/powerslave.id,30532749.obj,np.ng.,thes.,html](http://www.netigator.de/netigator/live/fachartikelarchiv/ha_artikel/powerslave.id,30532749.obj,np.ng.,thes.,html).
- Strohmaier, M. B. (2005): B-KIDE: A Framework and a Tool for Business Process Oriented Knowledge Infrastructure Development, Dissertation, Graz University of Technology, Graz, Austria

- 
- Sturdy, A. (2003): Knowing the Unknowable? A Discussion of Methodological and Theoretical Issues in Emotion Research and Organizational Studies, in: *Organization*, 10(1), S. 81-105.
- Sveiby, K. E. (1997): *The new organizational wealth - managing & measuring knowledge-based assets*, Berrett-Koehler, San Francisco, California.
- Sveiby, K. E., Lloyd, T. (1987): *Managing Knowhow*, London.
- Sycara, K., Paolucci, M., van Velsen, M., Giampapa, J. (2003): The RETSINA MAS Infrastructure, in: *Autonomous Agents and Multi-Agent Systems*, 7, S. 29-48.
- Szierpinski, E. (2007 (in Vorbereitung)): *Kompetenzmanagement*, PhD, Universität Karlsruhe, Karlsruhe
- Szulanski, G. (1996): Exploring internal stickiness - Impediments to the transfer of best practice within the firm, in: *Strategic Management Journal*, 17(Winter Special Issue), S. 27-43.
- Szulanski, G. (2000): The process of knowledge transfer - a diachronic analysis of stickiness, in: *Organizational Behavior and Human Decision Processes*, 82(1), S. 9-27.
- Terán, O. (2004): *Understanding MABS and Social Simulation - Switching Between Languages in a Hierarchy of Levels*, in: *Journal of Artificial Societies and Social Simulation*, 7(4), S. <http://jasss.soc.surrey.ac.uk/7/4/5.html>.
- Teubner, A., Nietsch, M. (2000): *Managing Knowledge in Medium-Sized Software Companies*, 8th European Conference on Information Systems (ECIS 2000), Vienna, Austria, S. 917-925.
- Thiesse, F. (2001): *Prozessorientiertes Wissensmanagement - Konzepte, Methode, Fallbeispiele* Dissertation, Universität St. Gallen, St. Gallen
- Thomas-Hunt, M. C., Ogden, T. Y., Neale, M. A. (2003): Who's really sharing? Effects of social and expert status on knowledge exchange within groups, in: *Management Science*, 49(4), S. 464.
- Thompson, M. P. A. (2004): Some Proposals for Strengthening Organizational Activity Theory, in: *Organization*, 11(5), S. 579-602.
- Timm, I. J., Scholz, T., Knublauch, H. (2006): *The Engineering Process*, in: S. Kirn, O. Herzog, P. Lockemann, O. Spaniol (Hrsg.): *Multiagent Engineering*, Springer, Berlin, S. 341-358.
- Tobias, R., Hofmann, C. (2004): Evaluation of free Java-libraries for social-scientific agent based simulation, in: *Journal of Artificial Societies and Social Simulation*, 7(1), verfügbar unter: <http://jasss.soc.surrey.ac.uk/7/1/6.html>.
- Trittmann, R., Brössler, P. (2000): *Effizienter Wissenstransfer in der Softwareentwicklung: Der sd&m Ansatz*, in: M. J. Eppler, O. Sukowski (Hrsg.): *Fallstudien zum Wissensmanagement. Lösungen aus der Praxis*, NetAcademy Press, St. Gallen, S. 163-188.
- Troitzsch, K. G. (1990): *Modellbildung und Simulation in den Sozialwissenschaften*, Westdt. Verl., Opladen.
- Troitzsch, K. G. (1997): *Social Science Simulation - Origins, Prospects, Purposes*, in: R. Conte, R. Hegselmann, P. Terna (Hrsg.): *Simulating Social Phenomena*, Springer, Berlin, S. 41-54.
- Tsai, W. (2001): Knowledge transfer in intraorganizational networks - Effects of network position and absorptive capacity on business unit innovation and performance, in: *Academy of Management Journal*, 44(5), S. 996.
- Tsai, W. (2002): Social structure of "coopetition" within a multiunit organization - Coordination, competition, and intraorganizational knowledge sharing, in: *Organization Science*, 13(2), S. 179-190.
-

- Tsui, E. (2003): Tracking the role and evolution of commercial knowledge management software, in: C. W. Holsapple (Hrsg.): Handbook on Knowledge Management, Springer, Berlin, S. 5-27.
- Tu, C.-H. (2002): The Measurement of Social Presence in an Online Learning Environment, in: International Journal on E-Learning, S. 34-45.
- Turner, S. F., Bettis, R. A., Burton, R. M. (2002): Exploring Depth Versus Breadth in Knowledge Management Strategies, in: Computational & Mathematical Organization Theory, 8(1), S. 49 - 73.
- Tversky, A., Kahnemann, D. (1986): Rational Choice and the Framing of Decisions, in: Journal of Business, 59, S. 251-278.
- Tynjälä, P. (1999): Towards expert knowledge? A comparison between a constructivist and a traditional learning environment in the university, in: International Journal of Educational Research, 31(5), S. 357-442.
- Ullenboom, C. (2005): Java ist auch eine Insel - Programmieren für die Java 2-Plattform in der Version 5, 5. Auflage, Galileo Computing, Bonn.
- Urban, C. (1997): Agenten-basierte Simulation eines Lotka-Volterra-Systems mit Hilfe von Simplex II, 7. Treffen des Arbeitskreises Werkzeuge für Modellbildung und Simulation in Umweltsystemen, Karlsruhe, Germany.
- Urban, C. (2000): PECS - A Reference Model for the Simulation of Multi-Agent Systems, in: R. Suleiman, K. G. Troitzsch, G. N. Gilbert (Hrsg.): Tools and Techniques for Social Science Simulation., Physica-Verlag, Heidelberg, New York, S. 83-114.
- Urban, C., Schmidt, B. (2001): PECS – Agent-Based Modelling of Human Behaviour, Emotional and Intelligent II - The Tangled Knot of Social Cognition, AAAI Fall Symposium 2001, North Falmouth, MA.
- van der Spek, R., Hofer-Alfeis, J., Kingma, J. (2003): The Knowledge Strategy Process, in: C. W. Holsapple (Hrsg.): Handbook on Knowledge Management, Springer, Berlin, S. 443-466.
- Vassiliadis, S., Wicki, Y., Seufert, A., Back, A., von Krogh, G. (1999): Knowledge Networks: Linking Knowledge Management to Business Strategy  
BE HSG / 11, Research Center KnowledgeSource, University of St. Gallen, St. Gallen.
- Verhagen, H. (1998): ACTS in action - Sim-ACTS - a simulation model based on ACTS theory, MABS.
- Verhagen, H. J. E. (2000): Norm Autonomous Agents, The Royal Institute of Technology and Stockholm University, Stockholm, Sweden
- Vogel, M. (2005): Die virtuelle Fabrikhalle wird zur Spielwiese, in: Computer Zeitung, 35(39), S. 16.
- Volken, T. (2002): Elements of Trust - The Cultural Dimension of Internet Diffusion Revisited, in: Electronic Journal of Sociology, 6(4), verfügbar unter: <http://www.sociology.org/content/vol006.004/volken.html>.
- Voß, A. (2004): Motivierte Wahrnehmung Selektive Aufmerksamkeit und entlastende Umdeutungen bei der Aufnahme valenter Informationen, PhD, Universität Trier, Trier
- Wah, Y. C., Menkhoff, T., Loh, B., Evers, H.-D. (2005): Theorizing, Measuring, and Predicting Knowledge Sharing Behavior in Organization - A Social Capital Approach, 38th Hawaii International Conference on System Science (HICSS 2005), Big Island, Hawaii.
- Walker, G., MacDonald, J. R. (2001): Designing and Implementing an HR Scorecard, in: Human Resource Management, 40(4), S. 365-377.

- 
- Waltert, J. (2002): Elektronische Kommunikationsforen als Elemente des Wissensmanagement - Eine Fallstudie bei genossenschaftlichen Banken, Dissertation, Universität Konstanz, Konstanz
- Walz, D. B., Elam, J. J., Curtis, B. (1993): Inside a Software Design Team - Knowledge Acquisition, Sharing and Integration, in: Communications of the ACM, 36(10), S. 63-77.
- Warne, L. (1997): Conflict as a Factor in Information Systems Failure, The 8th Australasian Conference on Information Systems (ACIS 97), Adelaide, South Australia, S. 387-391.
- Warne, L. (2003): Conflict and Politics and Information Systems Failure: A Challenge for Information Systems Professionals and Researchers, in: S. Clarke, E. Coakes, M. G. Hunter, A. Wenn (Hrsg.): Socio-Technical and Human Cognition Elements of Information Systems, Information Science Publisher, Hershey et al., S. 104-134.
- Watson, J. A., Keats, D., Yoong, P. (2005): Online knowledge sharing and media selection in a community organisation - An application of the Theory of Media Synchronicity  
16th Australasian Conference on Information Systems, Sydney, Australia.
- Watson, R. (2003): Data Management.
- Watzlawick, P., Beavin, J. H., Jackson, D. D. (2000): Menschliche Kommunikation - Formen, Störungen, Paradoxien, 10., unveränd. Auflage, Huber, Bern.
- Wegge, J. (2001): Gruppenarbeit, in: H. Schuler (Hrsg.): Lehrbuch der Personalpsychologie, Hogrefe, Göttingen, S. 383-507.
- Weggeman, M. (1999): Wissensmanagement - der richtige Umgang mit der wichtigsten Ressource des Unternehmens, MIT Press, Bonn
- Weinert, A. B. (1998): Organisationspsychologie - Ein Lehrbuch, 4. Auflage, Psychologische Verlags Union, Weinheim.
- Weinert, E. F., Waldmann, M. R. (1988): Wissensentwicklung und Wissenserwerb, in: H. Mandl (Hrsg.): Wissenspsychologie, Psychologie-Verlags Union, München-Weinheim, S. Kapitel7.
- Weise, P., Brandes, W., Eger, T., Kraft, M. (2002): Neue Mikroökonomie, 4. Auflage, Physica-Verlag, Heidelberg.
- Weiß, G., Jakob, R. (2005): Agentenorientierte Softwareentwicklung, Springer, Berlin.
- Wendelken, C., Shastri, L. (2002): Combining belief and utility in a structured connectionist agent architecture, 24th Annual Conference of the Cognitive Science Society (COGSCI 2002), Fairfax, VA.
- Wenger, E. (2002): Communities of practice - learning, meaning, and identity, repr. Auflage, Cambridge Univ. Press, Cambridge et al.
- Wenger, E. C. (2000): Communities of Practice and Social Learning Systems, in: Organization, 7(2), S. 225-246.
- Wenger, E. C., Snyder, W. M. (2000): Communities of Practice - The organisational frontier  
in: Harvard Business Review, S. 139-145.
- Wexler, M. N. (2001): The who what and why of knowledge mapping, in: Journal of Knowledge Management, 5(3), S. 249-263.
- Wickenberg, T., Davidsson, P. (2002): On Multi-Agent-Based Simulation of Software Development Processes, MABS 2002, Bologna, Italy, S. 171-180.
- Wiig, K. M. (1993): Knowledge management foundations - thinking about thinking : how people and organizations create, represent and use knowledge, Schema Press, Arlington, Tex.
-

- Wiig, K. M. (1997): Integrating Intellectual Capital and Knowledge Management, in: Long Range Planning, 30(3), S. 399-405.
- Wiig, K. M. (2003): A knowledge model for situation-handling, in: Journal of Knowledge Management, 7(5), S. 6-24.
- Wilkening, F. (1988): Zur Rolle des Wissens in der Wahrnehmung, in: H. Mandl (Hrsg.): Wissenspsychologie, Psychologie-Verlags-Union, München-Weinheim, S. 203-224.
- Willke, H. (2001): Systemisches Wissensmanagement, Lucius & Lucius, Stuttgart.
- Wilson, T. D. (2002): The nonsense of 'knowledge management', in: Information Research, 8(1), S. 144, verfügbar unter: <http://InformationR.net/ir/8-1/paper144.htm>
- Winkler, K., Mandl, H. (2003): Wissenschaftsmanagement in Communities : Communities als zentrales Szenario der Weiterbildungslandschaft im dritten Jahrtausend, Inst. für Pädag. Psychologie und Empirische Pädag. Lehrstuhl Prof. Dr. Heinz Mandl, München.
- Winograd, T., Flores, F. (2000): Understanding computers and cognition - a new foundation for design, 15th printing Auflage, Addison-Wesley, Boston [u.a.].
- Wong, R. M., Dalmadge, C. (2004): Media Choice for Complex and Knowledge-Intensive Processes, 37th Hawaii International Conference on System Sciences (HICSS 2004), Big Island, Hawaii.
- Wooldridge, M., Jennings, N. R. (1995): Intelligent Agents - Theory and Practice, in: The Knowledge Engineering Review, 10(2).
- Wooldridge, M. J. (2002): An introduction to multiagent systems, Wiley, Chichester.
- Wooldridge, M. J., Jennings, N. R. (1994): Agent Theories, Architectures and Languages - A Survey, ECAI94 Workshop on Agent Theories Architectures and Languages, Amsterdam, The Netherlands, S. 1-32.
- Wrobel, S. (1998): Data Mining und Wissensentdeckung in Datenbanken, in: KI - Künstliche Intelligenz, 12(1), S. 6-10.
- Xian, W., Lei, Z., Yong, Y. (2006): Exploring Social Annotations for the Semantic Web, Proceedings of the 15th international conference on World Wide Web, Edinburgh, Scotland, S. 417 - 426.
- Yahya, S., Goh, W.-K. (2002): Managing human resources toward achieving knowledge management, in: Journal of Knowledge Management, 6(5), S. 457-468.
- Young, R. M., Lewis, R. L. (1999): The Soar Cognitive Architecture and Human Working Memory, in: A. Miyake, P. Shah (Hrsg.): Models of Working Memory: Mechanisms of Active Maintenance and Executive Control, Cambridge University Press, New York.
- Zack, M. H. (1999): Managing Codified Knowledge, in: Sloan Management Review, 40(4), S. 45-58.
- Zafirovski, M. (2003): Some Amendments to Social Exchange Theory: A Sociological Perspective, in: Theory & Science, 4(2), verfügbar unter: [http://theoryandscience.icaap.org/content/vol004.002/01\\_zafirovski.html](http://theoryandscience.icaap.org/content/vol004.002/01_zafirovski.html).
- Zafirovski, M. (2005): Social Exchange Theory under Scrutiny - A Positive Critique of its Economic-Behaviorist Formulations, in: Electronic Journal of Sociology, 8, Tier Two, verfügbar unter: <http://www.sociology.org/content/2005/tier2/SETheory.pdf>.
- Zelewski, S., Siedentopf, J. (1999):
- Zucker, L. G. (1987): Institutional Theories of Organization, in: Annual Review of Sociology, 13, S. 443-464.

## Anhang

### Anhang A: Einflussfaktoren für Wissensweitergabe nach Quelle

#### Explorative work

source	research object	identified factors
(Cross et al. 2000)	40 managers, 38 consultants	knowing what others know, an appropriate access to them, people willing to actively engage in problem solving, a safe relationship to ask important questions
(Handzic et al. 2004)	tertiary education and research institution	organized meetings, atmosphere of openness, friendliness and trust, engender confidence, promote a sense of community, facilitate interaction between employees, IT providing connectivity and supporting collaboration

#### Investigative work

source	influencing factors	research object	confirmed factors
(Szulanski 1996; Szulanski 2000)	<i>identification of relevant knowledge (+), unknown value of knowledge, lacking motivation, lacking confidence in the source or recipient, absorptive capacity (+), retentive capacity, organizational context, strained social relation between source and recipient (-)</i>	8 companies, e.g., AT&T, Chevron, Rank Xerox	identification of relevant knowledge, absorptive capacity, social relation
(Cummings, Teng 2003)	<i>articulability, embeddedness, physical distance, knowledge distance, norm distance, learning culture, project priority, organizational distance, transfer activities</i>	R&D in 15 industries	articulability, embeddedness, norm and knowledge distance, transfer activities

<b>source</b>	<b>influencing factors</b>	<b>research object</b>	<b>confirmed factors</b>
(Ensign, Hébert 2004)	<i>reputation of recipient and her group</i> , (nature, duration and number of past interactions) + (predictability, reciprocity and obligations)	R&D in the pharmaceutical industry	reputation of recipient and her group
(Sarker et al. 2003; Joshi et al. 2004; Joshi et al. 2005)	<i>communication</i> , credibility, competence, culture	IS students	communication is strongly confirmed, the other factors partly
(Boer et al. 2002b; Boer, Berends 2003)	<i>relation model</i> (market exchange, authority ranking, communal sharing, equality matching), culture, type of knowledge, technology, activity, division of labor	theoretically derived, industrial R&D	all relation models, with market exchange being least important
(Song et al. 2003)	co-location, information technologies, <i>lead user and supplier networks</i> (+), <i>individual commitment</i> (+), formal rewards, R&D budget, <i>long-term orientation</i> (+), <i>organizational redundancy</i> (-), <i>organizational crisis</i> (+), risk taking behavior, management support for integration	277 US high-technology firms	lead user and supplier networks, individual commitment, long-term orientation, and organizational crisis, organizational redundancy

<b>source</b>	<b>influencing factors</b>	<b>research object</b>	<b>confirmed factors</b>
(Wah et al. 2005)	organizational concern, <i>reward &amp; recognition (+)</i> , impression management, competence, <i>open mindedness (+)</i> , prosocial motives, personal compatibility, <i>expected costs of hoarding knowledge (-)</i> , benefits of knowledge sharing, <i>expected costs of knowledge sharing (-)</i>	tertiary educational institution in Singapore	reward & recognition, open mindedness and cost concerns with regard to both knowledge hoarding and sharing are the strongest predictors
(Hendriks 1999)	<i>sense of achievement (+)</i> , <i>sense of responsibility (+)</i> , <i>recognition of job done (+)</i> , operational autonomy, <i>promotional opportunities (+)</i> , challenge of work	two ICT companies and one management consultancy firm,	sense of achievement, sense of responsibility, recognition of job done, promotional opportunities,
(Thomas-Hunt et al. 2003)	<i>social status (+)</i> , <i>expert status (+)</i>	undergraduate students in engineering or business at a private US university	social status, expert status
(Ladd, Ward 2002)	<i>culture of: openness to change and innovation (+)</i> , <i>task-orientation (+)</i> , bureaucracy, <i>competition and confrontation (-)</i>	23 organizations from the US Air Force	culture of openness to change and innovation, task-orientation, competition and confrontation

Literature reviews

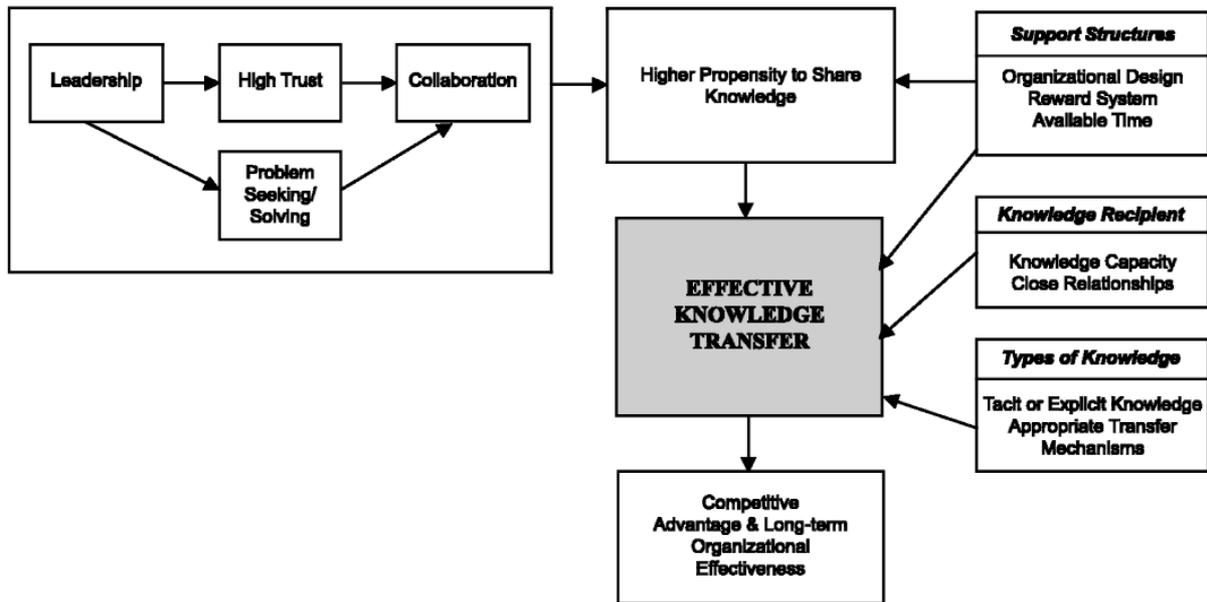


Abbildung A-1: Einflüsse für Wissensweitergabe (Goh 2002, S. 28)

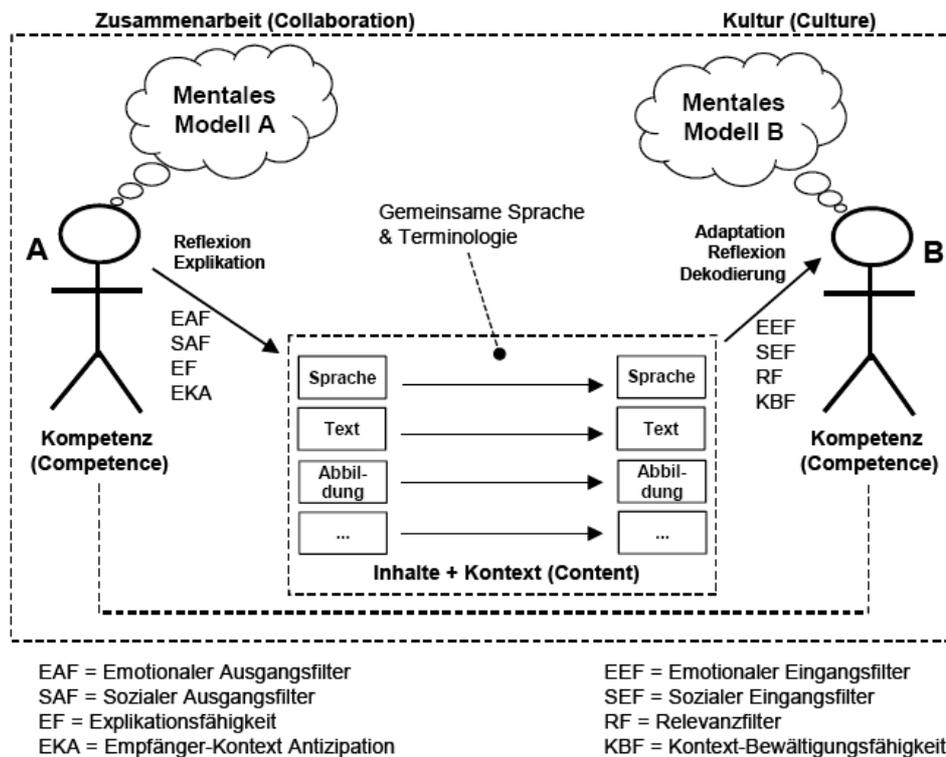


Abbildung A-2: Modell der Wissensweitergabe (Riempp 2003, S. 6)

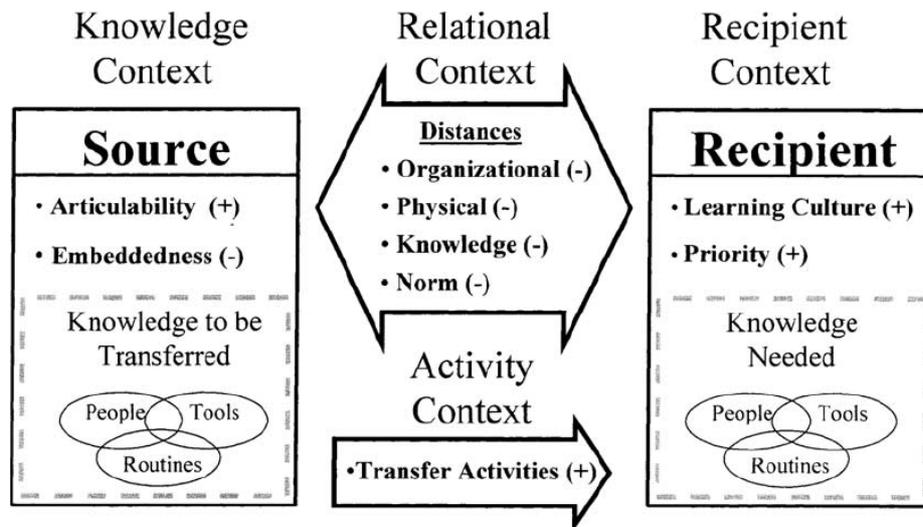


Abbildung A-3: Einflussfaktoren für Wissensweitergabe (Cummings, Teng 2003, S. 40)

Becker and Knudsen identified the following factors from the literature:

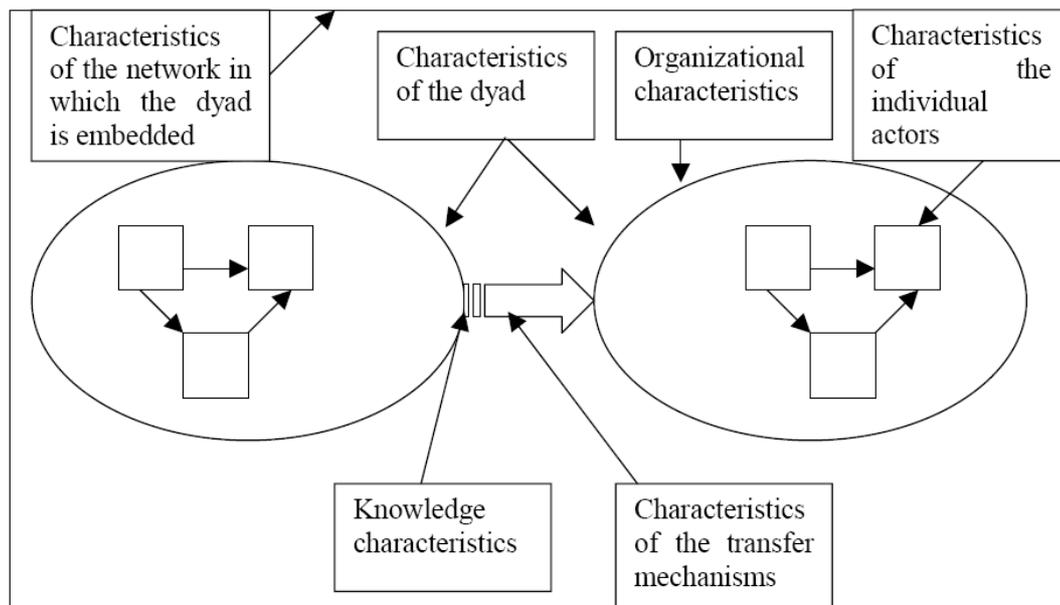


Abbildung A-4: Einflussfaktoren für Wissensweitergabe nach (Becker, Knudsen 2003 S. 15)

- Characteristics of the network in which the dyad is embedded:  
context, legal form
- Characteristics of the relationship (dyad):  
similarity (between individuals, between the knowledge they hold), types of relationships (horizontal/vertical, oneway/ two-way), and characteristics of relationships (knowledge of what other persons know, access to other persons, willingness to engage in problem solving, safety in the relationship that promotes

learning, kind of relationship (arduous), difference in access to template routines, tie strength, relational embeddedness, and knowledge redundancy

- Characteristics of the individual organization:  
formal (hierarchical structure, autonomy, control, competition between internal subunits for market share or internal resources), organizational practices (personnel movements, training, managerial incentive systems, organizational structures or practices that have an impact on communication, organizational structures or practices that have an impact on the replication of routines), absorptive capacity, collective identity, informal lateral relations, opportunities for communication with suppliers and customers
- Characteristics of the individual actor:  
individual capabilities (capacity of observation, transfer ability, and how much attention is required in the transfer process), attitudes (willingness), ‘sender’ (motivation to share knowledge, perceived as reliable), ‘receiver’ (motivation, absorptive capacity, motivational disposition to acquire knowledge, retentive capacity),
- Characteristics of the knowledge transferred:  
knowledge characteristics (tacitness, causal ambiguity, unprovenness, and degree of knowledge dispersion), knowledge level (organizational memory levels, intensity of knowledge production and knowledge absorption of subsidiaries), sources of knowledge (network-based, cluster-based), how knowledge is dealt with in the organization (the organizational instruments and conditions, existence and richness of transmission channels), characteristics of the task that the organization attempts to learn (frequency, heterogeneity, and causal ambiguity)
- Transfer mechanisms:  
patents, technology and reverse engineering

Influencing factors according to the literature review (Ladd, Ward 2002)

1. Relational channels: frequency and depth of two-way human-to-human contact)
2. Partner similarity: degree of similarity (e.g., interests, background, or education) between individuals
3. Depreciation: loss of knowledge after transfer (retentive capacity)
4. Organizational self-knowledge: what do individuals know
5. Divergence of interests: congruency of individual and organizational goals

**Anhang B: UML-Diagramme**

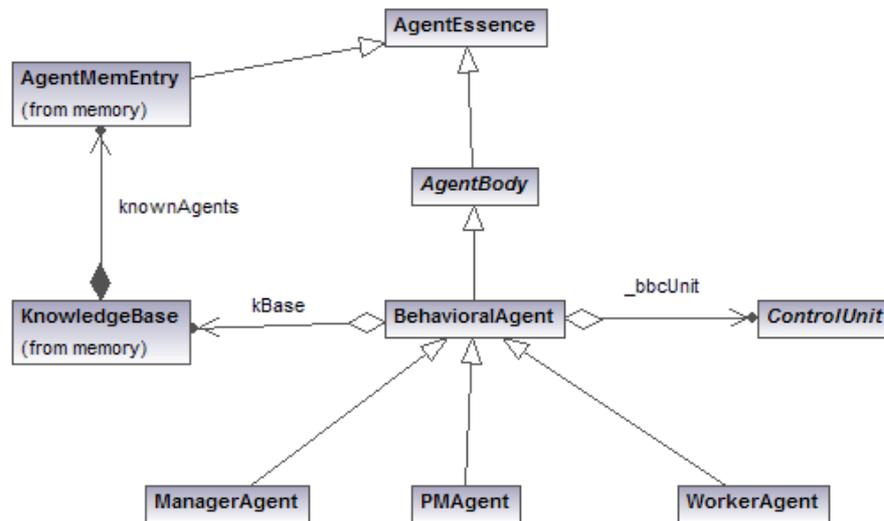


Abbildung B-5: Klassenübersicht der Agentenklassen

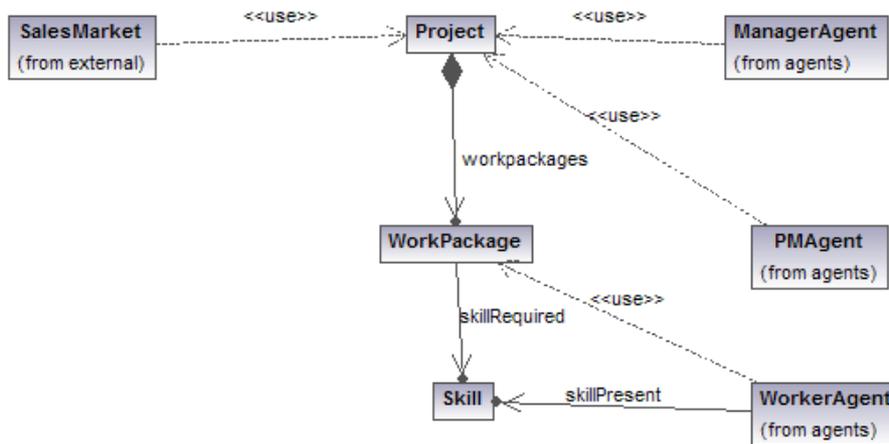


Abbildung B-6: Klassenübersicht Projekt-Workpackage-Skill

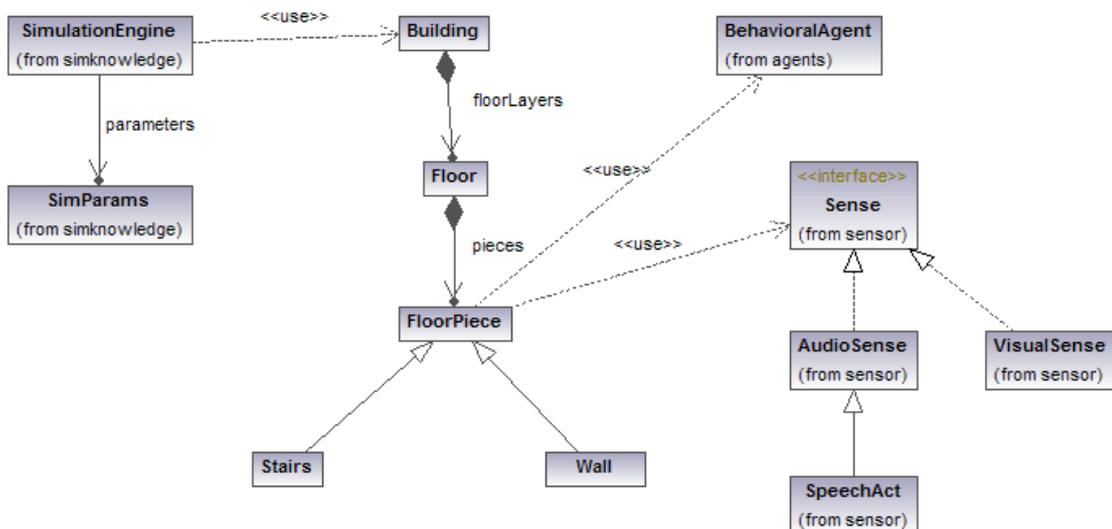


Abbildung B-7: Klassenübersicht Simulationsumgebung und Wahrnehmung



Abbildung B-8: Klassendiagramm SimulationEngine und Parameter

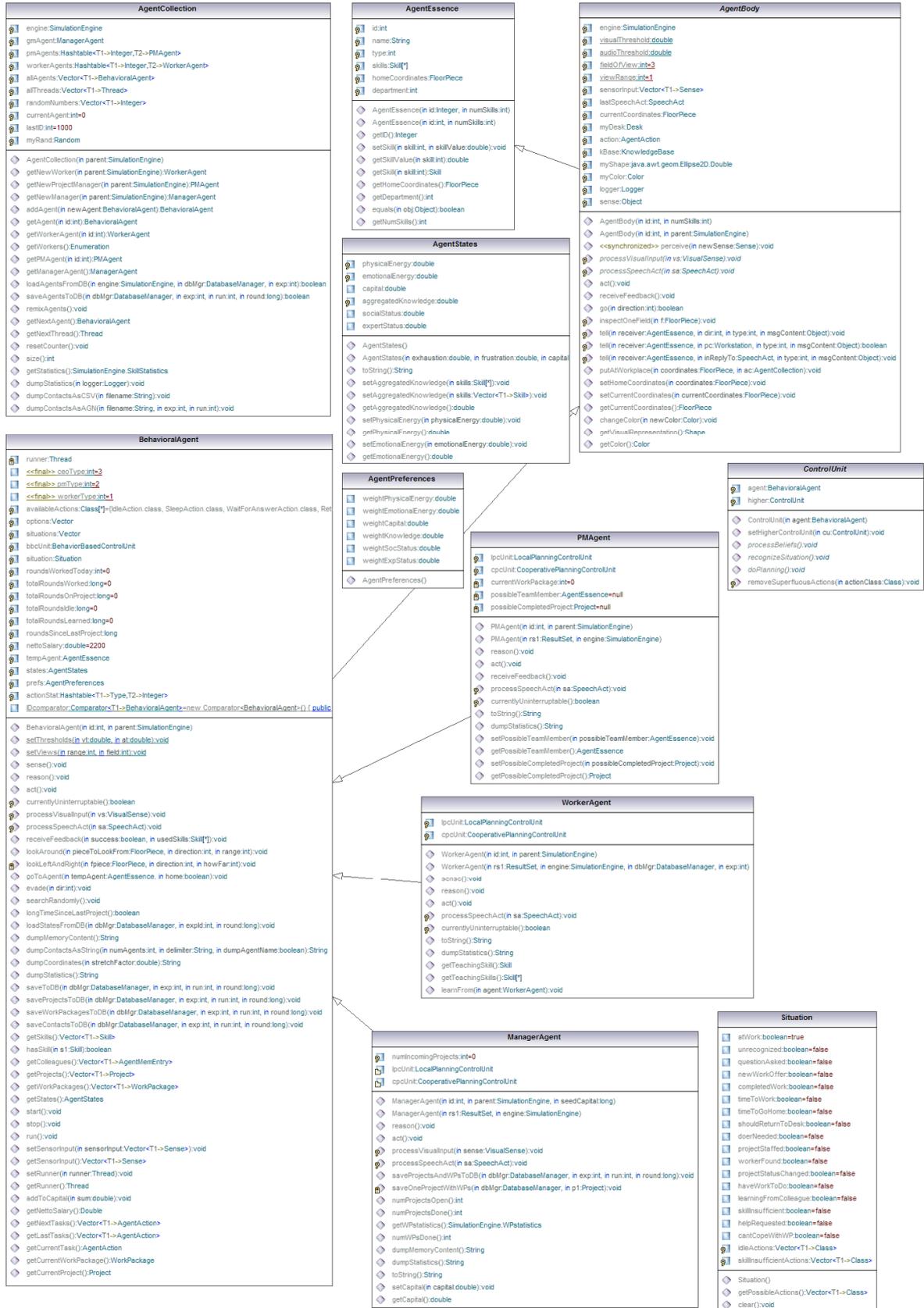


Abbildung B-9: Klassendiagramm der Agentenklassen

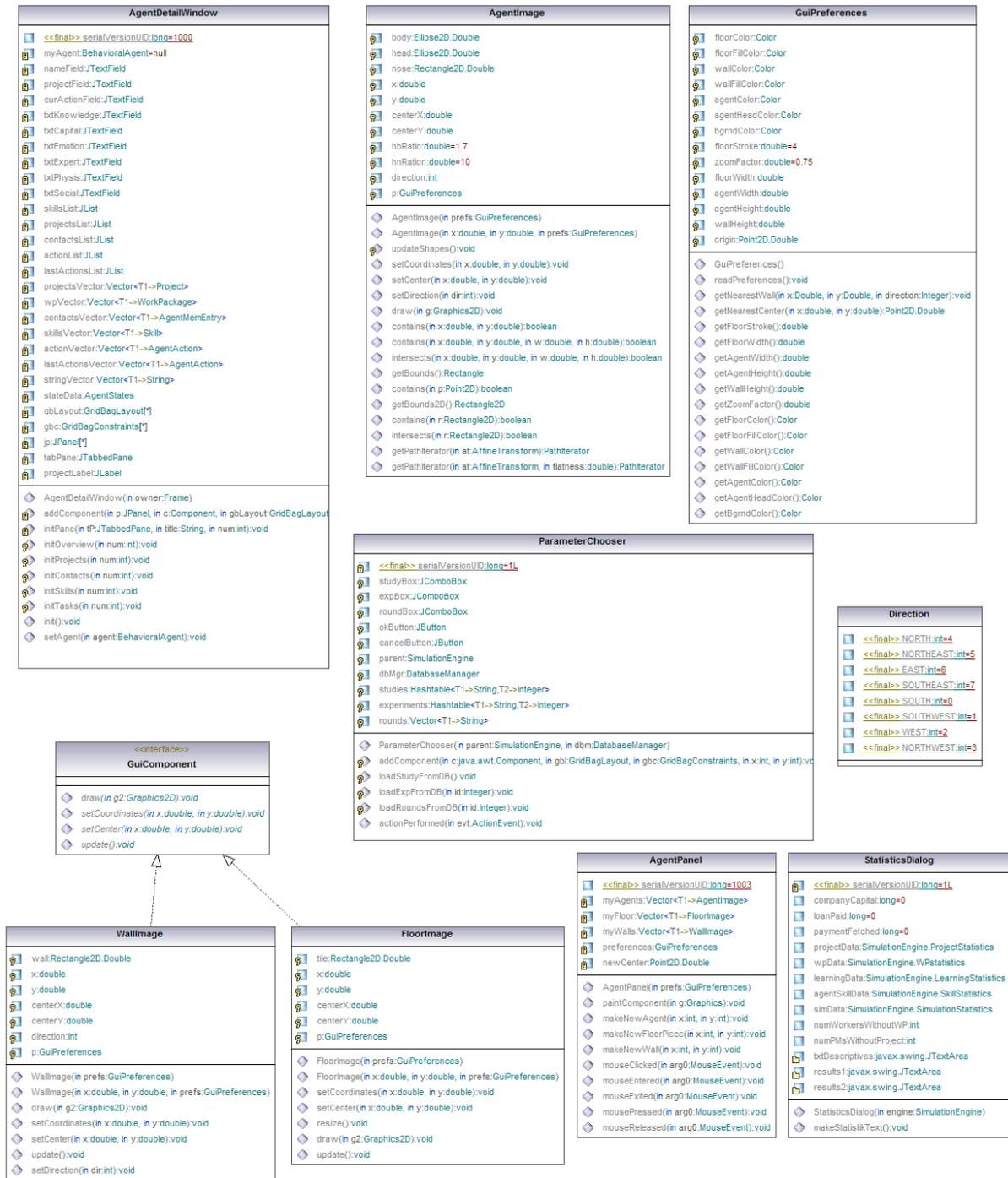


Abbildung B-10: Klassendiagramm der Anzeigeelemente

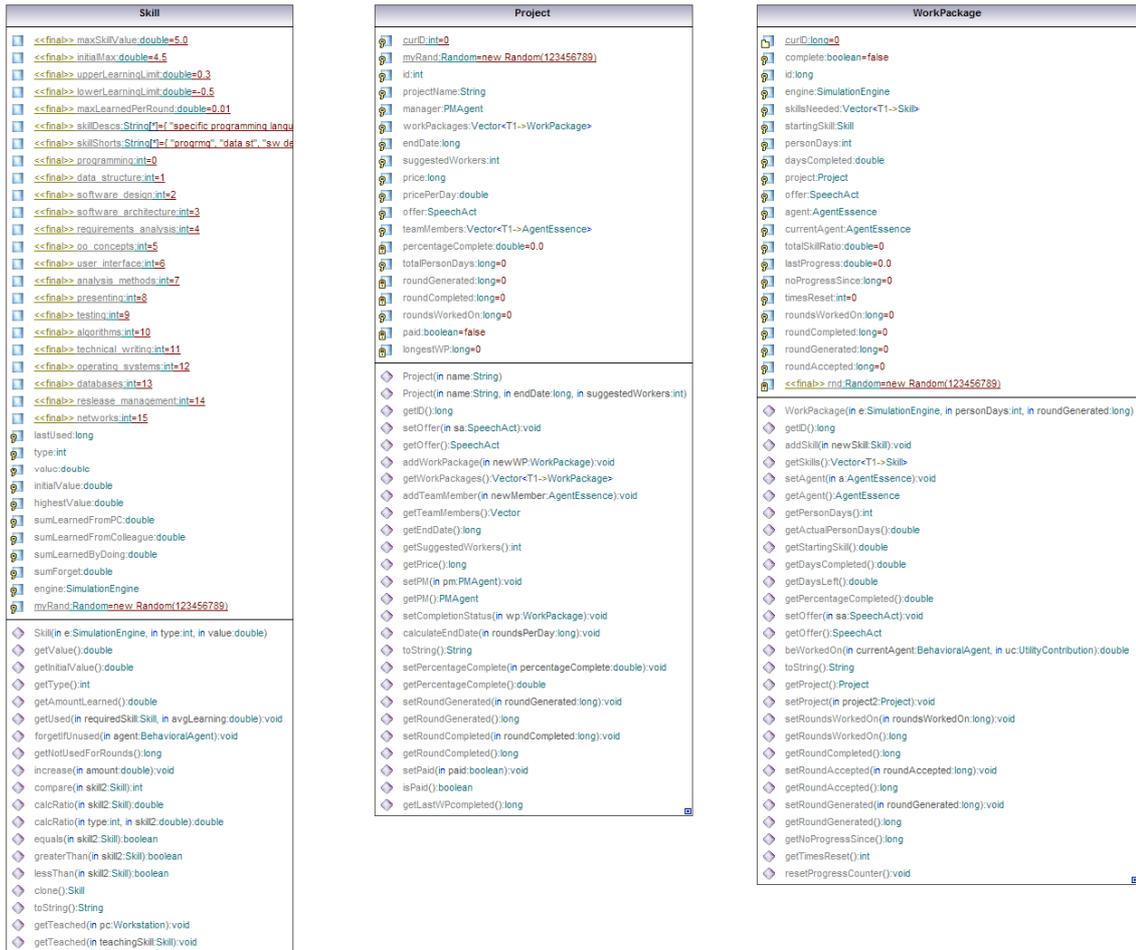


Abbildung B-11: Klassendiagramm Projekt-Workpackage-Skill

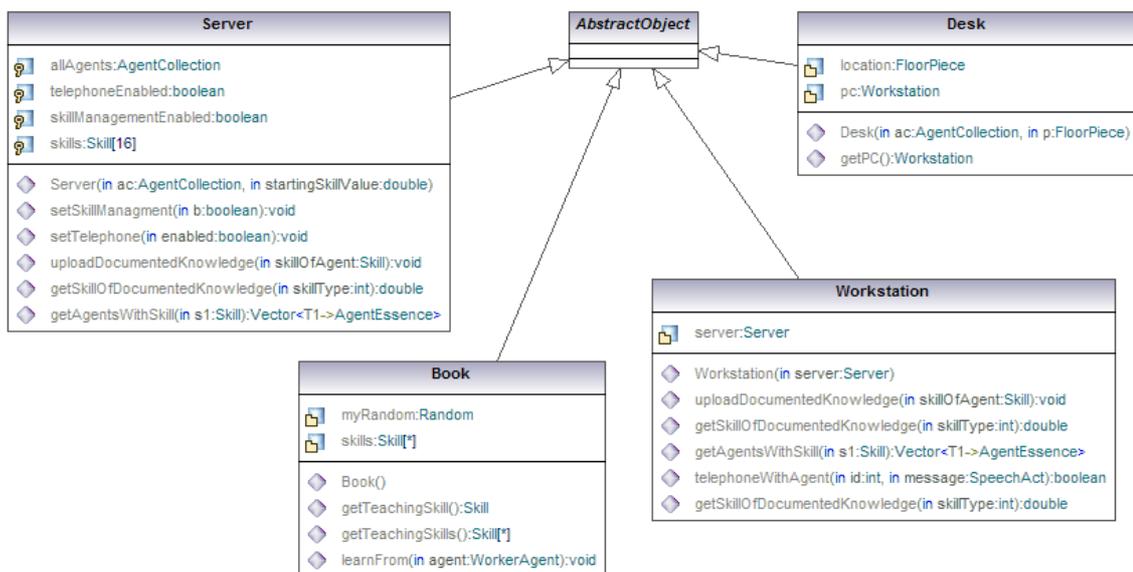


Abbildung B-12: Klassendiagramm der Lernumgebung

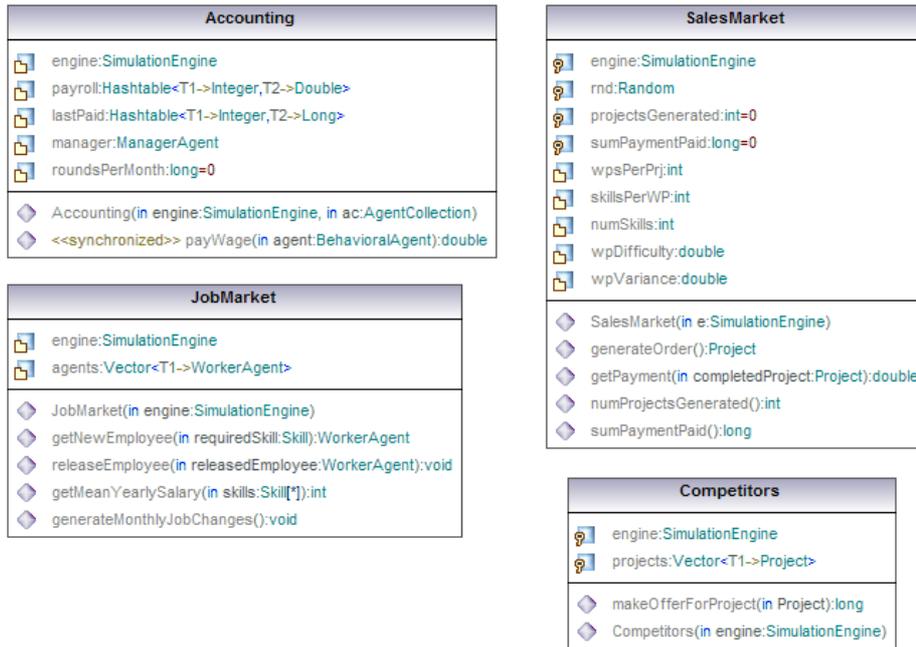


Abbildung B-13: Klassendiagramm der externen Einflüsse

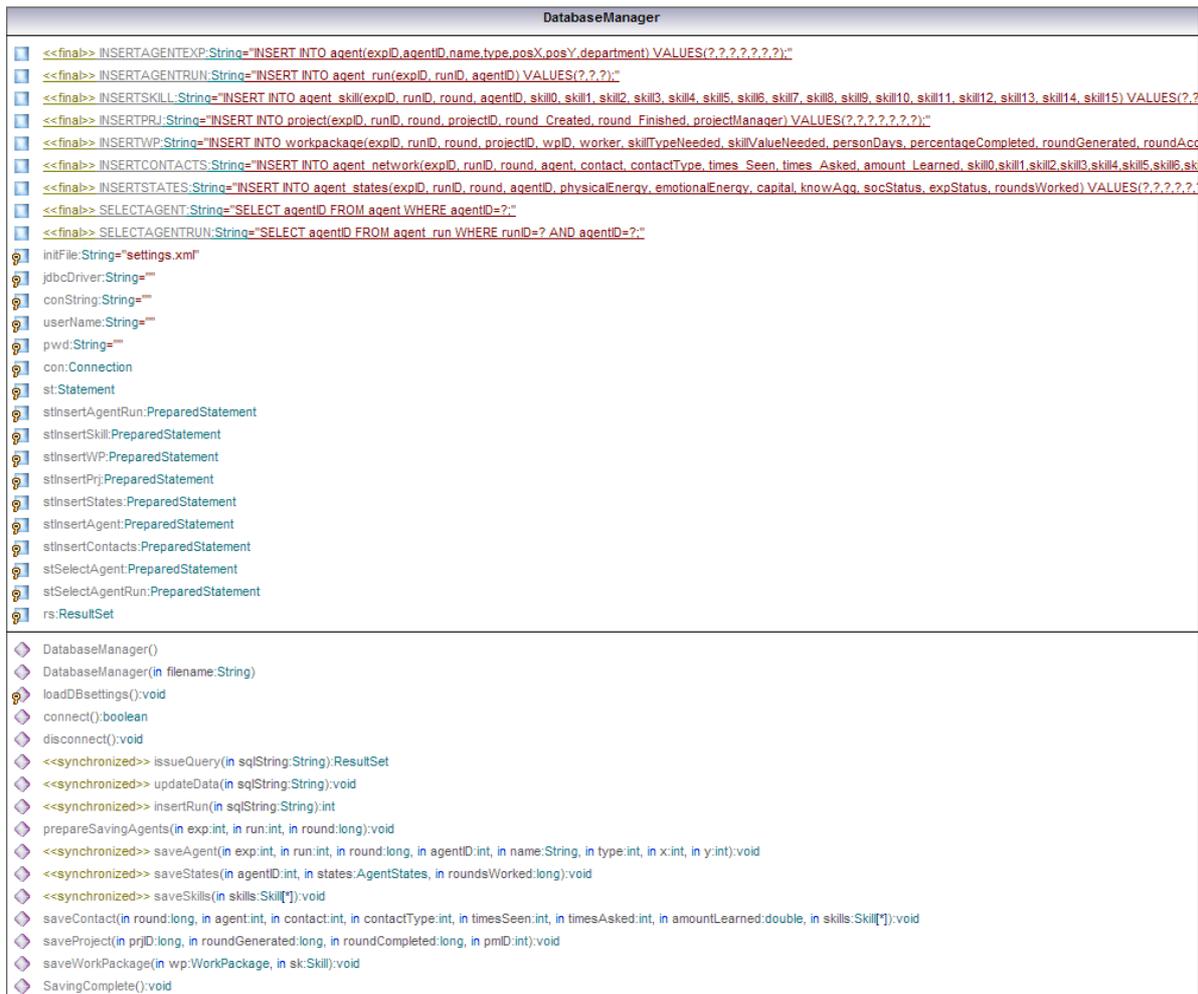


Abbildung B-14: Klassendiagramm der DatabaseManager-Klasse

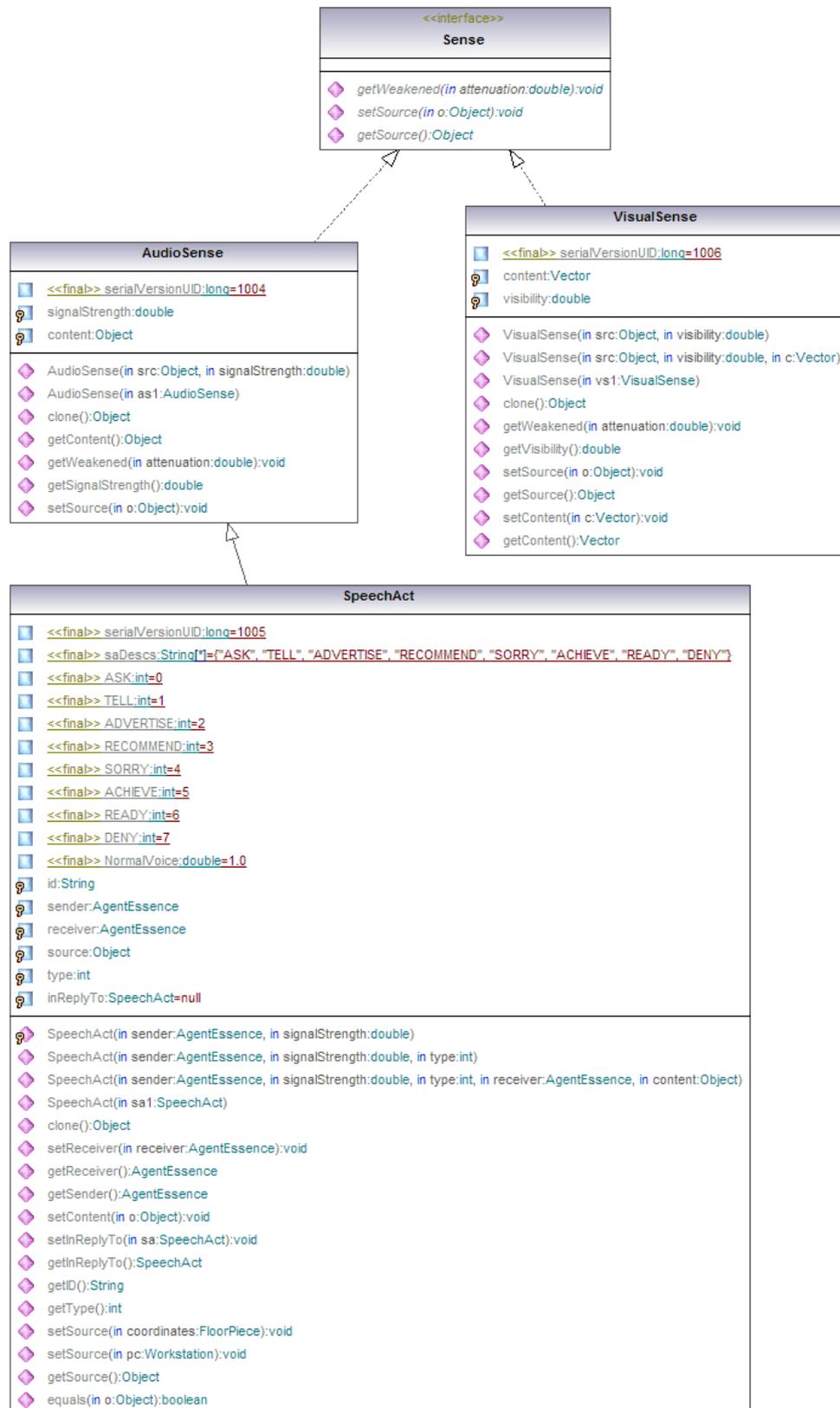


Abbildung B-15: Klassendiagramm der Wahrnehmungsklassen



Abbildung B-16: Klassendiagramm der Umgebungsobjekte

---

## **Anhang C: Ergebnisse des Interviews mit dem Geschäftsführer von itCampus**

Datum der Durchführung: 05.05.2004

1. **Welche Organisationsstruktur liegt vor (am besten Organigramm)?**

2-stufige Hierarchie mit GL, 2 Stabsstellen, Vertrieb + Buchhaltung, sowie 5 Abteilungen (genannt Bereiche) mit je einem Bereichsleiter die das Projektgeschäft abwickeln.

2. **Wie viele Mitarbeiter hat itCampus insgesamt?**

36 Festangestellte, 7 freie Mitarbeiter, die aber dauerhaft für die Firma arbeiten, sowie 18 Studenten mit unterschiedlicher Monatsarbeitszeit

3. **Wie viele Mitarbeiter davon sind in der Verwaltung beschäftigt?**

6 Mitarbeiter sind in der Verwaltung (inkl. GL) und 5 in Vertrieb und Marketing beschäftigt

4. **Wie hoch war der Umsatz im letzten Geschäftsjahr (2003)?**

2,2 Mio Euro

5. **Wo werden diese Umsätze gemacht?**

rund 40% in der mitteldeutschen Region, der Rest im gesamten deutschsprachigen Raum, z.B. auch rund 25% in der Schweiz

6. **Wie viele Projekte wurden im letzten Geschäftsjahr abgewickelt?**

rund 50, davon 10 größere (>50T€) und 5 sehr große(>100T€)

7. **Nennen Sie einige typische Projekte (inhaltlich nicht Kunden)?**

Idealtypisch wickelt itCampus individuelle Softwareprojekte ab und ist dabei branchenunabhängig. Das Vorgehen bei der Analyse ist prozessorientiert. Projekte können nur Analyse, nur Konzeption, nur Implementierung oder auch alles umfassen. Die Lösungen bestehen oft aus Modulen für bestehende Software, Customizing von Standardsoftware und Schnittstellen zwischen Systemen. in 2003 wurden ca. 5 Projekte abgewickelt wo selbstentwickelte Software die komplette Lösung dargestellt hat. Es werden seit kurzem auch 2 Produkte entwickelt, die dann beim Kunden in die vorhandene Softwarelandschaft integriert werden.

8. **Inwieweit kann der Auftragnehmer die Technologien aussuchen mit denen implementiert wird?**

Alle Formen kommen vor, von absoluter Freiheit seitens der Implementierer bis hin zu vorgeschriebenen Programmiersprachen. Meist wird aber nur die DBMS- und WebServer-Plattform vorgeschrieben, damit die Softwarelandschaft beim Kunden nicht

unnötig heterogen wird. Bei Programmiersprachen wird oft Java verlangt, seit kurzem auch verstärkt C#.

**9. Wie werden Schlüsseltechnologien und Technologietrends erkannt?**

**(Systematische Beobachtung? Institutionalisiert?)**

Eigene Stelle dafür eingerichtet (Strategische Technische Ausrichtung), der selber den Markt beobachtet und in unregelmäßigen Abständen mit den Bereichsleitern Schlüsseltechnologien definiert, Mitarbeiter besuchen regelmäßig Konferenzen, es sind diverse Zeitschriftenabos vorhanden, v.a. aber wird das Internet genutzt

**10. Welche Skills werden von der GL aus gefördert / gefordert?**

Proaktiv gefördert werden nur Java und C# einerseits und Kommunikationsfähigkeiten und Projektleiterwissen andererseits. Im Bereich ERP steht eine Entscheidung für den strategischen Wissensaufbau noch aus. Allgemein wird die Situation als in der Konzentrationsphase befindlich beschrieben (A.L.: „Wir sind noch viel zu breit aufgestellt!“). Alle übrigen Skills werden projekt-induziert ausgebaut.

**11. Werden Weiterbildungsmaßnahmen gefördert? Welche? Wie?**

Weiterbildung wird immer von der Firma bezahlt. Seminare werden hauptsächlich für Softskills genutzt, universitäres Engagement wird unterstützt (Hochschulkontakt, 2. Studium, Promotion), für ganz neue Themen oft: 1 Mitarbeiter besucht externe Schulung + Bücherkauf => interne Schulung zum Multiplizieren des Wissens, Bücher können von Mitarbeitern unkompliziert und teilformalisiert angefordert werden, Internetzugang steht allen Mitarbeitern uneingeschränkt zur Verfügung

**12. Welche Medien werden firmenweit bevorzugt? Gibt es Internet- oder Zeitschriften-Abos? Werden öfters Bücher gekauft? Werden interne Schulungen gehalten, externe Schulungsbesuche genehmigt?**

Als Quelle wird hauptsächlich das Internet genutzt, aber auch Zeitschriften und Bücher, sowie interne und externe Schulungen

**13. Gibt es systematische WM-Maßnahmen? Werden Projektnachbesprechungen gehalten? Wird das Skillmanagementsystem regelmäßig gepflegt?**

Erfolgskontrolle findet für Aufwände statt, sonst gibt es keine Projektreviews Skillprofile wurden erst vor 6 Monaten erhoben, es gibt derzeit aber keine festgelegten Aktualisierungszeiträume und auch keine Erfolgsüberwachung (Nutzungskontrolle). Es gibt jährliche Mitarbeitergespräche bei denen Lernziele festgelegt werden und die Erfolge vom letzten Jahr überprüft werden.

---

Es gibt darüber hinaus eine Literaturdatenbank, sowie einen Fileserver für die Ablage von Projektdokumentation. Die Gliederung ist hierarchisch nach Kunden und Projekten

**14. Gibt es Besprechungsräume oder Sitzecken für die Kommunikation?**

Ein Essensraum für das gemeinsame Mittagessen steht zur Verfügung und wird regelmäßig genutzt. Es gibt einen Besprechungsraum und einen Entspannungsraum im Keller mit Flipper und Kicker. Die Büros sind für 2-4 Personen. Bereiche sitzen räumlich nah zusammen. Bei Projektteams aus mehreren Abteilungen werden die Arbeitsplätze für die Projektlaufzeit verlegt, um räumliche Nähe zu gewährleisten. Gemeinsame Freizeitaktivitäten werden gefördert. Ein Computerspiel wurde angeschafft und alle 2-3 Wochen werden „LAN-Parties“ veranstaltet. Es wird im Sommer gemeinsam gegrillt und 1x im Jahr gibt es einen 3-tägigen Firmenausflug

**15. Wie gehen Sie mit dem Wissensverlust durch Fluktuation um?**

Fluktuation ist auf sehr niedrigem Niveau. Seit der Gründung 1999 haben nur 6 Mitarbeiter das Unternehmen verlassen, davon wurden 4 gekündigt. Ein Mitarbeiter ist zur Promotion an die Universität gewechselt. Bei den zwei Kündigungen von Mitarbeiterseite ist jeweils eine längere Zeit zur „Übergabe“ genutzt worden.

**16. Wie werden Neueinstellungen aus Wissenssicht gehandhabt?**

Neueinstellungen hauptsächlich auf Projektdruck hin. Strategischer Erwerb von Wissen hauptsächlich über Studenten (Praktika, Diplomarbeiten, hohe Chance auf anschließende Übernahme). Fachliche Qualifikation wird im Gespräch mit Bereichsleiter festgestellt. Die Geschäftsleitung urteilt über fachübergreifende Qualifikationen.

**17. Gibt es Anreizsysteme für Leistung? für Wissensweitergabe?**

Es gibt variable Gehaltsanteile pro vom Kunden bezahlten Personentag, den der Mitarbeiter gearbeitet hat. Ist das Projekt schneller beendet als veranschlagt, wird trotzdem die volle Summe ausgeschüttet. Dauert es länger, wird dieselbe Summe vergeben. Mitarbeiter, die an der Entwicklung der 2 Produkte beteiligt sind, bekommen anteilige Vergütung in Abhängigkeit von den Lizenzerlösen. Die Höhe des variablen Anteils pro Tag / Lizenzverkauf hängt wird bei der Einstellung verhandelt und hängt hauptsächlich von der Risikobereitschaft des Mitarbeiters ab. Je niedriger der fixe Gehaltsanteil, desto höher die Leistungsvergütung.

Beim jährlichen Mitarbeitergespräch werden zudem Gehaltserhöhungen verhandelt, deren Höhe und Genehmigung in erster Linie von der Leistung im letzten Jahr abhängt. Für Wissensaufbau oder Weitergabe gibt es keine Anreizsysteme.

## **Anhang D: Fragebogen für itCampus**

Lehrstuhl für Wirtschaftsinformatik,  
insbes. betriebliches Informationsmanagement

Martin-Luther-Universität Halle-Wittenberg

Anschrift: Barfüßer Str. 17, 06099 Halle

Telefon: 0345 / 68588-22

**Sekretariat: 0345 / 68588-11**

**E-Mail: [peinl@wiwi.uni-halle.de](mailto:peinl@wiwi.uni-halle.de)**



**– Untersuchung der Wissensentwicklung  
bei IT-Mitarbeitern**

Halle, den 15.06.2004

Sehr geehrte Damen und Herren,

Wir führen zurzeit eine Untersuchung zum Thema „Wissensentwicklung bei IT-Mitarbeitern“ im Rahmen einer Dissertation durch. Schwerpunkt stellt dabei die „Multiagentensimulation der Wissensentwicklung bei IT-Dienstleistern“ dar. In der Untersuchung geht es vor allem um die Erhebung von Parametern (z.B. bevorzugte Wissensquellen), die bei der Simulation verwendet werden sollen. Ziel der Simulation ist ein tieferes Verständnis der Wirkung von Wissensmanagement-Instrumenten in Organisationen.

Wir sind deshalb an Ihrer Meinung sehr interessiert und würden uns freuen, wenn Sie den Fragebogen ausgefüllt an uns zurücksenden würden. Das Ausfüllen des Fragebogens wird ca. 30 Minuten in Anspruch nehmen. Ihre Teilnahme an dieser Befragung ist selbstverständlich freiwillig. Wir hoffen dennoch, dass Sie sich zu einer Teilnahme entschließen können, denn nur wenn möglichst alle ausgewählten Personen an der Befragung teilnehmen, sind die Ergebnisse repräsentativ.

Dieser Fragebogen wurde an alle Mitarbeiter der itCampus Software- und Systemhaus GmbH gesendet, die direkt an der Umsetzung von Projekten beteiligt sind. Ihre Angaben werden anonym behandelt, d. h. die Antworten werden ohne Namen und Adresse ausgewertet. Die Forschungsarbeit unterliegt den Regelungen der Datenschutzgesetzgebung. Es ist sichergestellt, dass Ihre Angaben nicht mit Ihrer Person in Verbindung gebracht werden.

Wir bitten Sie, den Fragebogen bis zum 30.05.2004 an folgende Email-Adresse zu senden:  
[peinl@wiwi.uni-halle.de](mailto:peinl@wiwi.uni-halle.de)

Für Ihre Mitarbeit bedanken wir uns ganz herzlich.

Mit freundlichen Grüßen

René Peinl

---

**Allgemeine Hinweise**

Die Beantwortung der Fragen erfolgt über das Ankreuzen einer der angebotenen Alternativen, bzw. durch Ausfüllen der Lücken. Fragen, bei denen mehrere Antworten möglich sind, sind explizit gekennzeichnet. Bei allen anderen Fragen wählen Sie bitte nur eine Antwortmöglichkeit aus.

Der Fragebogen gliedert sich in fünf Teile. Im ersten Teil werden Fragen zum Lernen und der Wissensweitergabe gestellt. Der zweite Teil befasst sich mit Produktkenntnissen. Der dritte Teil erhebt Wissen im Themengebiet Softwareentwicklung. Im vierten Teil geht es um Kenntnisse in diversen anderen, für Softwareentwickler relevanten Bereichen. Der fünfte und letzte Teil erhebt schließlich einige demographische und tätigkeitsbezogene Daten.

**Teil I: Lernen und Wissensweitergabe**

1. Wie hoch schätzen Sie die Wichtigkeit folgender Kenntnisse für Ihre Tätigkeit ein?

	sehr wichtig	wichtig	unent- schieden	weniger wichtig	unwichtig
Kenntnisse über die Installation und Konfiguration bestimmter Softwareprogramme	<input type="checkbox"/>				
Kenntnisse über die Anwendung bestimmter Softwareprogramme	<input type="checkbox"/>				
Kenntnisse des Softwareentwicklungsprozesses	<input type="checkbox"/>				
Programmierkenntnisse	<input type="checkbox"/>				
Kenntnisse über Netzwerke	<input type="checkbox"/>				
Projektmanagementkenntnisse	<input type="checkbox"/>				

2. Wie viel Prozent ihrer gesamten Arbeitszeit verwenden Sie im Jahresdurchschnitt direkt für Projektarbeit? (Bitte schätzen Sie den Anteil)

0%

3. Sind Sie als Projektmitarbeiter schon einmal auf Situationen innerhalb des Projektes gestoßen, in denen Sie mit Ihrem Wissen nicht weiter gekommen sind, so dass Sie sich zur Lösung des Problems neues Wissen aneignen mussten?

ja  nein (wenn nein, dann weiter zur Frage 6)

4. Welche Wissensquellen nutzen Sie wie oft, um während der Projektarbeit eine Lösung für ein Problem zu finden?

	häufig (jede Woche)	oft (3-1x pro Monat)	gelegentlich (11-4x pro Jahr)	selten (höchstens 3x pro Jahr)	nie
Bücher	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zeitschriften	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Intranet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Internet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Personen (z.B. Kollegen)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interne Schulungen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Externe Schulungen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gezieltes Ausprobieren (z.B. Testinstallation eines Produktes)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

5. Wie viel Prozent der Projektarbeitszeit verwenden Sie auf Weiterbildung, die nötig ist um ein Projekt abschließen zu können?

- bis 10 %                       11-20%  
 21-30%                       31-40 %  
 41-50%                       über 50%

(Hinweis: 10% entsprechen bei 3 Monaten Projektarbeitszeit 6 Arbeitstagen)

6. Wie viel Abstimmung mit anderen Projektmitarbeitern, oder anderen am Projekt Beteiligten (Kunde, Partner, Lieferant) ist während eines Projekts nötig?  
(Bitte schätzen Sie den Anteil der Projektarbeitszeit)

0%

Im Folgenden geht es um selbstmotivierte Weiterbildung, die nicht unmittelbar wegen Projektdruck erfolgt.

7. Bilden Sie sich außerhalb von konkreten Projekten weiter, z.B. durch Lesen von Fachzeitschriften, oder Beobachten von Themen im Internet?

ja       nein (wenn nein, dann weiter zur Frage 11)

8. Wie viele Stunden *Arbeitszeit* verwenden Sie durchschnittlich pro Monat auf Weiterbildung außerhalb von Projekten?

9. Wie viele Stunden *Freizeit* verwenden Sie durchschnittlich pro Monat auf Weiterbildung?

10. Welche Wissensquellen nutzen Sie wie oft, um Ihr Wissen aktuell zu halten?

	häufig (jede Woche)	oft (3-1x pro Monat)	gelegentlich (1-4x pro Jahr)	selten (höchstens 3x pro Jahr)	nie
Bücher	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Zeitschriften	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Intranet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Internet	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Personen (z.B. Kollegen)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interne Schulungen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Externe Schulungen	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Gezieltes Ausprobieren (z.B. Testinstallation eines Produktes)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

11. Aus welchen Gründen verwenden Sie Arbeits- oder Freizeit, um Ihr Wissen aktuell zu halten und zu vertiefen?

	Trifft voll zu	Trifft eher zu	Unent- schieden	Trifft eher nicht zu	Trifft nicht zu
Ich erwarte, die nächsten Projekte mit weniger Aufwand erledigen zu können	<input type="checkbox"/>				
Ich erhoffe mir bessere Aufstiegschancen (z.B. Beförderung zum Abteilungsleiter)	<input type="checkbox"/>				
Ich erhoffe mir bessere Chancen auf eine Gehaltserhöhung oder Sonderprämien	<input type="checkbox"/>				
Ich erhoffe mir durch mein Fachwissen den Respekt meiner Kollegen als Experte	<input type="checkbox"/>				
Die Inhalte interessieren mich auch privat	<input type="checkbox"/>				
Sonstige, nämlich (bitte angeben)	<input type="checkbox"/>				

12. Haben Sie bei itCampus schon einmal jemanden um fachlichen Rat gebeten?

ja  nein (wenn nein, dann weiter zur Frage 18)

13. Wie viele verschiedene Personen haben Sie in den letzten 3 Monaten um fachlichen Rat gefragt?

14. Wenn Sie jemanden um fachlichen Rat bitten, dann ...

(bitte kreuzen Sie die Aussage an, die am häufigsten zutrifft)

- ... findet sich meist ein Kollege, der eine Idee hat, die zur Lösung führt.
- ... findet sich meist ein Kollege, der mir die Lösung direkt am Produkt zeigt.
- ... findet sich meist ein Kollege, der mir den Lösungsweg erklärt.
- ... reicht es meistens schon über das Problem zu sprechen, um die Lösung zu finden.
- ... findet sich meist ein Kollege, der mit mir direkt am Produkt die Lösung erarbeitet.

15. Haben Sie sich dafür entschieden jemanden um fachlichen Rat zu bitten, dann ...

	Trifft voll zu	Trifft eher zu	Unentschieden	Trifft eher nicht zu	Trifft nicht zu
... frage ich einen Mitarbeiter von derzeitigen oder früheren Kunden	<input type="checkbox"/>				
... frage ich einen Mitarbeiter von derzeitigen oder früheren Partnerunternehmen	<input type="checkbox"/>				
... frage ich einen ehemaligen Kollegen oder Kommilitonen, bzw. einen Freund oder Bekannten	<input type="checkbox"/>				
... frage ich einen derzeitigen Kollegen	<input type="checkbox"/>				

16. Wenn Sie einen Kollegen um fachlichen Rat fragen wollen, dann fragen Sie einen Kollegen ...

	Trifft voll zu	Trifft eher zu	Unentschieden	Trifft eher nicht zu	Trifft nicht zu
..., der gleich nebenan im Büro sitzt	<input type="checkbox"/>				
... aus dem aktuellen Projektteam	<input type="checkbox"/>				
... aus meiner Abteilung, der nicht im aktuellen Projektteam ist	<input type="checkbox"/>				
... aus dem selben Bürogebäude, aber nicht aus Team oder Abteilung	<input type="checkbox"/>				
... von einem anderen Standort	<input type="checkbox"/>				

17. Sie bitten den Kollegen um fachlichen Rat, ...

	Trifft voll zu	Trifft eher zu	Unentschieden	Trifft eher nicht zu	Trifft nicht zu
..., der mir schon öfters geholfen hat	<input type="checkbox"/>				
..., mit dem ich mich gut verstehe	<input type="checkbox"/>				
..., von dem ich weiß, dass er weitreichende Kenntnisse über den fraglichen Themenbereich hat	<input type="checkbox"/>				

18. Das Wissen über die Kenntnisse und Fähigkeiten Ihrer Kollegen beziehen Sie ...

	immer	oft	gelegentlich	selten	nie
... aus dem Skillprofil im Intranet	<input type="checkbox"/>				
... von anderen Kollegen	<input type="checkbox"/>				
... aus gemeinsamer Projektarbeit	<input type="checkbox"/>				
... aus eher zufälligen Gesprächen	<input type="checkbox"/>				
sonstiges, nämlich (bitte angeben)	<input type="checkbox"/>				

19. Von wie vielen verschiedenen Kollegen wurden Sie in den letzten 3 Monaten um fachlichen Rat gefragt?

20. Wie viele Stunden verwenden Sie pro Woche, um Kollegen fachlichen Rat zu geben?

21. Wie viele Stunden pro Woche arbeiten Sie direkt mit einem oder mehreren Kollegen zusammen (z.B. Paired Programming, gemeinsame Code Reviews, ...)?

22. Wovon war Ihre berufliche Entwicklung im letzten Jahr hauptsächlich geprägt?

	Trifft voll zu	Trifft eher zu	Unentschieden	Trifft eher nicht zu	Trifft nicht zu
Ausbau der fachlichen Kompetenz in Ihrem Spezialgebiet, in dem Sie zwar vorher schon viel wussten, das sich aber laufend weiterentwickelt.	<input type="checkbox"/>				
Aufbau <i>einer</i> speziellen fachlichen Kompetenz in einem Thema, über das Sie vorher noch wenig wussten, das aber zukunftssträftig schien.	<input type="checkbox"/>				
Aufbau <i>mehrerer</i> fachlicher Kompetenzen, in Themen, über die Sie vorher noch wenig wussten, die aber zukunftssträftig schienen.	<input type="checkbox"/>				
Weiterbildung fast ausschließlich durch Erfahrungen bei der Projektarbeit.	<input type="checkbox"/>				
Aufbau von Meta-Kompetenzen (z.B. Soziale Kompetenz, Kommunikationsfähigkeit)	<input type="checkbox"/>				
Beibehaltung des Status Quo, da für Sie eine Weiterentwicklung nicht erforderlich war.	<input type="checkbox"/>				
Weiterbildung war für Sie leider nicht möglich, weil keine interessanten Weiterbildungsangebote vorhanden waren.	<input type="checkbox"/>				
Sie konnten kaum wertvolle Erfahrungen machen, da Sie zu sehr ins Projektgeschäft eingebunden waren und in dem Projekt nur Standardtechnologien eingesetzt wurden.	<input type="checkbox"/>				
Sonstiges, nämlich (bitte angeben)	<input type="checkbox"/>				

**Teil II: Produktwissen**

In diesem Abschnitt geht es um Wissen über Installation, Administration und Konfiguration der Produkte (1-4), sowie Anwenderkenntnisse (5 und 6). In Klammern finden Sie jeweils Beispiele für die Produktkategorie, bzw. relevante Versionen des Produktes.

Für die Einschätzung des aktuellen Wissensstandes bewerten Sie bitte nur die Version und das Produkt, von dem Sie am meisten wissen.

Die Antwortkategorien bedeuten dabei:

- 0 Ich weiß nichts darüber
- 1 Ich bin vage damit vertraut
- 2 Ich kenne die Grundlagen
- 3 Ich habe solides Wissen, ausreichend zur selbständigen Arbeit im Themengebiet
- 4 Ich weiß sehr viel darüber
- 5 Ich weiß nahezu alles darüber

		aktueller Wissensstand					
		0	1	2	3	4	5
<b>Betriebssysteme</b>							
Windows	(9x, ME, NT, 2000, XP, 2003)	<input type="checkbox"/>					
Mac OS	(9.x, X)	<input type="checkbox"/>					
Linux	(Suse, Redhat, Debian, ...)	<input type="checkbox"/>					
Unix	(Solaris, AIX, HP UX, ...)	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					
		aktueller Wissensstand					
		0	1	2	3	4	5
<b>Datenbanksoftware</b>							
Oracle	(8i, 9i, 10g)	<input type="checkbox"/>					
DB2	(7.2, 8.1)	<input type="checkbox"/>					
MS SQL Server	(7.0, 2000)	<input type="checkbox"/>					
SAP DB	(7.3, 7.4, MaxDB 7.5)	<input type="checkbox"/>					
mySQL	(3.2x, 4.0, 4.1, ...)	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					
		aktueller Wissensstand					
		0	1	2	3	4	5
<b>Infrastruktur-Server</b>							
Content Management Systeme	(phpNuke, NPS, Pirobase, ...)	<input type="checkbox"/>					
Dokumentenmanagement Systeme	(Documentum, DominoDoc, ...)	<input type="checkbox"/>					
Workflow-Engines	(wfmOpen, Enhydra Shark, ...)	<input type="checkbox"/>					
Versionsverwaltungs Systeme	(CVS, Subversion, ...)	<input type="checkbox"/>					
Groupware	(MS Exchange, Lotus Notes, ...)	<input type="checkbox"/>					
XML-Middleware	(MS BizTalk, WebMethods, Seeburger, ...)	<input type="checkbox"/>					

<b>Infrastruktur-Server</b>		0	1	2	3	4	5
Application Server	(JBoss, WebSphere, WebLogic, ...)	<input type="checkbox"/>					
Web Server	(IIS, Apache, ...)	<input type="checkbox"/>					
Mail Server	(Sendmail, HP Open Mail, Postfix, ...)	<input type="checkbox"/>					
Directory Services	(Open LDAP, Active Directory, Sun DS, ...)	<input type="checkbox"/>					
File Services	(Samba, NFS, NTFS, ...)	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					
	aktueller Wissensstand						
<b>Unternehmenssoftware</b>		0	1	2	3	4	5
SAP R/3	(3.1, ..., 4.6, mySAP ERP)	<input type="checkbox"/>					
Oracle Applications	(9, 10, ...)	<input type="checkbox"/>					
Sage KHK	(2.x, 3.0, 3.2)	<input type="checkbox"/>					
Peoplesoft	(inkl. JD Edwards)	<input type="checkbox"/>					
Microsoft Business Applications	(inkl. Navision + Great Plains)	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					
	aktueller Wissensstand						
<b>Grafik- und Videosoftware</b>		0	1	2	3	4	5
3D Konstruktion	(AutoCAD, Cinema 4D, Maya, ...)	<input type="checkbox"/>					
Bitmapgrafik	(Photoshop, Fireworks, ...)	<input type="checkbox"/>					
Vektorgrafik	(Freehand, Illustrator, ...)	<input type="checkbox"/>					
Layout & Desktop Publishing	(InDesign, Framemaker, ...)	<input type="checkbox"/>					
Multimedia	(Flash, Director, ...)	<input type="checkbox"/>					
Video	(Premiere, Video Studio, ...)	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					
	aktueller Wissensstand						
<b>Officeanwendungen</b>		0	1	2	3	4	5
Textverarbeitung	(MS Word, ...)	<input type="checkbox"/>					
Tabellenkalkulation	(MS Excel, ...)	<input type="checkbox"/>					
Desktopdatenbanken	(MS Access, ...)	<input type="checkbox"/>					
Präsentationssoftware	(MS Powerpoint, ...)	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					
	aktueller Wissensstand						
<b>Softwareentwicklungswerkzeuge</b>		0	1	2	3	4	5
Integrierte Entwicklungsumgebungen	(Eclipse, Visual Studio, ...)	<input type="checkbox"/>					
DB-Entwurfswerkzeuge	(Power Designer, ERwin ...)	<input type="checkbox"/>					
CASE-Tools	(Rational Rose, Oracle JDeveloper ...)	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					

### Teil III: Kenntnisse der Softwareentwicklung

In diesem Abschnitt geht es um zentrale Fertigkeiten für die Softwareentwicklung.

In Klammern finden Sie z.T. Beispiele für die jeweilige Kategorie. Für die Einschätzung des aktuellen Wissensstandes bewerten Sie bitte nur die Sprache oder Bibliothek, von dem Sie am meisten wissen.

Die Antwortkategorien bedeuten dabei:

- 0 Ich weiß nichts darüber
- 1 Ich bin vage damit vertraut
- 2 Ich kenne die Grundlagen
- 3 Ich habe solides Wissen, ausreichend zur selbständigen Arbeit im Themengebiet
- 4 Ich weiß sehr viel darüber
- 5 Ich weiß nahezu alles darüber

		aktueller Wissensstand					
		0	1	2	3	4	5
<b>Software-Entwicklungsprozess</b>							
Analyse- und Design-Methoden		<input type="checkbox"/>					
Anforderungsanalyse und -definition		<input type="checkbox"/>					
Software-Architektur		<input type="checkbox"/>					
Algorithmen	(z.B. <i>Sort, Search, ...</i> )	<input type="checkbox"/>					
Software-Design und Patterns		<input type="checkbox"/>					
Datenbankdesign und Entity Relationship Modellierung	( <i>ERM</i> )	<input type="checkbox"/>					
Prozessanalyse und –modellierung		<input type="checkbox"/>					
Objektorientierte Analyse und Design	(z.B. <i>mit UML</i> )	<input type="checkbox"/>					
Testen, Testmethoden und Qualitätsmanagement		<input type="checkbox"/>					
Performance-Messung und -Analyse		<input type="checkbox"/>					
Konfigurations- und Releasemanagement		<input type="checkbox"/>					
Schulungen	( <i>insb. Präsentation</i> )	<input type="checkbox"/>					
Dokumentation	( <i>insb. technisches Schreiben</i> )	<input type="checkbox"/>					
Software-Wartung		<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					

		aktueller Wissensstand					
		0	1	2	3	4	5
<b>Programmiersprachen</b>							
Objektorientierte	( <i>C++, C#, Java, ...</i> )	<input type="checkbox"/>					
Strukturierte	( <i>C, Cobol, ABAP, ...</i> )	<input type="checkbox"/>					
Rapid Application Development Tools	( <i>Visual Basic, Delphi, ...</i> )	<input type="checkbox"/>					

## Anhang

---

Datenbanksprachen	( <i>SQL, X-Query, ...</i> )	<input type="checkbox"/>					
Web-Sprachen	( <i>PHP, ASP, JSP, ...</i> )	<input type="checkbox"/>					
Skriptsprachen	( <i>Flash Actionscript, JavaScript, Perl, Python, ...</i> )	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					

aktueller Wissensstand

### Programmierbibliotheken

0 1 2 3 4 5

Frameworks	( <i>.NET, J2EE, ...</i> )	<input type="checkbox"/>					
Graphical User Interfaces	( <i>Swing, GTK+, MFC GUI-Klassen, ...</i> )	<input type="checkbox"/>					
Java Enterprise	( <i>JAAS, JMS, JMX, JNDI, ...</i> )	<input type="checkbox"/>					
Webentwicklung	( <i>CGI, Servlets, WebServices, ...</i> )	<input type="checkbox"/>					
Generische Bibliotheken	( <i>STL, ...</i> )	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					

aktueller Wissensstand

### Internet Markup-Sprachen

0 1 2 3 4 5

HTML		<input type="checkbox"/>					
Cascading Style Sheets (CSS)		<input type="checkbox"/>					
XML-Basis		<input type="checkbox"/>					
Schema-Sprachen	( <i>DTD, XML-Schema, ...</i> )	<input type="checkbox"/>					
XSLT und XPath		<input type="checkbox"/>					
Sonstige XML-Standards	( <i>SVG, Docbook, ...</i> )	<input type="checkbox"/>					
Wireless Application Protocol (WAP)		<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					

aktueller Wissensstand

### Design von grafischen Benutzeroberflächen

0 1 2 3 4 5

2D-Animation		<input type="checkbox"/>					
3D-Visualisierung		<input type="checkbox"/>					
Fotoproduktion		<input type="checkbox"/>					
Bildbearbeitung		<input type="checkbox"/>					
Illustration		<input type="checkbox"/>					
Videoproduktion		<input type="checkbox"/>					
Interface-Design		<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					

---

**Teil IV: Sonstige Kenntnisse**

In diesem Abschnitt geht es um Kenntnisse, die nicht direkt mit der Softwareentwicklung zusammen hängen, in der IT-Projektarbeit jedoch hilfreich sein können.

Die Antwortkategorien bedeuten dabei:

- 0 Ich weiß nichts darüber
- 1 Ich bin vage damit vertraut
- 2 Ich kenne die Grundlagen
- 3 Ich habe solides Wissen, ausreichend zur selbständigen Arbeit im Themengebiet
- 4 Ich weiß sehr viel darüber
- 5 Ich weiß nahezu alles darüber

	aktueller Wissensstand					
	0	1	2	3	4	5
<b>Netzwerkprotokolle</b>						
TCP/IP	<input type="checkbox"/>					
HTTP	<input type="checkbox"/>					
SMTP	<input type="checkbox"/>					
POP3	<input type="checkbox"/>					
LDAP	<input type="checkbox"/>					
Sonstige, nämlich	<input type="checkbox"/>					

	aktueller Wissensstand					
	0	1	2	3	4	5
<b>Sicherheit</b>						
Verschlüsselung	<input type="checkbox"/>					
Viren und Antivirensoftware	<input type="checkbox"/>					
Firewalls	<input type="checkbox"/>					
Intrusion Detection	<input type="checkbox"/>					
Digitale Signaturen	<input type="checkbox"/>					
Virtual Private Networks (VPN)	<input type="checkbox"/>					
Secure Socket Layer (SSL) und Transport Level Security (TLS)	<input type="checkbox"/>					
Sonstige, nämlich	<input type="checkbox"/>					

		aktueller Wissensstand					
		0	1	2	3	4	5
<b>Module betrieblicher Standardsoftware</b>							
Einkauf	(Purchasing)	<input type="checkbox"/>					
Verkauf und Vertrieb	(SD)	<input type="checkbox"/>					
Kundenmanagement	(CRM)	<input type="checkbox"/>					

<b>Module betrieblicher Standardsoftware</b>		0	1	2	3	4	5
Lieferantenmanagement	(SRM)	<input type="checkbox"/>					
Produktlebenszyklus Management	(PLM)	<input type="checkbox"/>					
Supply Chain Management	(SCM)	<input type="checkbox"/>					
Finanzbuchhaltung	(FiBu)	<input type="checkbox"/>					
Personalwirtschaft	(HR)	<input type="checkbox"/>					
Materialwirtschaft	(MM)	<input type="checkbox"/>					
Produktionsplanung und –Steuerung	(PPS)	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					
		aktueller Wissensstand					
<b>Projektmanagement</b>		0	1	2	3	4	5
Grundsätze des Projektmanagements	(Erfolgsfaktoren, ...)	<input type="checkbox"/>					
Institutionelles Projektmanagement	(Organisation, Rollen, ...)	<input type="checkbox"/>					
Projektmanagement der Prozesse	(Planung, Steuerung, ...)	<input type="checkbox"/>					
Qualitätsmanagement	(insbesondere ISO 9001)	<input type="checkbox"/>					
Risikomanagement	(Analyse, Planung, Maßnahmen)	<input type="checkbox"/>					
Projektleitung	(fachliche und personenbezogene Führung)	<input type="checkbox"/>					
Sonstige, nämlich		<input type="checkbox"/>					

---

**Teil V: Demographische und tätigkeitsbezogene Fragen**

23. Welchem Geschlecht gehören Sie an?

- Männlich       Weiblich

24. Welcher Alterklasse gehören Sie an?

- |                                       |  |
|---------------------------------------|--|
| <input type="checkbox"/> bis 25 Jahre | <input type="checkbox"/> 26-30 Jahre   |
| <input type="checkbox"/> 31-35 Jahre  | <input type="checkbox"/> 36-40 Jahre   |
| <input type="checkbox"/> 41-45 Jahre  | <input type="checkbox"/> 46-50 Jahre   |
| <input type="checkbox"/> 51-55 Jahre  | <input type="checkbox"/> Über 55 Jahre |

25. Welche Abschlüsse haben Sie bisher erzielt? (Mehrfachnennungen möglich)

- |   |  |
|---|--|
| <input type="checkbox"/> Hauptschulabschluss                      | <input type="checkbox"/> Mittlere Reife        |
| <input type="checkbox"/> Fachabitur                               | <input type="checkbox"/> Abitur                |
| <input type="checkbox"/> Fachhochschulabschluss                   | <input type="checkbox"/> Universitätsabschluss |
| <input type="checkbox"/> Doktorgrad                               | <input type="checkbox"/> Kein Abschluss        |
| <input type="checkbox"/> Abgeschlossene Berufsausbildung, nämlich | (bitte angeben)                                |

26. In welchem Jahr haben Sie den letzten dieser Abschlüsse gemacht?

27. Falls Sie einen Studienabschluss besitzen, in welchem Fachgebiet haben Sie Ihren Abschluss erzielt? (Mehrfachnennungen möglich)

- |   |  |
|---|--|
| <input type="checkbox"/> Betriebswirtschaft | <input type="checkbox"/> Volkswirtschaft       |
| <input type="checkbox"/> Informatik         | <input type="checkbox"/> Wirtschaftsinformatik |
| <input type="checkbox"/> Mathematik         | <input type="checkbox"/> Physik                |
| <input type="checkbox"/> Sonstige, nämlich  | (bitte angeben)                                |

28. Wie groß ist Ihre Berufserfahrung in der aktuell ausgeübten Tätigkeit?

Jahre

29. Wie lange sind Sie schon bei itCampus beschäftigt?

Jahre

30. Wie hoch war Ihr Bruttoeinkommen inkl. aller Sonderleistungen in 2003?  
(in Tausend Euro)

- |                                 |                                |                                  |
|---------------------------------|--------------------------------|----------------------------------|
| <input type="checkbox"/> bis 20 | <input type="checkbox"/> 21-30 | <input type="checkbox"/> 31-40   |
| <input type="checkbox"/> 41-50  | <input type="checkbox"/> 51-60 | <input type="checkbox"/> 61-70   |
| <input type="checkbox"/> 71-80  | <input type="checkbox"/> 81-90 | <input type="checkbox"/> über 90 |

31. Welcher Teil dieses Einkommens waren vertraglich vereinbarte variable Anteile?

(Angaben bitte absolut in Tausend Euro oder relativ in Prozent)

32. In welcher Art von Tätigkeit sind Sie bei der Projektarbeit eingesetzt?

(Schätzen Sie bitte den Anteil an der gesamten Projektarbeitszeit)

- 0% Anforderungsanalyse, Softwareentwurf
- 0% Implementierung
- 0% Qualitätssicherung und Softwaretest
- 0% Einführung, Schulung, Kundensupport
- 0% Softwarewartung
- 0% Projektleitung
- 0% Kundenberatung
- 0% Sonstiges, nämlich (bitte angeben)

33. An wie vielen bereits abgeschlossenen Softwareprojekten haben Sie teilgenommen, seit Sie bei itCampus sind?

(bei 0 abgeschlossenen Projekten beziehen Sie sich bitte im Folgenden auf das aktuelle Projekt, ansonsten auf das letzte abgeschlossene Projekt)

34. Wie lange dauert(e) das Projekt von der Vertragsunterzeichnung bis zum Abschluss?

- |   |  |
|---|--|
| <input type="checkbox"/> bis 3 Monate         | <input type="checkbox"/> über 3 bis 6 Monate   |
| <input type="checkbox"/> über 6 bis 12 Monate | <input type="checkbox"/> über 12 bis 18 Monate |
| <input type="checkbox"/> über 18-24 Monate    | <input type="checkbox"/> über 24 Monate        |
| <input type="checkbox"/> Projekt läuft noch   | <input type="checkbox"/> weiß nicht            |

35. Wie groß war/ist der Umfang des Projekts? (Schätzen Sie bitte den von itCampus geschriebenen, nicht auto-generierten Softwareteil in 1000 Lines of Code [KLOC])

- |  |                                       |
|--|---------------------------------------|
| <input type="checkbox"/> bis 10 KLOC                 | <input type="checkbox"/> 11-25 KLOC   |
| <input type="checkbox"/> 26-50 KLOC                  | <input type="checkbox"/> über 50 KLOC |
| <input type="checkbox"/> kein Implementierungsanteil | <input type="checkbox"/> weiß nicht   |

36. Wie viele Mitarbeiter (einschließlich Ihnen selbst und dem Projektleiter) haben an dem Projekt gearbeitet / arbeiten an dem Projekt?

37. Welche Kenntnisse haben Sie für das Projekt benötigt?

(Maximal 7 Nennungen nach Wichtigkeit für den Projekterfolg! Sie können sich an den Kategorien aus den Fragebogenteilen II bis IV orientieren)

---

## Anhang E: Liste der Skills aus dem Skillmanagementsystem von itCampus

- Programmierung
  - Bibliotheken
    - AntLr
    - Castor
    - JAAS
    - JMS
    - JMX
    - JNDI
    - MFC
    - STL
    - Swing
    - SWT
  - Programmiersprachen
    - ABAP
    - C
    - C#
    - C++
    - COBOL
    - Delphi
    - Flash Actionscript
    - Java
    - JavaScript
    - Perl
    - PHP
    - Python
    - SQL
  - Tcl
  - UNIX Shell Scripts
  - VB
- Betriebssysteme
  - Betriebssysteme
    - AIX
    - Linux
    - MAC OS 9, X
    - Solaris
    - UNIX
    - Windows NT, 2000, XP
- Produkte
  - Datenbanken
    - DB2
    - MS SQL Server
    - MySQL
    - Oracle
    - Postgres
    - Power Designer
    - Sybase
  - Application Server
    - BEA WebLogic
    - JBoss
    - Jetty
    - Tomcat
    - WebSphere

- Content Management
  - In4meta
  - NodeVision
  - NPS
  - postNuke/phpNuke
  - Velogate
- Grafik
  - AutoCAD
  - Cinema 4D
  - Director
  - Fireworks
  - Flash
  - Framemaker
  - Freehand
  - Illustrator
  - InDesign
  - Maya
  - Photoshop
  - Premiere
  - Studio Tools
- Office Anwendungen
  - Crystal Reports
  - MS Access
  - MS Excel
  - MS Word
  - Open Office
  - Visio
- Know-How
  - Softwareentwicklung
    - .NET
    - COM+
    - DB-Design
    - EJB
    - OOD
    - OOP
    - Servlets
  - Web
    - CSS
    - Docbook
    - DTD
    - HTML
    - SOAP
    - SVG
    - WAP
    - XML
    - XML-Schema
    - XPath
    - XSLT
  - Grafik
    - Animation
    - Druckvorstufe
    - Fotoproduktion
    - Illustration
    - Kalligrafie
    - Videoproduktion

- 
- Server
    - ADS
    - Apache
    - Blind
    - CommuniGate
    - CVS
    - HP Open Mail
    - LDAP
    - Lotus Notes
    - MS Exchange
    - MS IIS
    - NFS
    - Postfix
    - Samba
    - Samsung Contact
    - Sendmail
    - VMWare GSX
  - Prozesswissen
    - ERP – CRM
    - ERP – Fibu
    - ERP – MaWI
    - ERP – PPS
    - Instandhaltung
    - Prozessanalyse
    - Prozessoptimierung
  - Sicherheit
    - Antivirensoftware
    - Backup
    - Firewallsysteme
    - Kryptologie
    - NIDS
    - SSL / TSL
    - VPN