Proactive Ontology-Based Content Provision in the Context of e-Learning

Dissertation
zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von: Dipl.-Inform. Steffen Mencke
geb. am 12.08.1978 in Brandenburg (Havel)

Gutachter:
Prof. Dr. Reiner Dumke
Prof. Dr. Dietmar Rösner
Prof. Dr. Stanimir Stoyanov

Promotionskolloquium: Magdeburg, Germany, 16.12.2008

Mencke, Steffen
Proactive Ontology-Based Content Provision in the Context of e-Learning
Dissertation,
Otto-von-Guericke University of Magdeburg,
2008.

Contents

Abstract

The provision of multi-type content was always a primary task of computer-supported information systems. In the days of the overwhelming data and information burst of the 21st century, new and innovative handling approaches are necessary. With every single day, the importance of the statement of John Naisbitt increases, who once said that the human mankind is drowning in information, but starving for knowledge. Because of this, mechanisms are necessary to adequately present content. Therefore, this dissertation focusses on solutions that base on technologies of the Semantic Web and are targeting a proactive, holistic quality-driven content provision.

For this goal it was necessary to define the process-oriented frame of this approach and to adequately model the subprocess of proactive content provision by taking into consideration the intended semantic and quality-related orientation. By this, it was possible to preserve the relationship of the specific solutions. To show the practical relevance, a primary focus was laid on the provision learning content within the domain of e-Learning.

For the first preparation step, presented solutions are related to the analysis of relevant processes and process description approaches. This was the basis for the derivation of ontology-based descriptions of didactical expertise and e-Learning process models. Additional topic where the individualization based on user models for the lifelong learning as well as the proactive preparation of content for further processing.

The second main phase was about the ontology-based provision of individualized content being enriched with additional information. The proactive presentation by graphical user interfaces was enhanced by agent technology concepts. Therefore, a common methodology as well as example scenarios have been developed.

This work is completed by the presentation of manifold application examples of the developed solutions as well as conceptional proactive systems.

The present dissertation is an important contribution within the area of tension of the proactive ontology-based respectively throughout quality-oriented provision of content. Especially for the domain of e-Learning methodologies, application scenarios, architectures and prototypical implementations were developed to implement and improve chosen aspects of the presented, innovative and quality-oriented QuaD^2-Framework.

Zusammenfassung

Die Bereitstellung von Inhalten jeglicher Art ist seit jeher eine der primären Aufgaben computergestützter Informationssysteme. In den Zeiten der Daten- und Informationsflut des 21. Jahrhunderts sind dafür neue, innovative Ansätze notwendig, da heute mehr denn je die Aussage von John Naisbitt gilt, dass die Menschheit in Informationen ertrinkt, aber nach Wissen hungert. Aus diesem Grund sind Mechanismen erforderlich, Inhalte adäquat bereitzustellen. Die vorliegende Dissertation fokussiert im Rahmen dieser Thematik auf Lösungen, die auf Technologien des Semantic Web basieren und auf eine proaktive, durchgehend qualitätsgetriebene Inhaltsbereitstellung ausgerichtet sind.

Für dieses Ziel war es notwendig, den prozessorientierten Rahmen dieser Vorgehensweise zu definieren und den Teilprozess der Inhaltsbereitstellung unter Berücksichtigung der intendierten semantischen und qualitätsbezogenen Orientierung geeignet zu modellieren. Mit dieser Vorgehensweise wurde die Aufrechterhaltung des Zusammenhangs der spezifischen Lösungen untereinander erreicht. Für die Darstellung der Praxisrelevanz wurde auf die Bereitstellung von Lerninhalten im Bereich des e-Learnings fokussiert.

Zu den Lösungen gehörten in der ersten Vorbereitungsphase die Analyse relevanter Prozesse und Prozessbeschreibungsmöglichkeiten, sowie die daraus abgeleitete ontologiebasierte Beschreibung didaktischer Expertise und e-Learning-Prozessmodelle. Weitere Schwerpunkte waren ihre Individualisierung auf der Basis von Anwendermodellen im Rahmen lebenslangen Lernens, sowie die proaktive Bereitstellung von Inhalten zur Weiterverarbeitung.

Die zweite wesentliche betrachtete Phase beschäftigte sich mit der ontologiebasierten Bereitstellung von individuell angepassten und mit Zusatzinformationen angereicherten Inhalten. Die proaktive Präsentation durch grafische, auf der Agententechnologie basierender Nutzerschnittstellen, fällt ebenfalls in dieses Themengebiet. Dafür wurden sowohl eine allgemeine Methodologie, als auch Beispielszenarien entwickelt.

Abgerundet wird diese Arbeit durch die Präsentation vielfältiger Anwendungsbeispiele der vorgestellten Lösungen, sowie konzeptioneller proaktiver Systeme.

Die vorliegende Dissertation stellt damit einen wichtigen Beitrag im Spannungsfeld der proaktiven ontologiebasierten bzw. der durchgehend qualitätsorientierten Bereitstellung von Inhalten dar. Speziell für den Bereich des e-Learnings wurden Methodologien, Applikationszenarien, Architekturen und prototypische Implementierungen vorgestellt, um ausgewählte Aspekte des präsentierten, innovativen und qualitätsorientierten QuaD^2-Frameworks umzusetzen bzw. zu verbessern.

Acknowledgments

Working on, writing and finishing a Ph.D. is an adventure that took me into the deepest universe of science with black holes slowing me down and wormholes pushing the speed of my learning. The beauty of the New and the wisdom of the Known form a unity, whose essence resulted in this dissertation and the experiences I made. It is not only boldly going where nobody has ever been before, it is also to know the roots where you are coming from.

The most important roots, are the people one can belong to. At the very first place, this is the family. I owe the deepest thanks for their extreme support and patience – especially from my wife Nicole. Without her I am nothing.

Although many teachers, professors, reviewers and mentors guided the preparation of this journey, traveling between the stars of open questions, nebula of unsolved mysteries and borders of the mind is a way that can be hardly gone without a compass.

So, I want to thank my adviser Prof. Dr.-Ing. habil. Reiner Dumke who was my compass, who called me in his group at the University of Magdeburg and gave me the opportunity and substantial support to make my Ph.D. I learned many things from him – not only professional expertise, but also many invaluable truths about the amenities and dangers of the world of research, science and academics.

A next thank goes to Prof. Dr. Stanimir Stoyanov. His readiness to evaluate my work provided substantial parts of my space trip's finishing line. I hope for many revealing conversations in the future.

Third, I want to thank Prof. Dr.-Ing. Cornelius Wille. His support and open mind were factors that I do not want to have missed. For him, I formulated the sentence: "Agents are not a new concept but their use in the field of ontology-based content provision constitutes a basis for consequential advances."

A space tripe alone is a hard trip. So, it was very helpful to talk with many friends and colleagues – thank you all! Without the possibility to be complete, I especially want to mention: Dr. rer. nat. Fritz Zbrog: I liked our technical term creation sessions. Dmytro Rud: Did you found your sentence in my thesis? Stephan Vornholt: Brainstorming, eh?! Martin Kunz: Quality is the future!

Finally, I want to thank all the members of the Software Engineering group at the University of Magdeburg as well as all my relatives and friends. We will meet behind the next star ... and dream the dream of the stars.

1 Introduction

1.1 Overview

Content provision is a fundamental process of the current information society – multimedial knowledge is presented on Web pages, e-Learning content can be learned in educational systems, virtual objects are visualized in Virtual Reality applications or Web services are used in business processes. The importance of content provision is beyond any controversy.

For this process exist multiple points of view with different focuses and different levels of detail (e.g. defined by [Government of Victoria, 2003], [Vidgen et al., 2001], [Holst, 2003], [Dias and Bidarra, 2007], [Roure et al., 2001]). In this work, proactive content provision is arranged within the process flow shown in Figure 1.1.

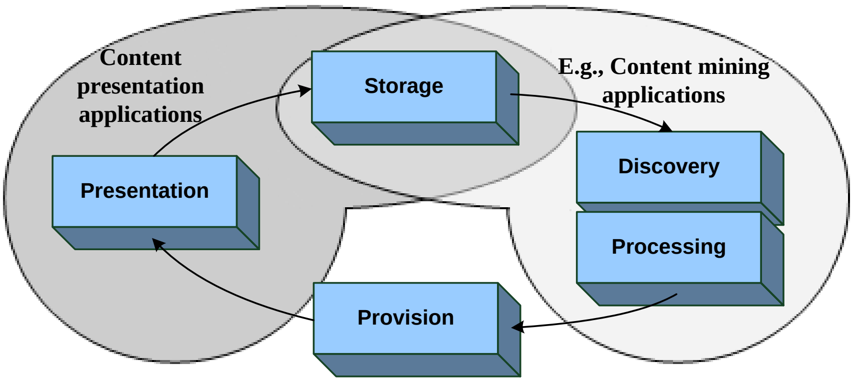


Figure 1.1: Content Provision in the Content Lifecycle

The increasing amount of available application vendors and content providers requires and enables ongoing research and development to meet the diverging requirements. Every consumer expects special solutions and products in order to meet his/her individual needs. Only high-quality products have the capability to *survive* on the international market.

Especially the provision of information content is targeted in this work. It is the step of the information life cycle where its value is at highest [Tallon and Scannell, 2007]. Expending amounts of information must be prepared, pre-selected and adaptively presented to the consumer in order to prevent an "information overflow".

This dissertation targets these problems by the usage of Semantic Web technologies to enable proactive high-quality content provision applications. Technologies of the Semantic Web, as envisioned by Tim Berners-Lee [Berners-Lee et al., 2001], are the one approach that is seen to shift the current Web 2.0 to a Web 3.0. Metadata integration and the semantic definition of content offer a tremendous potential [Hendler, 2008].

In this context e-Learning is chosen as a special use case. Education is a very advantageous application area ([Sampson et al., 2004], [Cristea, 2004], [Aroyo and Dicheva, 2004], [Aroyo et al., 2004], [Henze et al., 2004], [Jovanović et al., 2007], [Shih et al., 2008]): "The Semantic Web opens a number of new doors and multiplies the prospects of Web-based learning." [Devedžić, 2006]. Especially ontologies as a major technology can augment content provision applications. They "fit so well with education, by building a strong platform for it, by bringing reflection and by interweaving everything." [Devedžić, 2006].

The purpose of proactivity adds another quality dimension. Waiting for something to happen is not always the best solution to solve a problem. It can be more efficient to take the initiative – to be proactive. So, proactivity is controlling a situation by causing something to happen rather than waiting to respond to it after it happened [American Heritage, 2003b]. By being proactive, an earlier reaction to anticipated events as well as the improved possible adaptation to certain system/user requirements becomes possible.

The proactive content provision is a process linking the phases of content preparation and content presentation/publication. It refers to the technological and methodological basis of making content accessible to the consumer. In this work, semantic technologies are used to enable the proactive provision of content following a quality-driven process.

1.2 Contribution

The general goals that are targeted with this thesis are the definition of proactive content provision within a more abstract frame as well as the development of a general framework that has a throughout focus on quality. Existing approaches mainly focus on selection Quality of Service attributes and do not provide a holistic consideration of quality. For this purpose, technologies with proactive and semantic characteristics should be used to provide substantial contributions.

The general outcome of this dissertation is briefly summarized as:

1. This work has a focus on the symbiosis of semantic technologies and e-Learning to provide new approaches for consequential advances in proactive content provision. Results are new metrics, tools and application scenarios.
2. Furthermore, an analysis of content provision frameworks in the e-Learning domain is performed. The result is a novel framework that has a focus on proactive technology. It is used to classify existing work in this field.
3. With a background on quality and quality measurement, a novel quality-driven content provision framework is proposed. Several aspects are analyzed more detailed in order to propose new contributions in these fields – namely initialization and content presentation.
4. The applicability of the presented ideas is promoted by the presentation of architectures, prototypical tool implementations and application scenarios for proactive content provision.

1.3 Outline

In order to enable an appropriate dissertation *content provision*, this work is structured and enriched with metadata as follows. Due to technical limitations a *proactive* realization was not possible – so, the prospective reader can find and read the interesting information in the subsequent chapters.

Chapter 2 provides the fundamental information being necessary in order to understand this dissertation. It focusses on content in general and introduces e-Learning content as a use case. Furthermore, a placement of the content provision process in a higher-order process is introduced. Other major aspects are the brief description of chosen approaches of semantics-based content assembly as well as the discussion of a possible, fundamental proactive implementation technology.

Chapter 3 introduces a classification framework with a special focus on proactive e-Learning content provision. It bases on established standards and extends their expression capabilities to enable the depiction of special properties. The framework is used to classify existing approaches within the chosen application domain.

Chapter 4 presents the QuaD^2-Framework. This is a novel, general framework for the automatic provision of high-quality content in every field of application. Existing quality-related information can be reused to optimize this aggregation of content to thereby always provide the best possible combination.

Chapter 5 targets the initialization phase of the introduced QuaD^2-Framework. It describes how content provision processes can be semantically described. Furthermore, it explains an approach for the integration of user-related information to enable individualized adaptation throughout lifelong processes like learning.

Chapter 6 elaborates the proactive presentation of content. It shows how semantic information and technologies can be used to enhance the presentation subprocess of the QuaD^2-Framework. Another key element of this chapter are proactive user interfaces.

Chapter 7 reviews two developed proactive applications scenarios. Namely, proactive educational courses as well as a proactive class schedule are presented.

Chapter 8 summarizes this dissertation, reflects achieved results and provides an outlook to further work that can be addressed on the basis of this work.

2 Proactive Content Lifecycle

"There is nothing good: Unless one does it." Erich Kästner

Waiting for something to happen is not always the best solution to solve a problem. It can be more efficient to take the initiative – to be proactive.

Definition 1

Proactivity *is controlling a situation by causing something to happen rather than waiting to respond to it after it happened [American Heritage, 2003b].*

The origin of the word is Latin: *pro* means *before* or *for*, meanwhile *activus* is *practical* or *active*, respectively. The main targeted advantages that can be anticipated by being proactive in the current domain are: the earlier reaction to anticipated events as well as the improved possible adaptation to certain system/user requirements.

2.1 Preamble

Proactivity is a characteristic of behaviors that can lead to more effectiveness. Certain requirements need to be met to enable an iterative process of proactivity. Namely, a model of prediction as well as a model of observation are at the borderlines of high-quality proactive content provision. Chosen details to answer this first questions are presented later in this chapter, meanwhile the second set of questions is targeted in the subsequent chapters.

Model of prediction: Because proactivity means that a future event must be anticipated, predictions are needed. For the domain of content provision it is necessary to know and to appropriately model:

- What (type of) content will be provided?
- What are the requirements of the target of provision that must be met?
- What characteristics of the user and the content to be provided need to be observed?

The first aspect is necessary, because for each content type different technologies for its provision are needed. It is a difference, if multimedial content in an e-Learning system or Web Services[1] should be provided. This dissertation focusses on high-quality multimedial content and its provision.

[1] A Web Service is a software system identified by a URI [RFC 2396], whose public interfaces and bindings are defined and described using the eXtensible Markup Language (XML). Its definition can be discovered by other software systems [World Wide Web Consortium (W3C), 2004e].

Furthermore specific requirements of the target of provision (user/system) exist and must be met. A model must be designed that is capable to depict user or system requirements and thereby provide a basis to fulfil the needs of the target of provision with appropriate content. For the provision of content, especially for e-Learning systems, such requirements are solved with user models.

Those user models additionally model the characteristics of the target of provision. To be able to identify the appropriate content for the special needs, semantic metadata and annotation technologies are needed as a basis of the provision. Those concepts are related to the Semantic Web as envisioned by Tim Berners-Lee [Berners-Lee et al., 2001].

Model of provision: Once, time and scope of proactivity are identified, the content can be provided. Individualization – the adaptation of content to certain users – is one major key challenge in this domain. The key requirements are listed below:

- What to provide?
- What is an appropriate provision process?
- What are possible adaptation mechanisms?
- What is an appropriate technological basis for implementation?
- How to ensure quality?

Content to be provided, user models and adaptation mechanisms provide the already existing technological basis for possible solutions. For the implementation itself, certain technologies are feasible. Next to classic client/server implementations, agent technology promises to be valuable for certain implementation aspects, especially for proactivity due to its inherent proactive and autonomous characteristics.

The throughout focus on quality during a content provision process is not described in detail in literature so far. Due to this reason, the definition of a new quality-oriented process is in the scope of this work. The subsequent chapters provide detailly described ideas, approaches and implementation prototypes for selected parts of the identified quality-oriented content provision process with focuses on semantic support as well as proactive mechanisms. They form the core of this work.

Model of observation: In order to enable future improvements and ensure ongoing high quality, the execution of the content provision process as well as the content need to be continually observed and analyzed. Those aspects are beyond the scope of this work.

- How to define quality?
- When to analyze content and process execution?

In this work, a major focus is laid on the model of provision – it will be described how certain content can be proactively provided with the help of semantic means. Therefore, this chapter describes needed fundamental technologies as well as chosen similar approaches after a placement of the proactive content provision in its frame.

2.2 Placement of the Proactive Content Provision

Content provision as a process is part of the content lifecycle. Following [Government of Victoria, 2003] its subprocesses are development, quality approval, publish/provision, unpublish and archive. Vidgen et al. list another cycle consisting of creation, review, store, publish/exchange/provision, archive and destroy [Vidgen et al., 2001]. Archiving, creation, deliver/provision, distribution and consuming is the underlying lifecycle process of the Sun ONE Content Services Platform [Holst, 2003]. Other resources reveal a more general view only focusing on creation, storage and provision [Dias and Bidarra, 2007].

So, there exist multiple points of view with different focuses and different levels of detail. For this work, proactive content provision is arranged within the process flow shown in Figure 2.1.

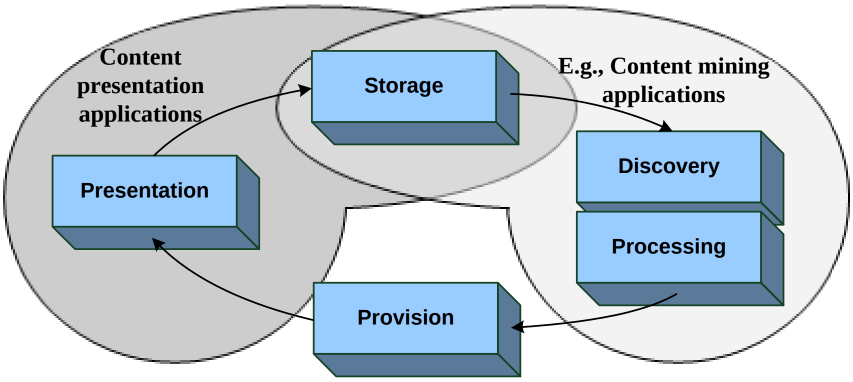


Figure 2.1: Content Lifecycle

Proactive content presentation automatically adapts the layout of content to special requirements of the user or to restrictions defined by his/her environment (e.g. bandwidth or display limitations of mobile clients).

The storage of content is the basis for its presentation in the context of content display applications like Content Management Systems (CMS) or Learning (Content) Management Systems (L(C)MS). The displayed and thereby necessarily stored content is also the starting point for content mining/access applications like search machines, data/Web mining tools etc.

Content discovery may involve such content mining/access applications. Another input is the manual authoring of content. A possible content processing step supports the preparation of content to be provided by certain tools or applications.

The proactive content provision is a process, linking the phases of content preparation and content presentation/publication. It refers to the technological and methodological basis of making content accessible to the consumer. Multiple aspects are involved in this step [Doyle et al., 2003], e.g.:

- Document management
- Content categorization
- Content delivery
- Content personalization
- Content platforms
- Content security
- Content syndication
- Content digital rights management

Especially for the first four aspects, Semantic Web technologies as well proactive application characteristics constitute a basis for consequential advances.

Document management and content categorization are aspects focusing on strategies to annotate documents and content with appropriate metadata to enable advanced retrieval mechanisms. Thereby, reuse, interoperability and accessibility are enhanced.

Content delivery involves technologies to make created content accessible. Next to technical aspects (like storage, networks, Web servers, ...) it refers to the process of delivery itself. Proactive characteristics can make it more efficient and adapted to special needs. Ontologies can provide additional support.

Content personalization is about manifold technologies later described in this work. Individualized content fits better to the different needs of the particular users. Targeted advantages are for example faster information acquisition, deeper understanding or increased user interest. Proactive and semantic technologies can provide a fundament for advanced solutions compared with the currently existing ones.

The remaining aspects are out of the scope of this work. Content platforms are larger applications comprising many functionalities to enable the other facets of content provision as described above. Content security is about the safe distribution of content, meanwhile content digital rights management exists to protect intellectual property. Content syndication tries to analyze and provide new content distribution channels like online marketplaces.

Chosen aspects in order to highlight challenges and opportunities of proactive content provision in contrast to “classic” content provision are presented in Table 2.1

After the placement of the proactive content provision, the next step is the description of possible content representation types.

"CLASSIC" CONTENT PROVISION	PROACTIVE CONTENT PROVISION
Content assembly based on user request	Content assembly is also based on user request but proactively individualized and enriched.
Enhanced provision based on predefined rules	Provision enhancements based on semantically-defined methodologies
Static provision	Behavioral system aspects for dynamic updates
Personalization based on predefined rules	Intelligent reasoning for improved results
A-priori knowledge about content objects is needed	Needed knowledge is determined and proactively provided by the system
Static content provision	Behavioral system aspects for dynamic updates

Table 2.1: "Classic" Content Provision vs. Proactive Content Provision

2.3 Selected Content Representations

Content can be data, information or knowledge encoded in a particular way. While data is the raw material of information and needs a context for a meaning, information is an interpretable message already being related with a meaning and thereby being relevant for the receiver [Endres, 2004]. Knowledge is even more complex – it can be defined as the general awareness or possession of information, facts, ideas, truths, or principles gained through experience or association [American Heritage, 2003a].

Content can have multiple characteristics. In its glossaries, the World Wide Web Consortium (W3C) defines it as:

Definition 2

> ***Content*** *is used to mean the document object as a whole or in parts [that is to be presented] to the user through natural language, images, sounds, movies, animations, etc. [W3C Glossary, 1999a].*

Another definition from ContentWatch, an organization for the promotion of content management theory and practice, defines content as:

Definition 3

> *Raw information becomes* ***content*** *when it is given a usable form intended for one or more purposes. Increasingly, the value of content is based upon the combination of its primary usable form, along with its application, accessibility, usage, usefulness, brand recognition, and uniqueness [Boiko, 2005].*

Especially for the targeted process of provision, its nature is similar to the classic communication mechanism as described by Shannon and Weaver [Shannon and Weaver, 1949] and depicted in Figure 2.2.

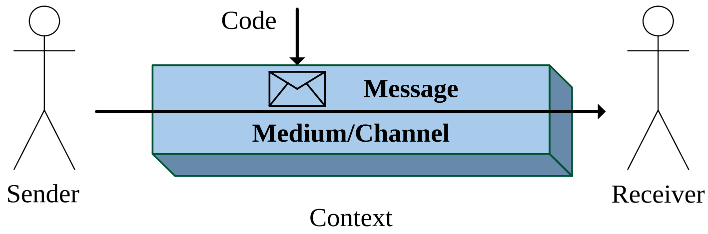


Figure 2.2: Classic Model of Communication Theory [Ferber, 1999]

This model consists of a sender, who encodes the message to be sent with a language and sending it via a communication medium/channel to a receiver who decodes it. The situation where both, sender and receiver, are placed in is called the context of the communication. Normally, communication's intentions are information and conversation.

In this context, the provision of content only focuses on information. Nevertheless, a consumer, a provider and a transportation medium of content exist. In this work, the consumer is the target of the content provision process (passive consumer) or the one acquiring the content (active consumer), respectively. The provider of content is a computer-based system providing interfaces for content input, output and maybe management. Chosen examples are Web pages, e-Learning systems or Web Service providers. Thereby, the context of content provision is defined, too. It is any computer-supported environment, maybe enhanced to distributed characteristics like the World Wide Web. The content provision channel/medium also results from the context – it can be a single workstation or distributed computers with network connections.

Content can be analyzed taking into account various points of view. Chosen ones are listed here.

Within the described scenario many possible usage options exist. This **domain-specific dimension** can be characterized on different levels of abstraction. Information systems, interaction systems and support systems are common domains, meanwhile e-Learning, e-Business, e-Health and e-Government are specific areas.

Another possible point of view is the **technological dimension** describing implementation architectures for content provision. Some important ones are briefly listed as client-server, Web Services, Grid and agents.

The **application dimension** describes the environment in/for which the content is provided. It can be distinguished in individual vs. organizational and covers content consumption at home, on the intranet, on the Internet, etc.

Figure 2.3 visualizes these identified characterization dimensions of content provision. In this example three possible characterizations of content provision are sketched – (a) client-server implementation of e-Government content on the Internet (b) client-server implementation of e-Learning content on a local intranet and (c) agent-based implementation of e-Learning content on the Internet.

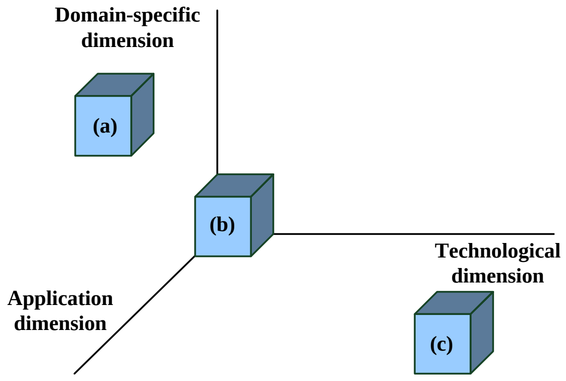


Figure 2.3: Characterization Dimensions of Content Provision

This scenario of content provision supports a possible range of content. This **technical dimension**, taking into account media types, is described below.

2.3.1 Classification of Content

For the identified context different types of content can be identified. A possible categorization evolves based on the different existing media types. Those types derive from the certain conventional human senses as there are sight, hearing, taste, smell and touch. To complete this list, thermoception, nociception, equilibrioception, proprioception/kinesthesia and the sense of time are sometimes seen as the other human senses.

The currently most used media types target the visual and auditive human senses. Chosen examples are:

- Text,
- Score,
- Graphic and picture,
- Object,
- Animations,
- Audio data (voice, sound, . . .) as well as
- Film and Worlds.

Thereby, the first five are discrete and the other are continuous information types [Dumke et al., 2003]. Another possible classification of presentation media content was introduced by Meder (see Figure 2.4).

Content in its various types can differ in format and structure.

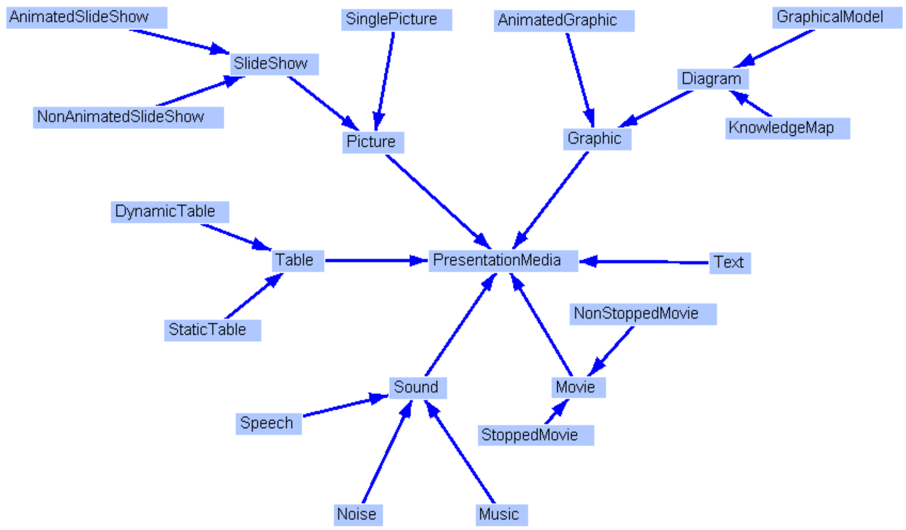


Figure 2.4: Presentation Media According to Meder [Meder, 2006]

2.3.1.1 Format of Content

The format of content thereby is the way it is encoded for the use by computer systems [Boiko, 2005]. The media types defined by the Dublin Core Metadata Initiative (DCMI) are collection, dataset, event, image, interactive resource, movingImage, physicalObject, service, software, sound, stillImage and text [Dublin Core Metadata Initiative, 2008]. The Internet Corporation for Assigned Names and Numbers lists content formats according to the MIME media types [The Internet Corporation for Assigned Names and Numbers, 2007]. They also provide an exhaustive introduction to possible content formats:

- Application, e.g. dicom, javascript, mp4, rdf+xml, smil, xhtml+xml, ...
- Audio, e.g. 3gpp, mobile-xmf, mpeg, ...
- Example, any subtype of other MIME type
- Image, e.g. gif, jpeg, tiff, ...
- Message, e.g. http, rfc822, ...
- Model, e.g. iges, mesh, vrml, ...
- Multipart, e.g. digest, mixed, report, ...
- Text, e.g. css, csv, html, plain, rtf, sgml, xml, ...
- Video, e.g. 3gpp, JPEG, mp4, mpeg, ...

Special, well known encoding standards are, e.g. Unicode for textual elements [Unicode Consortium, 2008], MP3 for audio [Fraunhofer-Institut für Integrierte Schaltungen IIS, 2008], X3D for Virtual Reality scenarios [Web 3D Consortium, 2004] or MPEG-4 for video [International Organisation for Standardisation, 2002].

2.3.1.2 Structure of Content

The problem of every content description and provision is the adequate syntactical description of the intended semantics. Therefore, different options exist [Kernchen, 2005]. Content can be provided in different ways – either unstructured or structured. The difference between them is the non-atomic nature of structured content. It is composed out of content atoms [Weglarz, 2004]. Thereby, the problem of unstructured content – its multiple possible interpretation [Spitta, 2006] – is intended to be avoided. In the following, a brief overview about chosen text-based examples for this classification is presented.

An adequate content description is the fundament for every content processing process like content provision. The basis of a content description is a language.

Definition 4

*A **language** is a system of signs for the articulation/exchange of information, where semiotics (relation of signs and the things they refer to), pragmatic (relation of signs to the impact to their users) and syntactics (rules of creating formal structures with them) are aspects of signs ([Saussure, 1915] and [Welte, 1974]).*

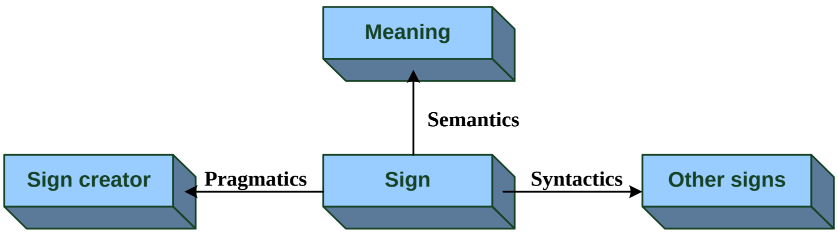


Figure 2.5: Aspects of Signs

The syntactical rules are used to create formal structures by arranging the signs – for textual content that will result in words and sentences.

Unstructured content reveals no or only minor structuring elements. Text-based examples result in simple highlighting like changed font types, empty lines or paragraphs.

Structured content achieves this goal by one- or two-dimensional characteristics. For texts that can be ([Pentland et al., 1999], [Kernchen, 2005]):

- One-dimensional
  - List
  - Classification
  - Pseudocode
- Two-dimensional
  - Table
  - Matrix

Graphical structuring is a next approach to handle content. Such representations result in certain diagrams ([Kernchen, 2005]), e.g.:

- **Pictograms:** The particular components are described by pictorial symbols. Relations are determined by lines or arrows.
- **Decomposition diagrams:** Organigrams, Venn diagrams, box diagrams as well as bracket diagrams describe the decomposition of a structure into its subcomponents.
- **Structure graphs:** This graphical description is a labeled connected graph of nodes and edges for the definition of components and relations.
- **Jackson diagrams:** Jackson diagram are tree-shaped structures for the description of decompositions with additional expressions for choices, sequences and repetitions.
- **Syntax diagrams:** This kind of diagram also described a decomposition. It can be used for the graphical representation of the syntactics of a rule set.
- **ER- and EER diagrams:** *Entity Relationship* diagrams and *Extended Entity Relationship* diagrams extend the expressiveness of structure graphs by an additional semantic. Here, cardinalities as well as multiple relationships are possible.

Formal structuring of content can be achieved by using logical formalisms and grammars [Kernchen, 2005]. The propositional logic as a logical formalism describes complex aspects as the connection of elementary expressions. This relation can be achieved for example by conjunctions, disjunctions, negations and implications. Propositions describing elementary expressions by variables having boolean values, can be used for checking certain expressions. The First Order logic is an extension of the propositional logic by predicate and function symbols. Additionally, variable parameters can be bound to value ranges by quantification.

Formal grammars are mathematical models of grammars for the creation of formal languages [Sipser, 1996]. They are devoted as Chomsky-Grammars and defined as:

Definition 5

*A **Chomsky-Grammar** is a 4-tuple (N, T, Z, P) with*
- *N as the (not empty, limited) set of syntactical variables (non-terminal symbols),*
- *T as the (not empty, limited) set of terminal symbols,*
- *Z as a special syntactical variable (start symbol) and*
- *P as the (not empty, limited) set of creation rules.*

Creation rules are replacing rules, where the syntactical variable on the right side of a rule R_j is identified in other rules R_i (i = j + 1, j + 2, ...) on the left side and is replaced with the right side of R_k, if it has a corresponding part within the rule R_k. A thereby created language is the complete set of all permitted terminal symbols that can be derived from the start symbol by using the given creation rules.

Regular expressions have the same expressiveness as Jackson diagrams. It is a special type of grammar – a Type-3 grammar [Sipser, 1996].

Definition 6

*A **regular expression** follows this rules:*
- *Every symbol of a given set of symbols is a regular expression (atom).*
- *With R1 and R2, (R1 | R2) is a regular expression, too (alternative).*
- *With R1 and R2, R1 * R2 is a regular expression, too (composition).*
- *With R, R* is a regular expression, too (closure).*
- *There are no more rules for the creation of a regular expression.*

As shown, many types of content exist and can be coded in multiple ways. The presented, very basic approaches represent a brief overview. Extended, text-based approaches may for example covered by markup languages like HTML [World Wide Web Consortium (W3C), 1999], XML [World Wide Web Consortium (W3C), 2003a] or the even more advanced Web Ontology Language OWL. A complete description is not the scope of this work.

2.3.2 Semantic Support

Semantics (from the Greek word *semantikos* = significant) in general is the meaning of something or more specifically the study of meaning [Encyclopædia Britannica - Online, 2008]. Often, additional information is needed to shift from information processing to knowledge processing. Those semantic annotations provide the technological basis for many advanced applications.

Semantic support is needed for different reasons, e.g.:

- To make the Web (and other information providing systems) more machine-understandable [Heflin and Hendler, 2001]
- To provide an infrastructure for intelligent agents [Hendler, 2001], [Cost et al., 2002]
- To explicitly declare knowledge for access, integration and extraction purposes [Gómez-Pérez and Corcho, 2002]
- To support automation, integration and reuse across applications and domains [Boley et al., 2001]
- To make Web Services computer interpretable [Narayanan and McIlrath, 2001]

The realization of annotations ranges from simple metadata to the definition of semantics in the vision of a Semantic Web [Antoniou and van Harmelen, 2004].

2.3.2.1 Metadata

Normally, content being sufficient for a human reader is not formatted satisfactory for computer-oriented processing. Instead of making the computer more intelligent for this information access purpose, the Semantic Web goes a different way. It introduces annotations for content description in the similar way like an HTML-page (The HyperText Markup Language [World Wide Web Consortium (W3C), 1999] is a markup-based publishing language for the Web.) is enriched with formatting information for presentation. Therefore, the most important goals of annotations are retrieval (information/content search), identification (information/content distinction) and access (use of information/content). This additional information can be used as a first step for improved systems and processes. More advanced approaches add semantic information and result for example in using ontologies.

Definition 7

Metadata *are "data about data [...], including but not limited to authorship, classification, endorsement, policy, distribution terms, and so on. A significant use for the Semantic Web" [W3C Glossary, 1999b].*

For almost every domain, purpose, resource/content type and specific features metadata schemes exist or can be created, respectively [Kelly, 2006]. Chosen metadata schemes are listed in Table 2.2.

NAME	TYPE	LINK
Dublin Core (DC)	All domains, resource types, and subjects	`http://dublincore.org/`
Data Documentation Initiative (DDI)	Social sciences, datasets	`http://www.icpsr.umich.edu/DDI/`
Anglo-American Cataloguing Rules (AACR2)	Library resources	`http://www.aacr2.org/`
Encoded Archival Description (EAD)	Archives	`http://www.loc.gov/ead/`
ISAD (G)	Guidelines for the preparation of archival descriptions	`http://www.hmc.gov.uk/icacds/eng/ISAD(G).pdf`
MARC 21	Libraries, bibliographic records	`http://www.loc.gov/marc/`
RSLP Collection-level description	Collections of all subjects, domains and types.	`http://www.ukoln.ac.uk/metadata/cld/`
SPECTRUM	Museum objects	`http://www.mda.org.uk/spectrum.htm`
Text Encoding Initiative (TEI)	Digital texts	`http://www.tei-c.org/`
VRA Core 3.0	Visual art images	`http://www.vraweb.org/vracore3.htm`

Table 2.2: Metadata Schemes [Kelly, 2006]

Every metadata scheme should be analyzed, e.g. in terms of granularity, interoperability, support, growth, extensibility, reputation, ease of use and existing experience to determine its usefulness for the targeted purpose ([Kelly, 2006], [Beall, 2006]). The extraction of metadata is an own field of research and not targeted in this work. Introductory information can be for example found in [Sheth et al., 2002a], [McCallum, 2005] or [Davies et al., 2006] and subsequent publications.

Exemplified, one of the most common metadata schemes – *Dublin-Core-Schema* (DC) of the *Dublin Core Metadata Initiative* (DCMI) [Dublin Core Metadata Initiative, 2008] – is described below.

The DCMI's goal is the development of interoperable metadata standards and specialized metadata languages for the intelligent information search in digital resources [Dolog et al., 2003a]. Thereby, the creation and maintainability of metadata should be supported. Additionally, they focus on the clearness of the semantic of resources across language borders. To be international as well as easy extendibility are other intended goals [Dublin Core Metadata Initiative, 2008].

Dublin Core has two levels: simple (*Simple Dublin Core*) and qualified (*Qualified Dublin Core*). Every *Simple DC*-element can be described with up to fifteen attributes. That are e.g. title, identifier, language and comments. *Qualified Dublin Core* has an additional element as well as certain refinement options for the better description of semantics. More detailed information can be found in [Dublin Core Metadata Initiative, 2008].

There exist three principles for the description of relationships among metadata:

- **One-to-one principle:** Metadata describe the instance of a resource. That means that a picture as well as its digitalization have their own metadata. The relationship between them should be depicted within the description.
- **Dumb-Down principle:** This principle defines the refinements of the *Qualified Dublin Core* being optional. The resource's semantics is not altered.
- **Appropriate values:** This aspect describes a trade-off between machine-readability and the readability for a human reader.

Dublin Core supports the metadata description on an abstract level. For special domains extensions are necessary to increase the expressiveness.

2.3.2.2 Semantic Web

The Semantic Web is no alternative to the current World Wide Web, but a logic extension. The machine-accessibility, as envisioned by Berners-Lee [Berners-Lee et al., 2001], as well as the implementation of metadata within and about the Web are necessary. They are a next step following the already outlined structured description of content.

In [Daconta et al., 2003] four different levels of data handling towards the Semantic Web are described.

- **Text and Databases:** Such data sets are application-specific. The automated usage is not always intended and automatic reasoning is rarely possible.
- **XML documents of a certain domain:** Thereby, application independence can be achieved for a special domain. Vertical XML standards support data exchange between applications of the same domain.
- **Taxonomies[2] and documents with several standards:** Data representations with taxonomies and multiple standards support the easy search and combination of data.
- **Ontologies and rules:** Ontologies and rules basing on them, allow the extraction of new knowledge based on existing data by logic reasoning. The data is atomically described with its relationships.

Figure 2.6 shows the layers of the Semantic Web design and vision according to Tim Berners-Lee. Every layer provides the necessary functionality for the layers above. XML is used to basically encode documents and RDF to define simple statements about the resource. They form the basis for more powerful ontology languages for more complex relationships. On the logic layer additional information can be defined to write application-specific declarative knowledge. The representation of proofs and proof validation can be performed on the proof layer involving a deductive process. The trust layer on top introduces trust for agents, agencies and consumers by defining digital signatures and other kinds of knowledge [Antoniou and van Harmelen, 2004].

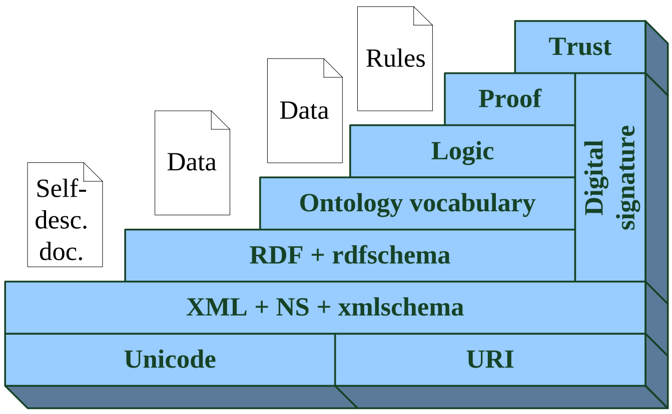


Figure 2.6: A Layered Approach to the Semantic Web [Berners-Lee et al., 2001]

2.3.2.3 Ontologies

Ontologies are a fundamental concept of the Semantic Web. The word *ontology* is originally Greek: *ontos* for being and *logos* for word [Devedžić, 2006]. It is the study of the categories of things within a domain [Sowa, 2000].

Together with explicit representations of the semantics of data for machine-accessibility, such domain theories are the basis for intelligent next generation

[2] A taxonomy is a hierarchical classification of information entities according to the relationships of the real entities which they are representing.

applications for the Web and other areas of interest [Devedžić, 2006] with a special focus on knowledge sharing and reuse. Furthermore they define, e.g.:

- A vocabulary for the unambiguous meaning [Chandrasekaran et al., 1999]
- A taxonomy for classification of entities [Devedžić, 2006]
- Content theory, due to the definition of classes of objects, relations and concept hierarchies [Chandrasekaran et al., 1999]
- Enabling consistency checking [Devedžić, 2006]

Top-level application areas identified by [Fikes, 1998] are collaboration, interoperability, education and modelling. Application domains are not limited, too. Ontologies are useful, wherever semantic information can enhance certain tools, products or processes (e.g. e-Learning ([Devedžić, 2006], [Mencke and Dumke, 2007d]), Virtual Engineering [Vornholt and Mencke, 2008], . . .).

Definition 8

Ontologies *are defined as a specification of a conceptualization [Gruber, 1993].*

Or in other words: they are the formal representation of an abstract view of the world. They include a vocabulary, taxonomy, instances, attributes, relations and axioms about a certain domain.

A vocabulary defines terms with unambiguous meanings. Furthermore, logical statements for the description of terms and rules for their combination and relation are provided. A taxonomy is part of the ontology concept for a hierarchical classification in a machine-processable form. Individuals/instances represent the objects of the ontology and thereby the available knowledge, while classes/concepts describe abstract sets of individuals. Attributes can be assigned to instances for description. They have a name and value. The last key element of ontologies is the relation. It can be described by using attributes and assigning another individual as a value. Common relation types are the is-a relation (subsumption relation) and the part-of relation (meronymy relation). The possibility to define domain-specific relations is a considerable additional value of the concept of an ontology. Axioms are always true and represent knowledge that is not inferable from other individuals.

It is possible to distinguish ontologies in two broad categories: lightweight and heavyweight ontologies. A lightweight ontology is described by individuals, classes, attributes, relations and axioms, meanwhile heavyweight ontologies are an extension of lightweight ones by the additional usage of axioms for a more detailed domain description.

Ontology creation is an iterative, creative process. There exists no *best* way to model a domain. Nevertheless, there exist certain methodologies to build, merge, re-engineer, maintain and evolve ontologies:

- Methontology Framework [López et al., 1999]
- Ontology creation following Fridman-Noy and McGuiness [Noy and Mcguinness, 2001]

○ Bottom-up construction following Van der Vet and Mars [van der Vet and Mars, 1998]
○ Ontology creation based on software engineering principles [Devedžić, 1999]
○ ...

There already exist many ontologies. Some are available via libraries like the DAML ontology library [DARPA Agent Markup Language (DAML) Program, 2008] and the SchemaWeb library [SchemaWeb, 2008].

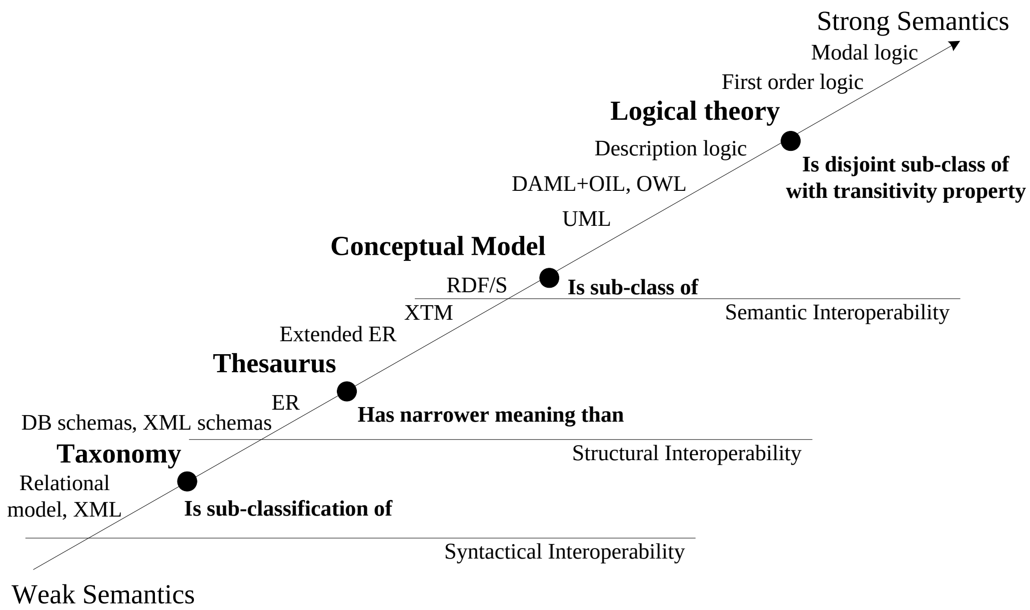


Figure 2.7: The Ontology-Spectrum: Weak to Strong Semantics (following [Daconta et al., 2003])

Technologies for semantic annotations differ in their semantic expressiveness. Figure 2.7 shows the ontology spectrum following [Daconta et al., 2003].

Complex tasks within the Semantic Web require standards for the representation of data and metadata. The most important standards are the Resource Description Framework and its extensions like the DARPA Agent Markup Language (DAML) + Ontology Inference Layer (OIL) and the Ontology Web Language.

Resource Description Framework The resource description framework is a language to describe data about entities in terms of object-attribute-value triplets. Every RDF resource is a list of such triplets, also called statements. Thereby, the object is a Web resource, the attribute one of its properties and the value a literal or other resource [World Wide Web Consortium (W3C), 2004b]. As being part of the layered Semantic Web approach (shown in Figure 2.6), RDF uses XML syntax. An example is given in Figure 2.8 (graphical representation) and Figure 2.9 (XML-based representation).

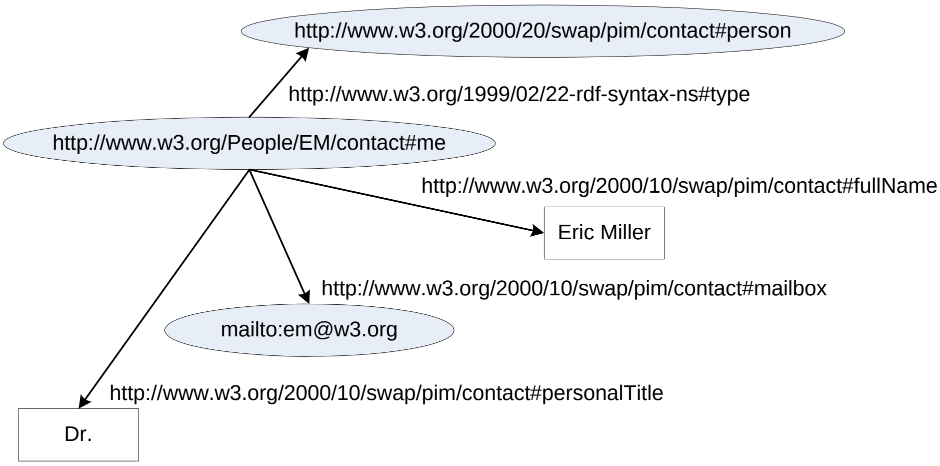


Figure 2.8: Graphical Representation of an RDF Example [World Wide Web Consortium (W3C), 2004b]

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
         <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
                  <contact:fullName>Eric Miller</contact:fullName>
                  <contact:mailbox rdf:resource="mailto:em@w3.org"/>
                  <contact:personalTitle>Dr.</contact:personalTitle>
         </contact:Person>
</rdf:RDF>
```

Figure 2.9: XML-Based Representation of an RDF Example [World Wide Web Consortium (W3C), 2004b]

Web Ontology Language The Web Ontology Language is an XML-based ontology language, too. It was developed by the W3C to replace the DARPA Agent Markup Language and Ontology Inference Layer (DAML+OIL), another – already deprecated – ontology language ([Cost et al., 2002], [Daconta et al., 2003], [Devedžić, 2006]). Additional information about DAML+OIL as well their root ontology languages DAML-ONT and OIL can be found e.g. in [Hendler and McGuinness, 2000] as well as [Fensel et al., 2001]. OWL can be seen as a set of RDF-triples of an OWL-specific vocabulary with an OWL-specific meaning [World Wide Web Consortium (W3C), 2004d].

OWL consists of three levels. The easier ones are completely part of the more complex ones:

- **OWL Full:** Is a complete OWL-language. For example, a class can be a set of individuals as well as an individual itself.
- **OWL DL (Description Logic):** OWL DL is a limited OWL. Here, e.g. classes cannot be individuals. Additionally, it is possible to use more expressive cardinalities.
- **OWL Lite:** It is the most simple OWL language. Here, only simple cardinalities are possible.

Already OWL Lite is capable to define an ontology of classes, attributes as well as instances of these classes and attributes. OWL DL extends this expressiveness by the possibility of more powerful cardinalities being not limited to only 0 or 1. Furthermore the creation of classes with boolean operators and lists is possible. OWL Full represents the complete capabilities of all OWL language layers. Classes for example, can be interpreted as sets as well as an individual [Daconta et al., 2003].

An example is given in Figure 2.10 (graphical representation) and Figure 2.11 (XML-based representation).

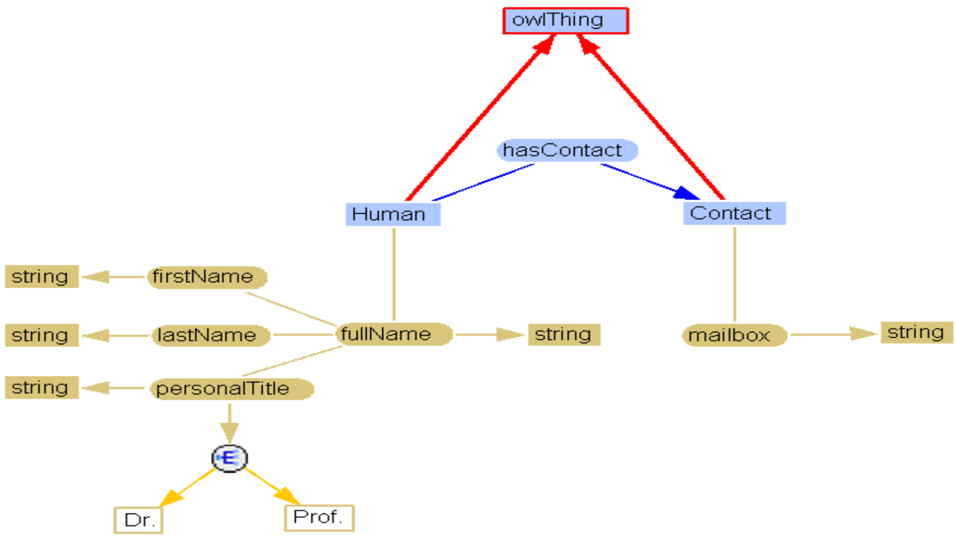


Figure 2.10: Graphical Representation of an OWL Example

```
...
  <owl:Class rdf:ID="Contact"/>
  <owl:Class rdf:ID="Human"/>

  <owl:ObjectProperty rdf:ID="hasContact">
        <rdfs:domain rdf:resource="#Human"/>
        <rdfs:range rdf:resource="#Contact"/>
  </owl:ObjectProperty>

  <owl:DatatypeProperty rdf:ID="fullName">
        <rdfs:domain rdf:resource="#Human"/>
        <rdfs:range rdf:resource="&xsd;string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="firstName">
        <rdfs:range rdf:resource="&xsd;string"/>
        <rdfs:subPropertyOf rdf:resource="#fullName"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="lastName">
        <rdfs:range rdf:resource="&xsd;string"/>
        <rdfs:subPropertyOf rdf:resource="#fullName"/>
  </owl:DatatypeProperty>
...
</rdf:RDF>
```

Figure 2.11: XML-Based Representation of an OWL Example

2.3.3 Use Case: Content in e-Learning Systems

Especially e-Learning is nowadays one of the most interesting of the "e-"domains available through the Internet [Anghel and Salomie, 2003]. In general, it refers to a wide range of applications and processes designed to deliver instruction through computational means [Juneidi and Vouros, 2005]. It is seen as a technology-based learning alternative respectively extension to the classic classroom model [Giotopoulos et al., 2005].

E-Learning is not intended to exclude existing methods and technologies. An appropriate use might complement them [Anghel and Salomie, 2003], e.g. as Blended Learning where e-Learning and traditional presence learning are merged [Schmietendorf and Mencke, 2008]. Current technical and technological progress and development lead to an increased usage of collaborative environments and distributed learning techniques. Nevertheless, it bases on classic learning theories and complements well established learning and teaching approaches.

Definition 9

Learning *[in general] is an enduring change in the mechanisms of behavior involving specific stimuli and/or responses that result from prior experience with similar stimuli and responses [Domjan, 1998].*

By extending this classic psychological point of view, the term e-Learning is defined below.

Definition 10

E-Learning *[is the electronic support of learning processes and] covers a wide set of applications and processes, such as Web-based learning, computer-based learning, virtual classrooms, and digital collaboration [American Society for Training & Development (ASTD), 2008].*

2.3.3.1 Why e-Learning?

It is not necessarily possible to simply apply certain technologies and pedagogical approaches to make people learn. A complete replacement is not the correct solution, too. Certain specific technologies and specific pedagogical principles are required to be adopted, developed and applied [Sadiig, 2005]. The aspects of e-Learning tend to be unique [Cerri, 2002]. Giotopoulos et al. state: "Many learning and technology professionals believe that e-Learning will take its place when we will stop referring to it using a separate name and regard it as an integral part of a complete learning environment." [Giotopoulos et al., 2005].

E-Learning is an already established concept. First roots can be traced back to the 60ties with the PLATO and TICCIT experiences in the USA [Cerri, 2002]. The first knowledge-based tutoring application appeared in domain of artificial intelligence in early 1970s. The first applications were simple automated instruction tools. Next fundamental steps were taken in the early 1990s. Authoring systems for intelligent tutoring systems were designed and developed. Furthermore, generic approaches were implemented, e.g. with the usage of task and domain ontologies [van Rosmalen et al., 2005].

The market for e-Learning products increased and became highly fragmented and less transparent. A wide array of products and concepts appeared, even more because of the Internet and its flexibility for learning and delivery detached from time and place.

The actual operating range of e-Learning includes many fields of application. The most common are [Garro and Palopoli, 2002]:

- Private self-improvement (e.g. for lifelong learning),
- Scholar learning,
- Vocational training,
- Support of university courses and other higher educational establishments and
- Business training as a component of Enterprise Knowledge Management.

General classes of vendors within the learning sector of e-commerce are technology providers, content providers and service providers. Certain full service vendors exist, too. Technology providers concentrate on learning platforms and portals, online conferencing systems, testing platforms, authoring tools and administration systems amongst others. Content providers generate and distribute differently specialized content as e.g. IT training, foreign languages or individual content. Services can be e-Learning consulting, application service provision, development, adaptation or training and support.

2.3.3.2 General e-Learning Architecture

A general view on e-Learning systems, involved roles and components is visualized in Figure 2.12. Technically, it can be for example implemented following the classic client/server principle, agent technology ([Mencke and Dumke, 2007a]), Grid-based approaches ([Pankratius and Vossen, 2003]) or with component-oriented techniques ([Rösner et al., 2007], [Piotrowski et al., 2007], [Amelung et al., 2008]).

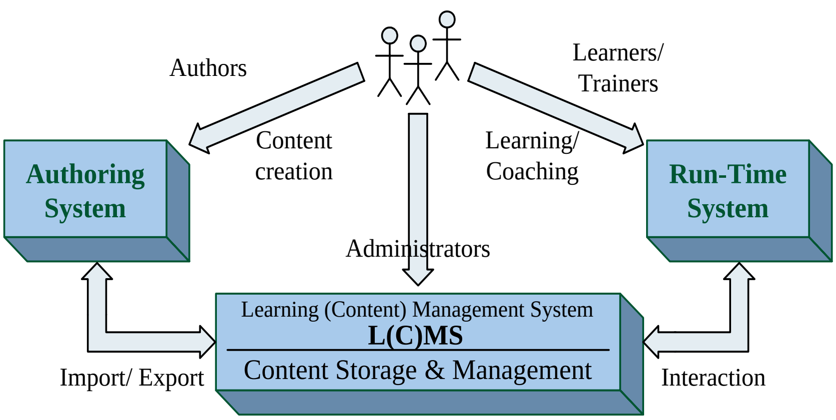


Figure 2.12: General e-Learning System and Involved Roles
[IMS Global Learning Consortium, Inc., 2003b]

In a typical learning environment learners, authors, trainers and administrators are the main groups ([Pankratius et al., 2004], [Giotopoulos et al., 2005]). Sometimes these roles overlap, so trainers and authors can be the same person, especially for small e-Learning systems. The content to be presented is created by the authors using authoring systems, stored in the learning management system (LMS) and thereby made available for the learner via a run-time system. The administrator's role is about the maintenance of the e-Learning system's core. He sets up, configures, updates and maintains the L(C)MS. Especially for larger applications additional roles can be identified e.g.: content expert, instructional designer, programmer, graphic artist and project manager [Giotopoulos et al., 2005]. The role of learners as content consumers is obviously clear.

E-Learning itself is a process containing two major phases: content development (additionally including planning, design and evaluation) and content delivery (additionally including maintenance). Its nature is iterative (see Figure 2.13). Evaluation is recommended for continuous improvement [Giotopoulos et al., 2005].

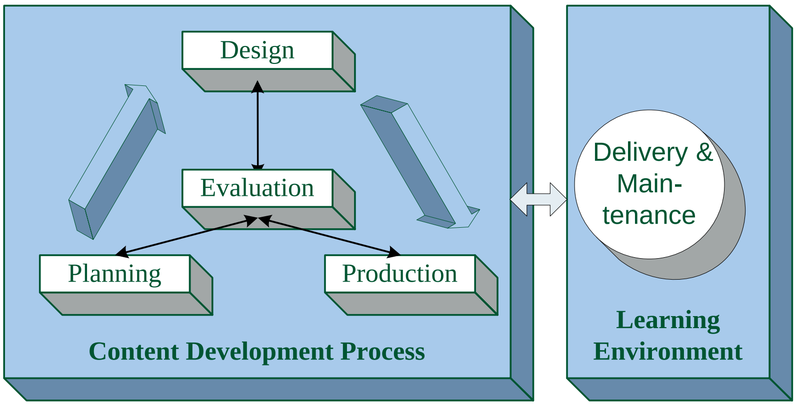


Figure 2.13: Iterative Process of e-Learning [Giotopoulos et al., 2005]

2.3.3.3 Primary Concepts and Definitions

There exist many terms within the context of e-Learning. This section provides an overview about primary concepts of this knowledge domain with a special focus on content and proactivity-related aspects.

Today mainly internet and intranet-based learning is entitled e-Learning. Following [SIVECO Romania SA., 2005] "the electronic resources permit the shift of the accent from *What are we teaching?* to *How are we teaching?*". Contrary to that, distance education (includes learning, teaching and training aspects) [UNESCO, 1987] describes a "variety of educational programs and activities... [where] the learner and teacher are physically separate but... efforts are made... to overcome this separation using a variety of media". It is important to know the difference. E-Learning definitions focus on instruction delivery technology meanwhile distance education is described in terms of physical separation (see Table 2.3). Based on this, e-Learning uses computational means to make distance education possible.

Instruction delivery technology	Physical separation yes	Physical separation no
Computational	Distance education & e-Learning	e-Learning
Other	Distance education	-

Table 2.3: Classification: Distance Education vs. e-Learning

Definition 11

Distance education *is an educational situation in which the instructor and students are separated by time, location, or both. Content is synchronously or asynchronously delivered to remote locations [American Society for Training & Development (ASTD), 2008].*

The independence of e-Learning in terms of spatial and temporal constraints is a primary aspect. Table 2.4 contrasts these properties with different e-Learning types.

	Synchronous (same time)	Asynchronous (different time)
Same place	Traditional learning (classroom) Face-to-face meetings with technology insertion (Computer Assisted Instruction (CAI) using computers, videos or Web-based material in PC-labs)	Asynchronous distance learning (Using learning centres or labs) Learning at own place in own time with organisation's facilities (Computer-Based Training (CBT) with CD-ROM, DVD, disks or tapes)
Different place	Real-time distance learning Live courses via high speed data links such as LANs, Satellites and the Internet (communication supported Web-Based Training (WBT), teleconferencing and Video Tele-Training (VTT))	Distributed learning Learning at own place in their own time, independent of geographic location (videotaped courses, WBT and CBT). Can incorporate aspects of the other quadrants.

Table 2.4: Time/Place Framework for Technology-Supported Distance Learning [Belanger and Jordan, 2000]

2.3.3.4 E-Learning Content Mediation

The term mediation within the context of e-Learning describes a type of content delivery. Its analyzable aspects are instructor and technology mediation [Belanger and Jordan, 2000].

Instructor mediation is focused on learning experience delivered by a human teacher/trainer, etc. He prepares the courses and presents the lectures. Furthermore, he chooses the content delivery technology. The main advantage is the possible interaction with the learner, e.g. to provide information, explanations, evaluation, feedback, human touch. The choice of a level of involvement has financial and teaching aspects to be considered and shown in Figure 2.14.

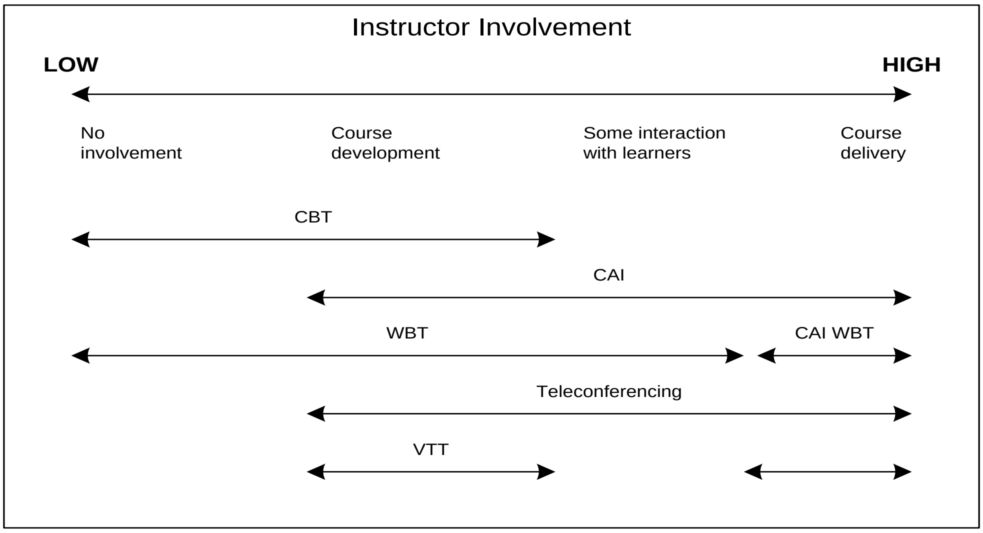


Figure 2.14: Instructor Involvement in e-Learning [Belanger and Jordan, 2000]

Technology mediation uses current advantages in information systems and Web technologies to facilitate e-Learning. Some sources claim that the performance does not significantly differ between traditional and e-Learning ([Russell, 2001], [Webster and Hackley, 1997]), but brings out several additional advantages. There are several factors to be considered like participation, satisfaction with learning environment, costs, reliability of technology and quality of delivery [Belanger and Jordan, 2000]. Technical aspects for several types of e-Learning are depicted in Figure 2.15.

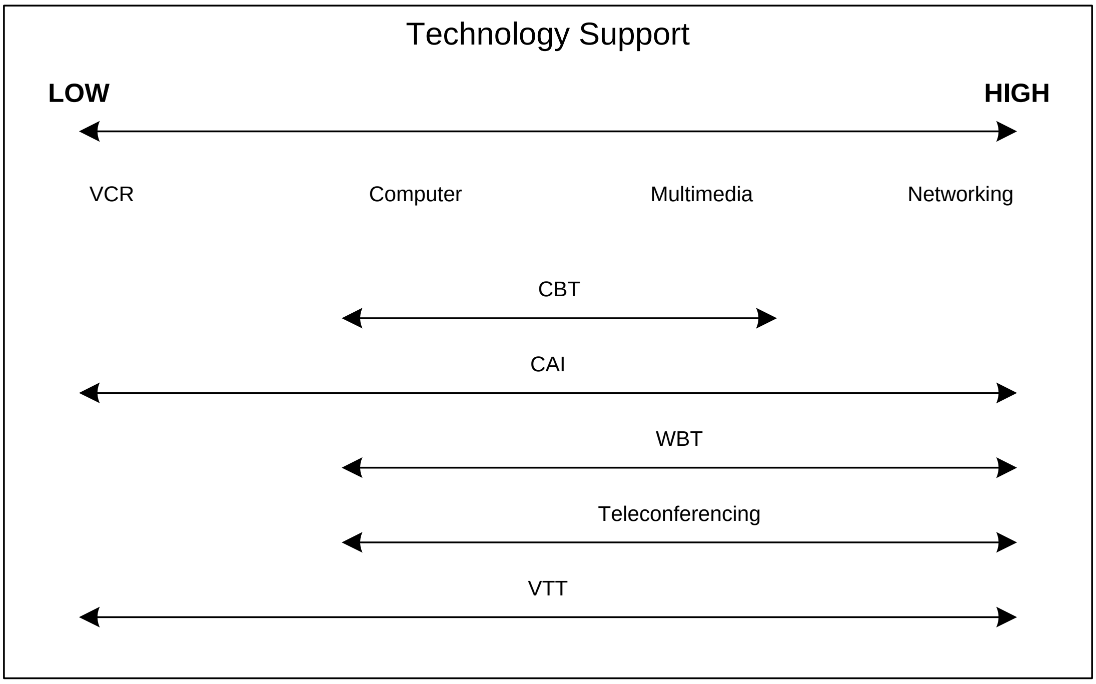


Figure 2.15: Technology Support in e-Learning [Belanger and Jordan, 2000]

2.3.3.5 Learning Objects

A *Learning Object* (LO) is an instructional component that represents a small piece of knowledge within the e-Learning domain [Garro et al., 2003]. Together with other associated LOs it forms the entire course. It should be usable in ways, contexts and for different purposes [Mohammed and Mohan, 2005].

Definition 12

Learning Objects *are defined as a reusable, media-independent collection of information used as a modular building block for e-Learning content [American Society for Training & Development (ASTD), 2008].*

A common aspect to all learning objects is the needed management for storage and (re-)combination. Therefore and for their description by metadata, several standards were already defined. The most important properties for a LO are ([Wiley, 2000], [Garro et al., 2003]):

Atomicity The atomicity of a LO describes the (self)consistent nature of a piece of knowledge.

Reusability A LO is reusable if it can be shared by multiple learning paths or in multiple courses.

Repurposability This term defines the ability to extract portions of a LO and to adapt them to new contexts.

Availability No temporal or spatial restrictions make a LO not available.

Granularity Granularity is the functional size of a resource.

Interoperability A LO is interoperable if it can be exchanged, reused and shared independent from its developer and its developer's organization.

For appropriate metadata of Learning Objects, a special standard (LOM - Learning Object Metadata) was developed by the IEEE Learning Technology Standards Committee (IEEE LTSC). Its goal is "to facilitate search, evaluation, acquisition, and use of Learning Objects, for instance by learners or instructors or automated software processes" [IEEE Learning Technology Standards Committee, 2003]. Nine categories of metadata are defined:

- General: for the definition of common information like title, used languages, keywords, descriptions, level of aggregation.
- LifeCycle: for the description of the history, state, version and list of contributors.
- Meta-Metadata: contains metadata about the metadata, because changes can be made not only by the author.
- Technical: describes technical aspects like format, size or requirements.
- Educational: for information about the recommended age of the learner, semantical density, degree of interactivity, etc.
- Rights: includes information about licenses, costs, copyright, terms of use, etc.
- Relation: to defines relations between objects.
- Annotation: for special remarks about the objects.
- Classification: for the classification of the objects with a taxonomic path or keywords.

The LOs described above typify a relative static nature. That represents a bottleneck for the general usage of LOs due to the needed manual re-purpose by specialists [Mohammed and Mohan, 2005]. *Reusable Learning Objects* (RLO) are intended to be automatically re-organized for multi-purpose usage in different contexts like certain courses [Belanger and Jordan, 2000].

Definition 13

Reusable Learning Objects *(RLO) are LO that can be transferred to various infrastructures or delivery mechanisms, usually without changes [American Society for Training & Development (ASTD), 2008].*

They represent modular reusable units of study, exercise, or practice and can be "consumed" in a single session [Pankratius et al., 2004]. They are intended to be authored independently from the target platform by an authoring system and to be stored within an LMS. The main intention for breaking up entire courses into RLOs respectively the summative creation of courses with certain RLOs is the interoperability, the possibility to independently reuse them on different target systems. Furthermore, they are more focused and can be developed by experts. Consistency is another advantage. The needed descriptive metadata ease the lookup and automatic course generation [Belanger and Jordan, 2000].

Modularization can be conducted using different ways depending on certain requirements.

- Type of knowledge chunk (e.g. definition, description, assessment)
- Type of representation (e.g. XML-files, picture, text)
- Different levels (e.g. in virtual worlds: geometry, scene, procedures, guided explanations/free usage [Mencke et al., 2008g])
- Aspired knowledge processing time (e.g. 5 - 15 minutes [Belanger and Jordan, 2000])

Following [Mohammed and Mohan, 2005] the next development step can be so called *smart learning objects* (SLO).

Definition 14

Smart Learning Objects *are defined as: "a structured aggregation of learning resources and the associated metadata encapsulated by a set of methods that provide intelligence, self sufficiency and platform independence while facilitating pedagogy" [Mohammed and Mohan, 2005].*

Based on the 'bucket model' suggested by [Nelson, 2000], SLOs represent an object-oriented container consisting of multiple packages and access methods and should be conform to the Shareable Content Object Reference Model (SCORM) [Advanced Distributed Learning (ADL), 2006b]. The packages themselves are aggregated by software files, image files, data sets and several other elements. They are described by metadata. The bucket model is implemented in Perl, but not limited to this technology.

The Grid computing paradigm is a new concept, essentially about the aggregation and unification of the view on existing software and distributed, heterogeneous hardware resources or remotely controlled instruments by using uniform interfaces. A Learning Object of the educational Grid (GLOB) extends the functionality of the classic notion of a Learning Object by Grid-specific functionality. It is wrapped by a Web Service for the easy integration in a Grid-based LMS. Those special functionality can be an interface for the access to a Grid application or access interfaces to other Grid services [e Cunha and Rana, 2006].

The Learning Objects described above respectively their combination to more complex course units result in different types that may occur isolated or combined [Klobas and Renzi, 2000]. Some examples are listed below and chosen ones are linked with several technical systems in Table 2.5.

- Lectures, presentations and
- Laboratories
- Seminars and tutorials,
- Consultations
- Interactive experiments
- Training scenarios in Virtual Reality [Mencke et al., 2008a]
- Educational games
- Assessments [Amelung et al., 2008]
- Documents, slides, simulations, role plays, questionnaires, pre-recorded lessons [Garro et al., 2003]

EDUCATIONAL STRATEGY	CHARACTERISTICS OF STRATEGY	CATEGORY OF WEB-BASED SOFTWARE FOR TEACHING AND LEARNING
Lecture or presentation	Teacher presents material to a class.	Readings or presentations prepared or converted to HTML format or web pages as index of downloadable material (text, tables, presentations) or audio, video material live or recorded and distributed via streaming technology.
Workshop or laboratory	Students complete set tasks designed to develop their skills; often live or recorded demonstrations presented or prepared by an instructor are included.	Activities prepared using WWW or other technology (including multimedia technologies), made available to students from a web page.
Self-guided instruction	Students work individually (often in geographical isolation), to complete assigned readings and exercises.	Readings, references and activities, prepared using WWW technology or distributed from a web page.
Seminar or tutorial	Students, working in relatively small groups, discuss set topics, cases or reading with the instructors guidance.	Discussion or conferencing software.
Consultation	Students (individually or in small groups) meet with the instructor to obtain answers or guidance on topics.	E-mail, chat, audio and video conferencing.
Collaborative learning	Students work together; the students learn through collaboration with one another rather than from material delivered by the teacher.	Discussion or conferencing software, e-mail, chat, audio/video conferencing, specific tools for community building and collaborative work.

Table 2.5: Web-Based Software for Teaching and Learning Strategies [Klobas and Renzi, 2000]

2.3.3.6 Content Provision in e-Learning Systems

The techniques of adaptive content provision can be refined and extended for the domain of e-Learning [Leutner, 1992]. The e-Learning system adapts to the requirements of the learner – the following general examples are about such micro adaptation. Macro adaptation describes the adaptation of technical system elements [Schmidtner, 1996].

The provision of information can depend on the *knowledge of the user*. For the comprehension of advanced concepts certain basis lectures can be needed. An adaptive system defines an adequate order of the individual presentation based on the user model and the modeled dependencies of the resource fragments.

Furthermore, the *creation of analogies* can be helpful for the comprehension of information. They support the recognition of similarities and differences by comparing known and new concepts. A possible technology is for example stretch text.

An additional method for teaching is the *adoption of instruction size and learning time*. Depending on individual characteristics, a training can be performed until the targeted knowledge level is achieved. Therefore, the knowledge state must be frequently determined.

The *adaptation of the instructional sequence* can be performed, too. This procedure depends for example on user mistakes at the current task. It can be an indication for the selection of a next information resource.

The *time to achieve a learning goal can be reduced* by adapting the task's difficulty. By this, the answering of questions of different difficulty levels determines the order of future questions. If a question with a certain level of difficulty is correctly answered, then the next step will provide a more difficult question. Wrong answers lead to easier questions.

The transition from declarative to procedural knowledge can be accelerated by the *adaptation of the task's presentation time* as well as by adaptively limiting the available time to answer. Wrong answers reduce the time for the next questions. Correct answers increase the available time.

The adaptive help for *exploratory learning* is another method. It explicitly points to an additional information source if it is found to be helpful and not yet identified by the user.

The *adaptive definition of new concepts* is an approach to support networked learning. New information is integrated into the scheme of already acquired information. By this, new concepts are adaptively introduced with already existing information. Older knowledge is repeated and new knowledge is learned much faster.

By *taking into account individual interests*, higher learning performances can be achieved, too. If such a domain is identified on the learner's side, then his/her knowledge about it can be adaptively increased.

2.4 Semantics-Based Assembly

Assembly is about the creation of a whole out of subcomponents. Recent research intends to provide metadata and semantically-related information in order to improve assembly results. The following descriptions provide a brief introduction about existing approaches from various domains. Other approaches for example focus on semantic-based content management systems as for example proposed in [Bügel, 2004], [Grunenberg, 2008], [Aumüller and Rahm, 2007] or [Sheth et al., 2002b]. The integration of semantics in Component-Based Software Engineering (CBSE) deals with various approaches to use semantic information for describing component semantics as well as the phases in the life of components as for example proposed in [Crnkovic and Larsson, 2002] or [Casey and Pahl, 2003].

2.4.1 Semantics-Supported Orchestration of Web Services

The basic idea behind Web Services is the provision of communication to enable cooperation and integration crossing application and organization borders. Therefore, technologies are needed like XML for content [World Wide Web Consortium (W3C), 2003a], XML schema for content description [World Wide Web Consortium (W3C), 2004a], SOAP for content packaging and transmission [World Wide Web Consortium (W3C), 2003b], WSDL for interface description [World Wide Web Consortium (W3C), 2007] and UDDI for central service access [OASIS, 2005].

Semantic Web Services possess not only a syntactical interface description, but an semantically enriched one. Thereby, discovery, selection, invocation and composition can be enhanced [McIlraith et al., 2001]. This assembly involves semantically-supported selection, composition and interoperation to achieve a higher goal. The components of semantic Web Services are shown in Figure 2.16

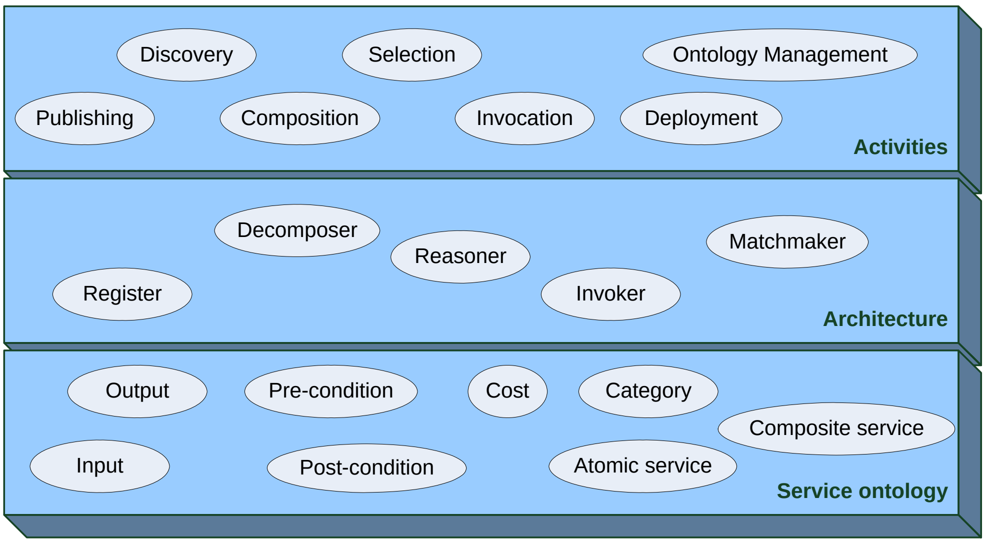


Figure 2.16: Components of Semantic Web Services [Cabral et al., 2004].

According to Cabral et al. ([Cabral et al., 2004]), the most important frameworks for this purpose are the Internet Reasoning Service (IRS-II) [Motta et al., 2003], OWL-S [World Wide Web Consortium (W3C), 2004c] and Web Service Modeling Framework (WSMF) [Fensel and Bussler, 2002].

IRS-II intends to support the semantic description and execution of Web Services [Motta et al., 2003]. They focus on the development of ontologies for domain models, task models, problem solving methods and bridges. Proposed key features are:

- Semantic Web Service description
- Mapping mechanism to link competence specifications to specific Web Services
- Capability-driven Web Service invocation
- Automatic wrapper generation

OWL-S is a set of ontologies to describe and reason about Web Services [World Wide Web Consortium (W3C), 2004c]. Assembly is achieved by OWL-S process models. They are used to reason about possible compositions/orchestrations (e.g. validation of possible compositions, determination of context-depending feasibility) as well as for the controlling of Web Service invocation.

The WSMF approach targets the domain of e-Commerce [Fensel and Bussler, 2002]. Therefore, it intends to decouple components of e-Commerce applications into services and to promote scalable communication by a mediation service. This project develops a service ontology with definitions of goals, mediators and Web Services in order to enable Semantic Web Service assembly.

Traditionally, service assembly with special focuses is tackled by using various planning algorithms (e.g.: [Zeng et al., 2003], [Qiu et al., 2006], [Lin et al., 2008]) by reducing the composition. These algorithms work on QoS-basis (Quality of Service) only.

2.4.2 Semantics-Supported Assembly on the Grid

The Grid [Foster and Kesselman, 1998] is a computing infrastructure basing on the idea of a *power grid* – the idea was, that it should be as easy to plug into this architecture as plugging into a power grid. Semantic Web technologies can provide intelligent support of seamless process automation, easy knowledge reuse and collaboration to extend it to a Semantic Grid. Thereby, the “Semantic Grid is characterized by an open system, with a high degree of automation, that supports flexible collaboration and computation on a global scale” [Tao et al., 2003].

The assembly of resources and services is supported by the definition of appropriate metadata and ontologies for certain goals.

2.4.3 E-Learning Content Assembly Approaches

Already established concepts for e-Learning content aggregation are existing standards and specification as well as educational modeling languages. They shift the focus from a content-oriented design to process orientation [Gruber, 1993]. Chosen examples are listed below.

2.4.3.1 IMS Learning Design

IMS Learning Design (IMS LD) bases on a set of IMS specifications to describe e-Learning content aggregation on three levels. The scope of content aggregation is more detailed in the next section, because the Sharable Content Object Reference Model (SCORM) uses specializations of IMS specifications [IMS Global Learning Consortium, 2003].

The core levels of IMS LD are:

- **Learning Design Level A:** basic vocabulary for the exchange and interoperability of e-Learning material
- **Learning Design Level B:** extension for personalization and sequencing
- **Learning Design Level C:** extension for notification.

To achieve compliance with these levels, several other IMS specifications were proposed, e.g. IMS Content Packaging, IMS Simple Sequencing, IMS/LOM Meta-Data, IMS Question and Test Interoperability, IMS Reusable Definition of Competency or Educational Objective, IMS Learner Information Package and IMS Enterprise [IMS Global Learning Consortium, 2003].

2.4.3.2 SCORM Content Aggregation Model

The Content Aggregation Model (CAM) – as part of the Sharable Content Object Reference Model (SCORM) specification – was developed for the collection and aggregation of atomic content sources to complex e-Learning courses [Advanced Distributed Learning (ADL), 2006a]. Appropriate metadata, based on the IMS Metadata Specification [IMS Global Learning Consortium, 2006], ensure interoperability as well as reusability of e-Learning content. An additional advantage is the easy extensibility for individualization purposes.

These metadata are as also valuable for the establishment of content repositories with advanced, semantically supported retrieval mechanisms. This is also supported by the differentiation into content (assets) and structure of those resources. The adapted presentation for different clients (e.g. for mobile devices, workstations, . . .) becomes possible.

The SCORM Content Aggregation Model defines a content model, metadata and content packaging.

A content model defines how reusable learning resources – in fact Learning Objects – are aggregated to higher-level instruction units. Therefore, the following parts exists:

- **Assets:** are electronic representations of e-Learning content, like text, media, images, sound, etc. The binding of assets to their metadata is described in the Content Packaging specification.
- **Sharable Content Objects (SCOs):** is a collection of assets including a special assets defining the communication between Learning Management Systems and the SCORM Run-Time Environment.
- **Content Aggregations:** are content structures defining a taxonomic representation of learning resources to sequence courses, chapters, etc.

A general example of the SCORM Content Aggregation is shown in Figure 2.17.

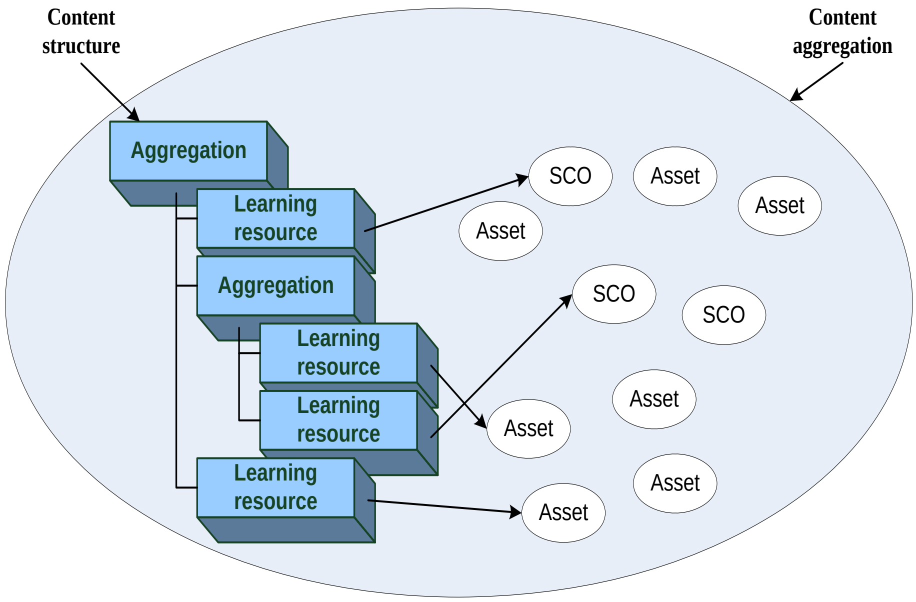


Figure 2.17: Example of SCORM Content Aggregation
[Advanced Distributed Learning (ADL), 2006a].

SCORM metadata are the basis for the aggregation of SCOs to larger structures. This specification directly bases on the IEEE LTSC LOM (see [IEEE Learning Technology Standards Committee, 2003] as introduced in Section 2.3.3.5) as well as on the IMS Learning Resource Meta-data XML Binding Specification [IMS Global Learning Consortium, 2006].

The intention of SCORM Content Packaging – directly basing on IMS Content Packaging [IMS Global Learning Consortium, Inc., 2007] – is to provide a standardized approach for the exchange of learning resources. Thereby, proprietary implementations of particular e-Learning systems can be displaced to ensure interoperability and reusability. This is achieved by the definition of:

- A manifest file describing the package with appropriate metadata, an optional organization section for the definition of content structure and behavior, and a list of references to resources of the package.
- A methodology for the creation of manifest files and
- Directions for packaging the manifest.

2.4.3.3 Educational Modeling Language

The intended usage of the Educational Modeling Language (EML) is to describe a learning design for automated processing [van Rosmalen et al., 2005]. It defines the learning process including activities (of students and staff) and resources/services. The Educational Modeling Language [Koper, 2001] is the basis for the IMS Learning Design Specification ([IMS Global Learning Consortium, Inc., 2003c], [Koper and Olivier, 2004]).

Its major implementation is an XML-based language and was developed to codify units of study, as e.g. courses, course substructures or study programs. Therefore, it provides structures for the content, roles, relations, interactions and activities of learners and students.

2.4.3.4 Learning Object Markup Language

The Learning Object Markup Language (LOML) was developed to define the structure of tutorials [Wu, 2002]. Its elements are

- **Title** of the learning object
- **Definition** of the core concept, the learning object is focused
- **Description** of the core concept
- **Example** about the core concept
- **Application** as simulation or demonstration to explain the core concept
- **Conclusion** about the core concept
- **Exercise:** to improve the transfer of knowledge and skills
- **Test** to evaluate the result of learning

2.4.3.5 Learning Material Markup Language

The Learning Material Markup Language (LMML) was developed for the structuring of the content of learning objects [Süss, 2000]. It is a meta-language using inheritance hierarchies to create discipline-specific markup languages, e.g. for computer science, music, finance.

2.4.3.6 PALO

PALO is a language to describe and design learning scenarios ([Rodríguez-Artacho et al., 1999], [Rodríguez-Artacho and Maíllo, 2004]). A corresponding reference framework provides five layers: management, sequencing, structure, activity and content, each identifying a group of related components of a learning resource. Different strategies can be created by defining special Document Type Definitions (DTD's).

2.4.3.7 Tutorial Markup Language

The Tutorial Markup Language is limited to specific learning scenarios, e.g. for questioning and problem-solving. It is an ISO SGML language for the creation of HTML-based learning materials in a platform neutral manner. Thereby, it separates delivery mechanism and content representation [Netquest, 1998].

2.4.3.8 Instructional Material Description Language

The Instructional Material Description Language is targeted towards instructional design and thereby limited to this special pedagogical design. It can be used to describe content, structure, assessments, user models and metadata in this context [Gaede, 2000].

2.4.3.9 Essen Learning Model

The Essen Learning Model is a development model to support the creation of computer-supported learning environments ([Pawlowski, 2000], [Pawlowski, 2001]). Therefore, it focuses on project management, quality assurance, process integration, curriculum development and learning sequence development. Another important aspect is the support for the specification of didactical models.

2.4.3.10 Ontological Support

The relevance of ontologies to describe learning scenarios is motivated in [Helic, 2005], too. Here, the author proposes an ontology-based configuration mechanism with the help of didactically-sound information. Additionally, a plan is described to integrate special ontology relations for the sequencing of teaching activities:

- Local to the learning scenario
  - Relations of didactical aspects to system features
- Global to the learning scenario
  - Temporal relations between activities (is-preceded-by, is-followed-by)
  - Structural relations to model compositions of activities (part-of)
  - Specializations of activities (kind-of)

Another difference is that the authors to provide a specialised extension of their WBT-Master system instead of targeting the provision of general expert knowledge.

In [Pawlowski, 2005] the author describes the Didactical Object Model (DIN DOM) developed within the German Standardization Body. It identifies the following major components to be important for the efficient exchange and reuse of didactical expertise:

- Context: to describe the (intended) environment for the scenario
- Actors: to model individuals, groups or agents within the scenario
- Activities: to describe the didactical concepts within an activity structure
- Resource: as the materials and services that are required for the scenario

Special needed sub-ontologies are already developed. As an example, there is a model of instructional objects in [Ullrich, 2004]. Each concept represents a particular instructional role of a learning resource. But these roles are only implicitly modeled. Figure 2.18 summarizes the instructional objects defined by the author.

Additional theoretical foundations are analyzed by ([Meder, 2001], [Meder, 2006] and [Borst, 2006]). There, additional usable taxonomies are described. They namely focus on knowledge types, presentation media, communication media, matter of fact relations, communication contribution cooperative objects as well as transactions/assignments. But for ontology-based assembly they are not yet suitable.

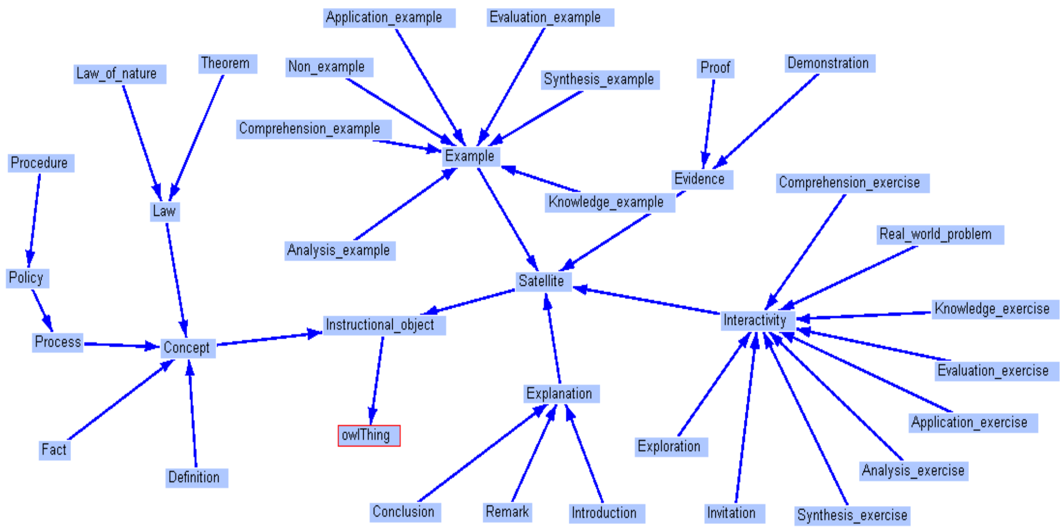


Figure 2.18: Class Hierarchy of Didactical Objects [Ullrich, 2004]

2.5 Proactivity

Following [Darbyshire and Lowry, 2000], proactivity is a peculiarity of autonomy – of having or having not the control about one's own actions ([Jennings and Wooldridge, 1998b], [Franklin and Graesser, 1996]). With decreasing complexity, the other ones are reactivity and passivity. Passivity thereby means not to become active due to certain events, while reactivity creates actions according to predefined schemata (time or events) ([Jennings and Wooldridge, 1998b], [Franklin and Graesser, 1996]).

This section introduces fundamental aspects about the proactivity-aspect of content. Therefore are presented: (a) the main adaptation techniques as one major part of proactivity in order to describe existing and additionally usable approaches in relation to this work, (b) user models as the technological basis of every adaptation and proactive action in order to provide fundamental knowledge for certain approaches described in this work as well as (c) the agent technology as the one technological implementation approach of proactivity because of its inherent proactive characteristic.

2.5.1 Adaptability

Content should be dynamic. Due to dependencies between content fragments and prerequirements of the user, static approaches are not feasible [de Bra, 2002]. Thereby, adaptability is a major topic for systems providing content. Especially within the research domain of adaptive hypermedia methods and techniques were developed to enable adaptability. Adaptive hypermedia systems are all hypertext or hypermedia systems that store information about the user in a model to adapt several visual system aspects to the user [Brusilovsky, 1996].

This section briefly introduces those approaches being applicable during content provision to enable individualized and thereby more efficient content assimilation processes at the user's side.

Definition 15

Content Adaptation *is defined by the World Wide Web Consortium (W3C) as the process of selection, generation or modification of resources in a given context. As an answer one or more content fragments are created (adapted from [World Wide Web Consortium (W3C), 2005]).*

In literature mainly two different types of adaptation are distinguished. Adaptive navigation (sometimes called link adaptation) is the modification of the availability or appearance of content navigation options – path modification. Thereby, for example important additional information can be highlighted, meanwhile information for which the user is not ready yet is hidden (e.g. described in [Wu et al., 1998], [Brusilovsky et al., 1998] or [Brusilovsky, 1999]).

Adaptive presentation (sometimes called content adaptation) is the presentation of content in different ways based on user model information (e.g. described in [Wu et al., 1998], [Brusilovsky, 1999] or [Devedžić, 2006]).

[Dolog et al., 2003b] and [Kernchen, 2005] propose a third class of content adaptation techniques. The adaptive content provision does not focus on the layout content, but on the content itself.

2.5.1.1 Adaptive Navigation

The navigation within the *hyperspace of resources* can become very complex and may lead to problems like being "lost in hyperspace" [Kernchen, 2005]. That is especially caused by the fact that classic system use identical navigation structures for the particular, possibly differing, user requirements. Techniques of adaptive navigation are designed to face this individualism of goals, knowledge and other user characteristics with adaptive navigation options. The intended goals are user guidance within the hyperspace of resources, the improvement of the user's orientation as well as the provision of an individual view on the resources.

[Brusilovsky et al., 1998] classifies different methods:

- **Global support** is for the information search with minimal effort based on link structures. Methodically it can be implemented using predefined and sorted links.
- **Local support** supports the user to identify the next navigation step. Based on the current location and user model information the user is directly forwarded to the next step or possible, prioritized navigation options are presented.
- **Local orientation** provides support depending on the user's current relative position. Possible methods are additional information about semantically related resources, the reduction of navigation options or the adapted presentation of additional resources.
- **Global support of orientation** describes the resources' structure as well as the absolute user position. Methodically, global and local navigation maps can be used to describe the structure of the navigation structure. Their adaptation can provide additional support. Position markers are used to uniquely identify resources and guided support introduces the user to the hyperspace of resources.
- **Management of personal views** allows that a small part of the hyperspace can be adapted and optimized. According to the current goal and user model information,

the view to available resources is individualized and only the most important are presented.

These methods can be implemented by single or combined usages of the following techniques [Brusilovsky and Nejdl, 2004]. Detailed information about them is provided in [Kernchen, 2005]:

- Direct forwarding to the next resource
- Sorting by relevance
- Predefined set of links
- Limitation of navigation space (deletion, hiding, fade out)
- Link typing
- Extension link presentation
- Additional information
- Adaptive local and global navigation maps
- Link generation (similarity-based, interest-based)

A possible implementation of the provision of adaptive navigation support is described in [Dolog and Nejdl, 2003]. Using a UML-based (Unified Modeling Language [Object Management Group, 2004]) representation extended by an XML-based metadata exchange, the creation of navigation sequences is possible. [Henze et al., 2004] focus on an approach for the generation of hypertext structures based on logic expressions. Xlinkit can automatically generate links, too. Based on standardized Internet technologies, functionalities for a rule-based link creation as well as consistency checks are possible [Nentwich et al., 2002].

2.5.1.2 Adaptive Presentation

Adaptive presentation differs from adaptive content provision in terms of other adaptation goals. Meanwhile adaptive presentation changes the layout of the content, adaptive content provision changes the provided content of the presented resources. The differentiated presentation of navigation structures targets the support of guided navigation as previously described.

For the adaptive layout of content several approaches can be distinguished:

- **Sorting of resource fragments:** Sorting of resource fragments influences the presentation flow based on certain user preferences. Short summaries can for example introduce an e-Learning course or end it.
- **Adaptation of presentation due to different media types:** The adaptation of presentation due to different used media types for the presentation of information can be necessary, if certain alternatively used media cause changes in the presentation's layout. Sometimes pure textual presentations need more space than a diagram.
- **Adaptive provision of content with different quality:** The need for resource's quality differences often occurs due to different client hardware structures. For example, a mobile device has limited presentation capabilities and requires low resolution pictures.

- **Adaptive provision of content due to different transmission contexts:** This and other technical requirements may cause the adaptation of content transmission. [Laakko and Hiltunen, 2005] describes a system using a proxy[3] for a rule-based adaptation of resources for mobile devices for an adequate presentation.
- **Adaptive resource provision due to different languages:** This approach's main domain of application is the Web due to the high user heterogeneity.
- **Color adaptations as well as font (size) changes:** Changes in color or variations of font type or color are based on user preferences as well.

A more basic approach of adaptive presentation is described in [Höök et al., 1996]. There, the already described techniques are combined to create controllable and predictable user interfaces. More common techniques and applications for the visualization of content, especially for the Semantic Web are described in [Geroimenko and Chen, 2002].

2.5.1.3 Adaptive Content Provision

Techniques of adaptive content provision focus on the content itself. From the implementation's point of view it can be placed on the side of the client, the server[4] or as a proxy. In the first case the client performs the adaptation meanwhile the server provides the content independently from particular client requirements. Server-side implementations result in the provision of content individually for each client. Proxies adapt server-provided content the special needs of the client and forwards it.

Again, methods of the adaptive provision of content are based on user model information [Dolog et al., 2003b]. Identified approaches are listed below:

- **Additional explanations** support the hiding and presentation of additional resources. Goal is the presentation of user-relevant information.
- **Fundamental explanations** must be understood before more advanced concepts can be accessed.
- **Comparing explanations** support an improved understanding of concepts by studying similar concepts.
- **Explanation variants** of different concepts can be created. Based on the user model, the most appropriate one is presented to the particular user.
- **Sorting of information fragments** places the most relevant information at the beginning of the presentation.

For the implementation of those methods certain techniques can be used:

- Provision of content variants differing semantic density [Smith, 1999],
- Frame-based provision,
- Provision of different media types,

---

[3] A proxy is a representative [Gamma et al., 2001].

[4] A client/server-system is a Web-based software system, where the roles between the service providing (server) and service consuming part (client) are clearly separated or structured, respectively (following [Dumke, 2003]).

- Stretch text [Smith, 1999],
- Text filtering (addition of information, information hiding) [Smith, 1999] and
- Adaptive Content Provision based on user classification (e.g. community models as the basis for Web directories) [Pierrakos et al., 2003].

An exemplified system is described in [O'Donnell, 1997]. Based on a metadata-annotated text, a detailedness-adapted document is created using pruning. All described methods again can be used in combination. For example, [Höök et al., 1996] presents a system using stretch text as well as frame-based techniques.

2.5.2 User Models

To be able to react in the correct way to certain events, an additional information source is needed. User models provide the necessary data about the user of the current system and are the basis for proactivity and individualized content provision. They reflect certain aspects of the individual knowledge state as well as goals and preferences of the current user. This and more information serve as a base for the heterogeneous user demands. Following Eklund and Zeilinger [Eklund and Zeiliger, 1996] the main tasks of a user model are:

- Identification of the current and relevant goals of the user.
- Saving and actualization of the user's knowledge about the system and its usage possibilities.
- Saving and actualization of the user's background knowledge.
- Analysis of the user's experience that can be useful for knowledge transfer.
- Saving and actualization of the user's preferences and interests.

Self identified four major goals: prediction and planning; diagnosis and remediation; negotiation and collaboration; interaction and communication [Self, 1994].

The ongoing user interaction with the system is used for the update of the user model [Dolog et al., 2003b]. The process of adaptation has the steps as shown in Figure 2.19.

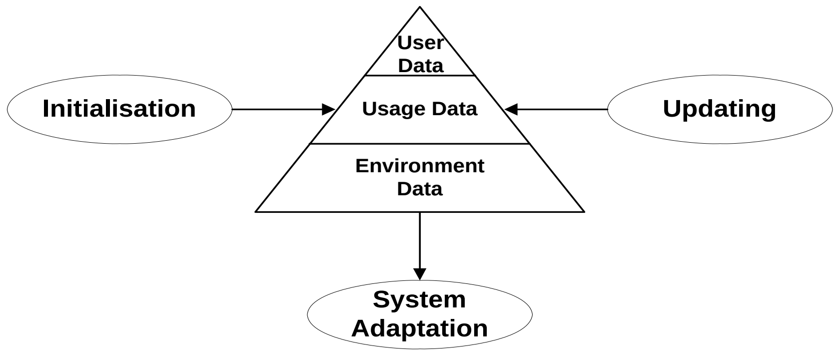


Figure 2.19: The Three Processes Involved With a User Model (adopted from [de Vrieze and van Bommel, 2004])

These steps can be either performed manually, automatically or semi-automatically. Although automatic approaches are the target of research and development, they can be error-prone [de Rosis et al., 1993]. Manual realizations suffer from high effort and are not effective. Semi-automatic approaches tend to be more effective. They allow a complete user model, because not all potentially usable information can be acquired automatically. The analysis of visited content, the number of visits and the average exposure time are for example automatically determinable but not sufficient. Sometimes interests or the personal background are needed but can only be acquired manually. Options to get this information are for example initial interviews, feedback questions, choices, tests or an manual user model adaptation by the user. Figure 2.20 sketches the creation and usage of user models.

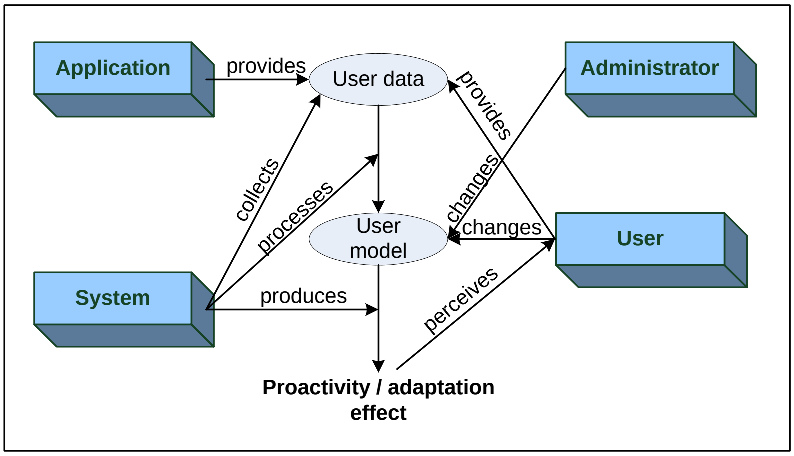


Figure 2.20: Creation and Usage of User Models (adopted from [Brusilovsky et al., 1998])

Certain data are needed to be able solve these tasks. According to [de Vrieze and van Bommel, 2004], they can be differentiated into user data, usage data and environment data. User specific data can be name, language, knowledge, interests, goals, preferred presentation medium, etc. Information about the user's interaction with the systems are usage data. They can be used as basis for e.g. decisions about future lectures. Environment data may include the position in time and space, socio-political aspects, the state of external resources and technological information.

[Cannataro and Pugliese, 2004] lists another categorization of possible information that can be stored in user models.

- External environment
  - Position in time and space
  - Language
  - Sociopolitical aspects
  - State of external resources
- Technology
  - Type of network and bandwidth
  - Quality of Service
  - Existing clients

User models themselves can be classified based on different characteristics. One is the basis for creation and usage depending on the targeted system as well as the development requirements as described by [Dolog et al., 2003a].

- **Resource-based models:** Those models evaluate e.g. the knowledge of the user based on the visited resources. Problems can occur if similar but differently classified resources of a domain are visited.
- **Ontology-based models:** They are similar to resource-based models, but their data modeling uses a conceptual model or an ontology as a basis. A possible problem can be the different depiction of the same information in different ontologies [Sosnovsky et al., 2007].
- **Models based on resources as well as ontologies:** Those combined models store resources as well information about their usage.

Another classification distinguishes the description of the user based on value-pairs or the membership to a group, respectively. Overlay models determine relevant user characteristics by appropriate metrics. Thereby, an overlay model is a vector of numerical values describing user attributes. An example can be the mastery of a certain topic. Thereby, the numerical values depict the user's knowledge about this concept [Devedžić, 2006]. All numerical values are continuously updated during user-system (and sometimes: user-user) interaction.

Stereotype models categorize users to predefined fixed vectors of numerical values. Thereby, user attributes are described, too. Stereotype models can be further distinguished. Next to assign a user to one group (pure stereotypes), he can also be assigned to multiple groups (multiple stereotypes) or the membership can be described using certain attributes (mixed stereotypes) [Cannataro and Pugliese, 2004].

In most cases, a mixture of overlay and stereotype models is useful. At first the identified group membership is used as the basis for proactivity and adaptation, but with an increasing amount of information about the user, a transition towards an overlay model is performed.

As in other domains, standards are helpful for an improved development process. For user models two main classic approaches can be identified in literature.

2.5.2.1 IEEE Personal and Private Information Project

The *IEEE Personal and Private Information Project* (PAPI) was developed with a special focus on the learner's performance during the learning process. That resulted in special categories. This user model standard concentrates on personal information and preferences (types of objects used by the learner), performance, security aspects, portfolio and a category to describe relationships to other people being involved in the learning process [IEEE Learning Technology Standards Committee, 2002c].

Furthermore, a differentiated presentation of information is possible. This feature was integrated to ensure that not every involved party (e.g. learner, teacher, teaching organization, ...) has access to each information, but only to the ones it needs access to.

Another standardization focus was the applicability in different systems. Next to external specification and internal data exchange specifications, data and control transfer mechanisms exist for the cooperation of different systems.

As an IEEE standard for learning technologies, IEEE PAPI is part of the same set of standards as the Learning Objects Metadata standard as described in [IEEE Learning Technology Standards Committee, 2002a].

2.5.2.2 IMS Learner Information Package

The user model specification of the IMS consortium – the IMS Learner Information Package (IMS LIP) – is more focused on management. Its main goals are the logging and management of learning process related events, goals and capabilities, learner support as well as the identification of learning advantages for the learner [IMS Global Learning Consortium, Inc., 2005].

Thereby, IMS LIP is intended to support interoperability between user models and Web-based learning systems. This standard is based on a so called *Learner Information Server* managing user information as well as the usage rights. Existing categories are:

- Access properties of the user,
- Learning acitivites,
- Relations between categories,
- Group memberships of the user,
- Competencies (knowledge, capabilities, . . .),
- Interests and goals,
- Identifiers (necessary biographic and demographic information),
- Certificates, qualifications and licences (about acquired knowledge),
- Security keys for system access by the user and
- Summaries about achievements.

For a short comparison and additional information for both models [Brusilovsky and Nejdl, 2004], [IEEE Learning Technology Standards Committee, 2002c] and [IMS Global Learning Consortium, Inc., 2005] are suggested for further reading.

2.5.3 Use Case: Proactive Agent Technology

Agent techniques and technologies with their very inherent proactive nature can be used in order to enhance the performance and the effectiveness of several aspects of providing and adapting content. Agents are not a new concept, but their use for certain aspects of providing content e.g. in the field of e-Learning it constitutes a basis for consequential advances.

To reasonable employ agent technology it is necessary to understand the underlying concepts. The agent idea goes back to works of Carl Hewitt in the field of artificial intelligence in 1977. He described an object “actor” being interactive, independent and executable. Furthermore it was intended to have an internal state and being able to communicate with other objects [Nwana and Ndumu, 1998]. Research on software agent technologies originated from distributed artificial intelligence and artificial life.

The first main discipline deals with the creation of organizational system for problem solving while the second one tries to understand and create models that describe life being able to survive, adapt and reproduce.

2.5.3.1 Why Agents?

The basic question when applying a technology is its usefulness [Dumke et al., 2009]. When is it possible and beneficial to integrate it? [Milgrom et al., 2001] answered this basic question for the agent-oriented paradigm by defining some guidelines validated by case studies ([Chainho et al., 2000], [Kearney et al., 2000], [Caire et al., 2001]). Their argumentation starts with a statement that agent-oriented design and implementation will have its greatest scope of applicability in systems with following general characteristics:

- Subsystems and subsystem components forming a system;
- High-level interactions between subsystems and subsystem components in terms of size and complexity;
- Changing interrelationships over time.

Common problems types that can be solved with agent technology where described in [Jennings and Wooldridge, 1998a] and [Ferber, 1999]. That may include system characteristics like dynamics, openness, complexity and ubiquitousness as well as problem qualities like physical distribution of components, data and knowledge. Agents can be helpful to solve these problems because of their scalability and their ability to improve latency [Anghel and Salomie, 2003].

The guidelines of [Milgrom et al., 2001] result in properties of solutions for complex software problems where the usage of software agent technology is expected to be useful.

Avoid overkill This principle refers to some philosophical background. It mainly concerns to the adjustment of requirements and solution. Not everything that is possible to design with agents should be implemented with them. Otherwise it is a waste of time and effort. “Always attempt to develop the simplest solution possible to any given problem.”

Need for distributed control Decentralized management of distributed systems can be appropriate due to platform, responsibility, privacy and physical constraints. For the first case this may emerge due to the intended integration of several applications running on incompatible platforms. Agents can be used to wrap existing functionality and enable their interrelation. Responsibility may cause effects that can be modeled explicitly by agent technology because complex software systems might work for different owners with different goals. Negotiation algorithms can offer a fair compromise at run-time. Privacy can be achieved by secure agents, privacy policies can be simply implemented. Physical constraints may require agent characteristics, too. A famous example is the complex robot control system for extraterrestrial deployment on missions to Mars.

Need for complex communications There exist many approaches to realize distributed systems (e.g.: n-tier architectures, Common Object Request Broker Architecture (CORBA), Enterprise Java Beans (EJB)). Their interaction style is mostly based on several assumptions. So the sender knows the intended receiver as well as his appropriate method/procedure to receive the message in addition to the message type to be sent. Agents are useful in situations with a more complex and flexible needed interaction. By limiting the set of message types and extending the included semantic it was possible to define communication patterns that are directly re-usable.

Need to concurrently achieve multiple, possibly conflicting goals Sometimes system behavior and the corresponding interaction schemes are too complex to be completely modeled at design time. Agent technology solves this problem by defining *how to decide* what to do instead of mapping inputs to outputs by defining *what to do*. By this approach a more flexible implementation becomes possible by adapting the behavior of the corresponding agents.

Need for autonomous behavior This need arises in the case of absence of explicit requests for action. Software is more flexible if it is able to perform certain actions in a goal-directed manner without continuous human supervision.

Need for high flexibility and adaptiveness Agent technology's advantage of intrinsic modularity and the possible cognitive capabilities lead to very effective and learning software systems. Agents can be added and removed at run-time and thereby lower costs because of the easy system expansion and modification.

Need for interoperability Sometimes systems are intended to interact with other software which specification is unknown during its own design. Using agents is a possible solution because they can provide services beyond their own capability due to their relations in a multi-agent system.

Non technical guidelines Technical aspects are not the only ones that need to be considered. Analysis and weighting of management issues is necessary, too [O'Malley and DeLoach, 2002]. That includes the cost of acquiring and adopting the methodology for use in an organization, the existence and cost of support tools, the availability of reusable components, the effects on existing organizational business practices, the compliance with formal or de facto standards as well as the support for tracing of changes during software life cycle.

2.5.3.2 Agent Definitions

There exists no single definition for agents, but a lot of discussion (e.g. [Wooldridge and Jennings, 1994], [Wooldridge, 1996], [Franklin and Graesser, 1996], [Castelfranchi, 1996]). Almost every author seems to propose own needs and ideas what leads to a variety of definitions depending on the targeted problem area. The expressed spectrum determines reasonable application areas as for example user interfaces, telecommunications, network management, electronic commerce and information gathering [Sánchez, 1997]. Russel and Norvig described this multiplicity aspect in this way [Russell and Norvig, 1995]: “The notion of an agent is meant to be a tool [. . .], not an absolute characterization that divides the world into agents and non-agents.” Nevertheless there are existing definitions [Dumke et al., 2009].

The Foundation for Intelligent Physical Agents (FIPA) provides a set of specifications representing a collection of standards which are intended to promote the interoperation of heterogeneous agents and the services that they can represent. Their definition is provided as an initial one.

Definition 16

An ***agent*** *is a computational process that implements the autonomous, communicating functionality of an application [Foundation for Intelligent Physical Agents (FIPA), 2006].*

Another classic definition of Wooldrigde and Jennings is based on technology features, too [Wooldridge and Jennings, 1995].

Definition 17

The ***Wooldridge-Jennings-Agent*** *is a software-based computer system with certain properties like autonomy, social ability, reactivity and pro-activeness.*

A next aspect of agent technology evolves from the following definition. [Maes, 1997] clearly states out that there is an environment needed for any autonomous action. [Franklin and Graesser, 1996] use a similar definition. Agents and their environmental context are shown in Figure 2.21.

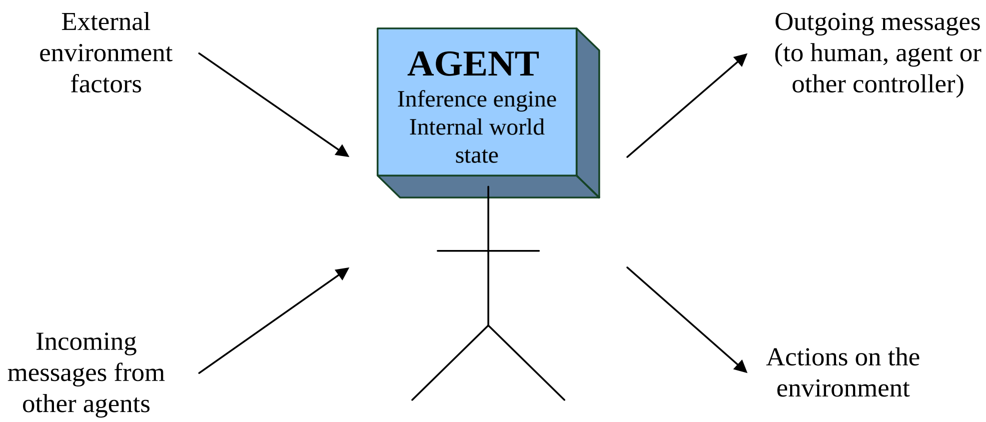


Figure 2.21: Agents and their Interaction with the Environment [Hayzelden and Bigham, 1999a]

Agents are situated in a certain environment which they are part of. Those agent platforms supply the needed infrastructure. A service directory, an agent directory, message transport and agent communication languages are those infrastructural elements as defined in the FIPA-Standard [Foundation for Intelligent Physical Agents (FIPA), 2006].

Definition 18

*An **agent platform** is the infrastructure being necessary for agent execution [Foundation for Intelligent Physical Agents (FIPA), 2006].*

For the implementation of certain agents, agent frameworks are provided. Next to infrastructural aspects they allocate Application Programming Interfaces (API's) and further specialized services for agent and multi-agent system (MAS) programming.

Definition 19

*An **agent framework** includes the necessary resources for the agent infrastructure as well as for the implementation of agents and multi-agent systems [Kernchen et al., 2006].*

2.5.3.3 Agent Properties

Almost all agent definitions have one aspect in common. They are based on certain properties. Every theoretician or developer proposes individual beliefs about potential benefits of his system or what is necessary to describe it. That explains the abundance of existing definitions [Kernchen, 2004].

Literature differentiates required and optional properties. The mostly referenced required ones are listed in the further course.

Autonomy The first and most important property is the autonomy. It is common to almost all agent definitions. Agents act autonomously when they perform their actions without direct interventions of humans or other agents. They should have control over their actions and their internal state. They significantly differ from "normal" objects in the sense of software engineering in having a behaviour. Agents have control over the execution of their methods ([Jennings and Wooldridge, 1998b], [Franklin and Graesser, 1996]).

Social ability This ability refers to the interaction potential of this technology. Agents need relations to other agents or humans to perform their actions or to help them performing their tasks ([Jennings and Wooldridge, 1998b], [Franklin and Graesser, 1996]). They are communicative for coordination and for exchange and validation of knowledge.

Reactivity Planning agents are widely known. But there is a need for instant reactions to changes in the environment, too. Therefore, they need perception capabilities ([Jennings and Wooldridge, 1998b], [Franklin and Graesser, 1996]).

Pro-activeness The property of pro-activeness is a counterpart of being reactive. Agents should reveal a goal-directed behavior and do something on their own initiative ([Jennings and Wooldridge, 1998b], [Franklin and Graesser, 1996]).

In reflection to special intended usage areas some more optional properties can be identified.

Adaptability Sometimes agents are characterized by their flexibility, adaptability and facility to set up their own goals based on their implicit purpose (interests). One of the major characteristics of agents is their ability to acquire and process information about the situation, both spatially and temporally. That results in non-scripted actions ([Hayzelden and Bigham, 1999a], [Franklin and Graesser, 1996]).

Agent granularity degrees Agents may have degrees of complexity. Most simple agents are characterized by the lack of intelligence regarding their behavior. These agents are called reactive. More complex agents are called cognitive or intelligent agents. They are characterized by their ability to know their environment, to act on themselves and on the environment; their observed behavior is a consequence of their perception, knowledge and interactions [Hayzelden and Bigham, 1999a].

Learning Either the agency itself may perform some learning ability (as society) or each individual agent may be embedded with a learning algorithm (e.g. a neural network or their re-enforcement algorithm). Learning often allows the agent to alter its future action sequences and behavior such that future mistakes can be alleviated. Learning is often a factor that provides an agent's ability to demonstrate adaptive behavior [Hayzelden and Bigham, 1999a].

Persistence An often as required defined property is persistence. It describes the retention of identity and internal state for a longer period of time as a continuous process ([Jennings and Wooldridge, 1998b], [Franklin and Graesser, 1996]).

Collaboration A major characteristic of agent technology is the system decomposition in smaller, more specialized components. One drawback or advantage (depends on the viewpoint towards this characteristic) is that not every agent has the complete functionality to solve a problem. The needed interaction to reach the goals is titled collaboration [Jennings and Wooldridge, 1998b].

Mobility Another major advantage of agents is their ability to migrate between environments over a network ([Jennings and Wooldridge, 1998b], [Franklin and Graesser, 1996]). It is an extension of the client/server paradigm of computing by allowing the transmission of executable programs between client and server. Mobile agent usage can reduce network traffic and allow asynchronous interaction, disconnected operation as well as remote searching and filtering. By this, bandwidth and storage requirements maybe positively impacted [DeTina and Poehlman, 2002]. Other fields of application are the access and administration of distributed information [Buraga, 2003] or the dynamic configuration of a entity network [Sadiig, 2005].

Character, personality This property refers to a believable personality and an emotional state ([Jennings and Wooldridge, 1998b], [Franklin and Graesser, 1996]). So it is describable within terms of an intentional stance in an anthropomorphic manner attributing to it beliefs and desires [DeTina and Poehlman, 2002].

Other detailed overviews about properties described in literature are given in [DeTina and Poehlman, 2002] and [Mencke and Dumke, 2007a].

2.5.3.4 Basic Agent Architectures

An often asked question refers to the difference between the concepts of agents and objects as well as between agents and actors.

Within the science of informatics an object is described by the concepts of a class-instance-relationship, inheritance and message transmission. The first concept esteems a class as a model of structure and behavior meanwhile an instance is seen as concrete representation of the class. By inheritance a class is derivable from another one and thereby able to use its properties. Message transmission allows the definition of polymorphic procedures whose code can be differently interpreted by different clients. By these common concepts of objects they cannot be interpreted as agents because they are not designed to fulfill certain goals or to satisfy a need. Furthermore message transmission is only a procedure invocation [Ferber, 1999]. Agents are able to decide about message acceptance and about an appropriate reaction.

Actors are parallel systems communicating by asynchronous buffered messages. They do not wait for an answer but order the receiver to send it to another actor. Actors are no agents due to the same reasons as explained above.

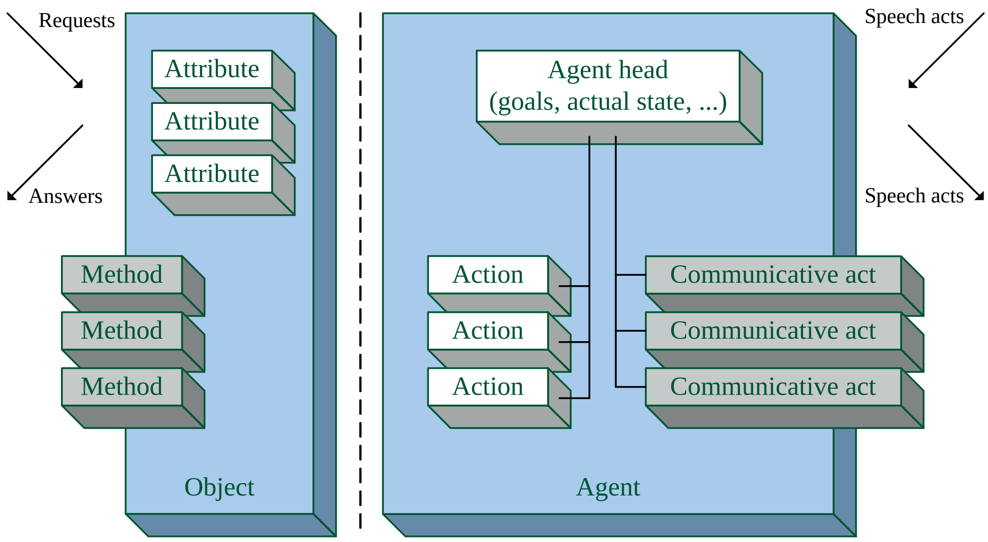


Figure 2.22: Comparison Agent and Object ([Ferber, 1999] and [Bauer and Müller, 2004])

Agent architectures represent the transition from agent theory towards their practical application [Kernchen and Vornholt, 2004]. Therefore, three main research and application directions exist.

Deliberative Agents Deliberative agents base on the classic Artificial Intelligence by explicitly requiring a symbolic model of the environment as well as the capability for logic reasoning. Fundamental aspects are described by Newell and Siman within their “Physical-Symbol System Hypothesis” [Newell and Simon, 1976]. This theory describes a system being able to recognize symbols which can be combined to higher structures. An additional intention is its capability to run processes for symbol processing. The symbols themselves can be used to create a symbolically encoded set of instructions. Their final statement is that such a system is capable to perform intelligent actions.

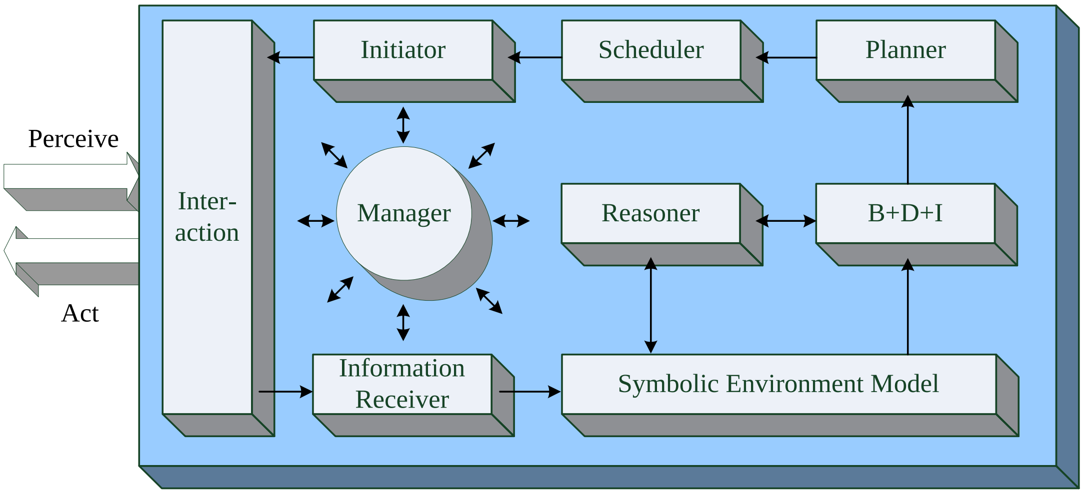


Figure 2.23: Deliberative Agent Architecture [Brenner et al., 1998]

Deliberative agents are the next step of this development. They contain an explicit symbolic model of the environment and decide following certain logical rules. The targeted types of problems to be solved are:

- **Transduction problems:** describing the translation of the real world into an adequate symbolic description.
- **Representation problems:** describing the symbolic representation of information about real world objects and processes and how agents reason with those data.

The vision, especially of representatives of the classic AI, was to create automatic planning, automatic reasoning and knowledge-based agents.

The most important deliberative architecture is the BDI architecture of Rao and Georgeff [Rao and Georgeff, 1991]. It is exemplary described below.

The basic elements of this architecture are the Beliefs, Desires and Intentions. They form the basis for the agent's capability for logical reasoning. Beliefs contain data about environmental information, action possibilities, capabilities and resources. An agent must be able to manage the heterogeneous, changeable knowledge about the domain of its interest. The agent's desires derive from its beliefs and contain "individual" judgements of future environmental situations from the agent's point of view. The desires can be mutional, non-realistic and even come into conflict with each other. The intentions are a subset of the agent's actual goals and point to the goal that is actually intended to be achieved.

Additional components completing the mental state of a BDI agent are its goals and plans [Brenner et al., 1998]. Goals are a subset of the agent's desires and describe its potential, realistic, not conflicting latitude. Plans subsume intentions and describe actions to solve a problem.

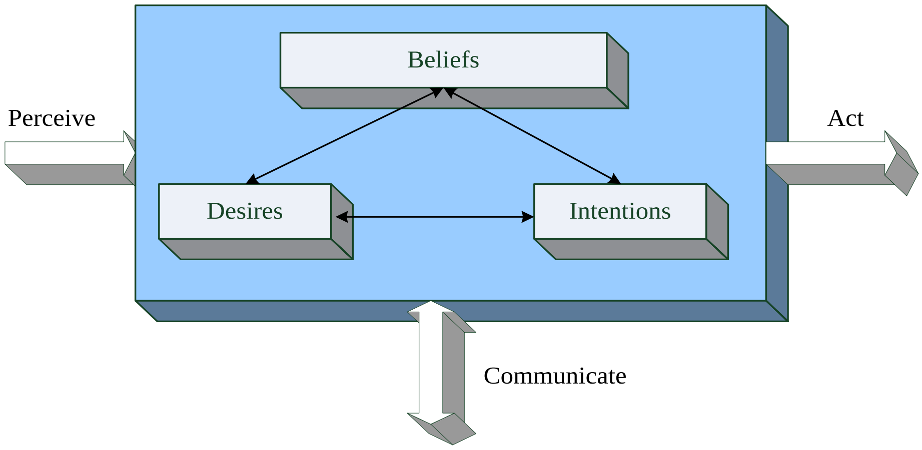


Figure 2.24: BDI Architecture [Rao and Georgeff, 1991]

The agent needs sensors to perceive data about its environment to create its world model (see Figure 2.24). These data need to be interpreted and may cause adaptations or extension of the agent's actual beliefs. Actuators are used to realize plans with certain actions. Thereby, the agent changes its environment in a goal-directed, methodical way.

Because of the high complexity of appropriate environmental representations, deliberative agents are rarely sufficiently applicable within dynamic environments.

Reactive Agents Reactive agents are an alternative approach to solve problems that are not or only insufficiently solvable with symbolic AI. Therefore, a reactive agent architecture does not include an explicit description of the environment and mechanisms for logical reasoning.

Reactive agents perceive their environment and immediately react to occurring changes. This interaction is the basis for their intelligence, in contrast to the internal representations of deliberative agents [Brenner et al., 1998]. The basic architecture of a reactive agent is shown in Figure 2.25. Even in complex situations, the agent only needs to identify basic axioms or dependencies. This information is processed by task-specific competence modules to create reactions. Again, actuators influence the environment based on the determined actions.

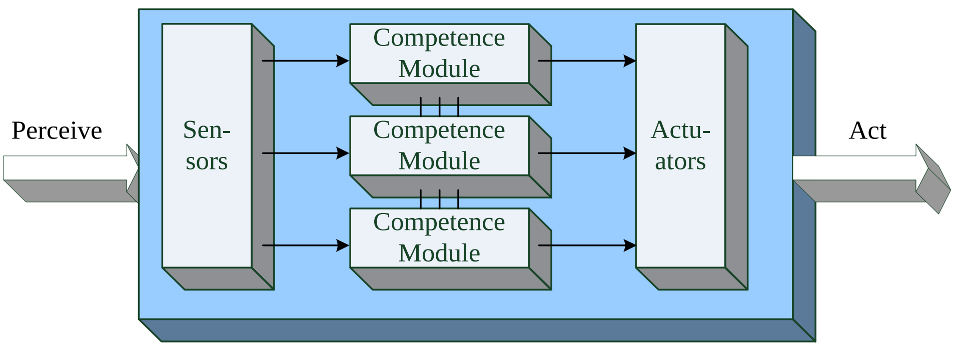


Figure 2.25: Reactive Agent Architecture [Rao and Georgeff, 1991]

A representative of reactive agent architectures is the Subsumption Architecture of [Brooks, 1991]. There every behavior is an almost independent process subsuming the behaviors of the lower behaviors (see Figure 2.26).

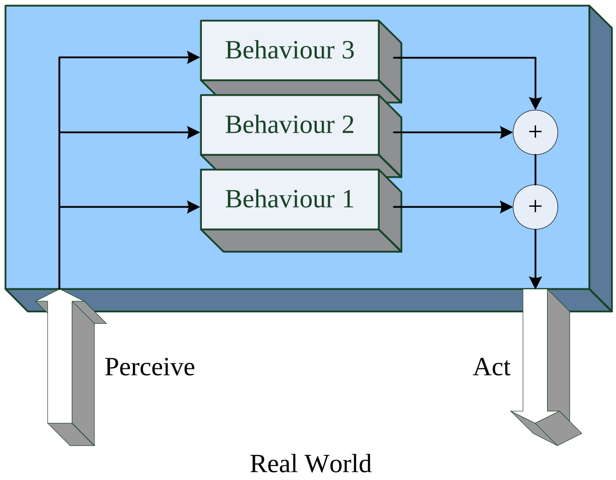


Figure 2.26: Subsumption Agent Architecture [Brooks, 1986]

Hybrid Approaches Hybrid architectures try to combine different architectural approaches to a complex system. The idea behind is to get all advantages but not the trade-offs of the particular approaches. Following Ferber, hybrid approaches can be classified according to the capacity of agents to accomplish their tasks individually as well as to plan their actions.

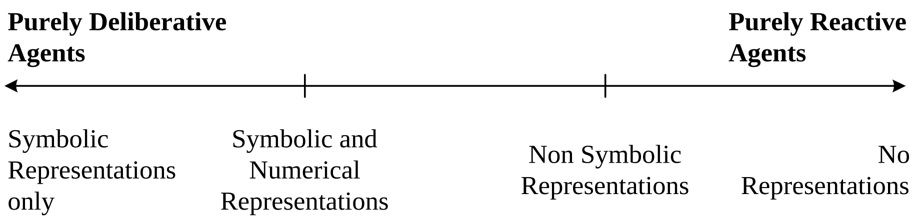


Figure 2.27: Hybrid Agent Architecture Classification [Ferber, 1999]

Literature like [Brooks, 1991] proposes horizontal as well as vertical levels, each with own functionality, in those complex systems. An example of a hybrid architecture is shown in Figure 2.28 and was developed by Müller in 1996.

One important advantage of agent technology is its possibility to find better solutions to problems due to the cooperation of many individuals. That leads to the concept of multi-agent systems.

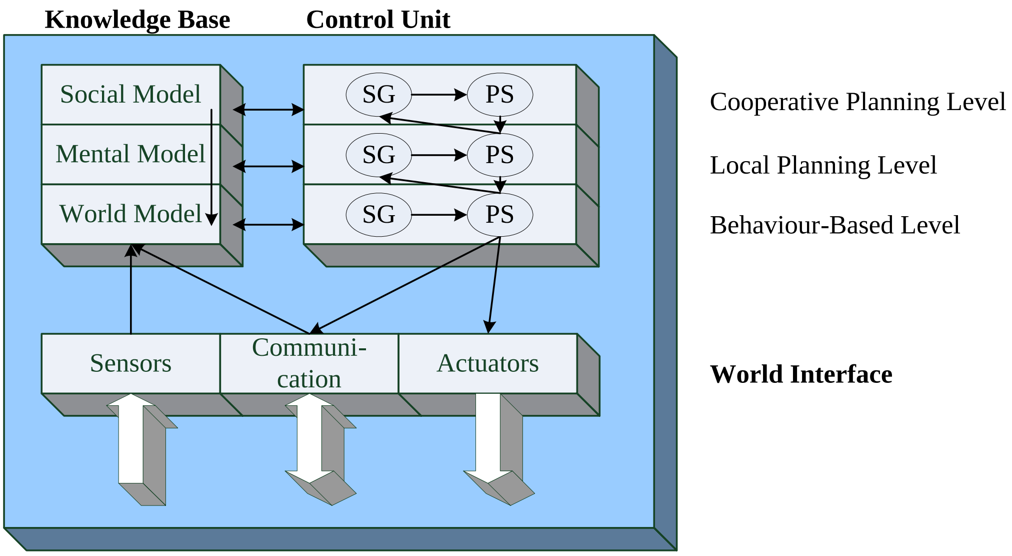


Figure 2.28: Hybrid Agent Architecture [Müller, 1996]

2.5.3.5 Multi-Agent Systems

The central approach of solving a given problem with a single agent may lead to certain restrictions ([Nwana, 1996], [Sycara et al., 1996]). Multi-agent systems are societies of a number of autonomous agents that work together to overcome them. It comprises their abilities and experiences an additional surplus value by the interaction among in individuals as this statement of Aristotle reflects: "The whole is more than the sum of its parts." Every agent of the MAS either can pursue its own goals and only communicate for information gathering or it can provide a coordinated, partial solution for the whole problem. But always the agent has a well defined task that it is responsible and especially appropriate for [Dumke et al., 2009].

Definition 20

*The term **multi-agent system** (or MAS) is applied to a system comprising the following elements [Ferber, 1999]:*

1. *An environment, E, that is a space which generally has a volume.*
2. *A set of objects, O. These objects are situated, it is possible at a given moment to associate any object with a position in E. These objects are passive, that is, they can be perceived, created, destroyed and modified by the agents.*
3. *An assembly of agents, A, which are specific objects ($A \subseteq O$), representing the active entities of the system.*
4. *An assembly of relations, R, which links objects (and thus agents) to each other.*
5. *An assembly of operations, Op, making it possible for the agents of A to perceive, produce, consume, transform and manipulate objects from O.*
6. *Operators with the task of representing the application of these operations and reaction of the world to this attempt at modifications, which we shall call the laws of the universe.*

Common areas of application are problem-solving, multi-agent simulation, the building of artificial worlds, collective robotics and program design [Ferber, 1999].

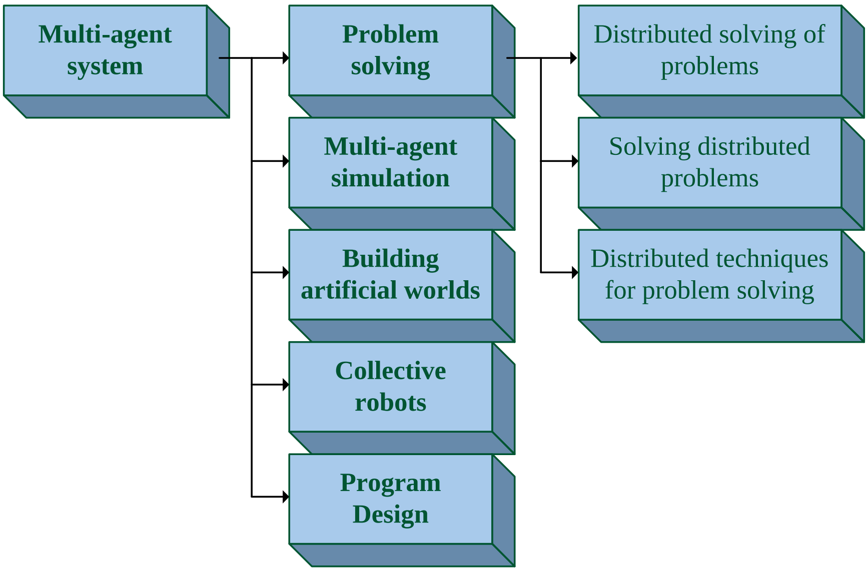


Figure 2.29: Classification of Application Types for Multi-Agent Systems [Ferber, 1999]

MAS have several advantages [Wille and Dumke, 2005]. [Hayzelden and Bigham, 1999b] listed the following ones:

- “To address problems that are too large for a centralized single agent, for example because of resource limitations or for robustness concerns (the ability to recover from fault conditions or unexpected events);
- To enable the reduction of processing costs – it is less expensive (in hardware terms) to use a large number of inexpensive processors than a single processor with equivalent processing power;
- To allow for the interconnecting and interoperation of multiple existing legacy systems, e.g. expert systems, decision support systems, legacy network protocols;
- To improve scalability – the organizational structure of the agents can dynamically change to reflect the dynamic environment – i.e. as the network grows in size the agent organization can re-structure by agents altering their roles, beliefs, and actions that they perform;
- To provide solutions to inherently distributed problems, e.g., telecommunications control, air traffic control, and workflow management;
- To provide solutions which draw from distributed information sources; and
- To provide solutions where the expertise is distributed.”

Following [Brenner et al., 1998] the most important restrictions of single agents, and thereby reasons for the creation of MAS are:

- Enormous amount of knowledge necessary for complex problems
- The problem can be so complex that there exists no actual technology that enables one single agent to develop a solution
- Many problems are distributed and require distributed solutions
- Domain knowledge and other resources are often distributed among different places
- Single agents can be bottlenecks in terms of processing speed, reliability, flexibility and modularity

This technology can be usefully applied to certain domains for information processing like for example e-Business and e-Learning. The later one is a very important one and was already introduced in this work.

2.6 Discussion

This chapter comprises information about the terminology and theoretical foundations in order to understand the described concepts of the subsequent chapters. Therefore, three key areas were identified and briefly sketched after introductory notes about the proactive content provision process as well as its arrangement within a superior process.

Due to this work's focus on content, related concepts were introduced at first. For this purpose, content and selected options for its representation were classified. Content-related concepts that are necessary for the remainder of this work, like the Semantic Web with especially ontologies, were described, too. Additionally, the throughout targeted use case of e-Learning was introduced.

Content assembly is a major topic in this dissertation. Because of this, chosen related work about the semantics-supported assembly of entity fragments was also part of this chapter.

Necessary basics about adaptability and proactivity were presented as well. User models as needed information sources as well as agent technology as an inherently proactive technology were described.

As introduced in the previous chapter, this dissertation focusses on semantic technologies for the proactive content provision. Summarized, this chapter presented the needed terminology as well as theoretical foundations.

3 Current Situation in Agent-Supported Proactive e-Learning

> "*Net generation* (those who learned to read after the Web) is qualitatively different in their informational behaviors and expectations; they are multi-task and expect their informational resources to be electronic and dynamic" [Marchionini, 2006].

Proactive content as defined and described in the previous chapter can be realized in various domains and ways. A very modern and dynamic implementation and usage of content is the application of agents and ontologies for the domain of e-Learning. Agent techniques and technologies can be used in order to enhance the performance and the effectiveness of several aspects of e-Learning systems. Agents are not a new concept but their use in the field of e-Learning constitutes a basis for consequential advances.

For the presentation of existing proactive content realizations, agent-supported e-Learning approaches are chosen and described below. Following the guidelines presented in the last chapter, agent technology can be applicable in the domain of e-Learning. This chapter describes several existing approaches.

Agent-supported e-Learning is defined as follows:

Definition 21

Agent-supported e-Learning *is the application of agent techniques and technologies in order to enhance the performance and the effectiveness of several aspects of e-Learning systems.*

For an appropriate classification, a special framework is developed in the following and used to categorize existing approaches [Mencke and Dumke, 2007c].

3.1 A Framework for Agent-Supported e-Learning

Literature provides several approaches for the application of agent technology for the domain of e-Learning. A "pedagogically neutral, content neutral, culturally neutral, platform neutral" [IEEE Learning Technology Standards Committee, 2002b] framework for the integration of possible architectural components is described below. It is intended to be used as an abstract representation of the functionality of e-Learning system artifacts that can be provided by a set of agents ([Mencke and Dumke, 2007b], [Mencke and Dumke, 2007c]). The main proposed key features are:

- Adaptable architectural components with extensive (additional) agent support.
- Identification of approaches for agent-based support for e-Learning systems.
- Separation and provision of basic and specialized services for reuse and optimized system development. Implementation aspects of basic aspects are hidden from the user.
- Improved focusing on key elements as e.g. pedagogical issues becomes possible.
- Exchange of application functionality between organizations and interoperability are eased.
- Extensive evaluation capabilities of users and system artifacts.

The developed framework is based on the abstract framework [IMS Global Learning Consortium, Inc., 2003a] of the IMS Global Learning Consortium, Inc. and the SUN Microsystems e-Learning Framework [SUN Microsystems, Inc., 2003]. It is further refined by several aspects of related architectures and models as for example the Open Knowledge Initiative [Open Knowledge Initiative, 2003], the ADL Sharable Content Object Reference Model (SCORM) [Advanced Distributed Learning (ADL), 2006b], the IEEE Learning Technology Systems Architecture (LTSA) [IEEE Learning Technology Standards Committee, 2002b] and the Learning Technology System Architecture of the Carnegie Mellon University [IMS Global Learning Consortium, Inc., 2003a]. Special requirements and advantages evolve from the intended application and integration of agent-based technology. Thereby, it is especially focused on adaptation, autonomy, support and flexibility.

The novel framework, visualized in Figure 3.1, takes into account the diversity of users involved in learning processes in contrast to the functional models of the abstract IMS framework [IMS Global Learning Consortium, Inc., 2003a].

Next to the main groups of learners, authors, trainers and administrators, support for content experts, instructional designers, graphic artists and project managers is needed [Giotopoulos et al., 2005]. Their requirements for an e-Learning system are grouped and depicted by several functional environments. Thereby, the *Presentation Environment (PE)* is the basic platform for the integration and display of the other environments. It is a basic element connected to all other environments, like the *Administration (AE)* and *Interaction Environments (IE)*, too. Appropriate and specialized access to functionalities for the learner is provided by the *Delivery (DE)* and *Working Environments (WE)*. Authors, trainers, content experts and instructional designers benefit from support of the *Learning Unit Environment (LUE)* and the *Content Environment (CE)*.

To guarantee flexibility, extension and interoperability, the framework is based on three *support layers*. They are differently specialized and are providing infrastructural support, common services and e-Learning services. Hereby, a service is defined as a functionality providing entity, which can be potentially used in different environments. Fundamental, needed and desirable services are horizontally integrated as provided by the support layers. The specific services can be ordered and used on demand. They also provide the basis for the connection and data exchange between certain implementations of the proposed framework. This abstraction of common facilities from the classic “LMS only” model was already proposed e.g. [IMS Global Learning Consortium, Inc., 2003a] and [SUN Microsystems, Inc., 2003].

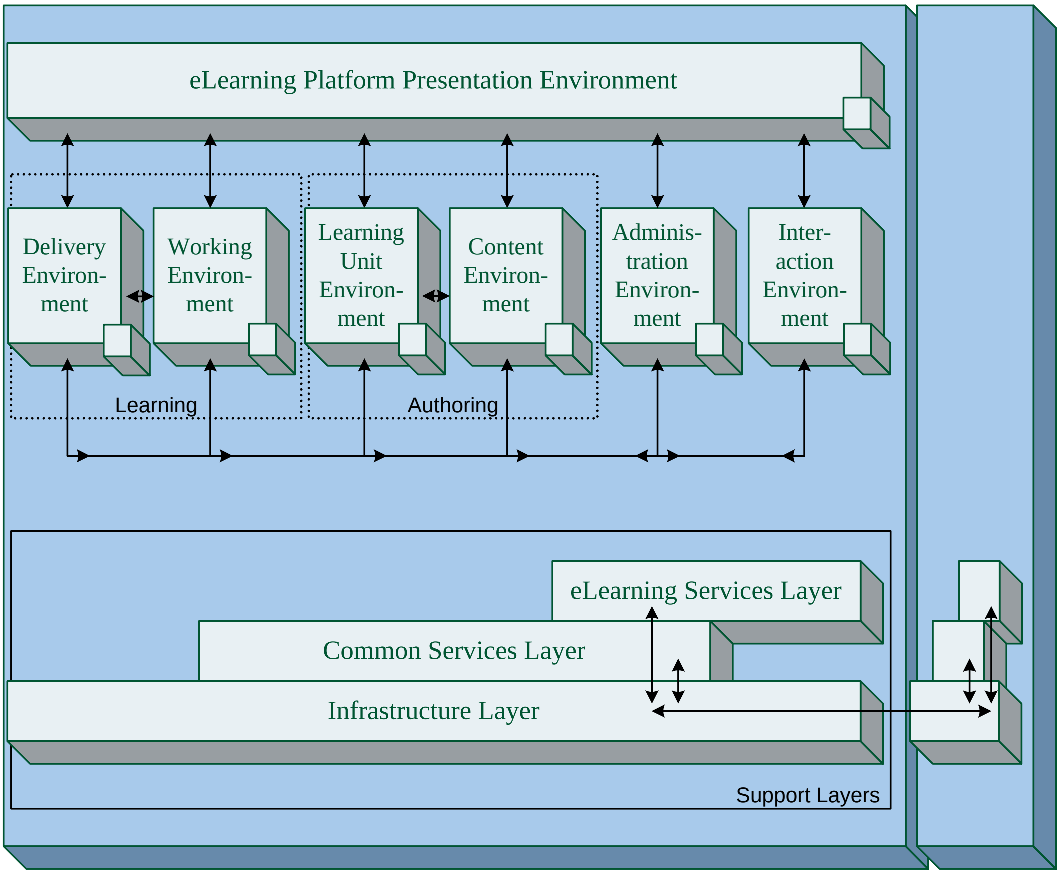


Figure 3.1: A Framework for Agent-Supported e-Learning [Mencke and Dumke, 2007c]

In the following sections, existing approaches are categorized according to the described parts of the framework for agent-supported e-Learning [Mencke and Dumke, 2007c].

3.2 E-Learning Platform Presentation

The e-Learning Platform Presentation Environment (PE) is the core of the Graphical User Interface (GUI) of every e-Learning system implemented following this framework. It provides personalized access for the different possible user groups. Exemplary use cases are visualized in Figure 3.2. It mainly provides access to the learning, authoring and administration environments (as described in [IMS Global Learning Consortium, Inc., 2003a]), as well as to the interaction environment.

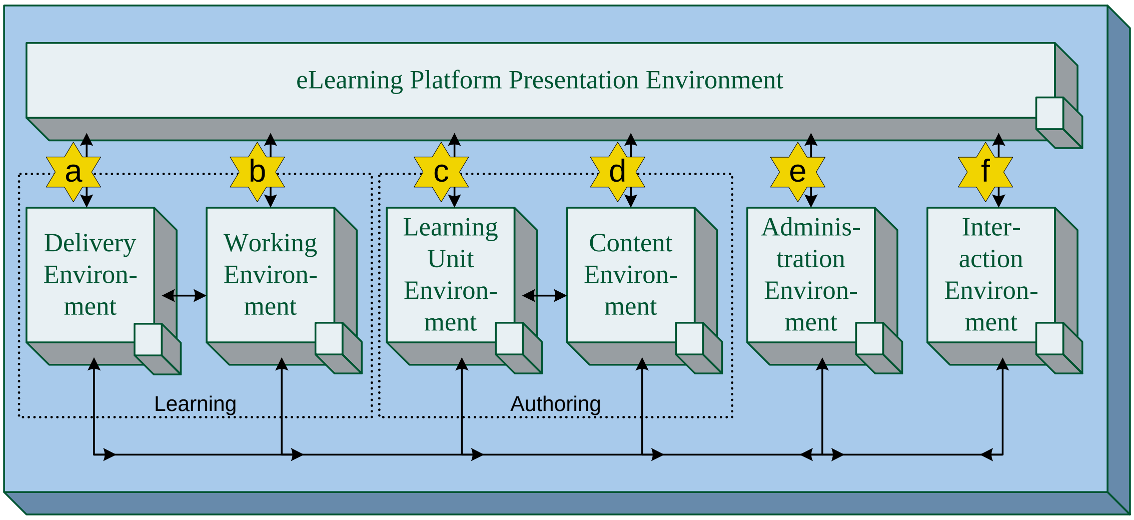


Figure 3.2: E-Learning Platform Presentation Environment [Mencke and Dumke, 2007c]

Use case a: Request and presentation of the next part of a course
Use case b: Request and presentation of personal annotations to a certain topic
Use case c: Creation and management of courses or certain course substructures
Use case d: Creation and management of Learning Objects
Use case e: Update of entries in a user model
Use case f: Interaction with other learners, tutors or experts

An important aspect of GUIs for e-Learning is the adaptability; the personalization of certain aspects based on collected information or assumptions about the user. That refers to all related environments and may result in adaptive navigation support, adaptive presentation and adaptive content [Kernchen, 2005]. Adaptive navigation support is related to the guidance of users and can be established by global and local support mechanisms, by local orientation, global support for orientation and by the management of individual views. Adaptive presentation can be achieved by the sorting of resource fragments, the adaptive content presentation due to different media formats and the adaptive provision of content because of differing quality, transmission contexts and different languages. Classic approaches like changes in font size, font type and font color can be used for adaptive presentation, too. Methods for adaptive content are e.g. basic, additional and comparing explanations, explanation variants and the sorting of information fragments [Kernchen, 2005]. Context adaptability is supported by the advantage to integrate different implementations of the proposed environments, extended with capabilities to receive and process context-sensitive information. By this, mobile, ubiquitous learning becomes possible.

The different environments themselves may interact with each other. A first primary relation exists between the two learning environments. The DE and WE are closely connected, because of the high possibility of exchanging data. Functionalities provided by the WE, like media processing, can be requirements of certain tasks of the course currently presented in the DE. Similar connections are needed for the LUE and CE. The

Learning Objects are integral parts of the courses being authored within the LUE.

Nevertheless, the AE and IE will exchange data with all other environments, because each one needs to be administered and collaboration between different users is always possible, too.

3.3 Knowledge Acquisition

The Delivery and Working Environments are grouping the functionalities of learning systems to enable the learning itself. Therefore, they mainly fulfill requirements demanded by learners. The DE presents the course, its structure, course metadata, enables course catalogue browsing, realizes the registration and is responsible for other all functionalities that are directly connected with the presentation of and working with learning content during the learning process.

The WE is grouping functionalities for the support of the learning process. That refers for example to classic requirements known from classroom learning. Components for Web search as well as for the access to certain repositories are needed to get additional information about the topic of the course. It is important for the personal learning progress – being able to make private annotations to the course content and to manage own additional information (for example: a list of links, a private file system). A scheduler for collaborative work and time management and the access to office tools are needed under certain circumstances. Figure 3.3 visualizes these chosen aspects for parts of the learning environments.

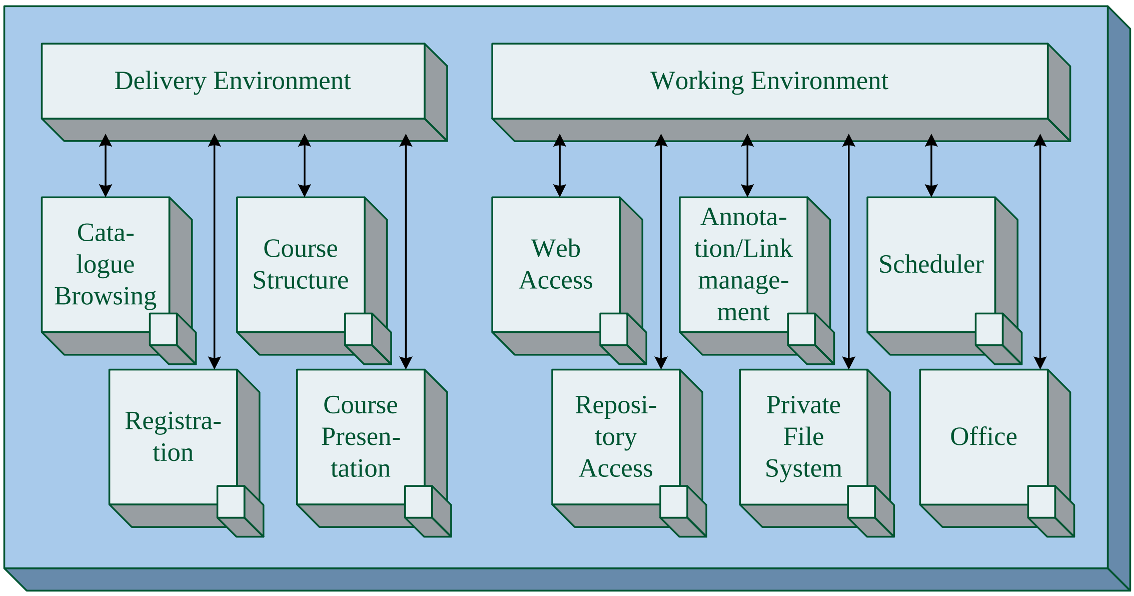


Figure 3.3: The Learning Environments [Mencke and Dumke, 2007c]

The learning environments need connections to the Administration and Interaction Environments and to the support layers. Administration for example is needed for the management of individual preferences, meanwhile interaction is fundamental for collaborative learning tasks. As for the other environments, the support layers provide access to basic information, repositories and functionalities that are needed for the functionality of the actual environment itself.

In the following, chosen approaches for the usage of agent technology within the domain of knowledge acquisition are sketched.

3.3.1 Agent-Mediated Online Learning

The agent-mediated online learning (AMOL) architecture targets the automation of an online learning process [Yi et al., 2001]. Therefore, the authors assumed three types of participants: the learners, the teachers and mediating education centers (see Figure 3.4). The difference to classic approaches is the existence of multiple education centers to provide the courses. A prototype was implemented with Aglet technology [IBM, 2002].

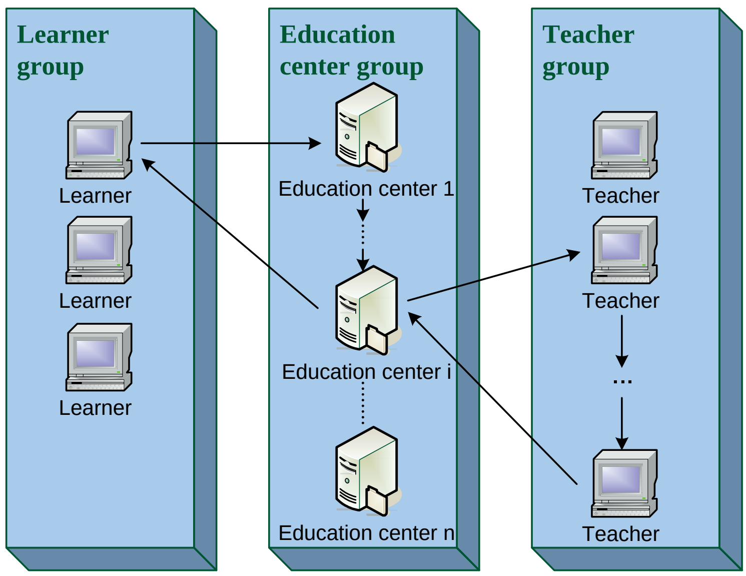


Figure 3.4: AMOL Architecture [Yi et al., 2001]

The implemented agents are mobile and their types are listed below:

- **Pegagocial agent:** tutoring based on task plan and feedback (answering the learner's questions and judging his answers)
- **Searching agent:** searching for appropriate learning content based on learner requests
- **Querying agent:** querying the various education centers for answers the pedagogical agent is not able to provide

3.3.2 Knowledge Assessment with JADE

A next architecture is described in [Anghel and Salomie, 2003]. It targets a special domain of e-Learning: the student assessment. The representativeness of this architecture is derived from its way of implementation. It is implemented by using JADE agent technology ([Telecom Italia, 2007]) in an applet of a Web site. Parts of the architecture are visualized in Figure 3.5.

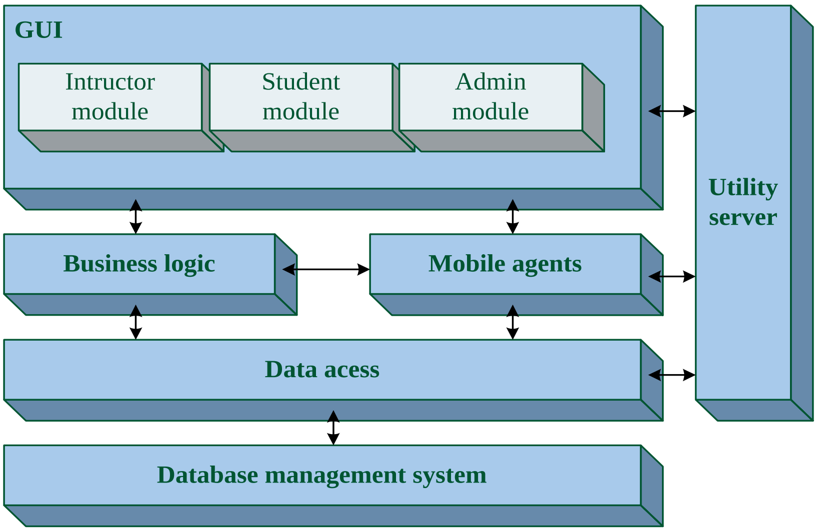


Figure 3.5: Architecture for Knowledge Assessment with JADE [Anghel and Salomie, 2003]

Agent technology was chosen because of scalability issues for many users and bandwidth/latency related problems of the classic client-server model. The authors identified the following tasks for agents in their domain of interest:

- **Personal assistant agent:** for human-computer-interaction
- **Server agent:** coordination of evolving tasks (e.g. handling self-assessment requests, generating corresponding evaluation engines)
- **Evaluation agent:** evaluating the tests based on test information (questions, answer options, correct answer) and assessment process information

3.3.3 File-Store Manipulation Intelligent Learning Environment

The File-Store Manipulation Intelligent Learning Environment (F-SMILE) was published in [Virvou and Kabassi, 2002]. It is intended to teach novices the usage of a graphical user interface. Therefore, it is protected and offers adaptive tutoring and help, based on the observed user actions. Used adaptation techniques are adaptive presentation and adaptive navigation support [Kabassi and Virvou, 2003].

Four agent types are implemented (see Figure 3.6):

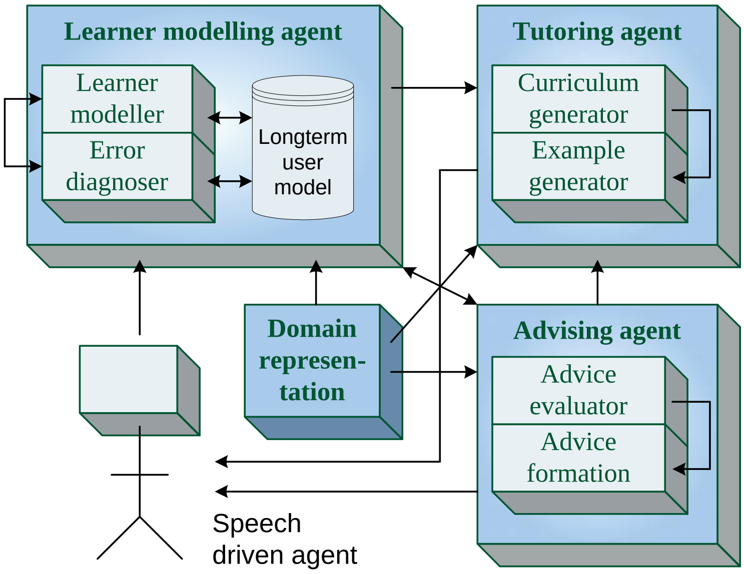


Figure 3.6: F-Smile's Architecture [Virvou and Kabassi, 2002]

- **Learner modeling agent:** observation of the learner's characteristics and identification of possible misconceptions
- **Advising agent:** simulation of a tutor's reasoning by the application of a defined formula that deals with the degree of similarity, typicality, degree of frequency, dominance to calculate the degree of certainty of the appropriateness of a given advice
- **Tutoring agent:** content, link and example adaptation based on learner information
- **Speech driven agent:** avatar for human-computer-interaction to provide entertainment and emotional function

3.3.4 Extended LMS "Samurai"

In [Ueno, 2005], the agent-based extension of the existing learning management system "Samurai" and an analysis of its usefulness is described. Agents are used to provide optimized instructional messages to a learner. Therefore, they identified nine primary variables of the user model as informational base for adapted message delivery.

A major part of their work was the comparison of courses held with and without the agent-based extension. The main results where:

- Reduced number of students, who gave up the course
- Improved test score
- Reduced variance of test score
- Increased learning time

3.3.5 Web-Based e-Learning Environment Integrating Agents and Computational Intelligence

A system for Web-based e-Learning integrating agents and computational intelligence is described in [Giotopoulos et al., 2005]. The platform front-end, the student questioner reasoner and the student model agent are connected with Web Services (Figure 3.7).

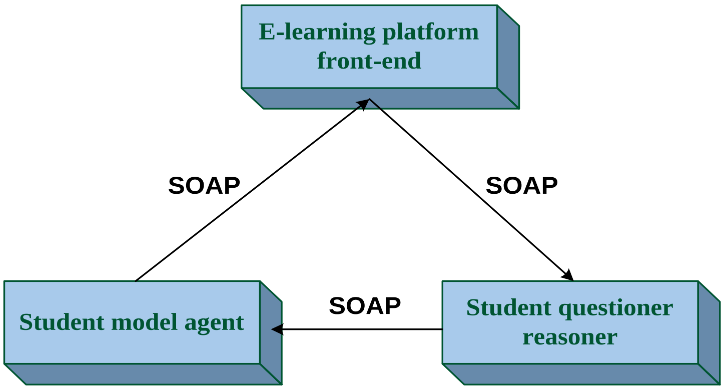


Figure 3.7: System Architecture with Web-Service-Based Interconnection [Giotopoulos et al., 2005]

The tasks of the student model agent are:

- Leading of the learner through the learning process
- Update of the learner model
- Access to possible interesting resources

3.3.6 Intelligent Learning Materials Delivery Agents

The Intelligent Learning Materials Delivery Agents (ILMDA) application was designed to deliver learning material to different students taking into account the content's usage history and the student's user profile.

The agents task is to learn from the available history data and to make assumptions about the appropriateness of learning material for certain students. The ILMDA architecture is sketched in Figure 3.8 [Soh et al., 2005].

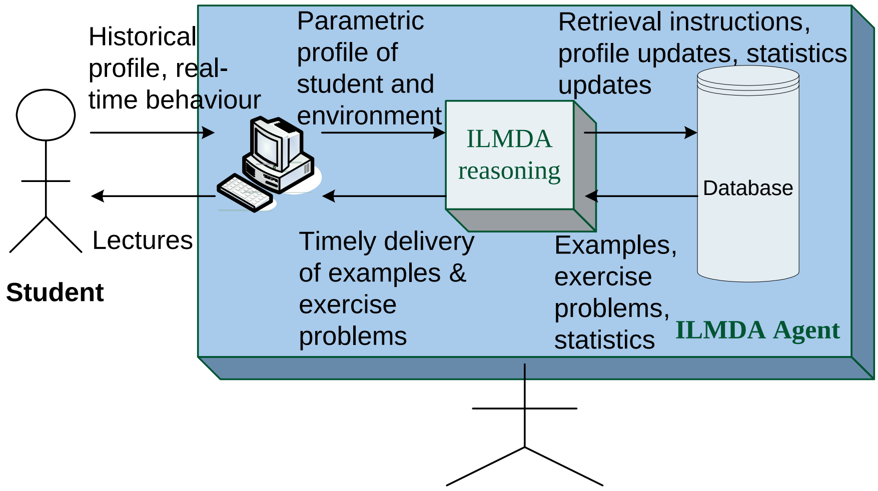


Figure 3.8: ILMDA Architecture [Soh et al., 2005]

3.4 Authoring in e-Learning

The Learning Unit Environment and the Content Environment focus on functionalities to support the authoring process of educational content (e.g. basic content, Learning Objects, assessments/tests, courses). The process' nature is iterative: the planning, design and production cycle is followed by a new iteration after an evaluation for continues improvement [Giotopoulos et al., 2005].

The CE provides functionalities for the planning, design, creation, assembly and management of basic content fragments. Thereby, different media types need to be taken into account. The LUE is focused on the processing of more complex content. Therefore, a learning unit is defined as a piece of information that is more complex than the content fragments and whose intended usage is targeting education. Entire courses and course substructures like assessments or tests are learning units.

The development and authoring of strategies for course assembly is a new key element of the proposed framework. Those, for example didactical, strategies are needed for the high quality of assembled learning resources, because they provide expert knowledge and user guidance for this complex task [Mencke and Dumke, 2007d]. Figure 3.9 is presenting chosen aspects of the CE and LUE.

Like the learning environments, the described authoring environments need connections to the Administration and Interaction Environments and to the support layers by the same token.

In the following, chosen approaches for the usage of agent technology within the domain of authoring in e-Learning are sketched.

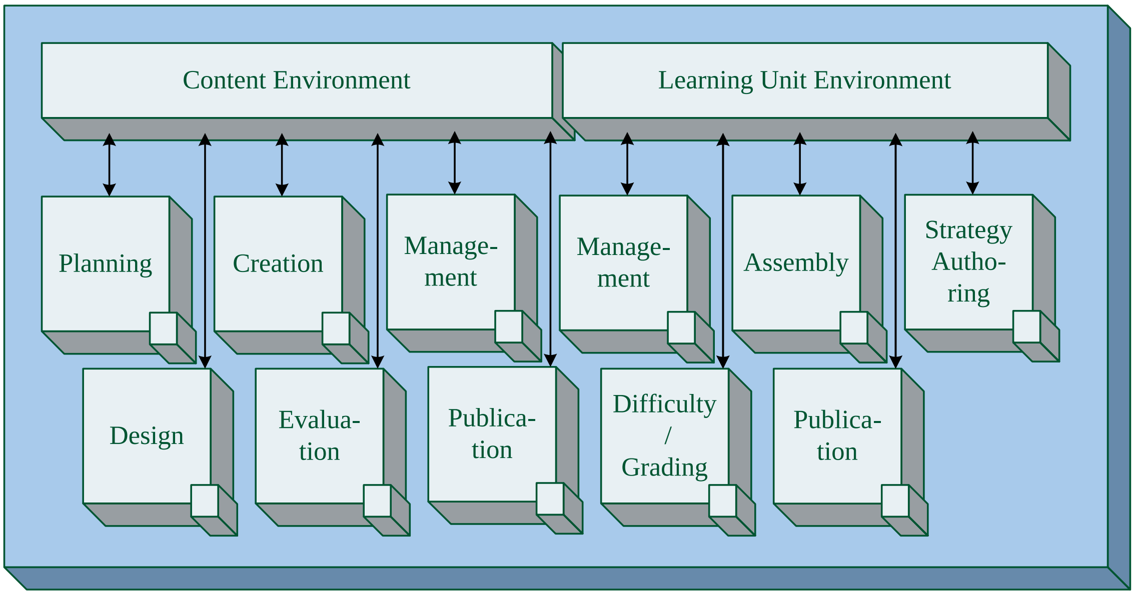


Figure 3.9: The Authoring Environments [Mencke and Dumke, 2007c]

3.4.1 ALFanet

ALFanet is a project intended to provide a framework to address the learners' need for activities and user-model-based content adaptation and the tutor's need for efficiency [van Rosmalen et al., 2005]. From an e-Learning point of view, the proposed architecture should base on available standards like IMS LD (see Section 2.4.3.1).

The resulting three tiers are the server layer, services layer and data layer. The server layer provides the user interface, manages application security issues and traces user interactions. The services layer is a composition of a set of application functionality and main logic providing services. Data management and storage are tasks of the data layer.

Agent technology is intended to be used for:

- Personalized guidance for learners
- Support based on an instructional model
- Support for course creators by monitoring the difference between design and current learning process

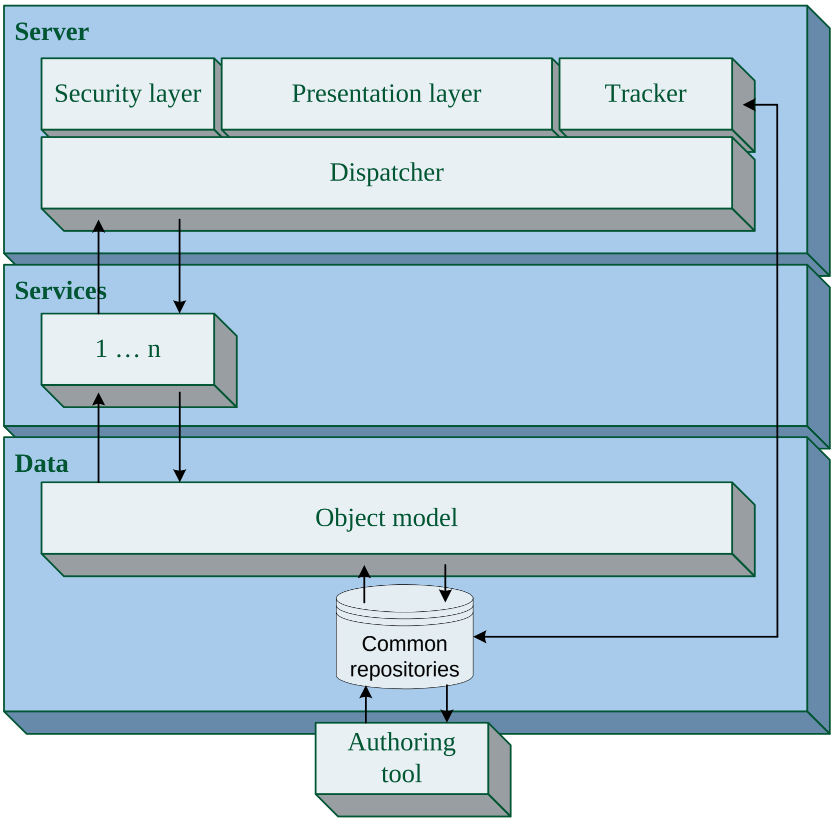


Figure 3.10: ALFanet Architecture [van Rosmalen et al., 2005]

3.4.2 MAS for Undergraduate Computer Science Education

Another multi-agent system in the domain of e-Learning is proposed in [Shi et al., 2000]. The authors are targeting the support of student-centered, self-paced and highly interactive learning in undergraduate computer science education. They are following a hybrid approach of a problem-based and case-based learning model to support creative problem solving and mechanical experience simulation.

From a technical point of view, they prototypically implemented a Web-based GUI additionally using Java RMI, JavaSpace and JATLite ([Jeon et al., 2000]). The architecture is sketched in Figure 3.11 and its main elements are several agents for certain purposes, a Web-based interface and a digital library for student profiles and course content.

The implemented agents and their assigned tasks are:

- **Course agent:** management of course materials and teaching techniques for a course
- **Teaching agent:** tutoring a course based on learning content and teaching strategy of the course agent
- **Personal student agent:** observation of the learner and management of his user profile

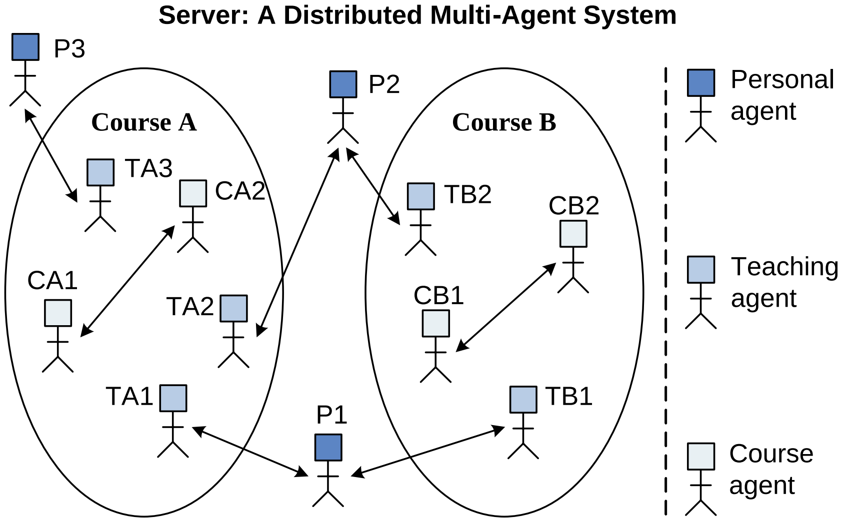


Figure 3.11: MAS for Asynchronous Learning [Shi et al., 2000]

3.4.3 Knowledge Intelligent Conversational Agents

The Knowledge Intelligent Conversational Agents (K-InCA) system was designed to help people to adapt new behaviors [Angehrn et al., 2001] (see Figure 3.12).

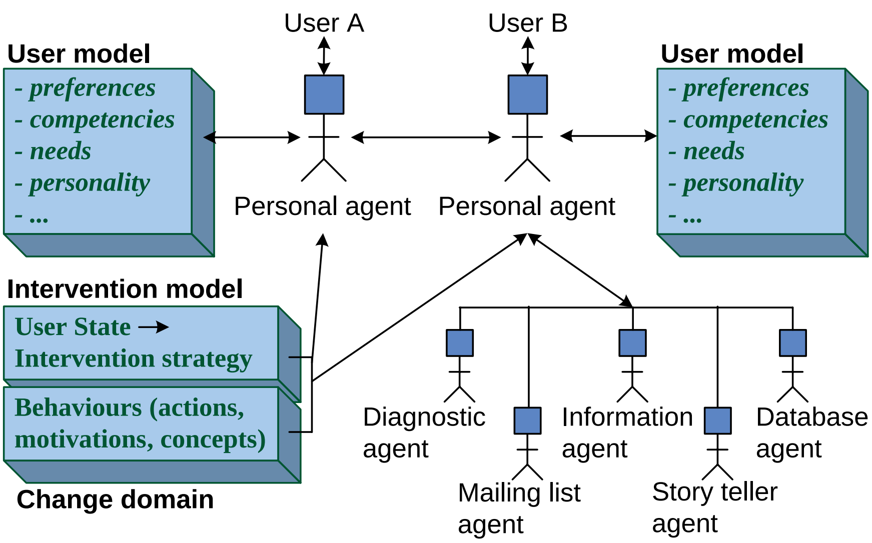


Figure 3.12: Overall K-InCA Architecture [Angehrn et al., 2001]

Therefore, the following methodology of human dealing with new behaviors was adopted within the implemented architecture:

- **Stage 1:** being unaware of new behaviors
- **Stage 2:** becoming aware of the new behaviors and the underlying concepts
- **Stage 3:** developing of interest in the new behaviors
- **Stage 4:** experimentation of how the new behaviors "work" for the human
- **Stage 5:** adoption of the new behaviors in the case of positive experience

The implemented agents fulfil several tasks: examination of user's actions, maintaining a "behavioral profile" (reflecting the level of adoption of the desired behaviors) and providing adaptive learner guidance for mentoring, motivation or stimulus. This agent-based adoption follows these steps:

- **Step 1:** observing the user's actions
- **Step 2:** activation of the diagnostic agent who updates the user model
- **Step 3:** selection a new current learning objective, solicitation of proposals from the expert agents to achieve the learning objective, proposal and selection of one or more intervention strategies
- **Step 4:** implementation of intervention

3.5 Interaction in e-Learning

Following Brown and Duguid in [Brown and Duguid, 2000], learning is "a remarkably social process. Social groups provide the resources for their members to learn." There are several social reasons for interactivity. It decreases isolation of the participants and increases the flexibility to adapt new conditions. Furthermore, it involves more human senses into learning and increases the variety of learning experiences (multicultural environments, communication capabilities, etc.). Interactivity builds a sense of group identity and community. Nonetheless, interaction sometimes is a fundamental requirement for certain courses [Belanger and Jordan, 2000]. Figure 3.13 is dedicated to chosen fragments of the Interaction Environment.

The proposed framework integrates multiple communication channels as technical support for human-to-human and human-to-computer interaction, respectively. It is extended by additional support tools. An avatar is used as a human representative for personalization, identification, anonymization and as backup in case of absence. Another component is the grouping tool, which is intended to form groups of learners for certain collaborative learning tasks based on user model information and appropriate psychological theories.

Interaction approaches can be distinguished in synchronous and asynchronous. Synchronous tools can provide text-, audio- or video-based chat, application/screen sharing, synchronous Web browsing, shared whiteboards, etc. Asynchronous tools can span for example email, wikis, forums, mailing lists or audio/video replay [SUN Microsystems, Inc., 2003].

The IE technically needs close connections to all other environments, because collaborative learning and working may occur in every proposed environment.

In the following, chosen approaches for the usage of agent technology within the domain of interaction in e-Learning are sketched.

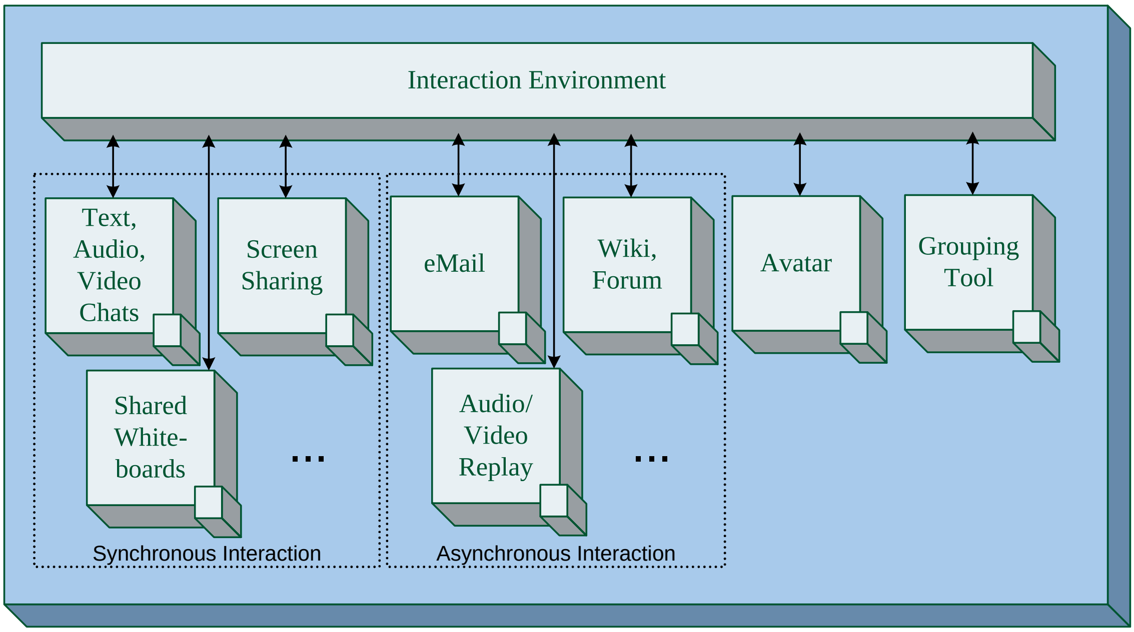


Figure 3.13: The Interaction Environment [Mencke and Dumke, 2007c]

3.5.1 Intelligent Multiagent Infrastructure for Distributed Systems in Education

The Intelligent Multiagent Infrastructure for Distributed Systems in Education (I-MINDS) was published by [Soh et al., 2004] and is intented to support cooperative learning among students in classic classroom teaching as well as in distance education. Therefore, the application establishes an agent-federated "buddy group": a close-knit student group where its members exchange messages and help each other understand the lectures.

The implemented agent types are teacher agent (see Figure 3.14) and the student agent:

- **Teacher agent:** assistant of the teacher to monitor the students and to adapt the class
- **Student agent:** interaction with teacher agent and other student agents to facilitate cooperative learning activities

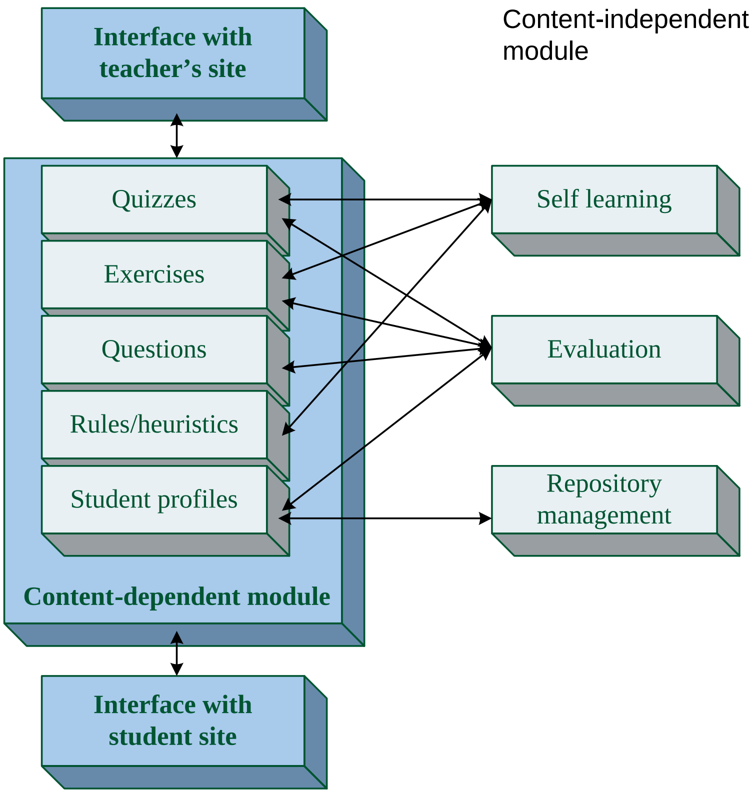


Figure 3.14: Structure of I-MINDS Teacher Agent [Soh et al., 2004]

3.5.2 Virtual Reality Game for English

The Virtual Reality Game for English (VIRGE) is an intelligent tutoring systems to teach English orthography and grammatical rules. Therefore, a virtual reality game was implemented to supply the opportunity to play a 3d-game [Virvou and Katsionis, 2003]. The architecture of the evolving MAS is presented in Figure 3.15.

Several agent types are implemented:

- **Animated agents:** for human computer interaction
  - Virtual enemy: asking questions to learners
  - Virtual adviser: showing empathy to the learners, help for learners
  - Virtual companion: appears when the student has declined much from his usual actions or has made a repeated mistake
- **Student profile agent:** collecting learner information and updating the user profile

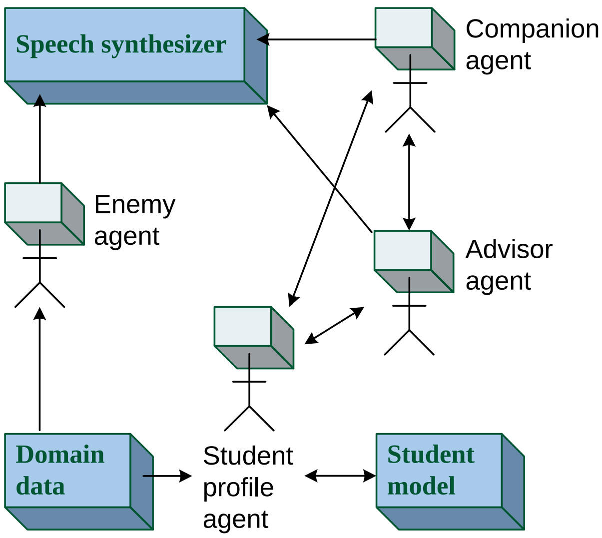


Figure 3.15: VIRGE MAS Architecture [Soh et al., 2005]

3.6 E-Learning System Administration

The administration environment provides access for the management of all environments, system components and support layers. The possibilities are ranging from simple observation to the integration of new components or the update of existing ones. The access to components and the provided functionalities is limited by the access restriction of a particular user.

The most extensive access is possible for the administrators. All other user groups have access to their specific objects and to the adjustment capabilities of the environments where they have access to.

A very important example of needed accessibility is the manageability of the user model for the depicted learner. If it is available and manageable for individuals it gives learner control and responsibility [Kernchen and Dumke, 2007a]. Thereby, it supports meta-learning activities like the monitoring of learning, the setting of personal learning goals; it is the basis for planning goals and supports the reflection about and the tracing of the learning progress by the comparison of set goals. As presented in Figure 3.16, the AE needs connections to all other environments.

Related functionalities are grouped in the user, institutional and technical area. Within the user area all aspects are pooled that are related to specific user tasks. Thereby, not only learners, but all possible users have access to administration functionalities that are targeted to them, their tasks or resources. Institutional management facilities provide access to services, functionalities and resources that are related to the management of meta-activities within the institution like user management, course management, class management and certification management. Management capabilities for the classic administrator role are pooled within the technical area.

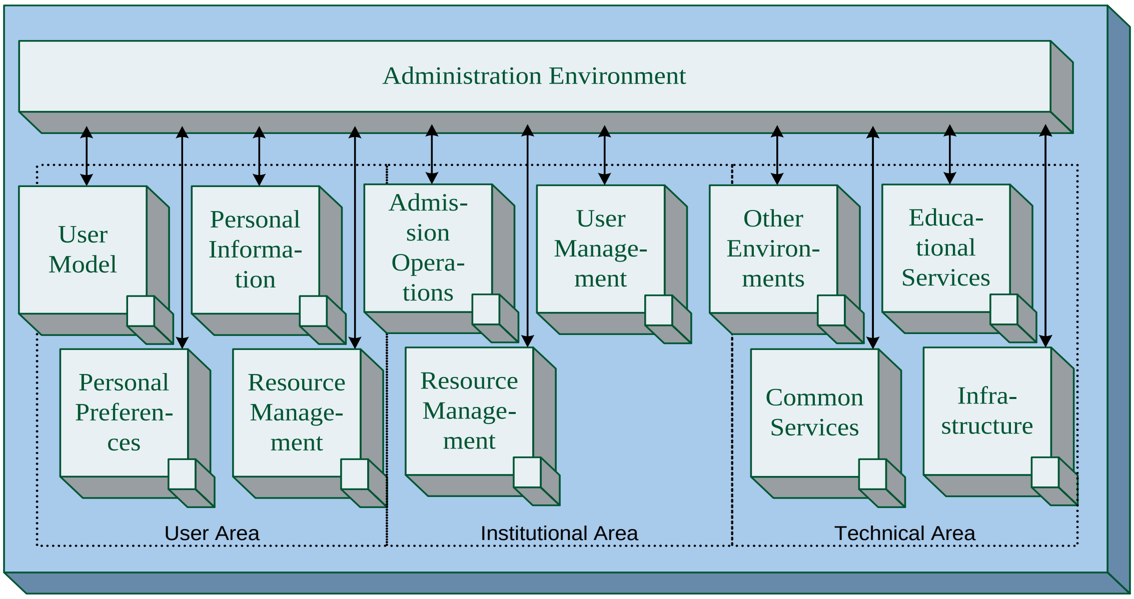


Figure 3.16: The Administration Environment [Mencke and Dumke, 2007c]

In the following, a chosen approach for the usage of agent technology within the domain of e-Learning system administration is sketched.

Multi-Agent System for e-Learning and Skill Management

The Multi-Agent System for e-Learning and Skill Management (MASEL) presented in [Garro and Palopoli, 2002], [Garro and Palopoli, 2003] and [Garro et al., 2003] targets the automation of certain tasks within the context of skill management for employees. That includes the individuation of student learning objectives, the evaluation of competence gaps, the control of learner improvements and the creation of the bridge between the individual learning objectives and the ones of the learner's organization. The system's architecture is presented in Figure 3.17.

In MASEL, agents are mainly used for communication between distributed components, for monitoring the environment, for autonomous operations, reasoning and to perform complex message-based operations. Therefore, this system was implemented in JADE ([Telecom Italia, 2007]) making extended usage of XML for ontology representation and handling as well as for communication. The created MAS itself contains seven agent types and consists of at least one CLO Assistant Agent, one Skill Manager Agent, one Content Agent, one Learning Paths Agent, one CCO Assistant Agent, one User Profile Agent and n Student Assistant Agents, that are described below.

The *CLO Assistant Agent* (CLO) supports the Chief Learning Officer in defining a learning strategy for the user in terms of roles and required competencies based on the organization's learning objectives. Therefore, the CLO supports the management of roles and competencies, the management of potential learners, the suggestion of suitable employees for certain roles, the definition of priorities and constraints as well as the presentation of individual learning activities, based on historical data of the employees.

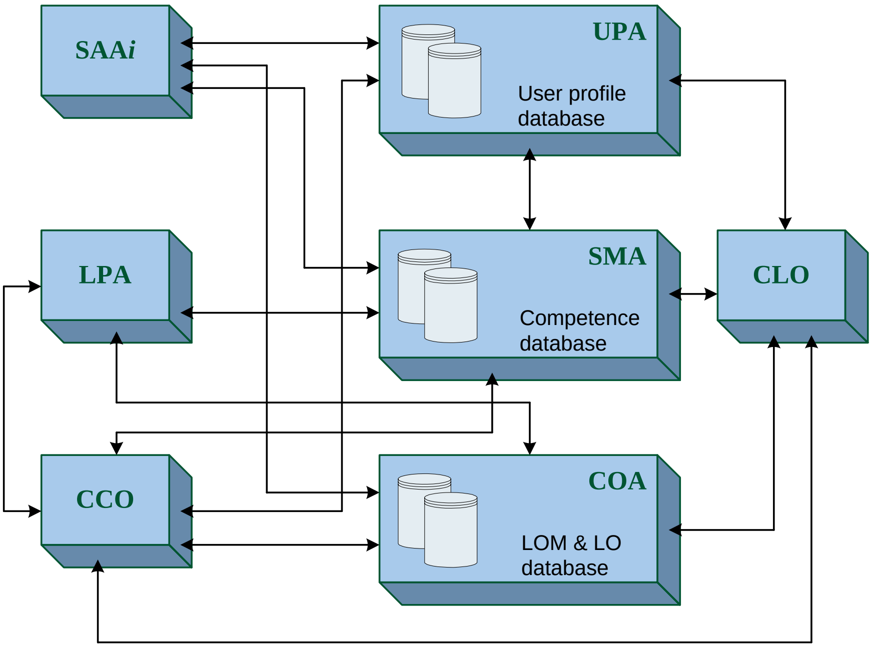


Figure 3.17: Architecture of the MASEL System [Garro et al., 2003]

The *Skill Manager Agent* (SMA) manages general skill information of the organization. Therefore, all related data, including the ones processed by the CLO, are stored in an XML document. Additional data are the individual roles and competencies of employees. This agent provides services to insert, delete and update individual and organizational role and competency information and functionality to query the data structures for certain reasons.

The *Learning Path Agent* (LPA) tries to create learning paths to fill identified competency gaps of the employees. It is related to the Student Assistant Agent (SAA) and it is used to create tests to identify and evaluate the missing skills, to enrich and modify the learning path and to inform the CCA Assistant Agent for missing Learning Objects.

The already mentioned *Student Assistant Agent* (SAA) is associated to an individual student and its task is supportive to fill his competency gap for a certain role. Therefore, it presents information about the identified competency gaps, presents the test created by the LPA, modifies the course based on user feedback and manages information about the learning progress.

The *Content Agent* (COA) manages the database containing the Learning Objects and thereby provides the content needed by the LPA and SAA to adapt a course. This agents inserts, deletes, modifies and queries the stored Learning Objects.

The *CCO Assistant Agent* (CCO) supports the Chief Content Officer in dealing with the Learning Object database. Therefore, it cooperates with the COA and can present the learning history of employees.

The last implemented agent type is the *User Profile Agent* (UPA) for the storage of needed user-related information. It manages the user's log-in, his profile information and updates his competency levels (together with the SMA and SAA).

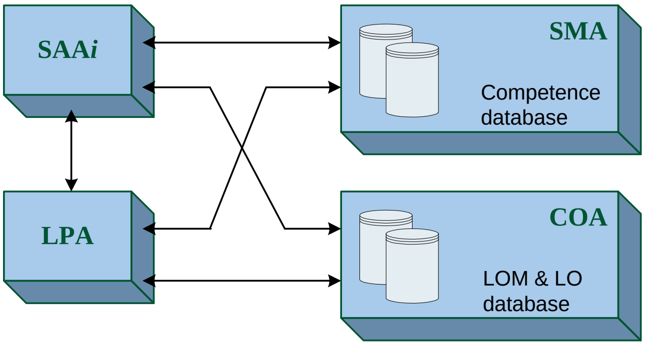


Figure 3.18: Personalized Learning Path in the MASEL System [Garro et al., 2003]

To create individual learning paths with the implemented agents (see Figure 3.18), different learning strategies can be applied, e.g. time minimization and knowledge maximization. The construction process is semi-automatic, has three steps and stops with a complete learning path, reaching the learning objective. Didactics is applied in terms of prerequirements that need to be fulfilled.

- **Step 1:** creation of a set of Learning Objects based on learning objectives
- **Step 2:** presentation of this set to the user
- **Step 3:** manual choice of appropriate Learning Objects as a subset

3.7 E-Learning Infrastructure and Common Services Layers

The infrastructure and common support layers provide basic functionalities for the e-Learning services layer and environmental parts. This separation idea was adopted from [Open Knowledge Initiative, 2003] and [IMS Global Learning Consortium, Inc., 2003a] and is based on the same motivation. The intended goals are twofold.

- Thereby, more complex functionalities of the upper framework elements do not need to re-implement already existing ones; redundancy is avoided.
- By the separation an easier intra-institution work sharing is possible, due to increased portability of the system.

This presented framework differs in the assignment of specific functionalities to certain support layers and environments, as described below. The *infrastructure layer* is responsible for basic networking and data transport, selected services are:

- ○ Exchange of data structures in terms of physical communications, messaging and transaction needs [IMS Global Learning Consortium, Inc., 2003a]
- ○ Support of complex multi-zone agent communication ([Schools Interoperability Framework (SIF) Association, 2007], [Ganchev et al., 2007])
- ○ Provision of the needed agent platform ([Foundation for Intelligent Physical Agents (FIPA), 2002a], [Foundation for Intelligent Physical Agents (FIPA), 2006])
  - Agent management
  - Message transport service
  - Agent directory
  - Services directory
  - Agent communication language (ACL)

The *common service layer* provides generic functionalities for the upper layer and the framework environments like (adopted and extended from [Open Knowledge Initiative, 2003]):

- ○ Authentication and authorization
- ○ Rights management, validation
- ○ Service discovery, database control for ([SUN Microsystems, Inc., 2003]):
  - Learning content
  - Learning meta data
  - Learning assessment
  - Learning administration
  - User repository
- ○ Filing and automated resource update
- ○ Logging of technical system aspects
- ○ Virtual centralization of remote resources

Additional possible services are summarized in a brief overview in [Wilson et al., 2004] as part of the ELF Initiative that is targeted towards a service-oriented approach for e-Learning.

In the following, chosen approaches for the usage of agent technology within the domain of infrastructure respectively common services are sketched.

3.7.1 Knowledge On Demand

The Knowledge On Demand (KOD) project is an initiative of a consortium consisting of five members from four European countries. Its target was the development of a platform independent solution for the publishing, brokering and delivering of Learning Objects and packages. Thereby, interoperation and interchange between different service providers and platform vendors should be enhanced [Trabucchi, 2001].

The presented solution argues to include all important e-Learning standards by the means of existing Web technologies as well as the agent technology [Sampson and Karagiannidis, 2002]. The architecture is visualized in Figure 3.19.

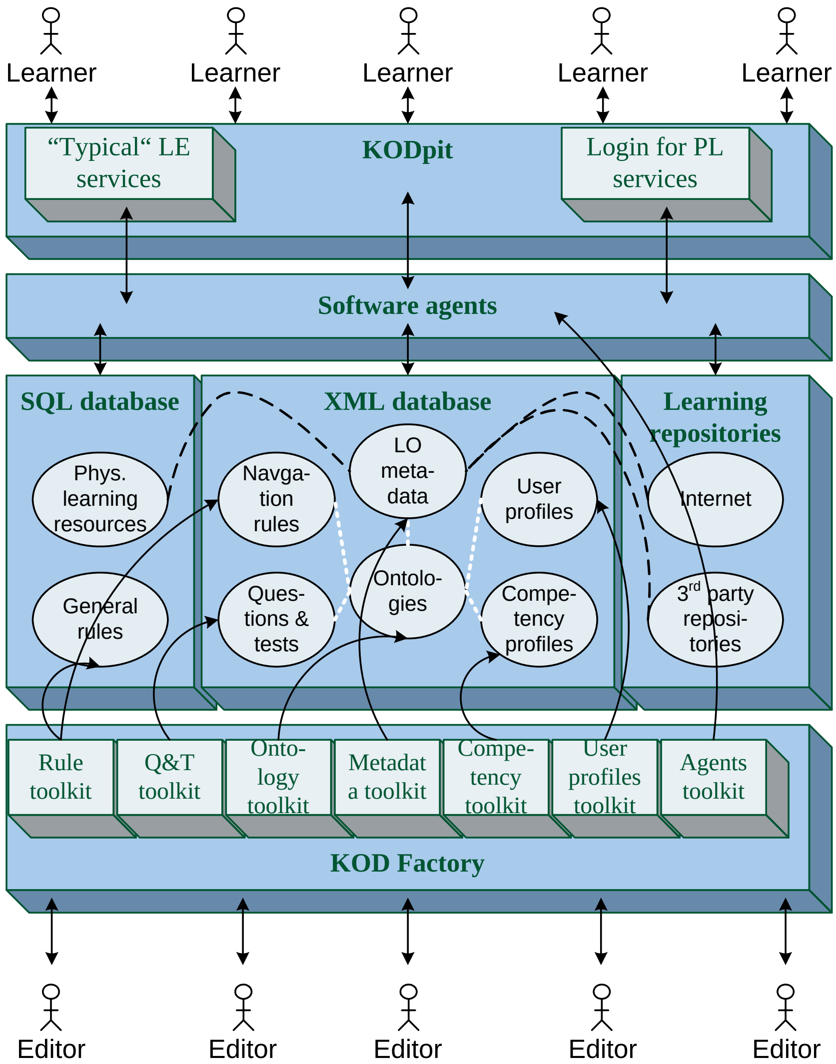


Figure 3.19: Knowledge On Demand Architecture [Sampson et al., 2002]

The intended main features are: individualized learning paths, user profiling, integration of multiple e-Learning standards and the integration of agent technology. Agents are addressed for the following tasks [Sampson et al., 2002]:

- Observing the learner
- Interaction of architectural components
- Search for information in internal and external databases
- Knowledge analysis, monitoring, generation, adaptation and delivery

3.7.2 Coaching FRED

Coaching FRED is an application targeting the organization and coordination of the lifelong learning process in a company [Smolle and Sure, 2002]. Agent technology is used for communication and interaction issues among different FRED-implementations.

The key objectives targeted by the project are the support of a skill-transition strategy, the active information of employees, the improved service for employees, the support of education staff and the general optimization of the learning process. The system's architecture is sketched in Figure 3.20.

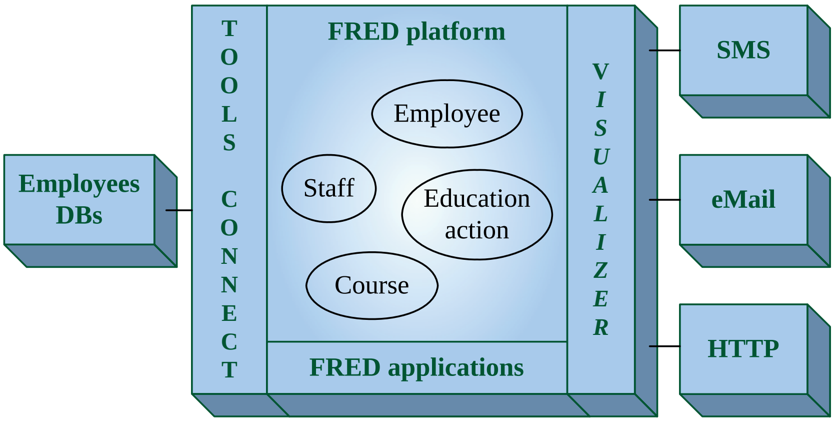


Figure 3.20: FRED Solution Concept [Smolle and Sure, 2002]

Several steps for the interaction with the system can be identified:

- **Step 1:** initialization of the application
- **Step 2:** user: creation of a personal task profile
- **Step 3:** application: offering of courses
- **Step 4:** user: optional creation of a personal interest profile
- **Step 5:** application: offering of additional topics
- **Step 6:** user: feedback of missing offers
- **Step 7:** application: information about new available courses
- **Step 8:** user: adaption of user profile; restart of process

3.7.3 Distributed e-Learning Center

Current research activities try to extend the Distributed e-Learning Center (DeLC) by agent technology [Stojanov et al., 2005]. Service-oriented e-Learning and e-Teaching should be extended for mobile support. Agents will serve as flexible personal assistants. Agent-related tasks for this second version of DeLC are:

- Intelligent interpretation of data
- Intelligent interpretation of exchanged content
- Communication with existing functional modules (Web Services)

Therefore, personal agents are developed for the processing of user profiles and the access of services. The service agents' tasks are the processing of profiles and models of existing services.

3.8 Specialized e-Learning Services

This layer provides specialized e-Learning functionalities. Therefore, they can base on services of lower support layers to provide related functionality to the upper environments. Thereby, the provided services reveal a fundamental educational and/or crossover nature for the certain environments.

As the most specialized support layer, this collection of e-Learning-specific services represents a second dimension of the proposed framework. The more vertically specialized functionalities of the environments are based on and are supported by multiple adopted implementations of the proposed services. In Figure 3.21 the hierarchy of environmental components is depicted in the upper boxes, meanwhile the dots within the net below visualize potential cooperation with the educational services.

To profit from the agent-supported realization of this framework, the implementation and offer of certain e-Learning-specific functionalities of the presentation environments as educational services is proposed. That relates to:

1. Content assembly and sequencing service ([SUN Microsystems, Inc., 2003], [Advanced Distributed Learning (ADL), 2006b])
2. Content adaptation service
3. Scheduling service [Open Knowledge Initiative, 2003]
4. Learning planner [IMS Global Learning Consortium, Inc., 2003a]
5. Annotation/link management service
6. Cataloging service ([IMS Global Learning Consortium, Inc., 2003a], [IEEE Learning Technology Standards Committee, 2002b])
7. Grouping tool
8. Interface to external office tools
9. Brokering service for educational material [Schools Interoperability Framework (SIF) Association, 2007]

More fundamental services are for example:

10. Evaluation (of e.g. learning progress, learning results, content usage, course usage, user preferences, strategy usage, ...) [IEEE Learning Technology Standards Committee, 2002b]
    - Collecting evaluation data: logging of education-related events, like learner profiling
    - Storing evaluation data
    - Processing evaluation data
    - Evaluation provision
11. Educational resource management (e.g.: content, learning unit, strategies)
12. Registration for new courses

13. Knowledge management
14. Report management
15. Dictionary [Open Knowledge Initiative, 2003]
16. Mobile learning management [MOBIlearn Project Consortium, 2005]
17. User model service (management, update, . . .)

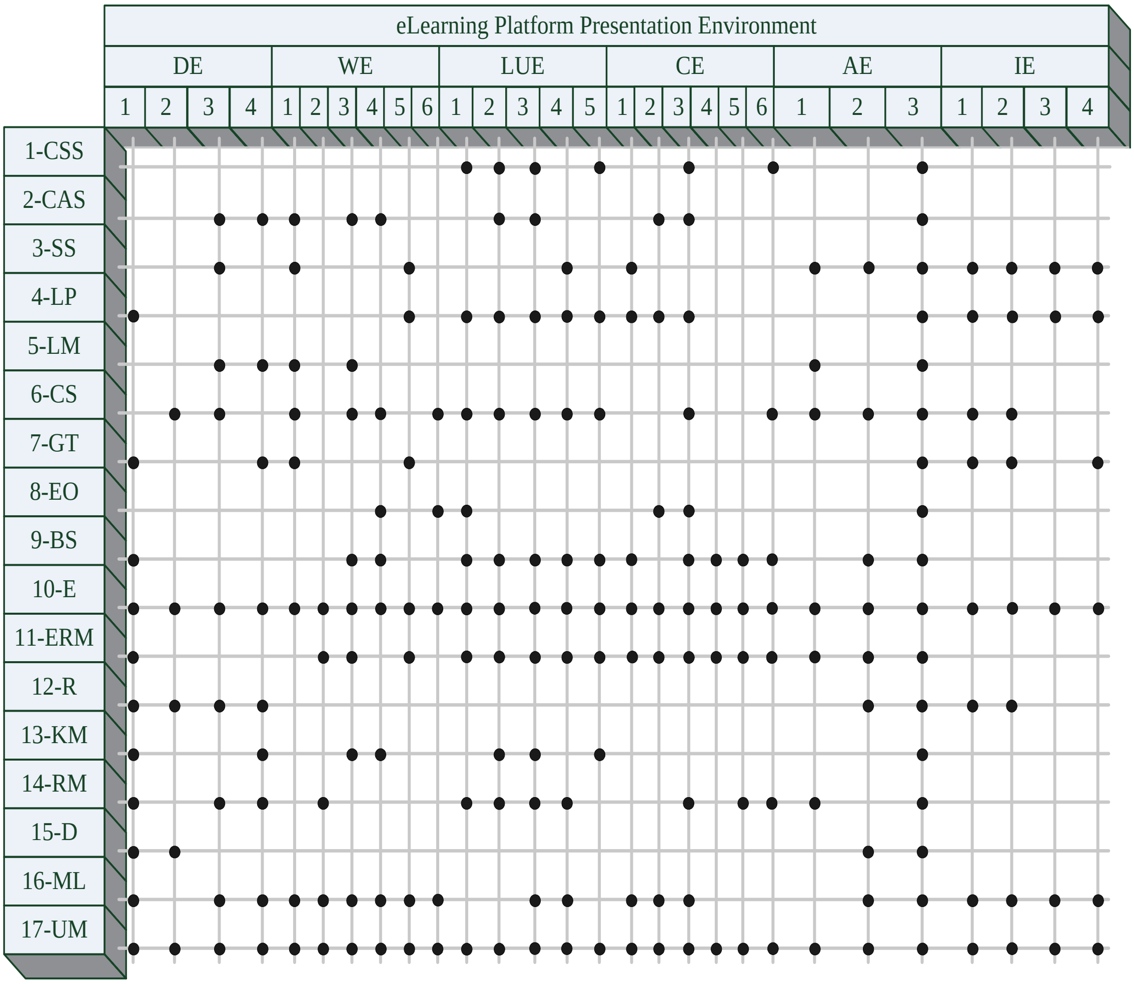


Figure 3.21: Two-Dimensionality of Environmental Functionalities (h) and Services (v) [Mencke and Dumke, 2007c]

In the following, chosen approaches for the usage of agent technology within the domain of e-Learning specific services are sketched.

3.8.1 Double Agent Architecture

The Double Agent Architecture for educational applications is presented in [Rahkila, 2001]. It is a user-centered and adaptive multi-agent architecture focusing on the identification of learners and the logging of their actions.

The architecture is named "Double Agent Architecture" because of the dual nature of the used agent that represents the learner as well as the teacher. A user request needs to be verified by an agent before it is processed by an agent. Fundamental aspects of the architecture are sketched in Figure 3.22.

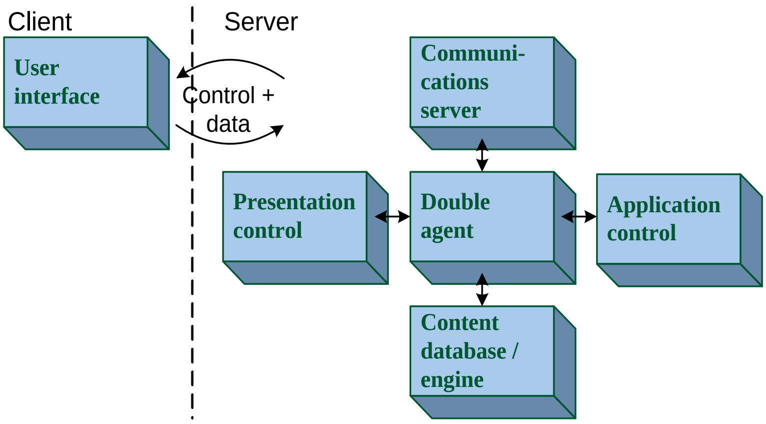


Figure 3.22: Double Agent Architecture [Rahkila, 2001]

3.8.2 User-Centered and Adaptive Interaction Multi-Agent Architecture

A user-centered and adaptive interaction multi-agent architecture is described in [Fernández-Caballero et al., 2003]. It is based on the idea that humans are different and the systems should adapt to them and not the other way around [Preece et al., 1994]. Agent technology artifacts of this architecture are intended to be used for certain aspects of e-Learning, e-Teaching and for interaction purposes. The proposed key features are:

- Social computing
- Logging of interaction and application of appropriate metrics
- Application of appropriate metrics for preference measuring

The interaction aspect of the proposed architecture is depicted by the interaction MAS and is visualized in Figure 3.23. The related task-specific agents are the upgrading agent (update the user interface with new information for the student), the preferences agent (logging of learner interaction preferences), the accounting agent (observing the learner's requests for other Web pages), the control agent (transferring learner preferences from preference agent to updating agent) and the performances agent (calculation of preference metrics).

A next MAS is the e-Teaching MAS including the teaching control agent that observes the learning system and provides suggestions to the teacher.

The learning MAS intends to maximize course learning. Therefore, the learning control agent is the information mediator for the other agents of this MAS. The theory agent delivers appropriate theory Web pages on requests of the control agent of this

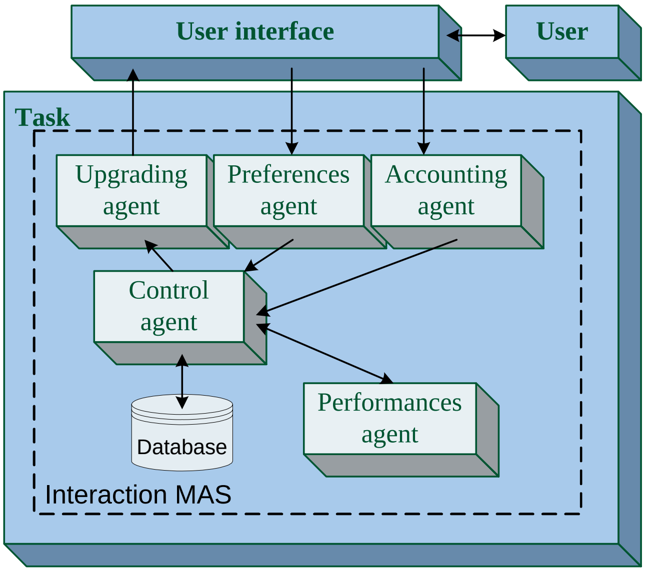


Figure 3.23: Interaction MAS Architecture [Fernández-Caballero et al., 2003]

MAS. The practice agent selects and delivers needed exercises and the test proposes agent requested tests.

3.8.3 Faded Information Field Architecture

The faded information field architecture is intended to handle a high rate of service provision and utilization requirements [Sadiig, 2005]. It is an approach for the improved provision of e-Learning by decentralizing it. Therefore, information provision is improved by communication improvement in a distributed environment. The architecture (see Figure 3.24) replicates content on demand to handle increased requirements in terms of service availability and utilization. Therefore, the amount of information that is stored as well as the information update frequency are inversely proportional to the distance of the node to the service provider. The authors distribute information across a network of nodes instead of storing it in a certain node.

Two major types of agents are suggested: pull agents (P1A) as mobile agents for acquiring and providing certain information for learners and push agents (P2A) that provide the services.

The authors strive for the following advantages:

- Increased reliability
- Reduced access time
- Autonomous determination of the amount of stored information
- Efficient content update
- Improved fault tolerance by the decentralization of information

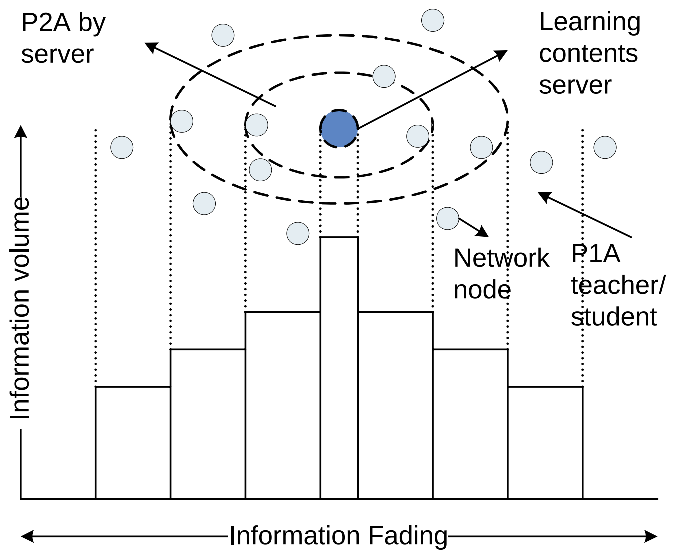


Figure 3.24: Faded Information Field Architecture [Sadiig, 2005]

3.8.4 Agent-Based Personalized Distance Learning System

A very light-weighted and abstract agent-based system for personalized distance learning is proposed in [Koyama et al., 2001]. It uses standard Web technologies with an agent-technology-enhanced server for content delivery (see Figure 3.25).

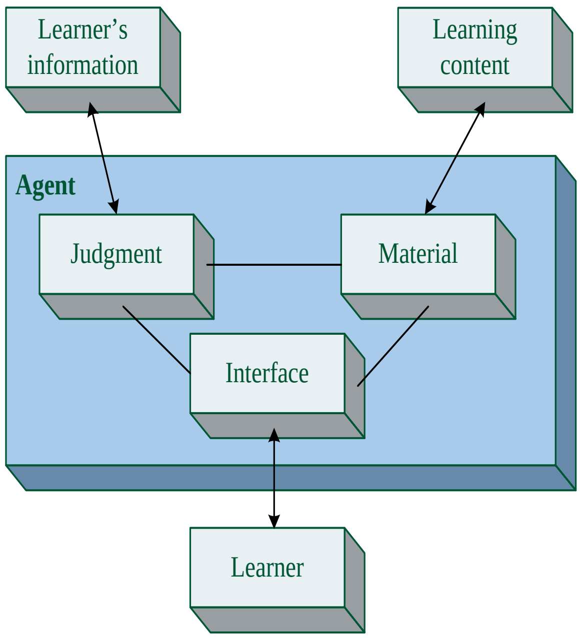


Figure 3.25: Architecture of an Agent-Based System for Personalized Distance Learning [Koyama et al., 2001]

The agent's main tasks are:

- Observing the learner and storing relevant information
- Management
- Analysis of information
- Judgment of the learner's progress
- Management of learning content
- Interaction with the learner

3.9 Discussion

Current research activities already led to the development of several e-Learning systems using agent technology. This chapter described chosen approaches and outlined the used agent types and characteristics.

Agents are implemented for different reasons and are affecting different target types. To the later belong the user itself, internal application components, internal databases and external applications (as proxy). Figure 3.26 elucidates the focus of current research towards user-centered agent technology for e-Learning. This observed information is a primary information source for proactive adaptation processes. About 50% of agents within existing approaches are targeting this aspect (see Figure 3.26).

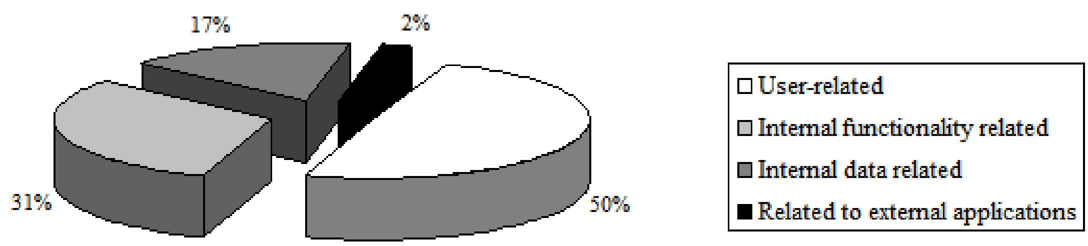


Figure 3.26: E-Learning Data Artefact Coverage by Agents [Mencke and Dumke, 2007c]

Figure 3.27 visualizes application options of agents for certain types of e-Learning functionality. Again, user-centered functionality is one main aspect for the usage of agents; that refers for example to knowledge delivery, notification, motivation and several objectives of human-computer-interfaces in general. Chosen observable targets are the user, learning objects, other knowledge resources and certain system artifacts. The "support" class of functionality summarizes aspects like decision taking, recommendations, tutoring and search capabilities. Furthermore, agents are used to manage knowledge, system components, learning activities and several aspects of user models, meanwhile another application area for this technology is the processing of several information as for example content, several learning units or evaluation data. Agents are used for adaptation and generation, too.

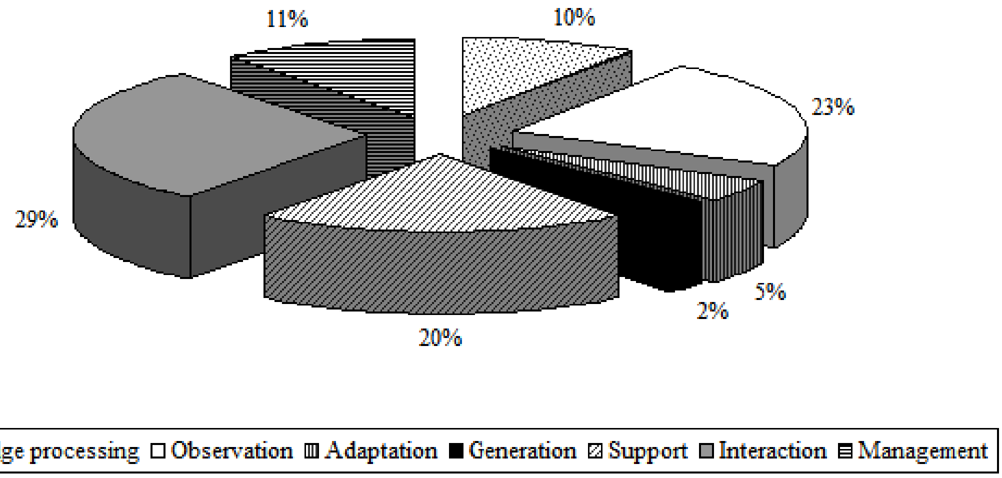


Figure 3.27: E-Learning Functionality Coverage by Agents [Mencke and Dumke, 2007c]

Table 3.1 summarizes the presented approaches for proactive content provision applications in the domain of e-Learning.

Obviously, the domain under survey is a current scientific research area. The future will reveal new trends and novel solutions, especially regarding proactive content.

FRAMEWORK COMPONENT	SELECTED APPROACHES
e-Learning Platform Presentation Environment	ABEL-GUI [Kernchen and Dumke, 2007c]
Knowledge Acquisition Environments	Agent-Mediated Online Learning [Yi et al., 2001]
	Knowledge Assessment with JADE [Anghel and Salomie, 2003]
	File-Store Manipulation Intelligent Learning Environment [Virvou and Kabassi, 2002]
	Extended LMS “Samurai” [Ueno, 2005]
	Web-based e-Learning Environment Integrating Agents and Computational Intelligence [Giotopoulos et al., 2005]
	Intelligent Learning Materials Delivery Agents [Soh et al., 2005]

FRAMEWORK COMPONENT	SELECTED APPROACHES
Authoring environments	ALFanet [van Rosmalen et al., 2005]
	MAS for Undergraduate Computer Science Education [Shi et al., 2000]
	Knowledge Intelligent Conversational Agents system [Angehrn et al., 2001]
Interaction Environment	Intelligent Multiagent Infrastructure for Distributed Systems in Education [Soh et al., 2004]
	Virtual Reality Game for English [Virvou and Katsionis, 2003]
Administration Environment	Multi-Agent System for e-Learning and Skill Management [Garro and Palopoli, 2002]
E-Learning Infrastructure and Common Services Layers	Knowledge On Demand [Sampson et al., 2002] [Virvou and Katsionis, 2003]
	Coaching FRED [Smolle and Sure, 2002]
	Distributed e-Learning Center [Stojanov et al., 2005]
Specialized e-Learning Services Layer	Double Agent Architecture [Rahkila, 2001]
	User-Centered and Adaptive Interaction Multi-Agent Architecture [Fernández-Caballero et al., 2003]
	Faded Information Field Architecture [Sadiig, 2005]
	Agent-Based Personalized Distance Learning System [Koyama et al., 2001]

Table 3.1: Existing Approaches in Agent-Supported e-Learning

4 Quality-Driven Content Provision – QuaD^2, A Holistic Framework

"But those who forget history are doomed to repeat it."
`James T. Kirk, The Flaming Arrow, p.42`

The importance of automatic provision of high-quality content in every field of application is beyond controversy these days. Unfortunately, existing solutions are mainly focusing on the automatism aspect. But for the success in the long run, the quality must be of substantial interest – it is an inherent characteristic of a product [Garvin, 1984]. Existing quality-related information can be reused to optimize this aggregation[1] of content to thereby always provide the best possible combination [Kunz et al., 2008b].

4.1 The Focus on Quality

The better is the enemy of the good. Why should somebody be satisfied with something, if he has the need and resources to achieve a better result? The answer is: he should not. And this is entirely about quality. A product's perceivable quality is a key factor for the long term success of a company [Buzzell and Gale, 1987]. For this work, quality is defined according to the definition of the ISO 9000 standard [ISO/IEC, 2004b]:

Definition 22

Quality *is the "degree to which a set of inherent characteristics fulfills requirements" [ISO/IEC, 2004b].*

A quality attribute is such a characteristic. To achieve quality in the field of software engineering, measurement is the fundamental basis. You cannot improve what you cannot measure. With software measurement it becomes possible to understand and communicate, to specify and achieve objectives, to identify and resolve problems as well as to decide and improve [Ebert and Dumke, 2007].

Definition 23

Software measurement *is the approach to control and manage the software process and to track and improve its performance [Ebert and Dumke, 2007].*

[1]In the following, the word 'assembly' is used for the common approach and the word 'aggregation' for the focus on content.

Figure 4.1 comprises general software measurement phases and methods.

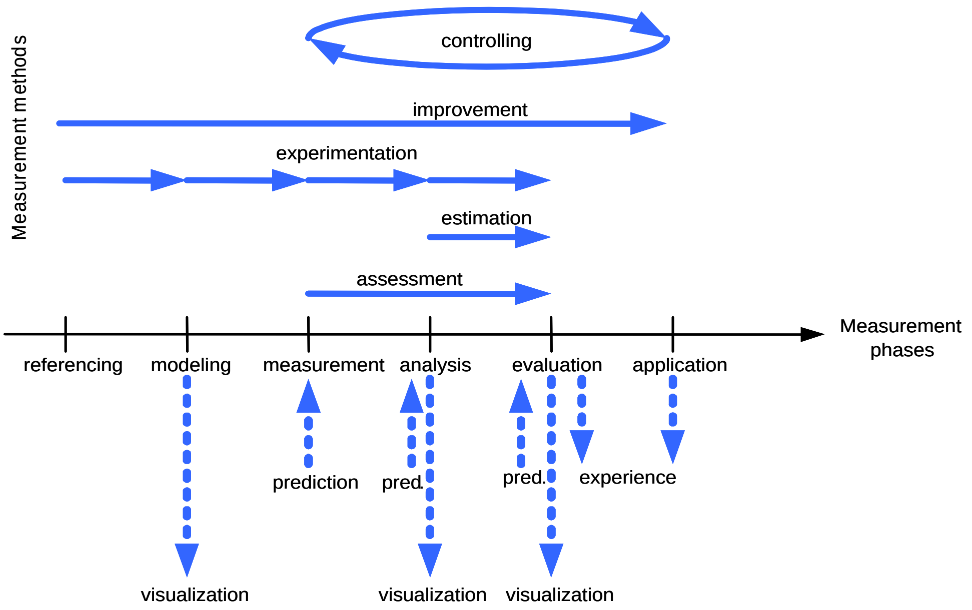


Figure 4.1: Software Measurement Phases and Methods [Ebert and Dumke, 2007]

Measuring certain attributes is only the first step. The interpretation of the results is important, too. It is necessary because the human mankind is rarely capable to directly comprehend the meaningful information of the real world (see Figure 4.2).

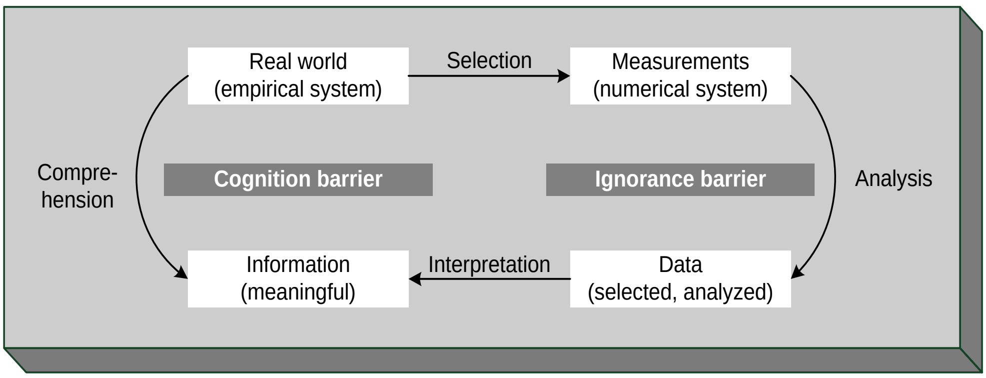


Figure 4.2: Measurement Helps to Comprehend the Real World [Ebert and Dumke, 2007]

Certain international activities were and are performed in order to standardize the expertise in this field. That ensures the usage and improvement of the current state of art on a global scale. The most important standards for software measurement are:

- How to do things (described in life cycle processes):
  - ISO/IEC 12207: Software Life Cycle Processes [ISO/IEC, 1995]
  - ISO/IEC 15288: System Life Cycle Processes [ISO/IEC, 2008]
  - SWEBOK: Software Engineering Body of Knowledge [Abran et al., 2001]
  - PMBOK: Project Management Body of Knowledge [Project Management Institute, 2004]
- How to do better (described in management systems and process improvement frameworks):
  - CMMI: Capability Maturity Model Integration [Ahern et al., 2008]
  - ISO 15504: Software Process Capability Determination [ISO/IEC, 2004a]
  - ISO 9001: Quality Management System [ISO/IEC, 2000]
  - ISO/IEC 9126: Software Product Quality [ISO/IEC, 2001]
- How to measure both:
  - ISO/IEC 15939:2002: Software Measurement Process [ISO/IEC, 2002]

Figure 4.3 shows the major software engineering standards and their relations. ISO standards are marked in grey.

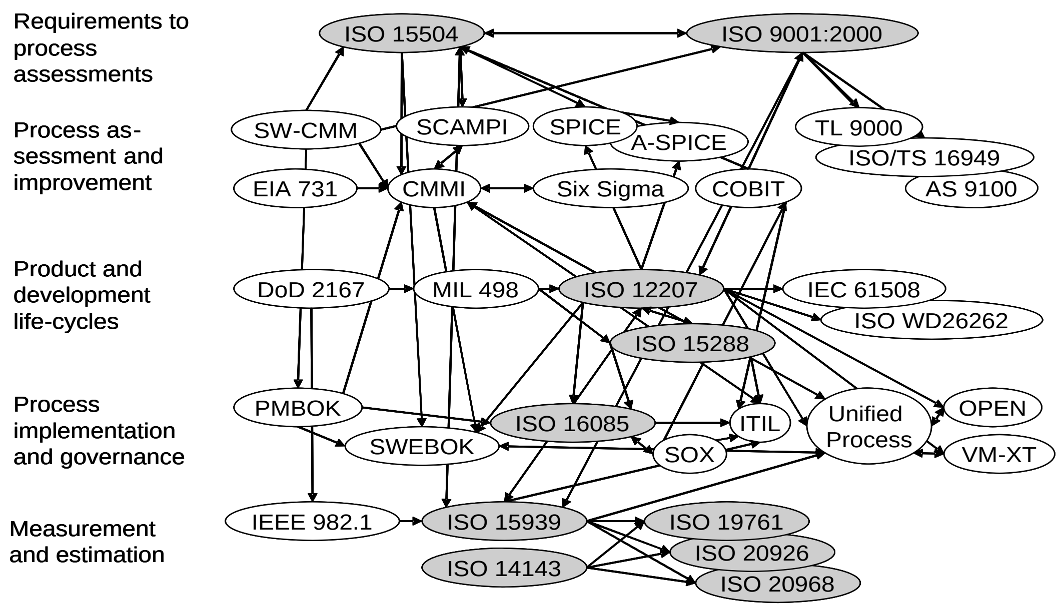


Figure 4.3: The Standards Quagmire: Standards Increasingly Line Up and Cross-Fertilize [Ebert and Dumke, 2007]

The most important quality-related standard for this work is ISO/IEC 15939:2002: Software Measurement Process [ISO/IEC, 2002]. It is about the improvement of measurement itself. The standard is depicted in Figure 4.4.

The *Establish and Sustain Measurement Commitment* is about the acceptance of the measurement requirements. Therefore, the scope of measurement needs to be identified. An agreement for this procedure must be achieved between management and staff. Next to this, the needed resources must be assigned.

The *Planning of the Measurement Process* is the next step. It is about the identification of relevant organizational units and the important information needs. Afterwards,

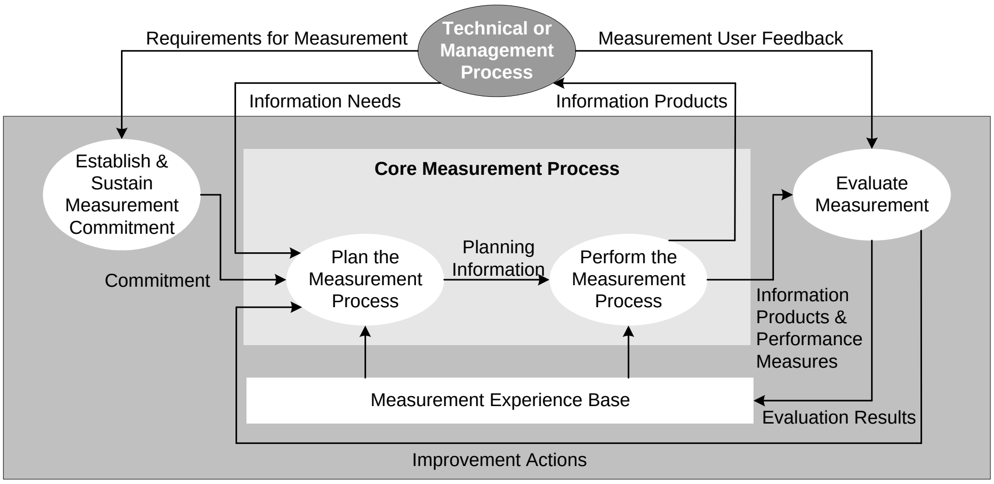


Figure 4.4: The ISO/IEC 15939 Measurement Standard [ISO/IEC, 2002]

the planning of the measurement procedure itself starts with the selection of measurement procedures as well as data collection, analysis and reporting procedures. In order to be able to appropriately evaluate acquired data, the criteria for the information products and the measurement process must be identified [Dumke et al., 2006a]. Acquiring supporting technologies and additional resources as well as the planning of the evaluation review process are other tasks of this step.

For *Performing the Measurement Process*, the substep of measurement procedure integration into the relevant processes is the first one. The collection of data, their analysis and the communication of the results follow ([Braungarten et al., 2005], [Wille et al., 2006]).

After having performed the measurement, the *Result Evaluation* follows. It targets information products as well as the measurement process. Starting points for further improvement should be identified as well ([Farooq et al., 2006c], [Dumke et al., 2007a]).

The ISO/IEC 15939 standard is widely accepted ([Dumke et al., 2005a]) and its focus on measurement makes it an ideal basis for quality-driven development. It is a cyclic process with the main steps of measurement agreement, measurement preparation, measurement performance and measurement evaluation. Evaluation results are the input for the next – now improved – cycle.

Such continuous improvement is also the goal for proactive, semantic entity provision as described in this work. Therefore, the quality-driven QuaD^2-Framework (**Qua**lity **D**riven **D**esign) is developed to reach this goal. It is shortly sketched and introduced below (see Figure 4.5). More detailed descriptions are presented in the subsequent sections as well as in [Kunz et al., 2008b], [Mencke et al., 2008e] and [Kunz et al., 2008c].

The presented QuaD^2-Framework reveals the same inner structure as the ISO/IEC 15939 standard. Only the Establish and Sustain Measurement Commitment is not explicitly modeled, because the framework's usage already implies this substep.

The entity provision initialization focuses on the functional preparation. Based on expert knowledge an appropriate process model is selected. It describes the functional flow of the proactive, semantic entity provision. A first agreement, about the quality that

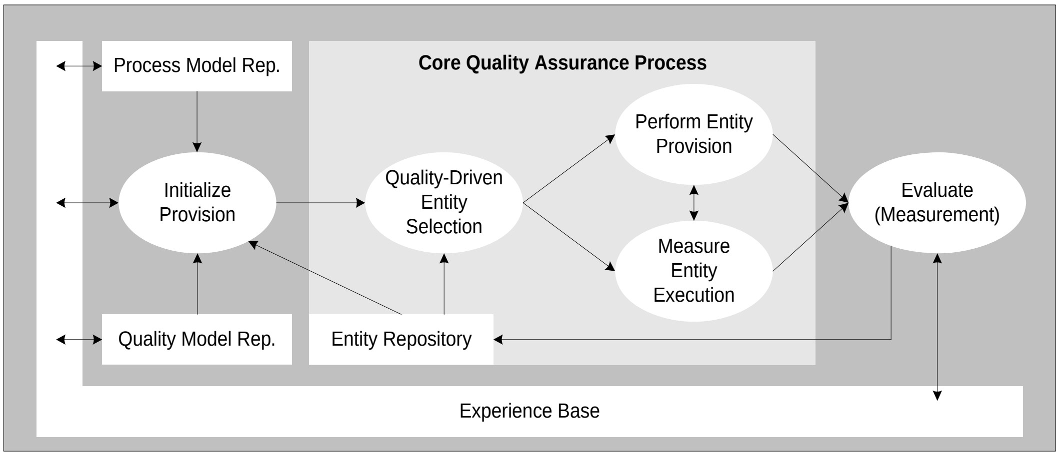


Figure 4.5: Quality-Driven Entity Provision

should be achieved, is made by the selection an appropriate quality model (set of quality attributes). Both, functional and quality-related information are used to determine the best entity.

In the standard, the measurement subprocess follows. QuaD^2 also performs its execution in the next step. Measurement is performed in parallel in order to allow evaluations. In both frameworks, evaluation is the last step. It is the basis for continuous improvements. The measurement and evaluation are not focus of this work. Related information can be found for e.g. service-oriented infrastructures ([Schmietendorf and Dumke, 2004], [Kunz et al., 2006b], [Farooq et al., 2006a], [Farooq et al., 2006b], [Rud et al., 2006a], [Rud et al., 2006b], [Rud et al., 2007c], [Rud et al., 2007d], [Zenker et al., 2008]), agent-oriented infrastructures ([Dumke et al., 2000], [Dumke et al., 2005b], [Wille and Dumke, 2005], [Wille et al., 2002], [Wille, 2005], [Kernchen, 2007]) or more general in ([Dumke, 2003], [Dumke, 2005], [Dumke et al., 2003], [Dumke et al., 2006b], [Dumke et al., 2006c], [Dumke et al., 2006d], [Dumke et al., 2007a], [Dumke et al., 2007c], [Wille and Dumke, 2007], [Kunz et al., 2005], [Braungarten et al., 2006a], [Ebert and Dumke, 2007]) in the cited references.

4.2 QuaD^2-Framework

As already described in Section 2.4, there exist approaches for entity assembly. Nevertheless, they suffer from certain problems. There exists no common model that is applicable in every situation where small parts are assembled to form a whole. Existing approaches focus on special domains like e.g. SOA or e-Learning. Another point of critique is the only sporadically emerging, throughout focus on quality. Existing knowledge is often not reused in contrast to information and data. Rarely expert knowledge is used to describe the assembly of entities ([Meder, 2006], [Pawlowski, 2005], [Helic, 2005], [Mencke and Dumke, 2007d]). Sometimes individual quality requirements are taken into consideration (e.g. for services : [Zeng et al., 2003], [Lin et al., 2008]). Their focus is on QoS only. No product-related quality attributes are continuously used.

A framework for the quality-driven assembly of entities, taking into the derived need for better solutions, is proposed in this chapter. Besides this quality-oriented characteristic, the usage of semantic knowledge and structured process descriptions enable an automatic procedure. Especially the combination of both is a promising approach. It will be shown, that the introduced QuaD^2-Framework is even more valuable due to the fact that it is not restricted to content aggregation, but also usefully applicable for all domains where a whole should be qualitatively assembled out of small parts. The current chapter follows this point of view to not to restrict the scope of the framework.

The concrete occurrence of the intended system uses elements that need to work together to provide the intended system functionality. These elements are abstractly defined as entities at this point.

Due to manifold advantages of high-flexible infrastructures compared to monolithic products, a lot of initiatives propose approaches for the integration of single components (e.g. services, content). Semantic metadata provide the basis for the automation of this process [Kunz et al., 2008a]. But for those approaches, either only functional requirements or single quality attributes are taken into consideration. A throughout consideration of existing and updated empirical data promises better solutions [Kunz et al., 2008b].

In contrast to existing approaches, the presented framework reveals a holistic orientation on quality aspects. It combines Semantic Web technologies for the fast and correct assembly of system elements and quality attribute evaluations for making the best assembly decisions possible. Therefore, complex quality models are considered as well as empirical evaluations. Both contain existing information that can be used to improve the assembly process. Furthermore, different types of quality evaluation like simulation as well as static and dynamic software measurement provide additional data. Combining them delivers a holistic quality view on entities and the flexibility enables a quality improvement of the targeted system by the exchange of single components, if the evaluation of their quality attributes decreases and fails quality requirements.

The presented general QuaD^2-Framework can easily be adapted to a lot of different fields of application, e.g.: e-Learning content provision, service-oriented architectures and enterprise application integration.

In general, the subprocesses of this empirical-based assembly process are the initialization, the feasibility check (checking the functional coverage), the selection process based on empiricism as well as the operation of the established application. Quality assurance is achieved by certain subprocesses that allow optimizations at initialization time as well as during runtime. Furthermore, measurement subprocesses are performed to update evaluation data – to get further information that can be reused to optimize the next application assembly.

The major goal of the described core process is an architecture consisting of single entities. Such an entity is metadata-annotated functionality and may be depicted by for example services, agents or content fragments in concrete applications.

In order to achieve the sketched goals, a special process is developed below. Its major use cases are introduced in Figure 4.6.

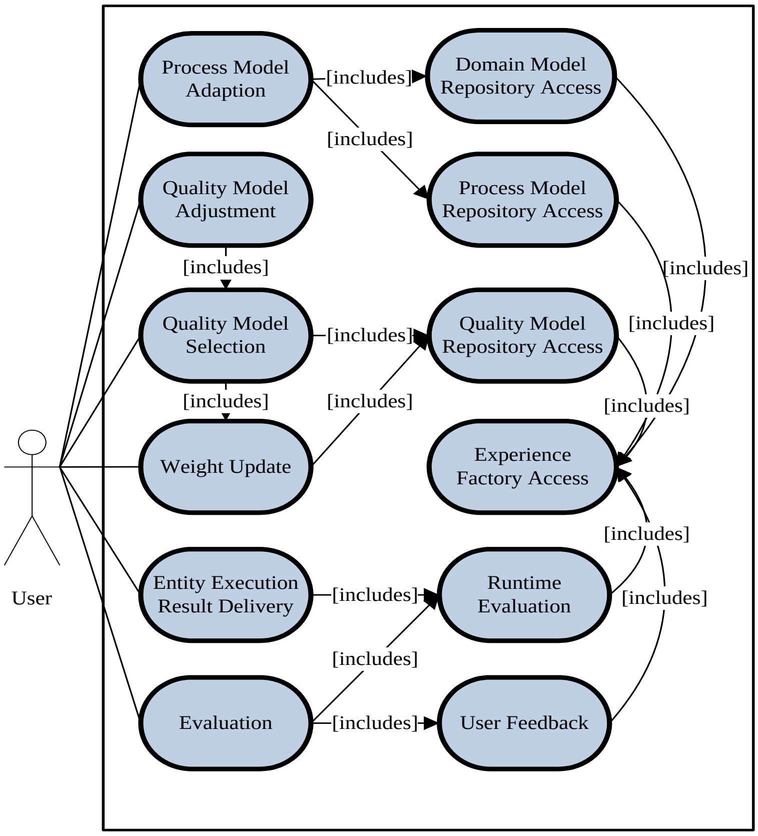


Figure 4.6: Use Case Diagram: Empirical-Based Entity Assembly Process

The basis of the presented framework is a collection of semantically-annotated sources: the process model repository, the entity repository, a quality model repository and furthermore an experience factory.

The process model repository is the source for process models that serve as descriptions for the functionality of the aspired distributed system. They depict a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a goal. Each process is enacted by a single organization, but it may interact with processes performed by other organizations [Weske, 2007]. Examples for such processes can be ISO/IEC 15939 [ISO/IEC, 2002] for the software measurement process or didactical approaches [Mencke and Dumke, 2008]. Technological realization may vary, too. That can result in UML [Object Management Group, 2004], BPMN [Object Management Group, 2005], ontologies [Lin et al., 2005], [Mencke and Dumke, 2007d], etc.

An important fundament for the intended reuse of quality-related information are quality models being provided by a quality model repository. The basis of a quality model's definition is an extensible list of quality attributes. The specification of a certain quality model is realized by selecting and weighting appropriate attributes. The evaluation and selection of appropriate entities bases on evaluation criteria for each included attribute. Such attributes can be e.g. cost, performance, availability, security and usability. The attributes and corresponding evaluation formulas are standardized e.g. in ISO/IEC 9126 [ISO/IEC, 2001]. Quality models are not a new concept. They have been extensively described in literature, e.g. [McCall et al., 1977], [Boehm, 1978], [Hyatt and Rosenberg, 1996] or [ISO/IEC, 2001].

The entity repository contains entities, their semantic description and up-to-date empirical quality evaluations regarding all defined quality attributes.

The selection and adoption of process models and quality models are difficult tasks which constitutes the need for guidance and support. Because of this, the presented framework proposes the usage of existing experiences and knowledge about previously defined and used process models and quality models to support both process steps. Based on the Quality Improvement Paradigm, Basili and Rombach proposed the usage of an Experience Factory which contains among others an Experience Base and Lessons Learned [Basili et al., 1994], [Basili, 1999].

In the presented framework, the Experience Factory is fed from the process evaluation process and is the major building block to save empirical data and the user's experiences with specific process procedures or with distinct quality attributes.

Figure 4.8 shows the entire developed QuaD^2-Framework. The used diagram elements are defined in Figure 4.7 – optional elements have a gray border.

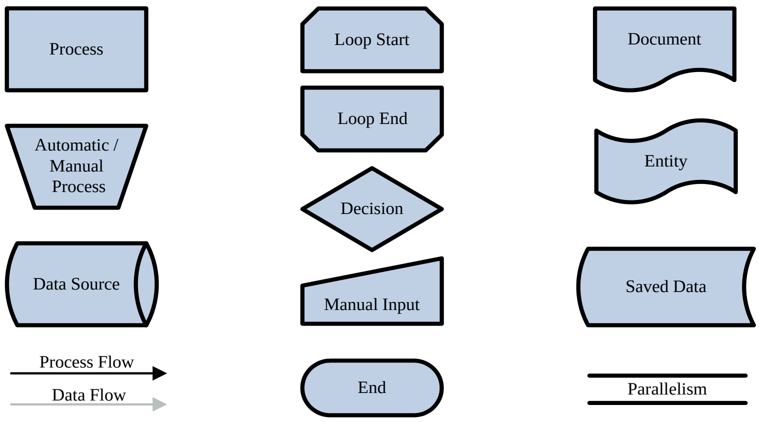


Figure 4.7: Definition of Used Diagram Elements

The focus on quality is a throughout property of the developed process and results in certain measurement and evaluation subprocesses that are introduced in the following general process description and described more detailed in subsequent sections. The derived results are directly used for optimization purposes.

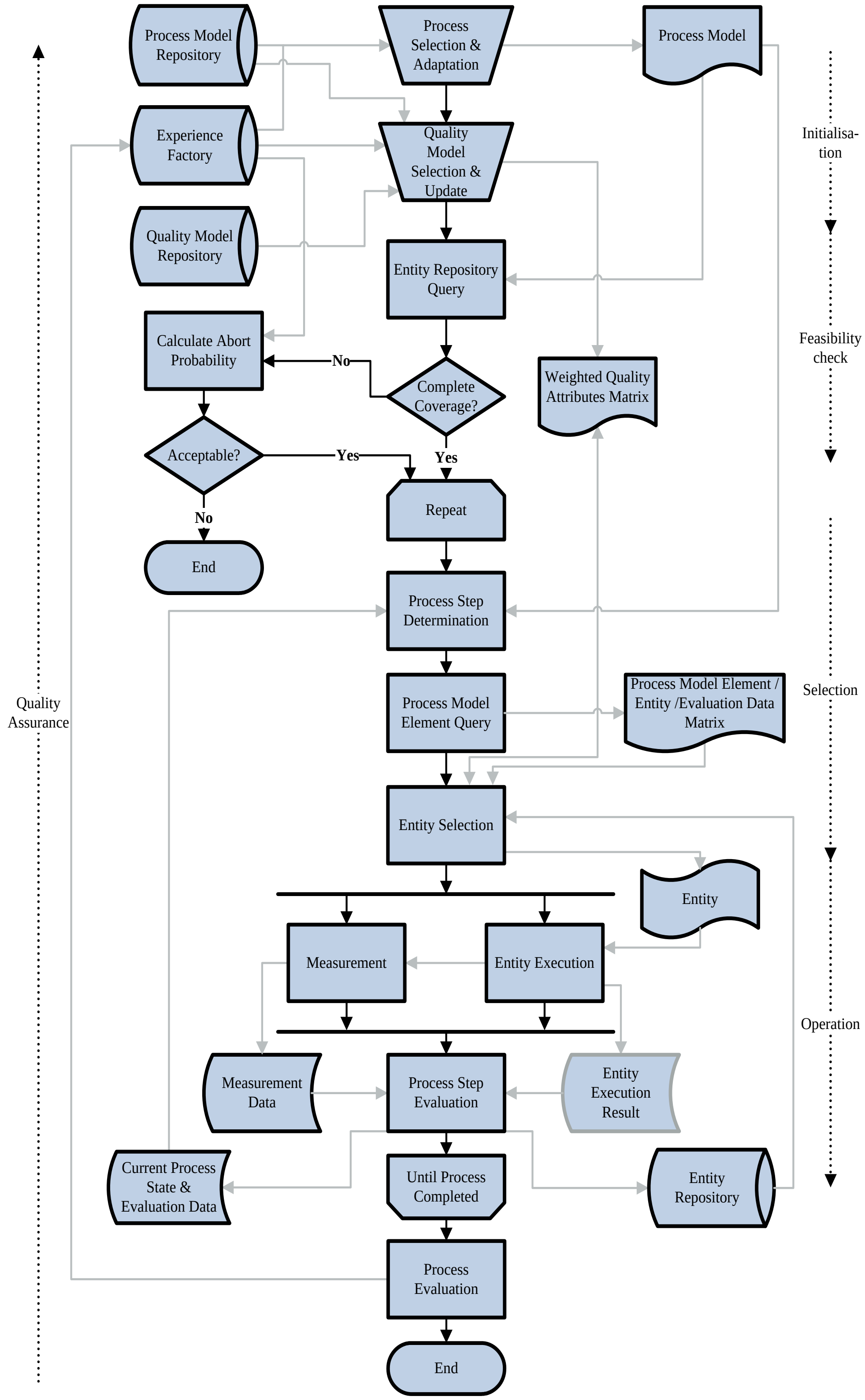


Figure 4.8: QuaD^2-Framework [Kunz et al., 2008b]

4.2.1 Initialization Phase

The selection of an appropriate process model that defines the functional requirements for the parts of the later distributed system is the first step of this phase. Due to the fact, that such a choice can be a manual process, it should be supported by an experience factory providing knowledge and experiences – lesson learned – for the decision for or against a specific process model for the current need. Information about previous assembly process' success factors are too important than to not use them for the next assembly process. The concept of an experience factory provides an appropriate basis for this purpose. The process model should essentially base on semantic metadata to allow the later automatic mapping of semantically-described entity functionalities to the functional requirements as specified by the process model. With the chosen process model, a set of concrete distributed systems is possible.

After the experience-supported selection of an appropriate process model, the next step of the presented approach is a selection of a quality model from a quality model repository. This is intended to be done automatically. For certain domains manual adaptations can be more efficient. A manual individualization of this predefined set of quality attributes as well as of their importance weighting is also possible. For these purposes, an experience factory can be helpful again. For practical aspects it is necessary to be able to retrieve current evaluation values from the entity repository, because they are needed to define appropriate quality thresholds.

As a result of this phase, a process model and importance-ranked quality attributes are defined. Those process models may vary in their basic structure according to the special, application-dependent requirements. Amongst others, the following types can be identified.

Sequential: Sequential process models are used for the modeling of sequential assembly and execution processes. Conditions are used to define functional decisions and to thereby create the adapted target system: maybe an adapted infrastructure, an e-Learning course or a measurement infrastructure.

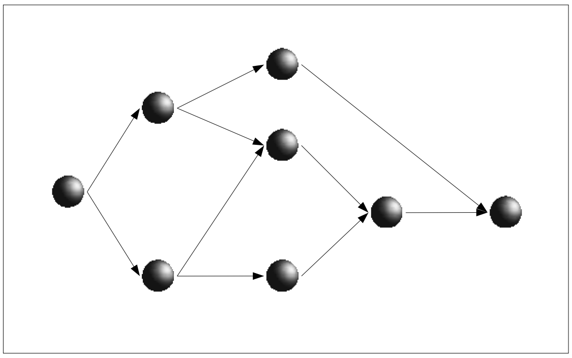


Figure 4.9: Sequential Process Models

Sequential with separated supervision: Sequential process models with separated supervision are used for the modeling of sequential assembly and execution processes, too. They additionally include downstream supervision process steps.

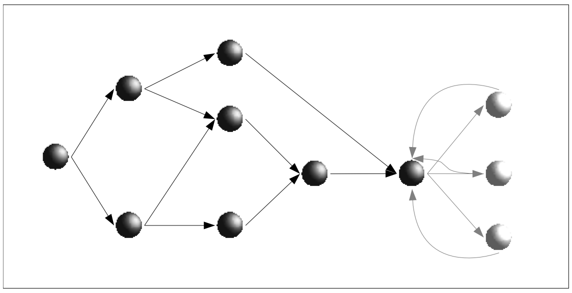


Figure 4.10: Sequential Process Models with Separated Supervision

Sequential with integrated supervision: Sequential process models with integrated supervision are similar to the one described above. In contrast, the supervision points back to the creation process steps.

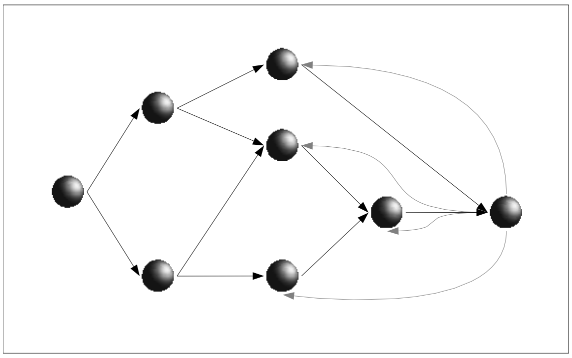


Figure 4.11: Sequential Process Models with Integrated Supervision

Supervision: Supervision process models only target the supervision of an existing system. Several conditions point away from a central event handling process step.

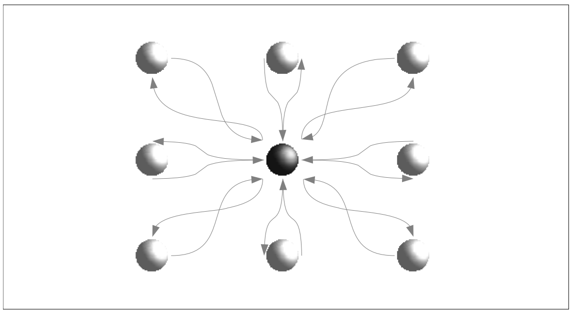


Figure 4.12: Supervision Process Models

Externally influenced: All types of process models being described above can be externally influenced by events outside the currently defined model. Thereby, meta-dependencies can be modeled.

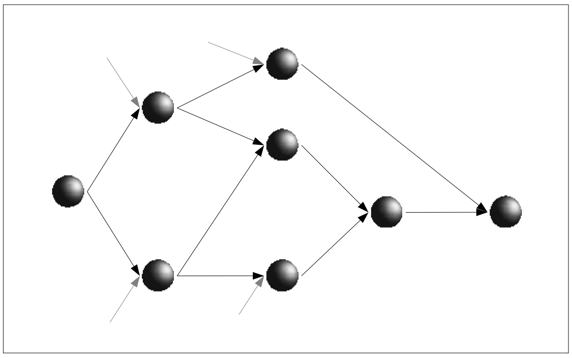


Figure 4.13: Externally Influenced Process Models

4.2.2 Feasibility Check Phase

With this information, it is possible to determine in the next process phase, if there exist enough available entities to provide an acceptable amount of functionality demanded by the process model. If there is no acceptable coverage after the negotiation subprocesses, then an abort probability based on already collected data can be computed. The user needs to decide whether he accepts the probability or not. If not, the distributed system provision process will be aborted. An automatic approach aborts the process, if the probability falls below a certain threshold.

In the case of an acceptable coverage, the runtime subprocesses of step 4 can start. The first of them determines the next process step to be executed following the process model. Therefore, information about the last process steps can be taken into consideration to optimize the next process step execution. Exception handling in case of aborted pre-subprocesses is a functional requirement and thereby should be covered by the process model itself.

Due to the fact that new entities can be added to the entity repository, another coverage check for the next process step is performed next. Now, up-to-date entity information, their evaluation values as well as the data of the quality model are available to identify the best entity possible.

4.2.3 Selection Phase

The weighting of the quality attributes during the initialization delivers weighted attributes. This procedure is not intended to be performed during runtime, because the executed distributed system should not be interrupted (abort costs, ...). The concrete selection process is described in detail in Section 4.3.

The result is a best possible distributed system – based on the existing entities as well as the specified quality model.

4.2.4 Operation and Evaluation Phase

Once the most optimal entity is identified, it can be executed. For the provision of multimedial content usable adaptation techniques were described in Section 2.5.1. In parallel to execution, measurement can be performed. These data are used to evaluate the last process step. The runtime subprocesses are repeated until: either all process steps of the process model are successfully executed or an abort due to missing entities took place. The last phase of the presented approach covers the evaluation of the entire process – serving as an input for the experience factory. It compares the achieved results with the desired ones.

4.2.5 Structure of Involved Data

For the already introduced and later detailed processes certain data are needed or created, respectively. The following Table 4.1 presents an overview about the structure of core data and possible implementation technologies.

NAME	TYPE	POSSIBLE TECHNOLOGY	ELEMENTS	DESCRIPTION
Process Model	Document	BPMN, OWL, RDF, XML	Action, state, description	Semantic description of a process
Process Model Repository	Set of documents	DBMS, XML, file system, Ontology Management System	Process models	Collection of process models
Entity	Source code, document	WSDL, C#, XML, ASCII, JPEG	Functional core, semantic metadata	E.g.: Web service, learning object, executable service
Entity evaluation data	Document	XML, ASCII	Entity ID, attribute, evaluation data	Stores an entity's evaluation data regarding quality attributes
Entity Repository	Set of entities	DBMS, XML, file system, Ontology Management System	Entity, evaluation data	Collection of entities and their evaluation values regarding all quality attributes
Interface	Document	IDL, SOAP, XML	Input, output, control options, version	Description of possible input, output and control options

NAME	TYPE	POSSIBLE TECHNOLOGY	ELEMENTS	DESCRIPTION
Interface Repository	Set of documents	DBMS, XML, file system, Ontology Management System	Interface	Collection of interface descriptions
Process Model Element / Entity / Evaluation Data Matrix	Relation	%	Process-model-element, entity, evaluation data	Collection of entities and their evaluation data which can provide the functionality which is needed for a process model element
Weighted Quality Attributes Matrix	Relation	%	Weights, quality attributes	Collection of weights of selected quality attributes
Measurement Data	Relation	%	Quality attribute, measured values, entity	Runtime values for every quality attribute and each entity
Current process state	Document	ASCII, SOAP	Name of current process model, list of completed process model elements, evaluation of completed process model elements	Collection of processed process steps and their report. For example: process step has to be repeated because entity was not available
Request	Document	%	Entity, interface, quality attribute, request messages	%
Entity / Interface Matrix	Relation	%	Entity, interface	Collection of interfaces that an entity provides
Wrapper	Source code	C#, Java, SOAP, XSLT	Input interfaces, functionality to wrap, output interfaces	Wraps one or more interfaces to others

NAME	TYPE	POSSIBLE TECHNOLOGY	ELEMENTS	DESCRIPTION
Experience Factory	%	%	%	Information about the usage and application of processes as well as quality models
Quality attribute	Document	%	Name, evaluation formula, range, direction of quality, normalisation formula, measurement frequency, textual description	%
Quality attributes list	Document	%	Set of quality attributes	%
Quality model	Document	XML, OWL, character separated strings	Set of quality attributes, weights	Weighted quality attributes matrix
Quality Model Repository	Set of documents	DBMS, OMS, XML	Set of quality models	%

Table 4.1: Structure of Involved Data

4.3 Quality-Based Entity Selection Core Process

In general, the entity selection has several steps. The first (Entity Repository Query) identifies possible entities according to the functionality defined within the process model. An additional step selects the identified quality model (Quality Model Selection and Update) that specifies what quality aspects are useful for the intended usage and how important they are for the initiator of the application to be assembled. Manual adjustments are possible, but not necessary and are performed during initialization, too. In exceptional cases a manual adjustment during runtime is reasonable.

Step three (Entity Selection in Selection phase) is the most important one and identifies the most appropriate entity for the next process step to be performed. It takes into account the weighted quality attributes as well the candidate entity set whose elements fit the functional requirements of the current process step. Figure 4.14 shows a diagram presenting the underlying process flow of this special Entity Selection Process.

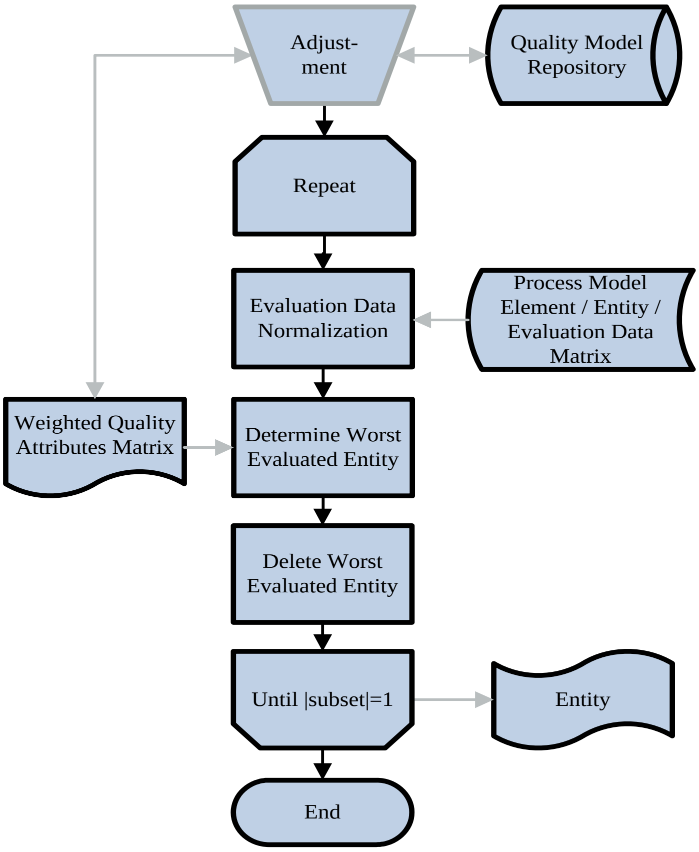


Figure 4.14: Entity Selection Process

For the Entity Selection Process, several approaches were analyzed. In general, they all focus on ranking several entities following several quality attributes to determine the one entity that fits best. The basic set of entities is determined by the selection of subprocesses focusing on the required functionality (defined by the process model).

A first decision to be made was: either to perform a pure ranking of entities based on an importance-ordered list of quality attributes or to decide based on relations between the quality attributes. The second approach was chosen because a more detailed specification of importance is possible. That means that the quality attributes (QA_i) defined in the quality model are weighted in the way that the sum of all weights is 1. The higher the defined weight is, the more important the related quality attribute is (example: three quality attributes weights $w(QA_1) = 0.7$, $w(QA_2) = 0.2$ and $w(QA_3) = 0.1$).

The weighted quality requirements matrix is manually created by selection needed quality attributes from a predefined set during initialization. Amongst others, the calculation formula and normalization directive are stored for all quality attributes to be able to determine the qualitatively best entity for the current need.

A next decision targeted the determination the correct interval for the normalization of the evaluation values. A normalization is necessary because the data have different ranges.

A first approach is to normalize the entity evaluation data over the whole set of entities within the entity repository. The observed problem is, that entities having never been used and maybe having nothing to do with the current required functionality get an influence to the selection process. Furthermore outliers get an extreme influence to the weighting procedure, because they cause major changes in the normalization procedure.

The next and better step is to normalize over the identified entity subset and to identify the best entity. Here, another classic ranking problem occurs. The addition of an entity with the same functionality but worse evaluation data changes the normalized values of better fitting entities. The ranking of better entities may change although their evaluation data remain the same (check the example in Table 4.2). This problem arises due to a needed re-normalization, which becomes necessary because the worse quality attribute evaluation value is beyond the range determined by the other entities.

	QA_1	QA_2	QA_3	
ENTITY 1	10	20	20,000	
ENTITY 2	5	80	5,000	
ENTITY 3	100	4	4,000	
Entity 4	*50*	*200*	*10,000*	
Ranking of 3 entities with normalization:				
	QA_1	QA_2	QA_3	RANKING
ENTITY 1	0.66	0.16	0.00	0.82
ENTITY 2	0.70	0.00	0.09	0.79
ENTITY 3	0.00	0.20	0.10	0.30
Ranking of 4 entities with re-normalization:				
	QA_1	QA_2	QA_3	RANKING
ENTITY 1	0.66	0.18	0.00	0.85
ENTITY 2	0.70	0.12	0.09	0.92
ENTITY 3	0.00	0.20	0.10	0.30
Entity 4	*0.37*	*0.00*	*0.16*	0.43

Table 4.2: Re-normalization Problem

Within the example, at first an evaluation and ranking of the first three entities is performed. Later a new entity 4 is included. The Table shows, that the re-normalization due to the bad entity 4 causes a ranking change of entities 1 and 2.

A predefinition of the possible value range is also not useful. How to define a range for costs? Is 1,000,000 the maximum cost or 100,000,000? A re-normalization may happen each time a new entity is added to the repository. Furthermore, extreme interval borders neglect the difference between the evaluation data values again.

So, ranking and normalization are necessary – because multiple criteria must be taken into account to determine the optimal entity and because the domains of the quality criteria are not comparable. An additional reason for normalized weights is the desired ensuring of stable weight influences. In the end the following algorithm is chosen as an adaptive, normalized, weighted indicator model.

Following the defined necessities and given data the entity selection is formally described below. For the following formulas let PM be the chosen process model. Function $f^{funct}(PM)$ specified in Formula 4.1 is used to determine the set of entities E from the entity repository. Each of them can deliver the functionalities specified within the chosen process model (see Formula 4.2).

$$f^{funct} : \text{ProcessModel} \mapsto \{\text{Entity}, \ldots\}. \tag{4.1}$$

$$E = f^{funct}(PM). \tag{4.2}$$

Using the classic normalization approach presented in Formula 4.3 (normalizing to the interval from 0 to 1), the evaluation values $v_{i,j}$ of quality requirements j defined in the quality model must be normalized for each entity i. These $v_{i,j}$ are the measurement/simulation values to anticipate the optimal decision for the next process step.

$$v_{i,j}^{norm} = \frac{v_{i,j} - min(v)}{max(v) - min(v)} * (max_{norm} - min_{norm}) + min_{norm}. \tag{4.3}$$

With the help of the weighted requirements matrix from the (maybe adjusted) quality model the last step – the identification of the optimal entity according to the empirical data and the quality model QM – can be performed (see Formulas 4.4 to 4.8). Formula 4.4 adjusts the normalized evaluation values to ensure proper calculation. If $v = 1$ describes the best quality level then no adjustments are necessary, otherwise a minimum extremum is desired and $1 - v$ must be calculated.

$$f^{mm}(v) = \begin{cases} v & \text{if a maximal } v \text{ is the best,} \\ 1 - v & \text{if a minimal } v \text{ is the best.} \end{cases} \tag{4.4}$$

$$f^{eval}(e_i) = \sum_{j=0}^{n-1} f^{mm}(v_{norm}^{i,j}) | e_i \in E \wedge n = |QM|. \tag{4.5}$$

$$V = \{f^{eval}(e_i) | \forall e_i \in E\}. \tag{4.6}$$

$$e^{worst} = e_{index} | index = min(\{x | v_x = min(V)\}) \wedge e_{index} \in E. \tag{4.7}$$

$$E' = E e^{worst}. \tag{4.8}$$

To determine the best evaluated entity, Formulas 4.5 to 4.8 are repeated until E' contains only one element. It provides the needed functionality and is the most appropriate one according to the specified quality model.

After the entity's selection it can be executed and measurement data about runtime behavior will be captured to get additional quality evaluations for this entity.

4.4 Entity Repository Management

As described in the general QuaD^2 Process, the presented quality-driven entity assembly framework requires an entity repository and semantic descriptions and evaluation data about all available entities. This section introduces chosen Entity Repository Management Processes.

The entity repository is responsible for storage, update and frequent evaluation of standardized as well as not standardized entities. The entities can be independent from the domains of application described within the process models of the process model repository to be collected for future process models.

A first overview is given in the following Use-Case Diagram (see Figure 4.15). Two different users are distinguished. The controller activates evaluation updates and the entity provider who can either add or update an entity. Every change of an entity forces a new evaluation regarding all defined quality attributes.

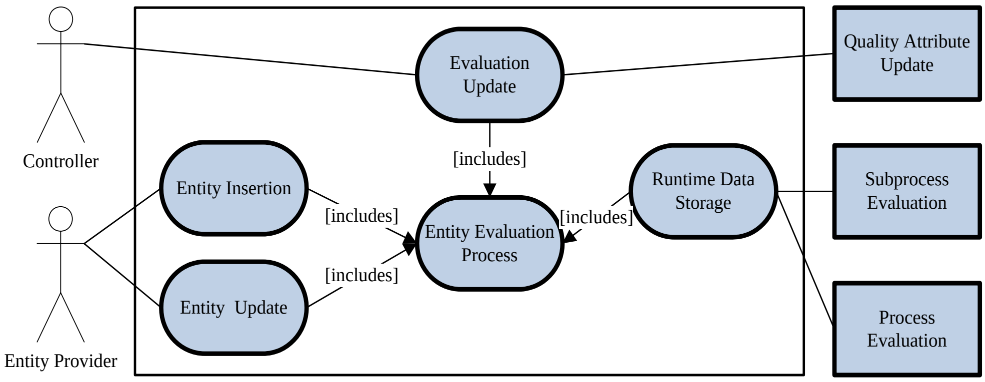


Figure 4.15: Use Case Diagram: Entity Repository Management

Due to the fact that the calculation formulas for each quality aspect are stored within the entity repository, the evaluation of entities can be executed by the proposed infrastructure without an intervention of the user. Furthermore the analysis of the measured value is performed using predefined thresholds.

By this, the holistic approach provides both: an easy to compass general view on quality (for example with a traffic light report) and a more detailed view by using the measured values over a long fraction of time.

Another key feature is the automated exchange of entities if a measured value is beyond the defined threshold and another entity fits the needed functionality as well as the quality needs.

4.4.1 Entity Insertion Subprocess

Amongst the described use cases, the entity insertion is a major one. The insertion is divided into three steps – consistency check, standardization check and evaluation. The first and second focus on functionality and the third is about quality. Functionality-related issues were already discussed in Section 2.4. The quality-driven QuaD^2-Framework adds a new dimension and shifts the focus towards the Entity Evaluation Process (see Figure 4.16).

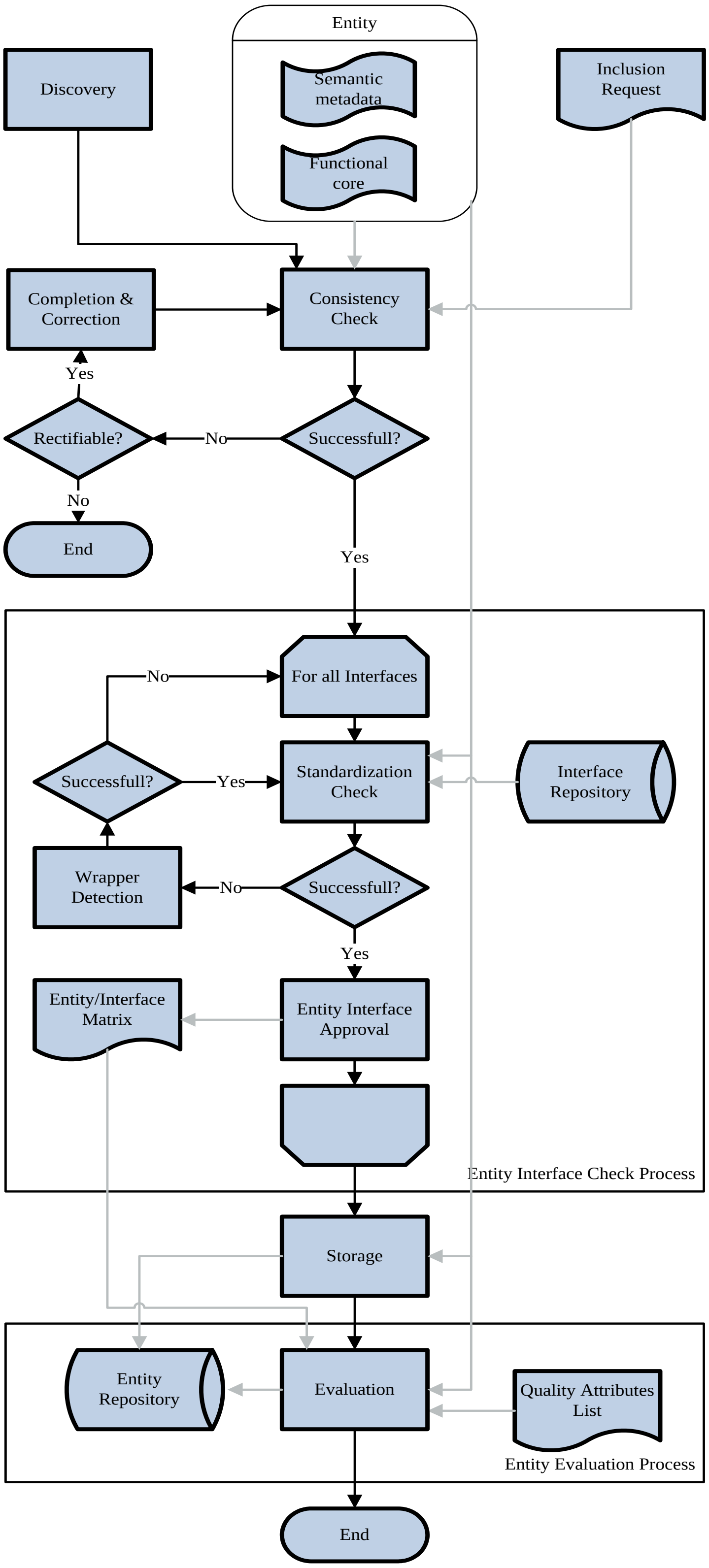


Figure 4.16: Entity Insertion Subprocess

Based on self-initiated discovery or an external inclusion request, a new or update entity with its semantic metadata and functional core is identified. A consistency check uses this information and performs a first analysis regarding standardization. If this check, eventually after certain completion and correction steps, is successful, then the Entity Interface Check Process takes place. For these purposes, separate subprocesses are defined maybe using internal or external tools. If necessary certain steps must be performed manually.

To check interface compatibility, a list of interface definitions is used to check which entity is conform to which interfaces. If it is not conform to a standard interface, it tries to identify appropriate wrappers. The goal is an as complete as possible standard coverage for a broad entity deployability. For entity execution it is unimportant whether a wrapper is used or not. Possible positive (e.g. standard conformance, performance) and negative (e.g. cost, performance, size, maintainability) quality criteria of the wrapper will become part of the entity and be evaluated by the according process. The list of interface definitions must be extensible, too. The management processes of this interface repository are similar to the ones of this entity repository.

As a last step, the entity is stored and evaluated according to the quality requirements definition and the results update the entity repository.

4.4.2 Entity Repository Query Subprocess

As already shown in the main process of the QuaD^2-Framework, the Entity Repository is queried multiple times.

Figure 4.17 shows this process. Possible concrete implementations were already discussed in Section 2.4.

A first check maps existing functionality (provided by the entities of the entities repository) with required functionality (defined within the process model steps). Based on this check, the Process Model Element / Entity / Evaluation Data - Matrix is updated. It defines which entities can cover the required functionality of which process model element. If there exists at least one entity for each them, the Feasibility Check is successful. If there were not enough entities, then the calculated abort probability must be evaluated. Only a positive feedback will result in starting the assembly and execution. A coverage verification is the result of the Entity Repository Query.

The second type of this process' execution is performed each selection phase. It updates the matrix to ensure the usage of up-to-date information about evaluation values and entity existence.

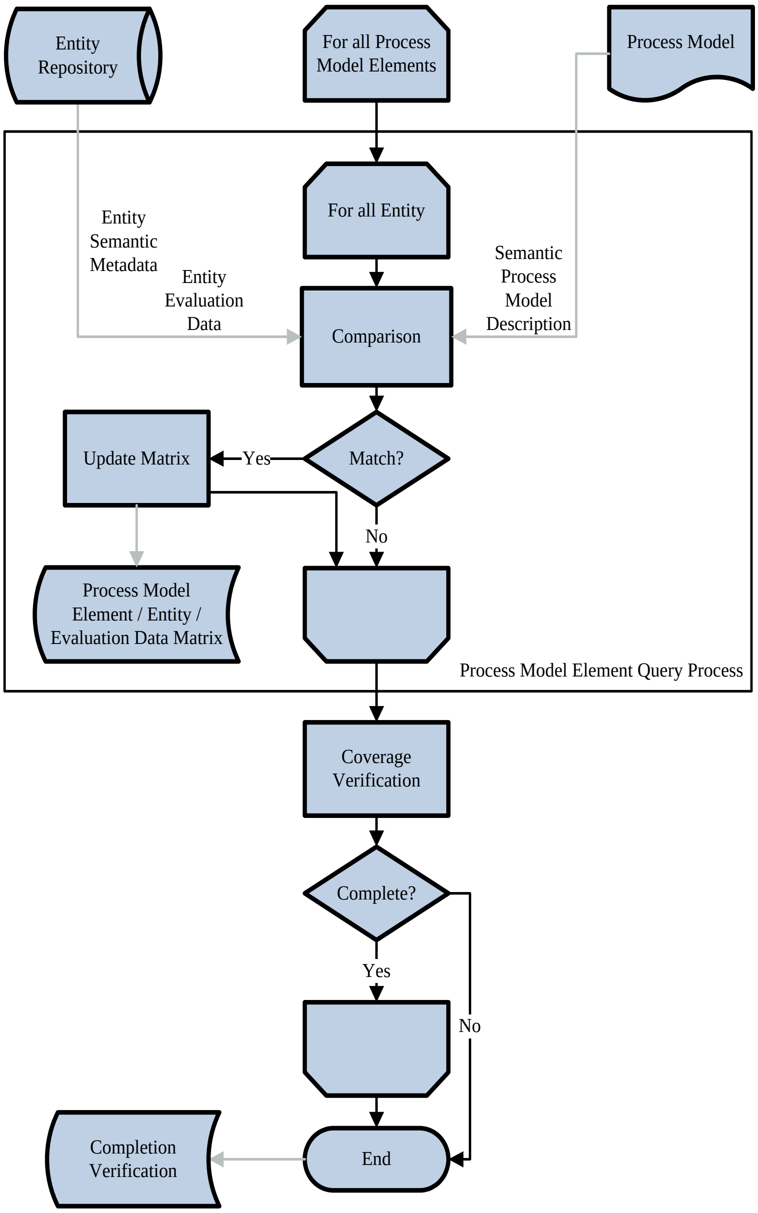


Figure 4.17: Entity Repository Query Subprocess

4.5 Quality Assurance

Following ISO/IEC 9126 [ISO/IEC, 2001], quality assurance can be distinguished into internal, external and quality in use. Latter is covered by the quality assurance during runtime. Internal as well as external quality is measured by the iterative Entity Evaluation Process as part of the Entity Repository Management Processes.

The focus on quality is a throughout characteristic of the developed process and results in certain measurement and evaluation subprocesses. The derived results are directly used for quality-driven improvement purposes, e.g.:

Evaluation points:
○ Quality attribute measurement during entity execution
○ Entity execution result (in case its functionality was to test/evaluate)
○ Process step evaluation
○ Process evaluation
○ Entity consistency checks
○ Continuous entity evaluation process
○ Meta-evaluation processes (e.g. usage of experience factory, entity update, . . .)

Quality-driven improvement approaches:
○ Use of process model repository
○ Use of quality model repository with quality models (quality attributes and weights)
○ Use of entity repository with evaluation data about the entities
○ Experience factory for process model selection
○ Experience factory for quality model selection
○ Update of quality requirements and weights in experience factory in case of aborted process steps (automatic and manual)
○ Process step determination based on success and evaluation data about the last process step (and experience factory data in case of aborted last process step)
○ Update and extension of process model repository
○ Update and extension of quality requirements
○ Update and extension of entity repository
○ Update and extension of interfaces and wrapper
○ Entity standardization with appropriate wrappers
○ Continuous entity evaluation process

Quality requirements are evaluated from the user's point of view. The provider of the distributed system defines a subset of for him acceptable requirements by the selection of available entities within the entity repository. The described process uses these available entities to create a result being as optimal as possible for the user.

4.5.1 Quality Attributes

Although most quality attributes are entity- and thereby domain-specific, there exist some common ones. In addition to cost, some selected ones are presented below.

Quality of Service
○ Availability (especial partial availability) [Rud et al., 2007d]
○ Performance ([Rud et al., 2007d], [Rud et al., 2007a], [Rud et al., 2007c])
○ Accessibility ([Rud et al., 2007d], [Malack et al., 2008])
○ Stability [Rud et al., 2007d]
○ (Data) security
○ Capacity
○ Integrity

ISO/IEC 9126 [ISO/IEC, 2001]
- Usability [Kernchen and Dumke, 2007d]
- Maintainability ([Rud et al., 2006b], [Rud et al., 2007b], [Kernchen et al., 2006])
- Functionality
- Reliability
- Efficiency
- Portability
- Changeability ([Farooq, 2005], [Farooq et al., 2005], [Farooq et al., 2006a])
- Interoperability

For every domain, specific quality attributes are useful and provide the opportunity for a more precise entity selection. Following [Devedžić, 2006] and [Vouk et al., 1999], e-Learning quality criteria can be differentiated into quality of educational content, quality of pedagogy and quality of the technical framework. High-quality educational content can be expressed by the quality of lessons, appropriateness of the teaching/learning paradigm, quality of user-system interactions and semantic interoperability. High-quality pedagogy can be achieved by adaptation to the needs of the learner (group), to the learning goals and to the preferred learning styles. Thereby, learning efficiency should be increased. The quality of the technical platform is important, too. Learner support and appropriate performance of the e-Learning system are pre-requirements for its acceptance and usability. The quality of the content can also be determined on an implementation dimension. Therefore, metrics for the multimediality of content exists [Wille and Dumke, 2007].

In [Hametner et al., 2006] some quality criteria based on a slightly different classification are listed. An advantage is their presentation of some initial thresholds to determine good quality. Due to the fact, that e-Learning systems provide multimedial content, related metrics can used to determine the current entity's quality.

- Metadata
  - System Requirements
  - User Guide
  - Didactical Guide
- Functionality
  - Print option
  - Bookmarks and list with links
  - Search option
  - Notices
  - Glossary and help
  - Download area
- Navigation
  - Sitemap
  - Navigation buttons
  - History lists
  - Abort buttons
  - Self-explaining navigation elements, e.g. self-explaining symbols, expectancy conformance, rollover effects, highlight effects, ...

Text-based recommendations are for example: font (no serifs), font sizes (min. 12 pt), highlighting (better bold than italic), line spacing (for displays 1.5 to 2), line length (eight to ten words = 60 to 80 characters) or paragraph alignment (better left-aligned).

For each of these requirements, the evaluation procedure as well as a measurement frequency must be available within its description. The entity's runtime evaluation data regarding quality requirements are determined using those evaluation procedures. Options for the measurement frequency are: only at entity functional core/version change (e.g. inheritance graph, maintainability), after every entity execution (e.g. performance of a database) or always and permanent (e.g. performance, availability). All information together forms the quality model that is stored within the quality model repository.

4.5.2 Quality Determination

The Entity Evaluation Process uses the defined formulas for each quality attribute being stored in the Quality Attributes List to calculate the evaluation values for every entity. Not for every attribute a mathematical formula is available, but at the attribute's definition time an evaluation procedure must be specified to allow quality assessment. Such evaluation procedures can be e.g. experiments, user surveys or certain simulations.

Ensuring quality is fundamentally based on measurement. Figure 4.18 classifies formal measurement approaches.

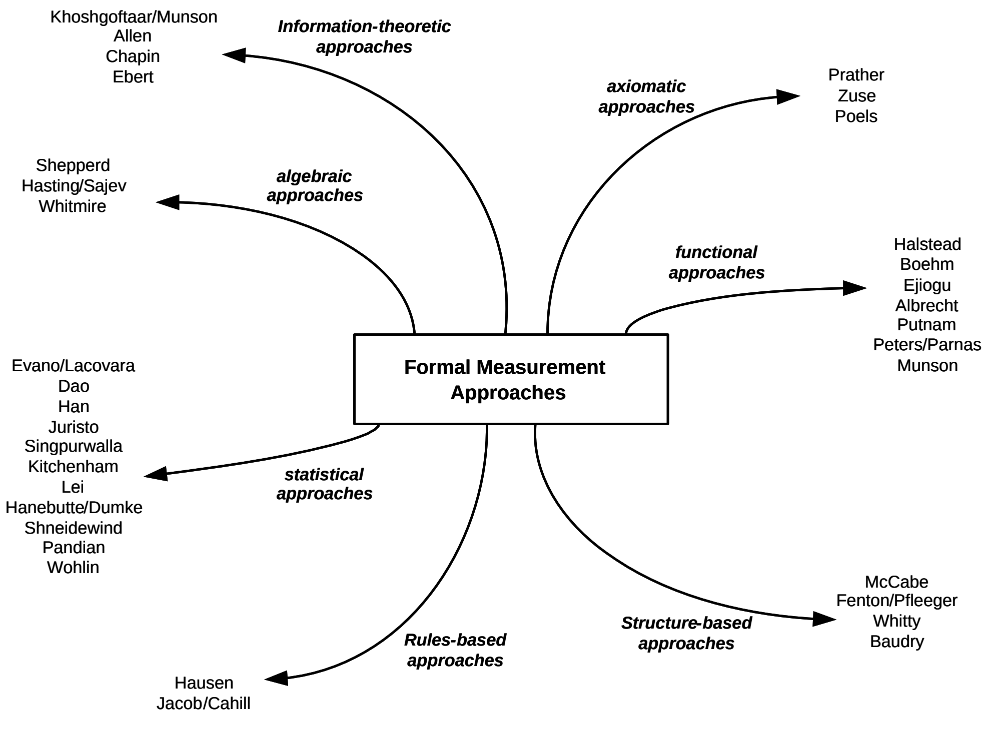


Figure 4.18: Classification of Formal Measurement Approaches [Ebert and Dumke, 2007]

For the proactive determination of quality aspects [Wille, 2005] is recommended for further reading. There, fundamental information and approaches about measuring of and measuring with agent technology are presented.

4.5.3 Process Step Evaluation

The Process Step Evaluation includes the measurement and empirical analysis of the process step's execution. Derived information is added on-the-fly to the entity repository to be available for the next process step according to the process model.

The Process Evaluation Step is something special. It is not performed after each process step but only once after the whole process has finished. Derived information is stored within the experience factory to be available for other process model choices. According information can be e.g.: process quality information (cost, duration, success rate, . . .), results of a user interview about the complete process, the teaching success of an e-Learning course or the measurement success of measurement processes. In general, this subprocess is performed as shown in Figure 4.19.

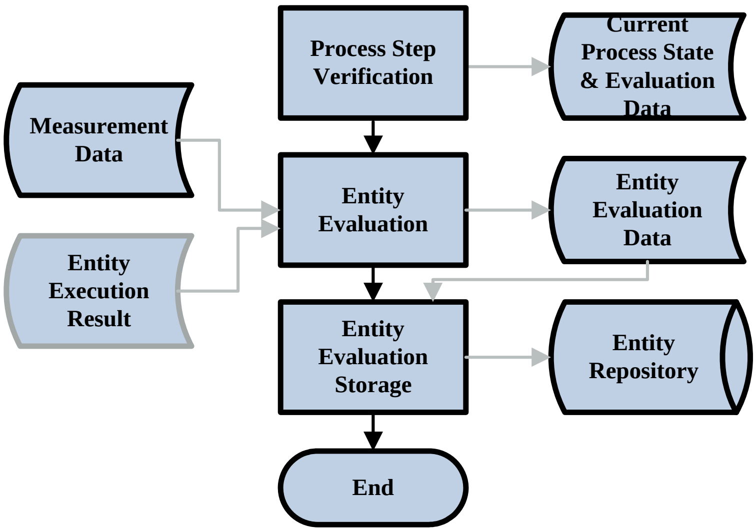


Figure 4.19: Process Step Evaluation Subprocess.

Process step evaluation is a major building block for an automatic quality measurement and evaluation. The collected measures about the runtime behavior of entities bears the capability of reducing manual evaluation processes especially where automated metrics-based evaluation before runtime is difficult or not possible and user opinions at runtime makes more sense.

For the presentation of entities to human users (as e.g. in e-Learning) a special aspect occurs. The evaluation of entities – especially Learning Objects – is also influenced by user characteristics. For example, his motivation, capabilities or his available time frame possibly change entity evaluation results. In [Jovanović et al., 2007] the authors describe a list of feedback levels that can be used as a starting point for functionality-based as well as quality-based adaptations:

- Better recognition of problems at a coarse-grained level: based on a quiz to identify motivation, cognitive disability or working habit problems
- Better recognition of differences between successful and unsuccessful learning trajectories: quiz- and performance-based analysis and identification of behavior patterns

- ○ Detection of content that was too hard for students to comprehend: calculating time and number of times the students spent on a lesson, to determine cognitive overload
- ○ Identification of student difficulties at a topic level: quizzes, semantic annotations and student interaction observation to identify difficult domain concepts
- ○ Identification of frequently discussed topics: analysis of exchanged messages
- ○ Identification of student's level of engagement in online interactions: analyze student behavior and activity, e.g. in exchanges messages, read/started discussions, ...

This work does not focus on these aspects, but redirects to further related literature as for example [Goldberg, 1996], [Merceron and Yacef, 2005], [Zinn and Scheuer, 2006], [Feng and Heffernan, 2007] or [Scheuer and Zinn, 2007].

4.5.4 Evaluation in Entity Evaluation Process

The results of the Entity Evaluation Process are used to create the optimal possible entity assembly. Because initial evaluations can change over time, updates are necessary. For this purpose, a runtime measurement can be performed parallel to entity execution. All runtime data prove, refine or disprove this simulated information.

Event and time triggered entity evaluations provide additional empirical data. This continuous entity evaluation ensures up-to-date information sources for the throughout quality assurance during the QuaD^2-Processes and enables high quality software products.

4.6 Discussion

The QuaD^2-Framework can be implemented using various technologies as for example ontologies, Web Services and agents. The presented quality-driven approach proposes the usage of semantic descriptions for process automation and supports different quality models and quality attribute evaluations. The easy extensibility of process models, entities, interfaces and quality models makes the presented framework deployable for many fields of application. Next to these parts, permanent measurement and evaluation are important information sources that are updated and reused for the substantial support of a throughout quality-oriented assembly of entities.

Both, the Entity Management Process and the Runtime Evaluation of the general QuaD^2-Process, are major building blocks for an automatic quality measurement and evaluation. The collected measures about the runtime behavior of entities bears the capability of reducing manual evaluation processes especially where automated metrics-based evaluation before runtime is difficult or not possible and user opinions at runtime make more sense.

The second major outcome of the QuaD^2-Framework regarding automated quality evaluation is the collection of empirical data in different model components (Entity Repository, Experience Factory) like the knowledge about processes, runtime behavior and measured quality evaluations. This meaningful data are used to automatically select entities, adjust processes and substitute elements.

This general framework is very abstract due to its intention to describe a usable approach for many domains. In the subsequent chapters, selected parts are analyzed further for proactive ontology-based content provision within the domain of e-Learning.

5 Appropriate Process Models for Content Provision

"Even the smallest [...] can change the course of the future."
Galadriel, The Lord of the Rings: The Fellowship of the Ring

This chapter mainly targets the initialization phase of the QuaD²-Framework. Therefore, the primary focus is on ontology-based process models for content provision as well as their storage within appropriate repositories. Furthermore, a special source of additional knowledge is presented that may influence process model selection as well as quality model selection and adaptation.

As a third key area, the proactive provision of entities is described in this chapter. As a part of the Entity Repository Management Processes it has an initiatory character, too.

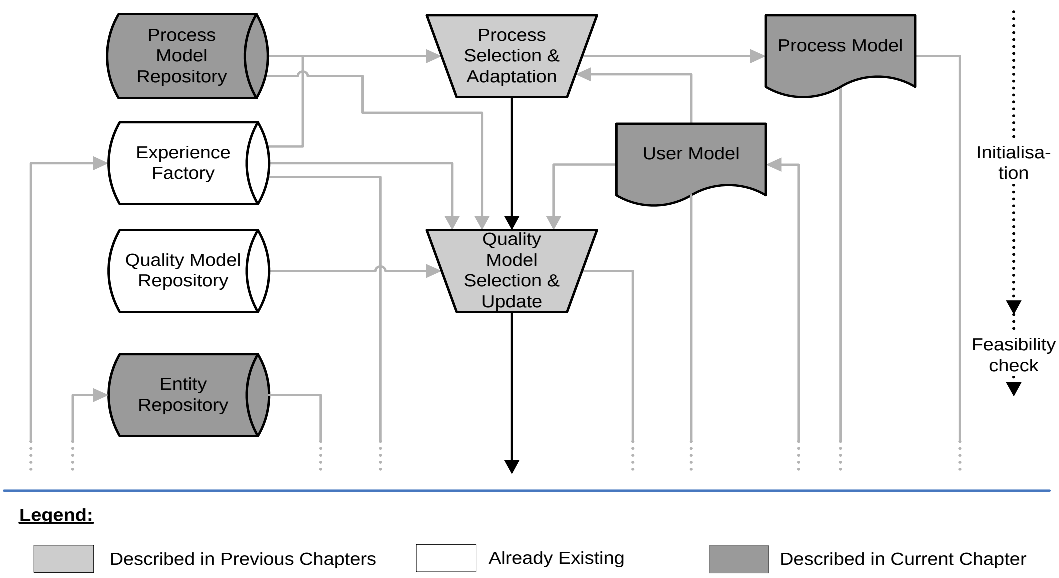


Figure 5.1: Research Questions of the QuaD²-Framework's Initialization Phase

Meanwhile the Process Selection & Adaptation as well Quality Model Selection & Adaptation were described in the previous chapter, the concepts of an Experience Factory and of Process Models are not new, but also introduced in Chapter 4. Additional information about functionality-related selection were given in Section 2.4.

For the following chapters, e-Learning is chosen as the major use case. The requirements and challenges regarding an appropriate and well-supported aggregation of content are especially shaped within this domain. Nevertheless, all presented approaches and solutions are transferable to other domains of interest.

5.1 A Hierarchy of Ontologies for Content Aggregation

This section analyzes possible process description approaches and proposes an appropriate technology to implement process models as the basis for content aggregation. Furthermore, an abstraction hierarchy of process models is introduced, which can be used at different points of content creation and aggregation.

5.1.1 E-Learning Processes

The term "process" has manifold significances within each domain. Formally it is defined as a set of activities associated with a set of events, where an event is an internal or external signal, message, variable, scheduling, conditional change, or timing that is specified in association with specific activities in a process [Wang and King, 2000].

An e-Learning process thereby is a special process, whose domain is e-Learning and the process transitions involve e-Learning-related activities to change certain states within this domain [Mencke et al., 2008k].

There exist many possibilities for the implementation of process models as required for the QuaD^2-Framework.

5.1.1.1 Chosen e-Learning Processes

E-Learning itself is a process containing two major phases: content development (additionally including planning, design and evaluation) and content delivery (additionally including maintenance). Its nature is iterative (see Figure 5.2). Evaluation is recommended for continuous improvement [Giotopoulos et al., 2005].

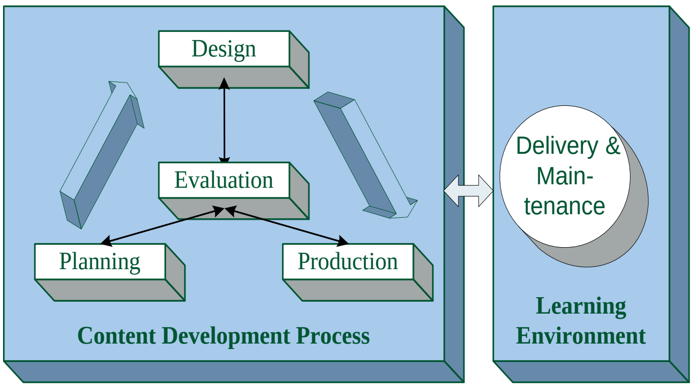


Figure 5.2: Iterative Process of e-Learning [Giotopoulos et al., 2005]

The stages of a **general e-Learning process** are planning, design, production, evaluation, delivery and maintenance, instruction stage and marketing [Khan, 2004].

There exist more specialized e-Learning processes being categorized in the following according to their domain of application. The proposed dimensions and some exemplified e-Learning processes are [Mencke et al., 2008k]:

- Technological dimension
  - E-Learning platform operation
  - Technical-enhanced dissemination process
- Organisational dimension
  - E-Learning establishment and process
  - Course and organization administration
  - Evaluation through the entire lifecycle
  - Coordination process
  - E-Learning innovation process
- Authoring dimension (refers to the classic content creation stage)
  - Learning Object and course design
  - Didactical design
- Learning and teaching dimension (refers to the classic content delivery stage)
  - Presentation process
  - Learning process

5.1.1.2 General Process Descriptions

Processes in general can be described using various approaches. The identified classes of process descriptions reveal an increasing degree of formalization – informal process descriptions, graphical process descriptions and formal process descriptions.

Some process description approaches were already used in the previous subsection about e-Learning processes, like e.g. textual descriptions. Other resources refer to activity lists, hierarchies or tables as informal approaches to depict processes with their states and activities.

More graphical approaches to model stepwise processes are diagrams and graphs as for example workflow-diagrams, UML-diagrams, PERT-nets or the representation shown in Figure 5.2 about the general e-Learning process.

A last class of approaches categorizes formal descriptions like algebraic possibilities, languages or rules. That may result in grammar-based, BPEL-based, BPMN-based or pi-calculus-based representations. Other examples base on petri nets, like the event-driven process chain or ontologies, for example describing the structure of didactical ontologies which can be used to model didactical process expertise [Mencke and Dumke, 2007d] or supporting certain aspects of Virtual Engineering [Mencke et al., 2008i].

5.1.1.3 Process Descriptions for e-Learning Processes

As already introduced, processes are determined by a sequence of states. This can be influenced on certain levels and depicts the nature of a process' activities – it reveals a system behavior. The states of the process base on certain domain objects. Within e-Learning that can be a Learning Object, a course, an e-Learning system, etc.

These dimensions – the nature of state domain objects and the activities nature – are used to categorize already identified e-Learning process classes [Mencke et al., 2008k].

ACTIVITIES' NATURE	STATE DOMAIN OBJECT'S NATURE	
	rigid	*flexible*
rigid	technological, organisational	
probabilistic		authoring, learning
situative		learning

Table 5.1: Classification of e-Learning processes according to the processes' nature [Mencke et al., 2008k].

According to the classification in Table 5.1 the rigid nature of activities as well as of state domain objects are used in closed processes. The activities and domain objects rarely change. Technological and organizational processes are categorized here. More open processes deal with flexible objects and flexible activities. Within e-Learning open processes are authoring and learning processes. Depending on the type of learning, learning processes can be completely open and therefore guided, but not specificly bounded to any predefined learning path space.

Closed e-Learning Processes Main evaluation criteria for closed e-Learning processes are stability, safety, being optimized and that they meet time constraints. Routine must be achieved. This process type is already well known and researched. There exist multiple proven approaches to meet the several levels of routine in different domains. Roadmaps defined on the basis of maturity models fit the requirements of process management. They should be conform to existing standards, measuring and measurement-based adaptation to understand, model and improve closed processes. An existing maturity model for e-Learning is eMM [Marshall, 2007]. Its levels are:

5. Optimizing: continual improvement
4. Managed: ensuring the quality of e-Learning resources and student outcomes
3. Defined: defined process for development
2. Planned: clear objectives for e-Learning
1. Initial: ad-hoc processes

Roadmaps of maturity models use the informal textual approach to define rules to help an institution to further develop their closed e-Learning processes. Graphical approaches are only used for the human reader for a better understanding. Executable formal approaches are not useful because closed e-Learning processes are not entirely automatically executable (e.g. the introduction of an e-Learning system can only be guided but not be automatically performed) [Mencke et al., 2008k].

Semi-Open e-Learning Processes Semi-open processes cannot be evaluated simply by the routine of its execution. Authoring and learning based on strict course structures reveal a rigid or maximal probabilistic activity nature. There are predefined degrees of freedom to choose different activities and to change the objects the states are based on. These objects, for example Learning Objects or learning steps, are flexible in the manner that they can and must be adapted to reflect the changes within the environment: new knowledge needs to be integrated and new teaching/learning approaches to be applied. Selected factors to be taken into consideration are:

- Relative completeness, e.g. in terms of extension, issue representation, maintenance conformity, avoidance of semantical thinning and individualization (concept overvaluation)
- Didactical preparation, e.g. in terms of comprehensibility, goal conformity, logical consistency

For semi-open e-Learning processes, applicable process descriptions exist, too. That refers for example to the PELO model for authoring [Müller et al., 2005]. The main steps are process modeling, process execution and process measurement. For the first step, the authors use a formal visualization technique, the Event-driven Process Chain that is based on Petri-net theory.

For the learning process guidance certain models exist (e.g. SCORM [Advanced Distributed Learning (ADL), 2006b]). They are not completely sufficient due to several reasons. So, they still lack from an appropriate definition of difficulty and a sufficient definition of usage rights and educational activities (because of the often used IEEE LOM [IEEE Learning Technology Standards Committee, 2003]). Furthermore, there is a subjective selection of educational material types or missing detailed specifications for some types of media [Simon, 2002]. Related approaches only describe a set of potential processes and thereby only provide fixed/limited support.

Open e-Learning Processes Open e-Learning processes are the most complex ones. There are high degrees of freedom for activities as well as for the state's objects. The nature of the objects as well as their types can extremely vary. For a learning process, there are for example different culture-related, individual disposition-related, intrinsic and extrinsic motivation or timely emotional influences. Other variables are the learning situation, the individual learning type and the learning content.

The main goals for open e-Learning processes are specificly directed to learning next to individual knowledge increase. It is not about to classify but to individually treat learners, to keep their motivation and to provide learning possibilities that can adapt to individuals and their specific situation. The learner is a partner within the process, not a target.

Some criteria for evaluation of process outcomes are:

- ◦ Content quality according to the learning goal:
  - Degree of the content's abstraction
  - Difficulty level of content
- ◦ Flexibility of the learning system according to individual learning and life situations
  - Method conformity
- ◦ Individual learning goal adaptations by the learner
  - Individual knowledge gain
  - Degree of content understanding, repetition and applicability
  - Achieving the didactical goal

Again, routine criteria and related process descriptions are not sufficient. So far, no single system provides sufficient process support that comprises all dimensions. Only an ontology-based approach can solve the occurring diversity to take account in teaching-, knowledge- and user-models ([Simon, 2002], [Mencke and Dumke, 2007d]). Semantic information is needed for the appropriate support of ad-hoc learning in its various dimensions.

5.1.2 Ontologies for Content Aggregation

As argued above, most process descriptions are not sufficient to model the complex influences that may occur within open e-Learning processes. A flexible and semantically defined approach is needed to guarantee applicability, reusability and extensibility.

Ontologies as described in [Mencke and Dumke, 2007d] and [Mencke and Dumke, 2008] are suggested to fulfill these requirements. Their usage in e-Learning can be useful for numerous goals [Mencke and Dumke, 2007d] – for example they serve as:

- ◦ **Didactical ontologies** for the categorization of learning goals,
- ◦ **Thematic ontologies** for the thematic categorization of learning material,
- ◦ **Rhetoric-semantic ontologies** for categorization of learning material for the creation of meaning contexts,
- ◦ **Relational ontologies** for the description of contextual dependencies and
- ◦ **Curricular ontologies** for the organizational categorization of learning material.

Didactics is a science targeting several directions, so it is the science of organized teaching and learning, the science of education or it is the application of psychological teaching and learning theories. Additionally, it is seen as the theory of education contents and the theory of controlling learning processes [Kron and Sofos, 2003].

In this work, the developed ontologies' tasks are manifold: providing a general scheme for process description, being a basis for automated content aggregation, describing didactical expert knowledge as well as serving as a starting point for process optimization.

The following descriptions address the organization and control of learning and teaching processes for e-Learning. A taxonomy of didactical approaches is presented in Figure 5.3.

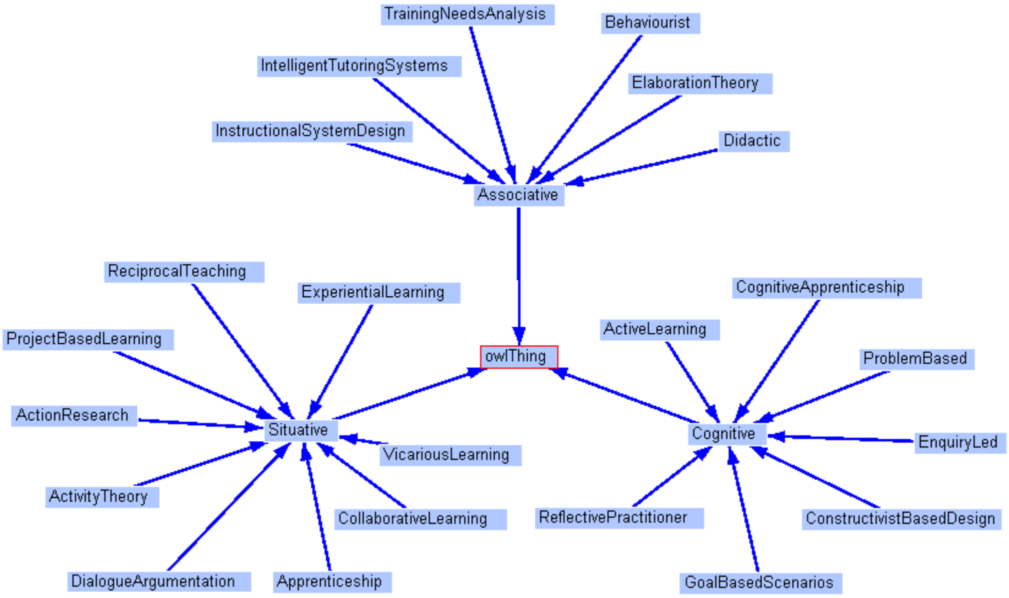


Figure 5.3: Taxonomy of Didactical Approaches [Mencke and Dumke, 2007d]

Meder [Meder, 2001] defines a didactical ontology as an approach to describe information for being able to structure cognitive learning processes. This work goes a step further and intends to use those information also for the ontology-based modeling of didactical expertise – didactical ontologies. With parts of the subsequent work, tools are introduced that mould learning environments for improved and adapted learning experiences. In the following, approaches for an ontology-based provision of didactical expertise as well as for course structure specification are introduced.

5.1.2.1 Hierarchy of Ontologies

For the hierarchy of ontologies, a 5-level structure is proposed to reach the intended advantages (see Figure 5.4 plus a level for e-Learning course description as the fifth level).

Level 0 contains the most general ontology of the proposed set. It depicts a general description of a didactic strategy. Its purpose is to define the scheme for an ontology-based realization of the order of learning content to achieve an optimal learning result as well as the description of didactical expertise. Human experiences with the learning and teaching processes can be integrated in those ontologies. These implicit quality aspects result in a substantial quality gain. Timed strategic elements need to be adaptively chosen to fit certain context, learner or teacher-defined requirements:

- ○ Abstract class for a learning step
- ○ Definition of an order of learning steps
- ○ Conditions for multiple learning paths
- ○ Metadata inclusion for runtime support

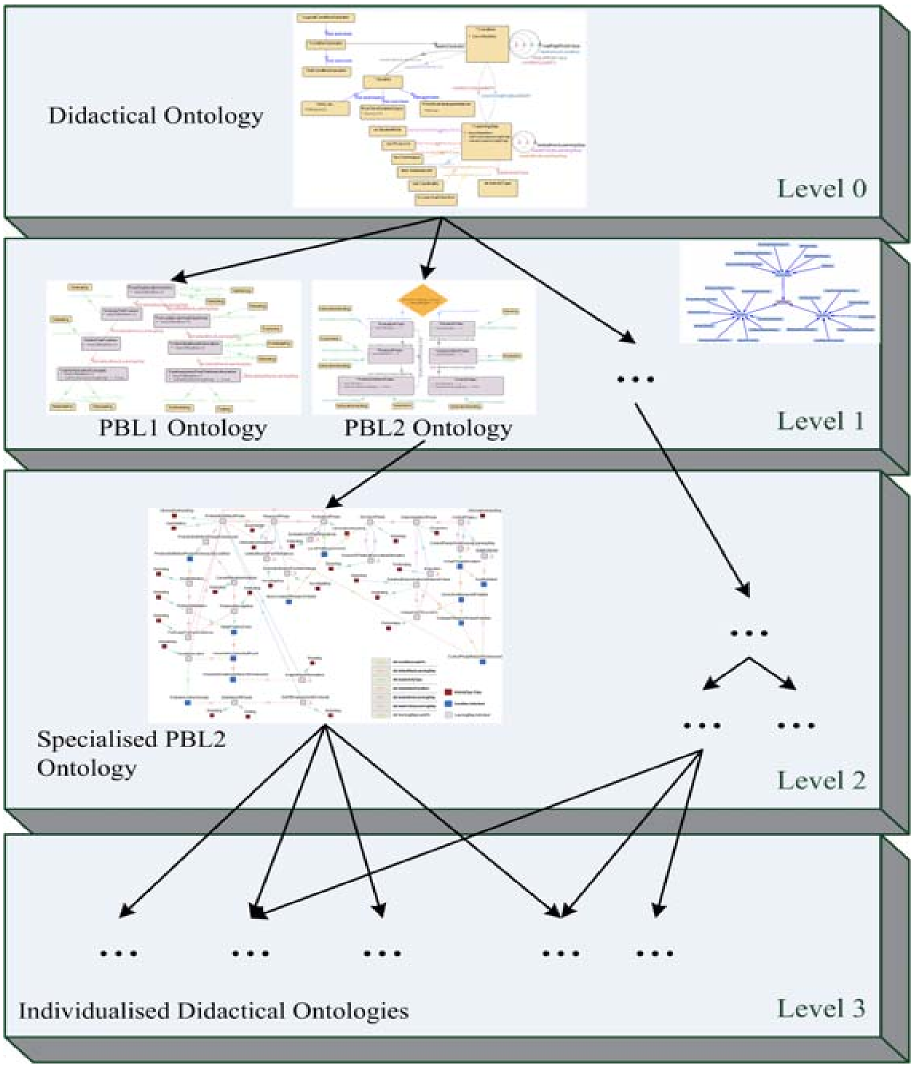


Figure 5.4: Hierarchy of Didactical Ontologies [Mencke and Dumke, 2007d]

Figure 5.5 presents the developed top-level ontology. The central concepts are the *LearningStep* and *Condition* class. A *LearningStep* is the reference to a part of a didactical approach. Further refinement is supported by the possibility to divide a learning step into several sub learning steps. Therefore, the relation *leadsToSubLearningStep* is created to point to the first *LearningStep* node that will compound the sub learning steps. The property *isFirstLearningStep* must be set true to mark this first node. According to this, the property *isLastLearningStep* must be set true for the last node. To permit a return to the main didactical flow, the sub nodes reference to their root node through the relation *hasAsRootLearningStep*. Additional relationships point to describing (sometimes taxonomic) ontologies:

- *hasActivityType* points to certain activities which the current learning step should cover.
- *hasLearningObjective* points to an ontology describing learning objectives
- *hasIntendedStudentRole* points to a description, where possible student roles a listed
- *hasIntendedResource* points (technical) resources that are intended to be used
- *hasIntendedTechnique* points special techniques/approaches for teaching
- *hasAssessment* points to suggestions for certain assessment types
- *hasIntendedCardinality* describes the type of interaction according to the number of participants

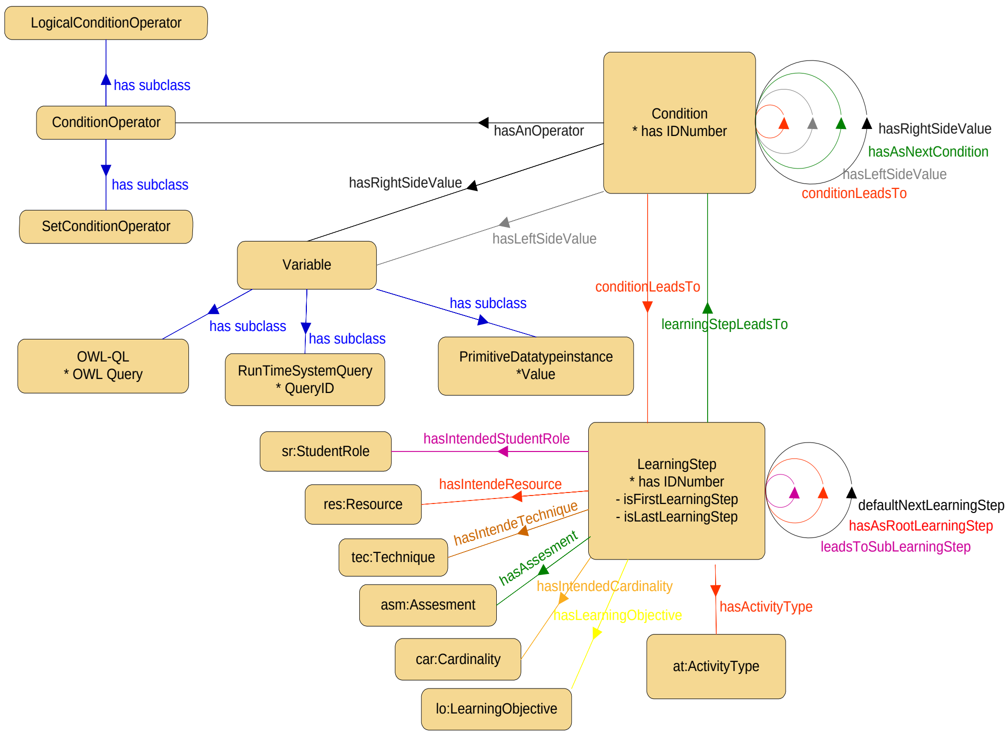


Figure 5.5: Level 0 Didactical Ontology [Mencke and Dumke, 2007d]

The condition concept is used to model restrictions to a path, permitting the runtime environment of an e-Learning system to decide the next appropriate path through the learning content for the current user in his specific context. Both main concepts are used to model a didactic in this way:

- Identify the first *LearningStep*
- Follow the learning path for the first condition that delivers a true result

Therefore, a LearningStep points to a *Condition* with a *learningStepLeadsTo* relationship. A *Condition* itself redirects the learning path to one other *LearningStep* with the *conditionLeadsTo* relationship, if its result is true. Multiple learning paths can be modeled by integrating multiple *Condition* individuals. To support those alternative ways through the e-Learning course, additional aspects are integrated into the ontology. A first one is a hierarchy of conditions. If one fails, the *conditionLeadsTo* relationship points to the next condition to be checked. Another one is the possibility to depict sequences of conditions by using the *hasAsNextCondition* relationship; the last condition of a sequence must point to a *LearningStep*. The default relationship *DefaultNextLearningStep* between two learning steps provides an alternative for the case where no condition is fulfilled and must appear only once. Figure 5.6 exemplary visualizes some aspects described above.

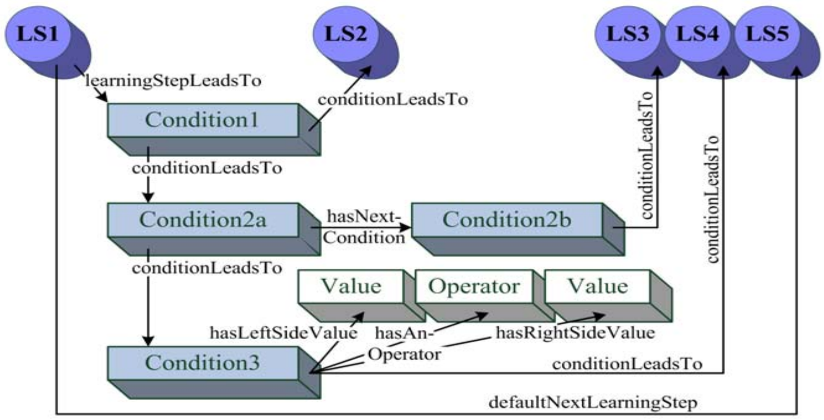


Figure 5.6: Hierarchical Conditions for Multiple Learning Paths [Mencke and Dumke, 2007d]

The conditions themselves are described by three (two, if a unary operator is used) additional relationships. The relationships *hasLeftSideValue* and *hasRightSideValue* point either to another condition or to a *Variable* that can be of type *PrimitiveDatatypeInstance*, *OWL-QL* or *RuntimeSystemQuery*. The first type has the anyType-property value and is used to model variables like the "5" within the following conditional expressions: "If (NumberOfTries) isGreaterThan 5". The *NumberOfTries*-variable is of type OWLQL and the query is stored as a string within the OWLQuery property. The *RuntimeSystemQuery* has a string-property, too. QueryID will be used by an e-Learning runtime system to locate an internal condition. That is internally analyzed and delivers back a Boolean value for the comparison. The ontology-intern condition must look like: "If (runtimeCondition1) isEqual true". The relationship *hasAnOperator* points to a *ConditionOperator* that defines the set and logical operators.

For conditions as well as for learning steps, the mandatory property *hasIDNumber* was created. These IDs are used to provide the runtime environment a way to identify the path that the user has gone through.

Level 1 may reveal an inner hierarchical structure, too. It is directed toward to description of general didactical strategies, based on the level 0 ontology. According to the taxonomy of didactical approaches that are presented in Figure 5.3, the problem-based learning (PBL) approach was chosen for further implementation. PBL is a didactic that begins with a presented problem and is followed by a student-centered inquiry process [Trevena, 2007]. Fundamental principles base on the work of Barrows and Schmidt ([Barrows, 1986] and [Schmidt, 1983]). Figure 5.7 visualizes the ontology focusing on Schmidt's seven steps in problem-based learning.

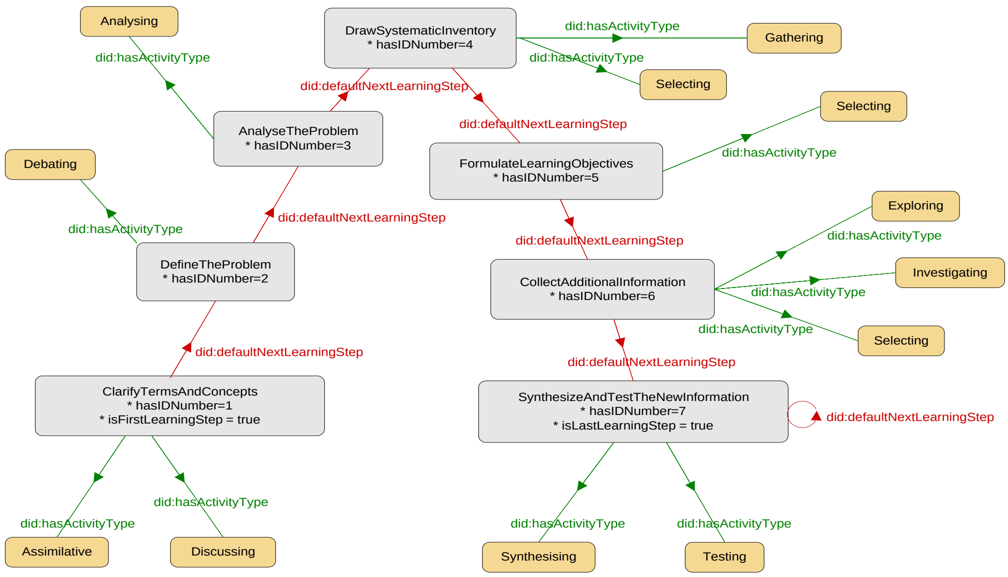


Figure 5.7: Problem-Based Learning Didactic Level 1 Ontology [Mencke and Dumke, 2007d]

This implemented PBL ontology describes the seven basic steps that a PBL didactical approach should have according to [Trevena, 2007], namely:

- Clarify terms and concepts
- Define the problem
- Analyze the problem
- Draw systematic inventory
- Formulate learning objectives
- Collect additional information
- Synthesize and test the new information

These steps are defined as individuals of a *LearningStep* and, as there is no special condition to the transition between them, only the *defaultNextLearningStep* relationship is used. The activity types for each *LearningStep* are chosen based on what should be performed by the learner.

Level 2 contains the leaf nodes of the hierarchy, each describing an applicable didactical approach. Here, for example the micro didactics of Meder [Meder, 2006] or the didactical models of [Flechsig, 1996] are integrated. Figure 5.8 defines an ontology for a special problem-based learning didactic. It is adopted from [Mertens, 2002] and bases on [Hahn, 1971].

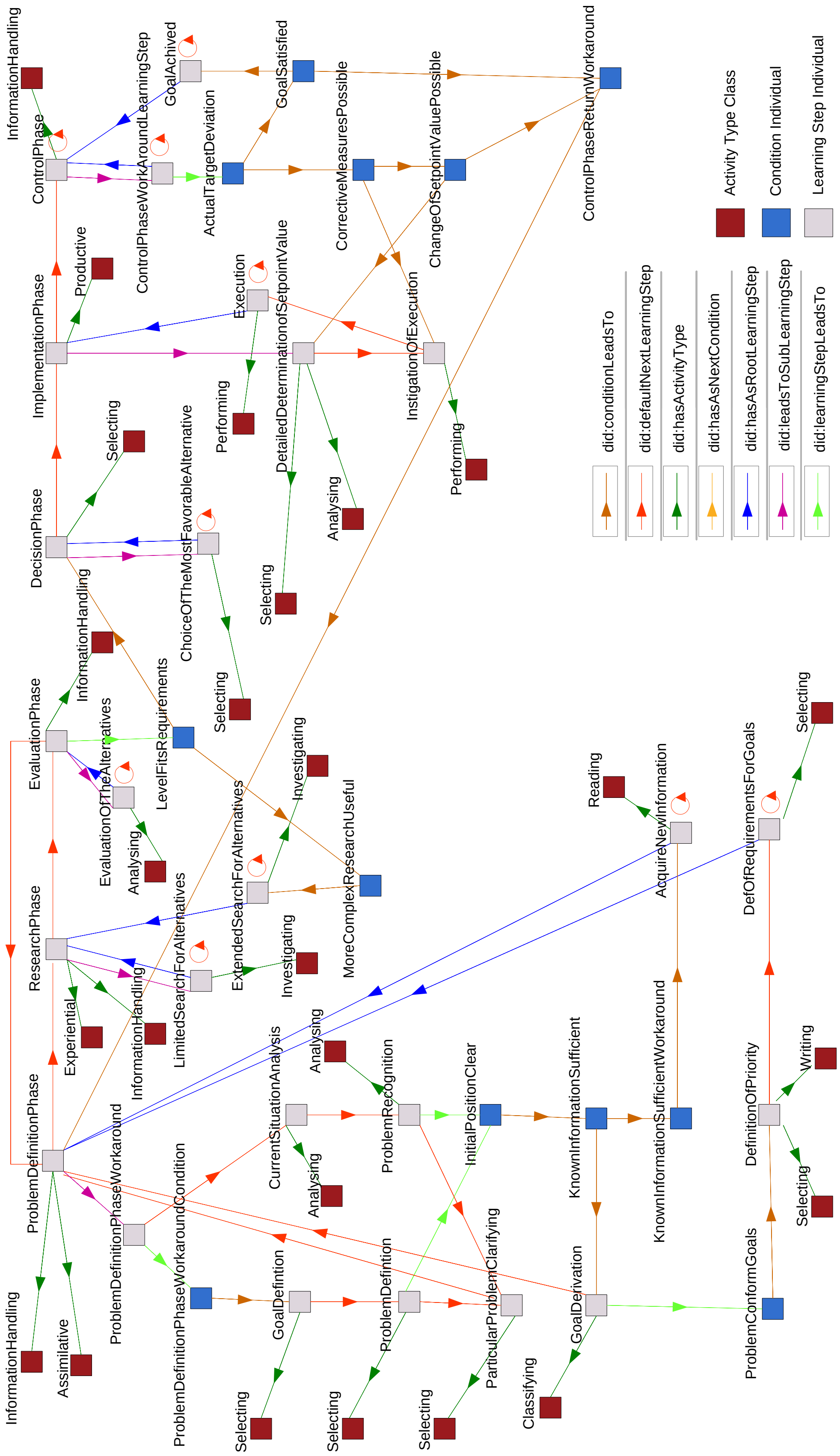


Figure 5.8: Problem-Based Learning Didactic Level 2 Ontology [Mencke and Dumke, 2007d]

The presented didactical approach consists of six main steps, namely problem definition phase, research phase, evaluation phase, decision phase, implementation phase and control phase. These main learning steps are further refined into sub learning steps and related to appropriate activity types. The needed conditions are integrated as *RuntimeSystemQuery*, because this example was not developed according to a specific existing e-Learning system.

Exemplified conditions are specified below and base on the structure *ConditionName(did:hasAnOperator; did:hasLeftSideValue; did:hasRightSideValue; did:conditionLeadsTo)*:

- *LevelFitsRequirements(did:EQUAL; RTSQ_LevelFitsRequirements; BooleanTrue; MoreComplexResearchUseful | DecisionPhase)*
- *ProblemDefinitionPhaseWorkaroundCondition(did:EQUAL; RTSQ_ProblemDefinitionPhaseWorkaroundCondition; BooleanTrue; GoalDefinition)*
- *CorrectiveMeasuresPossible(did:EQUAL; RTSQ_CorrectiveMeasuresPossible; BooleanTrue; InvestigationOfExecution | ChangeOfSetpointValuePossible)*

Level 3 is directed to the approach of individual (recombined) adapted didactics. The idea behind is, that individual approaches of specific teachers, tutors or scientists should be made available and usable, too. The trivial usage is to identify sub elements of the course that are didactically decoupled or only loosely coupled. These *(Sub-) LearningSteps* are affiliated with each other with the standard *defaultNextLearningStep* relationship or reusable relationships that for example point forward, if the current *Learning Step* was successfully completed. The more complex problem is the identification of inter-didactic relationships within certain contexts and their ontology-based modeling.

To be able to depict the specific structure of an e-Learning course, the level 0 ontology is extended by an additional concept and certain properties (see Figure 5.9).

The *LearningObject* is integrated from a developed LOM-Ontology comprising metadata instances of existing Learning Objects (see Section 6.1.5.1). This ontology forms the basis of the Entity Repository in this exemplified instantiation of the QuaD^2-Framework. Next to the *hasIDNumber* variable, storing an ID of the currently described Learning Object, this concept has two datatype properties – namely *isFirstLearningObject* and *isFirstLearningObject*. In addition to them object properties are integrated to point from a *LearningStep* to a *LearningObject* as well as from a *LearningObject* to another *LearningObject*. By this procedure, it is possible to refine a *LearningStep* as a sequence of *LearningObjects*.

In order to show the usefulness of the developed approach a prototype was developed. It bases on the 3-level ontology Framework later introduced in Section 6.1.5.1. Using the ontology shown in Figure 5.9, an ontology for the description of Learning Object metadata and a domain ontology, courses can be individually assembled. The provision is parametrized by didactical expertise defined within the didactical ontology.

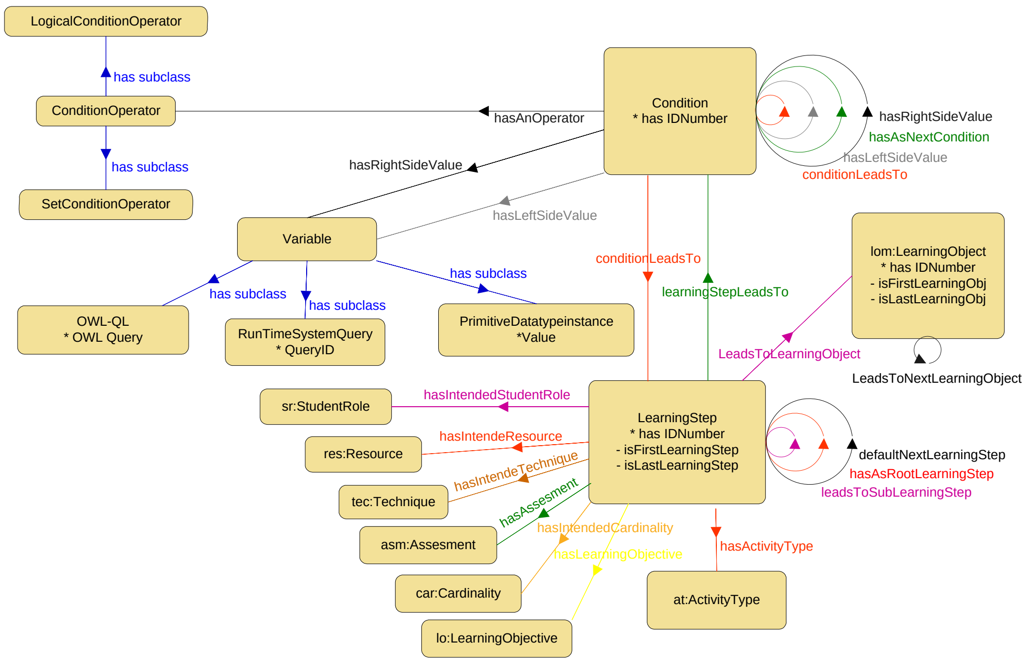


Figure 5.9: Level 4 Didactical Ontology (extended from [Mencke and Dumke, 2007d])

Figure 5.10 shows an initial login screen, where chosen parameters can be defined. Based on this and other data, a series of course fragments including assessments is presented to the learner, adapted to her/his user model.

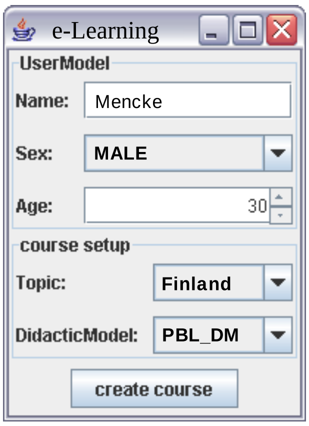


Figure 5.10: Login Screen for Individual, Semantics-Based e-Learning Courses

Figure 5.11 shows two excerpts of an individual semantics-based e-Learning course about Finland.

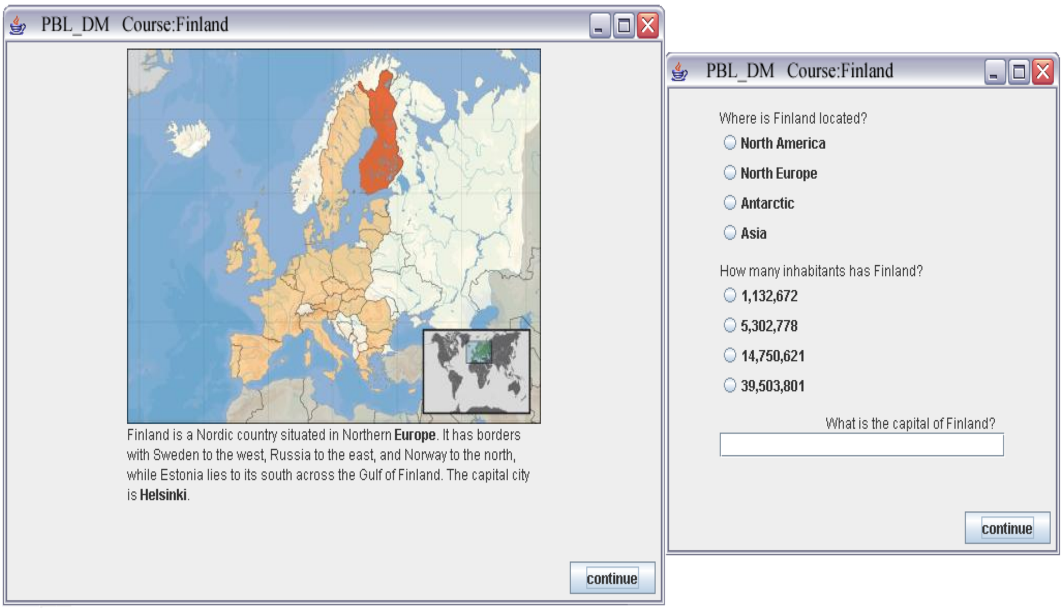


Figure 5.11: Chosen Course Fragments of an Individual, Semantics-Based e-Learning Course

Figure 5.12 visualizes the usage of a prototypical tool for the creation of a level 4 didactical ontology.

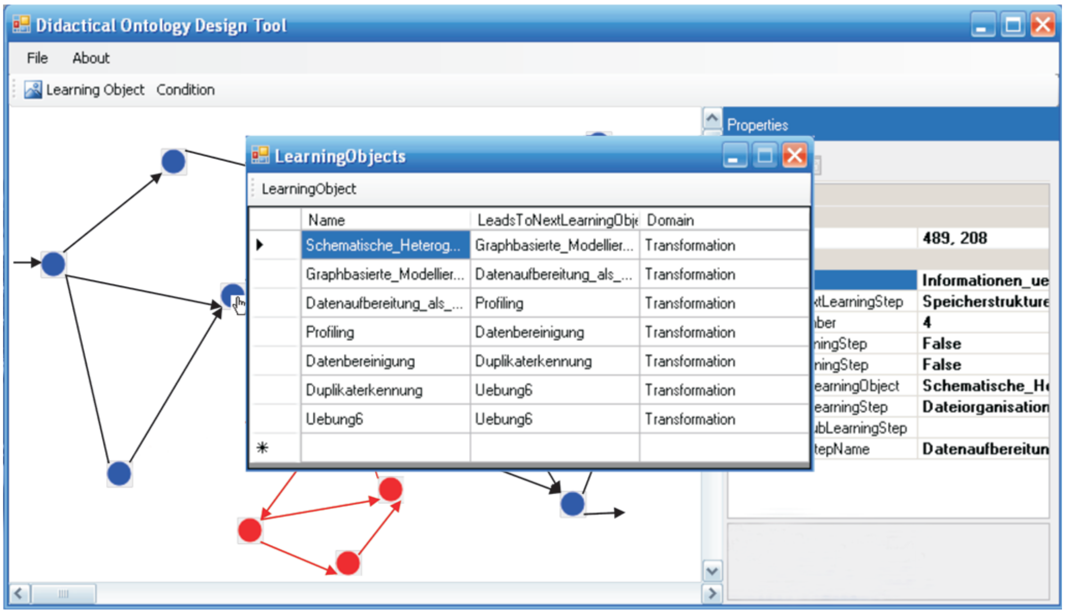


Figure 5.12: Creation of a Level 4 Didactical Ontology

5.1.2.2 Process Model Repository for Didactical Ontologies

With the didactical process models it is possible to reuse existing information and expert knowledge. Reusability is important, because the provision of expert knowledge is a goal. According to [Pawlowski, 2005] these requirements need to be met:

- ◦ Formal representation, like a metadata model
- ◦ Widely accepted representation format
- ◦ Available repositories for search and retrieval
- ◦ Semantics need to be understandable in different contexts

Reuse of existing standards and ontologies is one of the most important aspects for ontology design. If everybody develops his own 'standard', the intention of ontologies to describe a certain semantic never can be achieved. Some contributions to this work base on the widely accepted ontologies of Meder (see Figure 5.13) to describe on knowledge types, presentation media, communication media, matter of fact relations, communication contribution cooperative objects as well as transactions/assignments ([Meder, 2001], [Meder, 2006]). Another foundation is for example the Learning Objects Metadata standard (LOM) [IEEE Learning Technology Standards Committee, 2003].

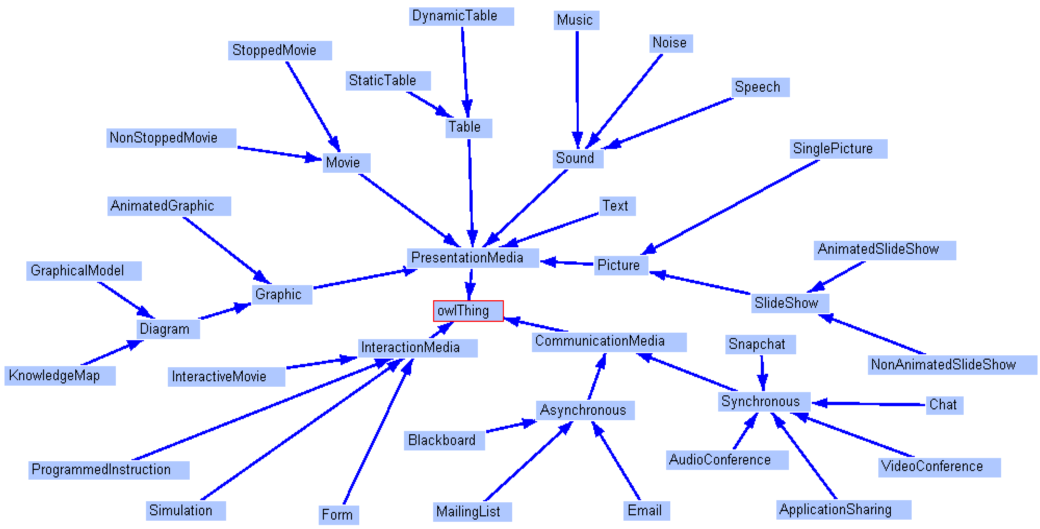


Figure 5.13: Presentation Media Taxonomy of Meder [Meder, 2006]

Applicability is another requirement. Therefore, an architecture of a system is presented that makes explicit and extensive use of the proposed ontologies. On the side of a centralized server, the authorized sets of didactical approaches can be hosted and maintained (location A in Figure 5.14) within a process model repository.

Of course, there may exist other approaches that are (not yet) proved for being applicable, complete – in general sufficient – for the intention of providing didactical expertise. Therefore, a collection pool is proposed (location B). This set serves as the basis of the work of a maintenance authority that analyzes, re-models, annotates, categorizes and releases (in repository A) the proposed didactical approach, if it is found sufficient for approved usage.

The clients, that can benefit from (a hierarchy) of didactical ontologies, are e-Learning systems. They need expertise for the didactically well-founded creation of learning units (location C). Together with other resources like the content itself or curriculum specification, this information is used to author high-quality e-Learning courses.

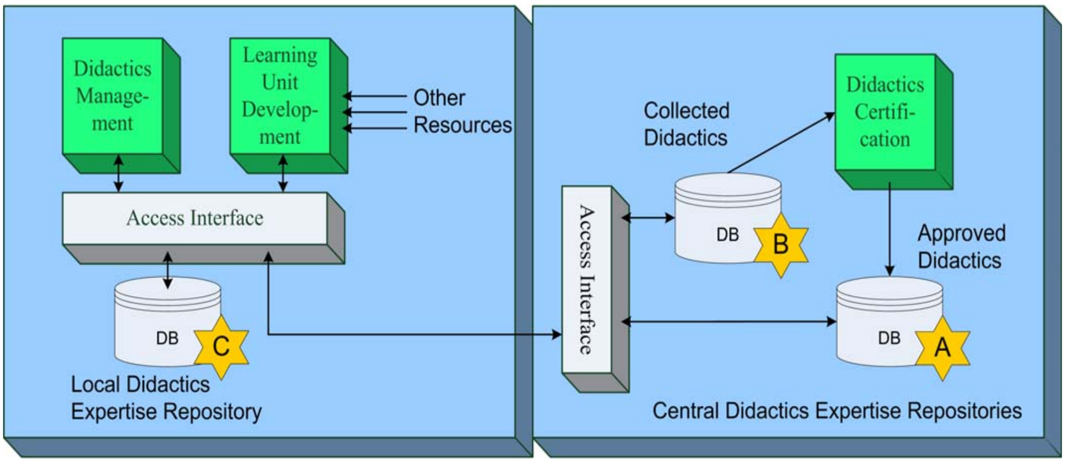


Figure 5.14: Architecture of Process Model Repository [Mencke and Dumke, 2007d]

Another possible application is the storage of individual didactical approaches of the specific learning system vendor or operator. By this, individual and group competencies can be collected, concentrated, further developed, maintained and made available. At each location A, B and C, the ontologies are used to store didactical expertise as well as to serve as a directory representation for categorization and search mechanisms.

Extensibility is an integral part of the proposed approach. New expertise must be addable and new usage scenarios applicable. Flexibility is almost always a goal in modern sciences. Nevertheless, openness should be restricted, otherwise a main goal of the approach (provision of approved didactical expertise) cannot be achieved and by this usability and applicability are reduced. This can be achieved for example by defining new aspects as optional.

This architecture is the basis for the provision of expert knowledge on different levels including specialized process models describing the assembly of entities – in this case e.g. the aggregation of Learning Objects based on functional criteria.

5.1.3 Quality Criteria for Learning Object Selection

The QuaD^2-Framework's next step is about the selection of a quality model. Therefore, the definition of appropriate quality attributes is necessary. Although quality was already defined above, the term quality needs to be interpreted in a special way in the context of e-Learning content provision. Its interpretation is fluid due to the involvement of a human within the whole process. His individual characteristics sometimes determine what is good and what is bad. For example, the amount of pictures of a LO can be a quality attribute. But maybe a certain learner learns better on pure text-based descriptions – the quality-oriented interpretation of certain attributes is not constant.

These facts lead to certain consequences that are defined in the following.

- User characteristics are important (the usage of user models is proposed later in this work)
- Every requirement that is seen as important for the author of the course must be modeled within the process model and thereby becomes a functional criterion for the

selection of LOs for the next process step.

- Every requirement that is not explicitly modeled within the process model can serve as quality attributes and will lead to quality-driven decisions about the LO to be presented.

Another facet of human involvement is, that not everything can be determined with automatically-processable metrics. Human experiences sometimes elude automatic observations. An example are human experiences with learning processes. Those quality aspects can be integrated during the creation of didactical ontologies and provide a substantial additional value that exists but cannot be determined automatically.

As introduced in Section 4.5.1, certain common and special quality attributes exist. In the following, some new approaches for the measurement of domain-specific quality attributes are introduced.

Didactical Appropriateness (DA) Didactical appropriateness is about how good the current didactical approach fits to the desired one. The more equal they, the better it is for the quality of the entire learning process. The metric basis on the taxonomy of didactical approaches as shown in Figure 5.3 as well as ontological distances as specified on page 169. Formula 5.1 presents the developed metric, C_F is the desired didactical approach and C_j is the current one. An additional limitation to $c^{abs}(C_F, C_j)$, $c^{spec}(C_F, C_j)$ and $c^{sib}(C_F, C_j)$ (see Formulas 6.25 to 6.27) is that 0 is defined as a result, if no such distances exists.

$$DA(C_F, C_j) = c^{sib}(C_F, C_j) + c^{spec}(C_F, C_j) + c^{abs}(C_F, C_j). \tag{5.1}$$

This metric sums up all applicable distance measures to define the distances of the concepts within the proposed taxonomy of didactical approaches. The lower the result is, the closer the concepts are related. A future extension may result in the addition of values, describing the similarity of the didactical approaches to each other.

Learning Object Consistency (LOC) Learning Object Consistency is about changes of a Learning Object. For the application of this quality attribute, a metadata history and the introduction of a special update process are necessary. The update process should continuously check the LO for consistency. History information should describe each check and update. The idea is, that a continuously maintained LO has a higher quality. The metric delivers back *true* for a positive consistency check and otherwise *false*.

Learning Object Success Indicator (LOSI) Learning Object Success Indicator may reveal some information about LO quality as well. It determines the relates successful learning processes of available courses $F_{success}$ to the not successful ones $F_{notsuccess}$. Each course must contain the LO under survey.

$$LOSI(LO) = \frac{|F_{success}|}{|F_{notsuccess}|}. \tag{5.2}$$

This metric can be a possible indicator, that the current LO has a high quality and is a key factor for learning success. The higher the result is, the better the quality is indicated.

Learning Object Interest Factor (LOIF) Learning Object Interest Factor indicates an implicit user evaluation about the interest that the LO can cause at the learner's side. It compares the LO's anticipated acquisition time t^{ant} (taken from the LO metadata) with the average acquisition time whenever it was presented to a learner in any course t^{avg}.

$$LOIF(LO) = \frac{t^{ant}}{t^{avg}}. \tag{5.3}$$

$LOIF$ indicates high quality, if the result is close to 1. Much higher and lower values indicate bad $LOIF$ quality.

Domain Coverage of the Course (DCC) Domain Coverage of the Course indicates, how much percent of the domain knowledge is covered by the e-Learning course. The results serve as a basis for other metrics. The formula is based on a ontology-oriented domain description. For the metric, the number of several ontological elements being covered by the course are counted: number of concepts C, number of attributes A, number of non-taxonomic relationships R and the number of instances I.

$$DCC(course, domain) = |C| + |A| + |R| + |I|. \tag{5.4}$$

Domain Coverage of the Learning Object (DCLO) The Domain Coverage of the Learning Object describes, how much percent of the domain knowledge of the course is covered by the current Learning Object. Each LO is about a subset of the domain of the e-Learning course. Because the LO is part of the course, the covered domain subsets of the LO is completely within the subset covered by the course. For the metric, DC the number of several ontological elements being covered by the LO are counted: number of concepts C, number of attributes A, number of non-taxonomic relationships R and the number of instances I.

$$DC(LO, course, domain) = \frac{DCC(course, domain)}{|C| + |A| + |R| + |I|}. \tag{5.5}$$

A quality indicator can be, that all Learning Objects have a similar DC value. Therefore, the DC of the current LO_i is compared to the average DC^{avg} of all LOs of the current course.

$$DCLO(LO, course, domain) = \frac{DC^{avg}}{DC(LO_i, course, domain)}. \tag{5.6}$$

$DCLO$ indicates high quality, if the result is close to 1. Much higher and lower values indicate bad $DCLO$ quality.

5.2 Integrating User Information

As explained, process models are not always completely sufficient to provide all necessary information for certain implementations of the QuaD^2-Framework. Under special circumstances, additional data sources are necessary to make appropriate decisions. For e-Learning and similar domains this requirement results in the usage of so called user models that store information about a certain user [Kernchen and Dumke, 2007a].

The concept is not new. But there exist challenges that focus on ubiquitousness and data modeling for the provision of content throughout a longer time frame. *Lifelong learning* is the related catchword in the context of e-Learning. It relies to the fact that learning is not limited to schools, universities or vocational education. With the opportunities of the World Wide Web, Web2.0, mobile devices and all other recent technological developments as well as the learning requirements set by industry and society, this field of application becomes more and more important for software development. User models are fundamental parts of learning tools. They store information about the user, his preferences and his learning progress. That are main information sources for adapted, individualized and thereby more optimal (content) assembly processes.

Unfortunately, there exist almost as many models as learning tools and only few common standards that are usable for the merging of user models or that support the integration of foreign ones. In this section, the core of a framework for the integration of different user models is presented. The first step is the structured collection of those models. In the vision of an established Semantic Web, their generalization is intended. The additional value for the society is the improvement of learning and teaching during lifetime. By the central storage and update of user-related learning information, a better and more easy system adoption is possible. The autonomous characteristics of the framework to be described will be helpful in this context [Kernchen and Dumke, 2007a].

5.2.1 The Need for Lifelong Information Collection

Lifelong learning is continuous education in everyday life. The corresponding idea was firstly articulated in modern times by Basil Yeaxlee [Yeaxlee, 1929]. Multidimensional changes need to be considered from a psychological point of view. Lifelong learning is influenced by biological, psychological and cultural developments: human abilities, adaptation capabilities and cognitive capabilities change over time. Cultural aspects refer to changes in roles and function [Kernchen and Dumke, 2007b].

Lifelong learning is not limited to the already mentioned aspects. Furthermore, it stands for example for the re-entry in education or for the certification of acquired but not formally evidenced competencies. Therefore, it recombines the existing segmented education areas to a complete system. That includes preschool, school, vocational education, higher education as well as common and advanced vocational education. There exist three key features of lifelong learning [Kernchen and Dumke, 2007b]:

- Lifelong education is seen as building upon and affecting all existing educational providers, including both schools and institutions of higher education.
- It extends beyond the formal educational providers to encompass all agencies, groups and individuals involved in any kind of learning activity.

◦ It rests on the belief that individuals are, or can become, self-directing, and that they will see the value in engaging in lifelong education.

An approach to partly support lifelong learning was presented by Maddocks et al. [Maddocks et al., 2000]. The presented tool is aimed towards students for the development of reflective learning skills by encouraging the adoption of an ongoing model of development from school, through higher education to professional membership within the construction industry. In contrast to the approach presented here, it is limited in terms of domain and time. Lifelong learning is a highly complex, rapid changing and very important aspect of everyday life for the society [Kernchen and Dumke, 2007b].

5.2.2 Lifelong User Models

Many competing providers of educational software, proprietary user models, different user model standards and different application domains are only few reasons to explain the variety of existing user models. Their merging and/or integration is important for lifelong learning.

There are various reasons that actually constrict the development of a general user model for lifelong learning [Kernchen and Dumke, 2007b].

◦ Rapid technological development: Seen on a temporal dimension there were different appropriate technologies for the realization of user models. According to the requirements and the particular once state-of-the-art log-files, databases, semi-structured files, XML-files or ontologies were used for the storage of user data. What are the technologies of the future?
◦ New or altered requirements due to new learning domains and new fields of application: A future domain for e-Learning may be nanotechnology or advanced space navigation. There is much ongoing research that is updating, restructuring or even creating domains. It is not possible for current models to fit all future requirements. E-learning may shift to new fields of applications, too. Ubiquitous learning is only one catchword. That may lead to new or changed requirements for user models.
◦ Vague descriptions: Even well-designed and updated domain descriptions are only partly helpful in this case. Until the establishment of the Semantic Web vision, no clear classification and thereby definition of relations is possible to cover learning incidents over the whole lifetime.
◦ Security implications: In the age of Big Brother and data espionage the storage of information about individuals and companies is a delicate problem.
◦ Interpretability: The bottom line is still the individual. Decisions based on computer-delivered information may give a hint. But they never can represent the whole person. That is why misinterpretations and over-interpretations due to missing, wrong or just overrepresented data are possible.

The overall goals are content aggregation and adaptation support as well as the improvement of learning and teaching during the whole life. More precise data support a more easy and faster system adaptation. There is no need to explicitly ask for information that is already available. Furthermore, new lessons to be learned can be better adapted. The advantage for society and individuals is a more effective learning. Another one is the repetition of knowledge. It is possible, for systems using lifelong user models, to predict required recapitulations of already learned knowledge to prevent oblivion.

There are also advantages related to companies. A first one are well-educated and up-to-date employees. It is efficient to teach them knowledge they do not know but they are interested in. Employees are more motivated, better educated and overall more effective. A more individualized course generation is possible, too. People or companies mainly pay for education they really need and want. Companies providing educational content are more flexible and can produce more individualized courses to obtain more customers.

Several approaches for user model support are identified for the lifelong learning optimizing – the current situation is sketched in Figure 5.15a.

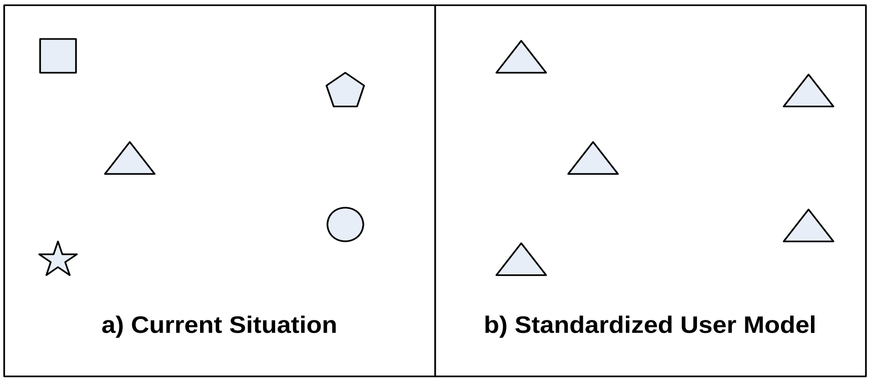


Figure 5.15: (a) Current User Model Situation, (b) One Standard for User Model Implementation [Mencke and Dumke, 2007d]

The implementation of a standardized user model is one possibility (Figure 5.15b). An achieved advantage is the exchangeability between different systems. Problematic realization aspects are the already mentioned imponderables of future developments regarding implementation technology, domain changes and creation as well as altering or new requirements. Another complexity is caused by the transition or reimplementation of user models in existing systems.

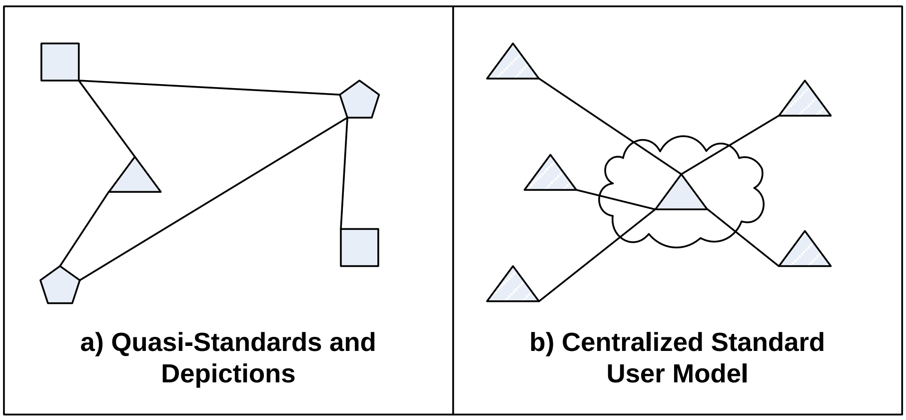


Figure 5.16: (a) User Models of Certain Quasi-Standards, (b) Centralized Extensive User Models [Mencke and Dumke, 2007d]

It is possible to use one or several distinct quasi-standards to achieve the deployment capabilities mentioned in the previous approach (Figure 5.16a). An arising problem is the definition and implementation of appropriate depictions to achieve usability until a standard is developed.

Centralized extensive user models are expected to concede additional advantages (Figure 5.16b). Learning optimization due to central storage and management as well as individual availability and manageability can be achieved. Technical optimization may result in distributed realization and updating mechanisms.

To support the usage of user models for livelong learning, a framework is proposed that defines the central storage of distinct user models as well as the depiction of not explicitly modeled information (see Figure 5.17).

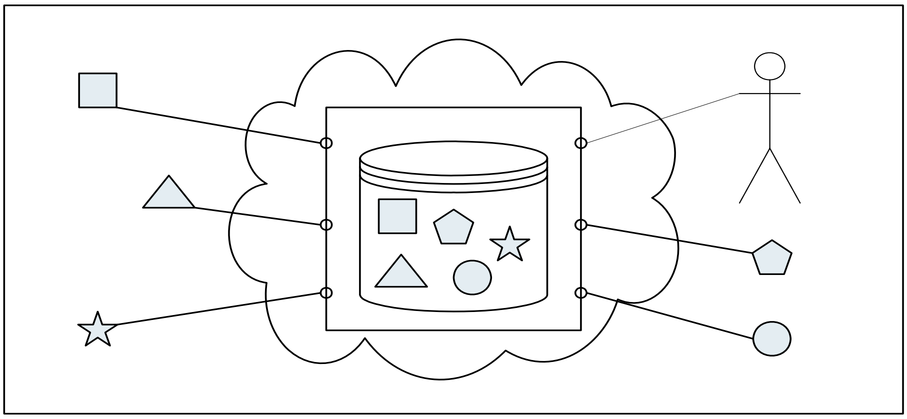


Figure 5.17: Framework for Central Storage, User Accessible and Autonomous Management of Distinct User Models [Mencke and Dumke, 2007d]

The proposed key features are:

- Provision of required information for content aggregation
- Central, uniform access to various user-related information
- User model that tolerates several points of view like temporal order, life phases, learning strategy, learning place, competencies and may include provided meta data
- Smooth transitions, overlaps, gradual characteristics of ontology attributes
- Changeable and extensible user model, consideration of new or changed standards, laws, . . . is possible (consideration of access rights)
- Integration of external domain descriptions
- Central storage and management: eased identification of existing user models and allocation for further processing
- Availability and manageability for individuals
- Data collection to apply appropriate algorithms for the extraction of meta information
- Depiction of learning results and processes that are not explicitly controlled like hobbies, spare time activities and sports
- Depiction of individually varying interests that can be proved for example by club memberships

5.2.3 Autonomous and Proactive User Model Management

In a first version, the framework is used for the central storage and user accessible and autonomous management of distinct user models using agent technology. Possible extensions can provide a (semi-) automated user model adaptation.

5.2.3.1 Architecture

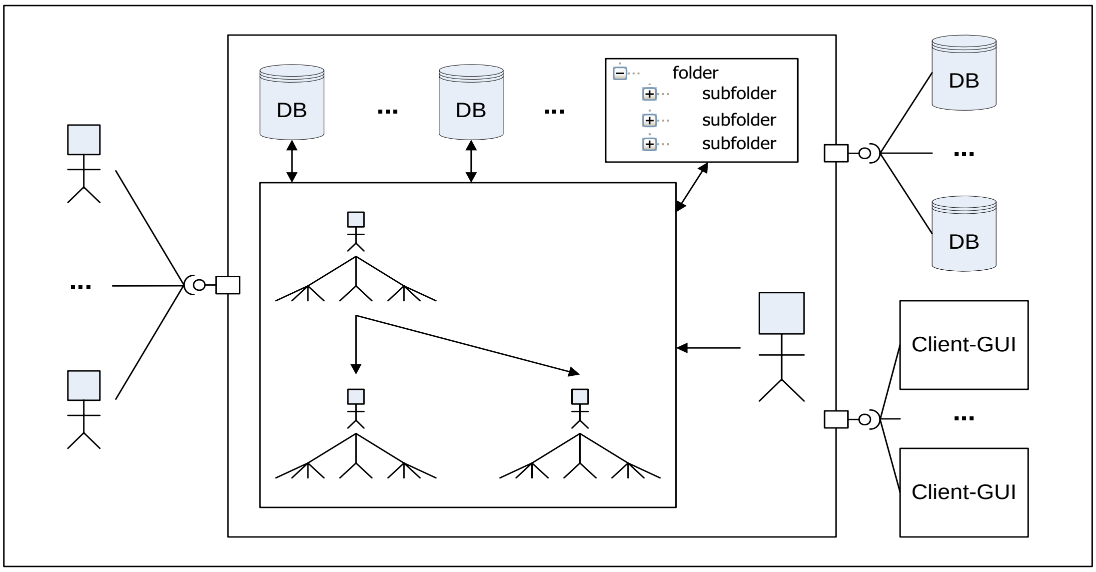


Figure 5.18: Architecture for Proactive User Model Management [Mencke and Dumke, 2007d]

Interfaces: In this framework, the integration of several functional differing interfaces is proposed. First ones are graphical front-ends for the manual access to supply and access data (for individuals and business entities). Because not all structuring decision can be automated so far (see [Morik, 1993], [Maedche and Staab, 2001]), graphical front-ends are needed to support the automated management processes in the case of algorithmic undecidable structuring problems.

Furthermore, automatic interfaces for access via agents or services can be provided. Other interfaces are remote access capabilities to access distributed data sources. The data transfer itself can be realized by communication or mobile agent-based message delivery [Kernchen et al., 2007c].

Data accessing entities may be the modeled user, business entities or administrators. Not everyone should be allowed to see or modify all data. Therefore, access rights need to be defined and granted. The user for example should be allowed to access all data affecting him [Roda et al., 2003], but he should not be able to modify assessment results. Business entities on the other hand can upload new user models or modify data of existing ones uploaded by them. They are allowed to read public information of other user models.

Storages: As already described, the current situation regarding user models is indifferent. They can be differently structured and implemented using different technologies. Because of this, it is argued to supply various storages types like an organized file system, various databases and so on. To ease data provision and to balance resource consumption, remote data access can be an option. The decision for or against this possibility must be carefully analyzed and weighted, because of possible connection problems.

Central agent management: The proposed system is intended to serve as a permanent service on the Internet or intranet. Therefore, failure repair and restart techniques are needed. This can be achieved by storing signatures and data of existing agents and the implementation of appropriate recovery mechanisms.

Processing agents: In the first version of the framework, one agent is intended to handle one special instance of a user model. Later, more advanced algorithms and interactions to partition a user model for processing it with more agents can be implemented.

Hierarchy/societies of agents: One of the major advantages of agent technology is the possibility for distributed problem solving. That relies to functional as well as to spatial distribution. Thereby, for example load balancing and local sub-problem solving can be achieved.

Meta-schema of a user model: As already described, there exist several approaches to realize user models. Here, an ontology is proposed to store the structure of the meta user model as well as the targeted data. In the following, especially process-oriented aspects are targeted. Some of them are: teamwork, effective communication, effective conflict management, team goal achievement, negotiation, group evaluation, cooperation, collaboration, conflict and dispute resolution, arguing a case, locating and collecting and analyzing and synthesizing of data, sharing ideas and making judgments [Kirkpatrick and McLaughlan, 2000]. These are only few ones, more will evolve over time. With this approach, user models can be created that are as complete as possible and that adapt to changes. For the framework, an initial meta-schema based on existing quasi-standards is provided. A remarkable possible extension is the inclusion of domain models like SWEBOK (SoftWare Engineering Body of Knowledge) [Wille et al., 2004] or to other standardized ontologies to determine the body of knowledge that was intended to be learned.

5.2.3.2 Primary Interaction Schemes

Several primary interaction schemes exist within the proposed framework. They are mainly initialized by external elements or other internal interactions.

a) Integration of a new user model following the existing meta model:
- Saving of (reference to external) user model
- Mapping of this model to the actual meta model

b) Integration of a new user model with differences to the existing meta model:
- Saving of (reference to external) user model
- Mapping of this model to the actual meta model leads to problems
- Creation of MAS for user model processing
- Communication and negotiation with (sub-)MAS of meta model and (sub-)MAS' of other user models to identify appropriate mappings or starting points for meta model

adaptation
- ◦ Usage of external resources as other agents (e.g. WordNet-based) or human supervisors

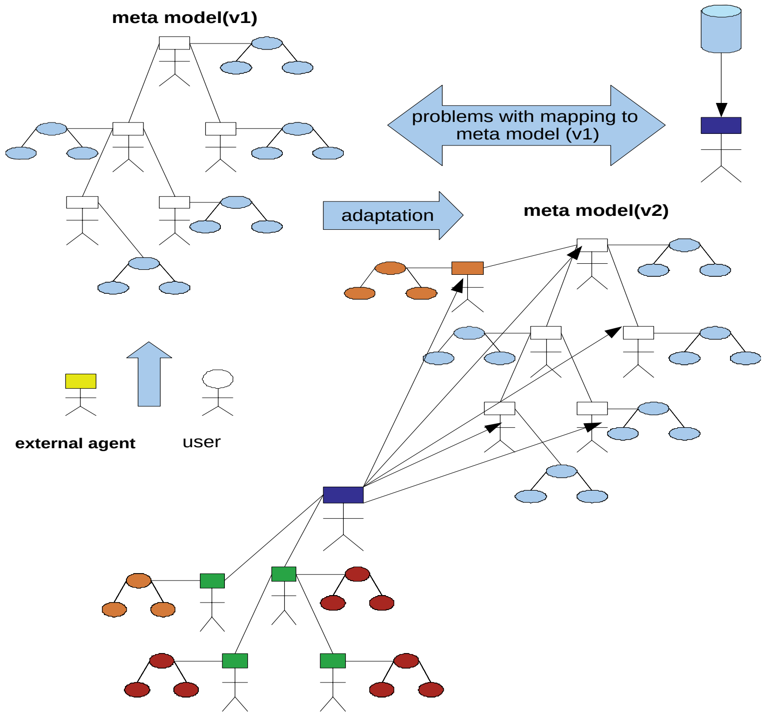


Figure 5.19: Interaction Following "Scheme b)" [Mencke and Dumke, 2007d]

c) Adaptation of already integrated user models:
- ◦ Initialized by b)
- ◦ The adapted meta model is basis for all user model agents

d) Request for user model:
- ◦ Request by manual (user, administrator, . . .) or automatic (Web services, agents, applications, . . .) means
- ◦ Creation of management agent responsible for model of the requested user

5.2.3.3 Implementation Aspects

The framework's basis is an initial user model. Therefore, IEEE PAPI (Personal and Private Information) standard [IEEE Learning Technology Standards Committee, 2002c] was extended by concepts of IMS LIP (Learners Information Package) [IMS Global Learning Consortium, Inc., 2001] in order to provide a substantial, initial basis.

Figure 5.20 visualizes an excerpt of the modeled ontology that represents the structure of the initial user model. It is used as a structure to map existing user models to an internal "standard". The next generation of this application should include automatic agent-based extension and adaptation of the meta user model.

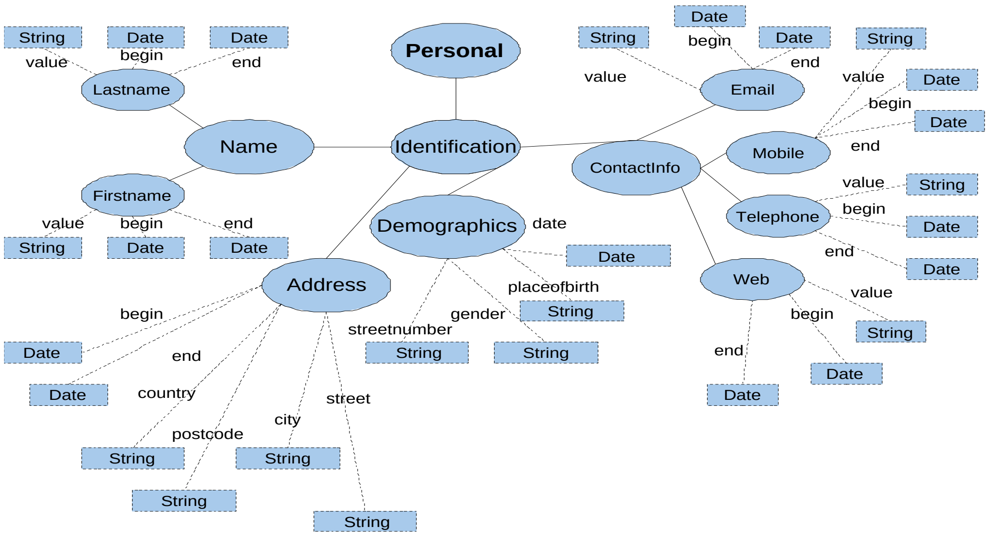
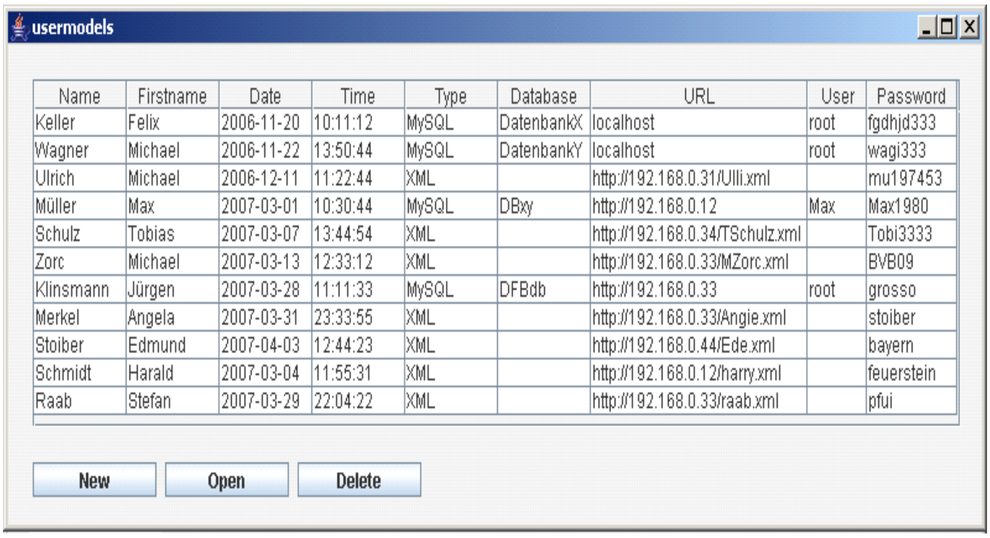


Figure 5.20: Excerpt of the Initial Meta User Model [Mencke and Dumke, 2007d]

Other mentionable aspects are graphical user interfaces for MAS maintenance purposes. Figure 5.21 presents such a GUI including some exemplary user models.



Name	Firstname	Date	Time	Type	Database	URL	User	Password
Keller	Felix	2006-11-20	10:11:12	MySQL	DatenbankX	localhost	root	fgdhjd333
Wagner	Michael	2006-11-22	13:50:44	MySQL	DatenbankY	localhost	root	wagi333
Ulrich	Michael	2006-12-11	11:22:44	XML		http://192.168.0.31/Ulli.xml		mu197453
Müller	Max	2007-03-01	10:30:44	MySQL	DBxy	http://192.168.0.12	Max	Max1980
Schulz	Tobias	2007-03-07	13:44:54	XML		http://192.168.0.34/TSchulz.xml		Tobi3333
Zorc	Michael	2007-03-13	12:33:12	XML		http://192.168.0.33/MZorc.xml		BVB09
Klinsmann	Jürgen	2007-03-28	11:11:33	MySQL	DFBdb	http://192.168.0.33	root	grosso
Merkel	Angela	2007-03-31	23:33:55	XML		http://192.168.0.33/Angie.xml		stoiber
Stoiber	Edmund	2007-04-03	12:44:23	XML		http://192.168.0.44/Ede.xml		bayern
Schmidt	Harald	2007-03-04	11:55:31	XML		http://192.168.0.12/harry.xml		feuerstein
Raab	Stefan	2007-03-29	22:04:22	XML		http://192.168.0.33/raab.xml		pfui

Figure 5.21: Screenshot of Prototypical Manual Administration Tool for User Models [Mencke and Dumke, 2007d]

The processing components of the proposed framework are societies of agents. Following the presented interaction schemes, distinct hierarchies of agents are evolving (see Figure 5.22).

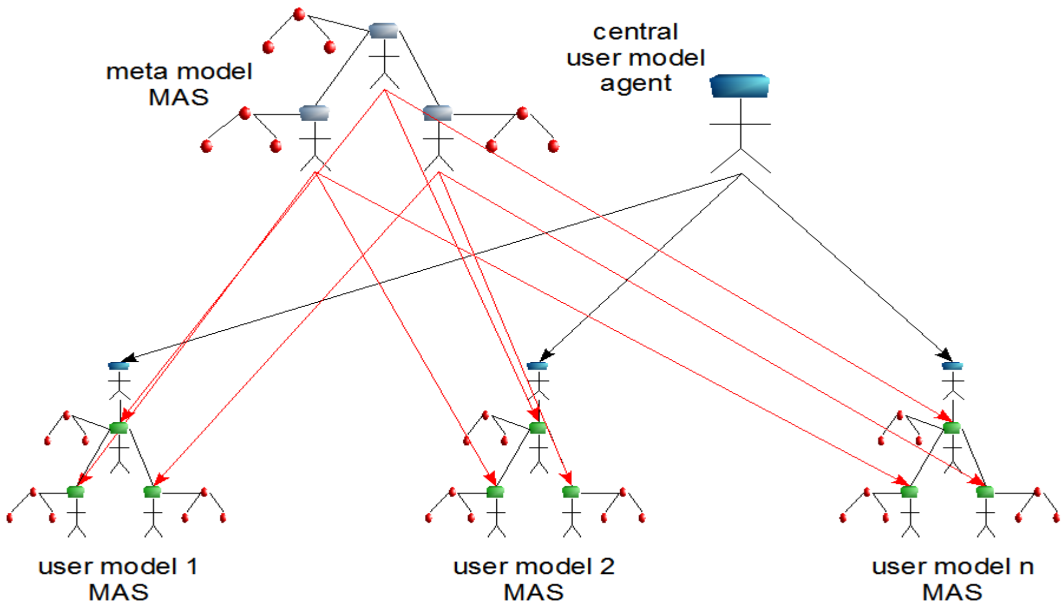


Figure 5.22: Evolving Agent Societies [Mencke and Dumke, 2007d]

Therefore, different points of view are possible.

- Learner focused: A society of agents processes user models of one special learner. One central agent exists that concentrates the information in a user model based on structural information from the agents managing this meta model (black arrows in Figure 5.22).
- User model focused: Each distinct user model of a particular user can be processed by a society of agents. Thereby, the mapping of user model and agent responsibility is based on the partition created for the meta model processing.
- Meta model focused: Another hierarchy of agent is responsible for the maintenance and adaptation of the meta user model.
- Topic focused: A society of agents is processing special aspects of user models. Central agents concentrate on evolving structural adaptations for this aspect and inform the main agent managing the complete meta model (red arrows).

Next to the already identified ones, there are more possible fields of application for agents for the distributed management of user models and the creation of a general meta user model.

That includes the temporal and spatial independent registration of learning processes by autonomous agents that can be located in several everyday life software or hardware entities: e.g. browsers, learning systems, personal assistants, mobile phones.

Agents for temporal and spatial independent individual user model development are possible, too. As the already mentioned registration agents, their deployment can vary to cover as many learning situations as possible. Those agents may work dependently and independently from the proposed framework.

Another mentionable approach is the processing of stored user model information by current and highly optimized (data mining) algorithms in an architecture as described in [Kernchen et al., 2007c]. There, confidential and modern processing capabilities can be "leased" from appropriate providers.

5.2.3.4 Advantages

An arising question is how to encourage users and business entities to participate. A clear answer is the motivation and definition of expected advantages end evolving products. In the following, advantages are outlined in terms of central management, availability and completeness of user models. A first and main positive aspect is the support of automated content aggregation. Another one is the possibility of creating more individualized and thereby more effective learning courses because of central available and complete user models.

Advantages Due to Central Management By a central management, learning experiences from different learning platforms, life phases, spheres of life, …can be taken into consideration and mapped to the extended user model [Kernchen and Dumke, 2007a].

A central management can provide the basis for anonymous comparisons to other individuals, groups or certain benchmarks and standards. That may involve experts, students and peers for example. That may serve as an indication for the individual learning progress.

Another aspect is the consideration of different learning strategies. A better analysis of learning style appropriateness becomes possible by comparing preferred learning approaches. That is directly coupled with learning performance and learning success [Mencke and Dumke, 2007d].

Furthermore, the summarized provision of preferences directly leads to better adaptation qualities of the learning management systems based on those data (e.g.: preferred type, size and color of font, aspects of accessibility).

A direct usage of central stored user models leads to applications that do not need to re-initialize the user model every time the user changes the client he uses.

A central storage is basis for the development and allocation of appropriate sources, tools and services [Sinitsa, 2000] and new products. Techniques as described in [Kernchen and Dumke, 2007e] can be easier applied.

With some limitations and extensions, the proposed approach may serve as a support tool to find a new job respectively a new employee.

Advantages Due to Availability If the user model is available and manageable for individuals, then it provides control and responsibility to learners [Kay, 2000]. The proposed approach thereby supports meta-learning activities like the monitoring of learning, the setting of personal learning goals; it is the basis for planning goals and supports the reflection about and the tracing of the learning progress by the comparison of set goals [Kernchen and Dumke, 2007a].

The learner gets an option to acquire an overall learning experience to develop learning abilities and qualities necessary for effective autonomous learning [Sinitsa, 2000].

The determination of the current learning status and the substantiated evidence of missing competences becomes possible for lifelong learning.

The central availability increases the responsibility of learners that enhances learning effectiveness and clarifies that learning and skill development is an iterative and continuous process [Maddocks et al., 2000].

It also supports the user in optimizing his learning environments by the central availability of data like contacts or bookmarks. Less initial interaction and an overall improved workflow with the learning system is another result in early stages of the interaction with the system.

Advantages Due to More Complete User Models Although lifelong learners' motivation tends to be higher than traditional student's motivation [Patel et al., 2000], it can be improved when they know why they do something at a particular time. The user gets feedback on the exploration process. That avoids the problem of mismatches between set learning goals and the learner's perception of them what may cause difficulties.

More complete user models include more exact information about existing knowledge. That includes whole domains as well as special aspects. Thereby, again adaptability and learner performance can be increased.

The storage of temporal learning information supports repetitions to avoid knowledge loss. Again, new applications become possible [Kernchen and Dumke, 2007a].

Summarized, the sketched approach of proactive user model processing for lifelong learning may provide substantial support in a later, improved implementation stage.

5.3 Proactive Entity Retrieval

So far mechanism were described that provide appropriate process descriptions with a process model repository. With this information entities can be assembled. Those entities are stored within an entity repository. Such an entity repository is defined as a technical infrastructure to store, manage and retrieve entities. It is more than the concept defined in [Cheng and Chang, 2007] where the whole World Wide Web (WWW) is seen as a Entity Repository.

The implementation of Entity Repositories may vary depending on the domain of interest. For content-related domains like e-Learning that can be databases (e.g. described and listed in [Porter et al., 2002], [Richards et al., 2002], [McGreal, 2004] or [Norman and Porter, 2007]). That are already established technologies.

Effective extensions may result in the usage of agent technology for the proactive processing and delivery of content. For this purpose an approach is presented below [Kernchen et al., 2007c]. The key idea is to introduce a specialized third party to ensure the success of such access mechanisms. Many functional and non-functional aspects must be taken into account for development of these mechanisms – e.g. performance, security, format conversion, data generalization, etc.

There are several reasons that make it inappropriate to dispatch raw data over the network:

- The data may be confidential and therefore need to be anonymized.
- The amount of the data can be too big. In this case, it could be advisable to generalize the data on the database provider node and to transfer only these generalization results to the customer.
- Furthermore, this special information processing can be not in line with qualifications of both the database holder and its clients. The usage of agents gives the possibility to outsource the development and installation of additional software to an external agent provider, which is then responsible for regular updates of the software considering new algorithms etc.

Using agents as mediators for human database access is a common idea. They act as proxies to create queries for database access ([Masuoka and Ohtani, 1999], [Elio et al., 2000]). That is motivated by assumptions like (a) highly complex database design so it is not possible for the human agent to specify a single, simple database query for which there is one single answer; (b) vague set of constraints when starting the search task, (c) multiple search goals, or (d) the interface is not visual or requires direct manipulation.

The realization of distributed databases is another important role for agents in this context. While working towards a certain goal, they may exploit concurrency, parallelism and distribution to thereby bequeath those functionalities to databases [Kernchen et al., 2007c].

Figure 5.23 sketches the application.

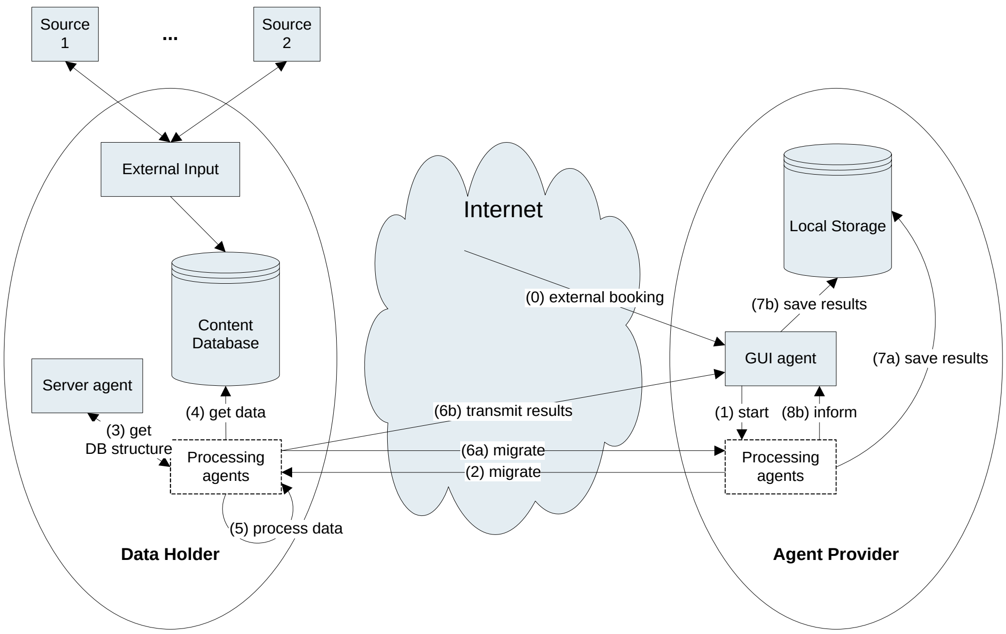


Figure 5.23: Agent-Based Entity Provision

5.3.1 Architecture

As shown, agent technology can provide substantial support for the proactive access and filling of the entity repository. In the following, an architecture sketching a possible business application, is derived.

Within the proposed model, the following parties exist. The data holder owns a large data amount that is to be processed. On the data holder side, there exist an interface for interaction with the agent provider, a service for configuration of arrived agents and the database mentioned above. The functionality tha t is necessary for database processing can be obtained from the agent provider in the form of an appropriate agent. Figure 5.24 shows component interaction steps within the architecture. These components will be discussed in detail below.

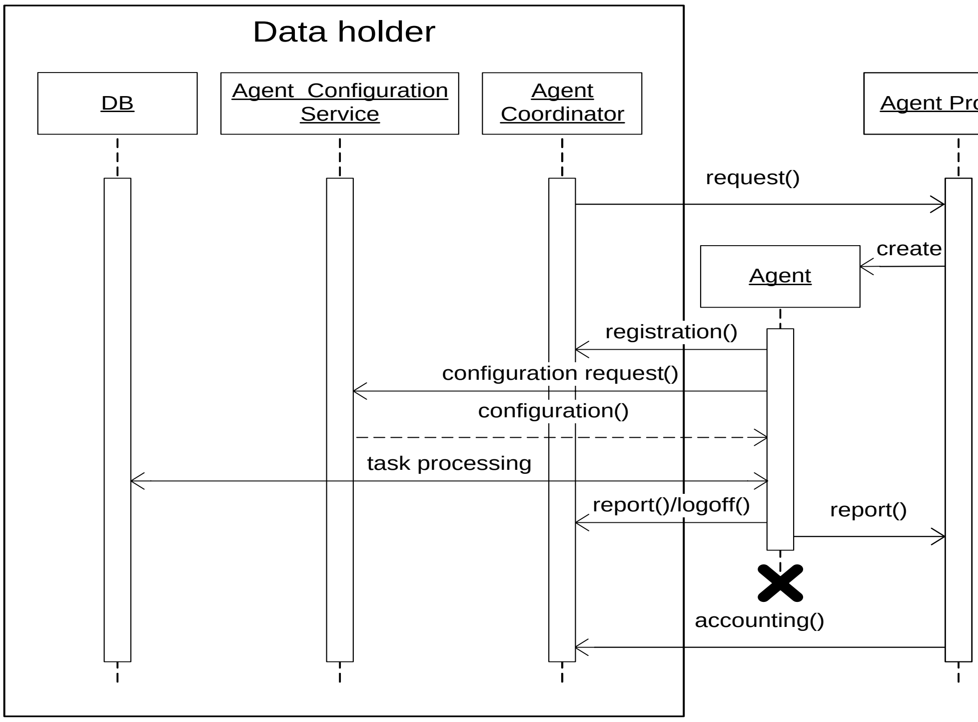


Figure 5.24: Interaction Flow of Involved Parties [Kernchen et al., 2007c]

5.3.1.1 External Entity Provider

The interface of the agent coordinator is dedicated for interaction with agent providers and their agents. It initializes the interaction, registers the generated agents and carries out management tasks. Special ontologies could be helpful to solve possible interoperability issues. It is important to standardize the relevant vocabulary and message exchange patterns used for requests to the agent provider, the registration of agents and the accounting process [Grütter, 2005].

Providing a uniform database accessing mechanism is the most important task of the agent configuration service, because there might be different databases and database schemes. This service supplies necessary information about access credentials and addresses of tables and views which need to be processed. Furthermore, by this the agent gets a location for the storage of intermediate results.

5.3.1.2 Proactive Functionality Provider

The service located on the side of the agent provider, provides the needed functionality. Therefore, it takes and checks orders of the clients and instantiates appropriate agents for autonomous data processing. Other tasks are the accepting of reports and accounting. Non-functional properties of both, the provider's interface and the delivered agents could be described using special ontologies.

By creating a single authority for this purpose, up-to-date functionality with the newest and high-performance algorithms as well as current data (e.g. about laws) are available for the customer due to the stress of competition amongst the functionality providers.

5.3.1.3 Agent

The agent is the main working component in the proposed architecture. It is instantiated by the agent provider and should support all requested features as specified in the client's order. A modular construction system and a predefined skeleton constitute the foundation of the generation process. After migration, the agent registers itself on the agent coordinator and is being configured for database access. Now it is ready to perform its tasks. Finally, it sends reports to the agent coordinator and the agent provider as a basis for service accounting.

Security aspects such as safe code distribution and sandbox-based execution should be respected by implementing the infrastructure. To avoid espionage network interaction is minimized, because main communication takes place on the data holder's side. Furthermore, the usage of secure transport protocols and authentication mechanisms is proposed.

The next question is, where these agents can be taken from. Below three possible answers are outlined and discussed with their advantages and disadvantages:

- Agents are implemented and provided by the database owner side. In this case, there is roughly the same development effort as if an additional analysis component would be implemented, only the implementation technology differs.
  - Advantages: (a) possibility to reuse such agents for other data holder; (b) no security risks and low agents configuration effort.
  - Disadvantages: Wesement is a research project, and there is neither interest nor resources to develop the agents.
- Agents are implemented and sent to the data holder's side by a data creation/provision facility.
  - Advantages: they have the full control of how their agents behave.
  - Disadvantages: (a) reuse of agents between many providers is impossible; (b) same as in previous case, the provider might not have all necessary resources to develop and manage the agents.

- Full division of labour – a third party works on behalf of the data holders and deals with development and management of agents.
  - Advantages: (a) possibility to reuse agents for many data holders; (b) possibility for providers to select agents from many offerings.
  - Disadvantages: management efforts.

Figure 5.25 shows the proposed approach in the context of the Wesement research project as described in ([Rud, 2006], [Schmietendorf et al., 2004], [Kernchen et al., 2007c]). Interaction steps of the three possible scenarios discussed above are numbered appropriately.

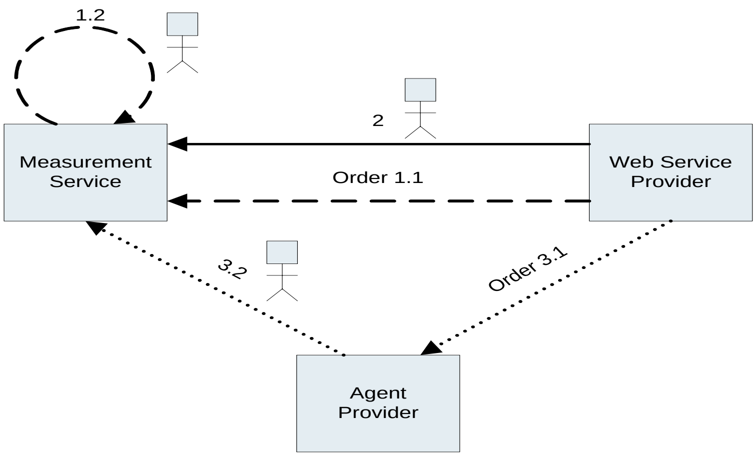


Figure 5.25: Agent Deployment Scenarios [Kernchen et al., 2007c]

5.3.2 Application Scenarios

5.3.2.1 Acquisition of Third-Party Know-How

Next to content provision, several other fields of application for this architecture are possible. In the scenario presented here, the large amount of data on the data holder side needs knowledge and know-how for analysis. These algorithms can be subject to change or proprietary. The proposed architecture makes the data holder independent of procurement and update of monolithic resource-consuming software products or expensive in-house development. Time and effort for development, optimization and maintenance is outsourced to specialized providers. In that way, service evolution is simplified, too.

The further advantages of using agents are minimized network load and increased security – no confidential information must be dispatched.

Other possible fields of application of agents in this case are: knowledge discovery in self-contained data volumes, automatic tax consultancy or the extension of the internal knowledge base of the agent or its providing service in a generalized and anonymized manner.

5.3.2.2 Case Study: Web Services Measurement Service

The architecture presented for the proactive content provision was implemented for the special domain of a Web-Service-based distributed infrastructure.

The Web Service measurement service Wesement, being operated by the SMLab-Team of the Otto-von-Guericke University of Magdeburg as part of the Web Service Trust Center research project ([Rud, 2006], [Schmietendorf et al., 2004]), is collecting performance statistics of third-party Web Services. Any Web Service provider can use Wesement for long-term availability, metadata stability and performance measurement of its services at no-cost. Collected statistics is being persisted in a relational database. The only available means go generalize the data in the Web-based Wesement frontend is to represent it graphically in form of per-day or per-month diagrams. That is unambiguously insufficient, because:

- Another presentation form – for example a Web Aervice provider, simply wants to get notified whether its Web Services are performing well on a given day (i.e. to get a boolean value instead of a PNG image) or to receive a list of services whose performance in this month has fallen below the preconfigured limit.
- Another time granularity – some providers may want to get weekly or yearly reports in addition to daily or monthly ones.

A possible solution of this problem would be to change the measurement service frontend itself. But this way is tied with enormous maintenance effort, because the frontend will need to be customized almost every time a new provider comes and registers its services. Another possibility for the providers is to get raw statistics (e.g. in XML format) and process it locally.

The most appropriate solution is to use agents, which work on the Wesement's side and thus have local access to the statistics database. Agent technology provides the highest degree of productivity, because it is the only possibility for clients to extend the infrastructure of the server. By this, they are able to remotely perform their tasks in direct adjacency to the possible source of events. Results of their calculations can then be sent to the corresponding Web Service providers. It is also advisable that agents work continuously and keep track of services' performance changes in "real-time" – if a negative trend is detected, the provider can be notified early enough.

5.3.3 Implementation Aspects

In the following, the most important aspects regarding the implementation of agents for remote processing and delivery of entities are presented. The focus was on measurement databases, but the described technology can easily be adapted to entity and thereby content provision in general. As aforementioned, the database of the existing service Wesement containing performance statistics of Web Services is used.

For the execution of agents an infrastructure is needed. JADE [Telecom Italia, 2007] was chosen as a Java-based agent platform to provide normative services like life cycle management, white pages service, yellow pages service and message transport service as defined by the according FIPA standard for agent platforms. Additional services are agent-software integration, an ontology service and human agent interaction.

Based on this technology, the multiagent system was created. Therefore, one JADE platform with several containers was launched. They are used to logically separate agents and are often referred as “agent cities” in literature. In JADE they can be distributed on several computers. In the present case, one container resides on the same computer as the database to allow its remote processing and to guarantee the aforementioned advantages. The others are distributed.

Entry point for agent creation is the ClientAgent. It is launched with the JADE platform and presents a Graphical User Interface (GUI) to the user. The main objectives of this agent are the selection of an appropriate processing agent type as well as the definition of required parameters. The ClientAgent is able to detect available processing agent definitions and presents parts of their corresponding GUI’s. Thereby, the separation of ClientAgent and processing agent functionalities is guaranteed. Each of the later agents provides an own GUI component for the definition of necessary parameters. Figure 5.26 visualizes the ClientAgent with a graphical component of a picture agent.

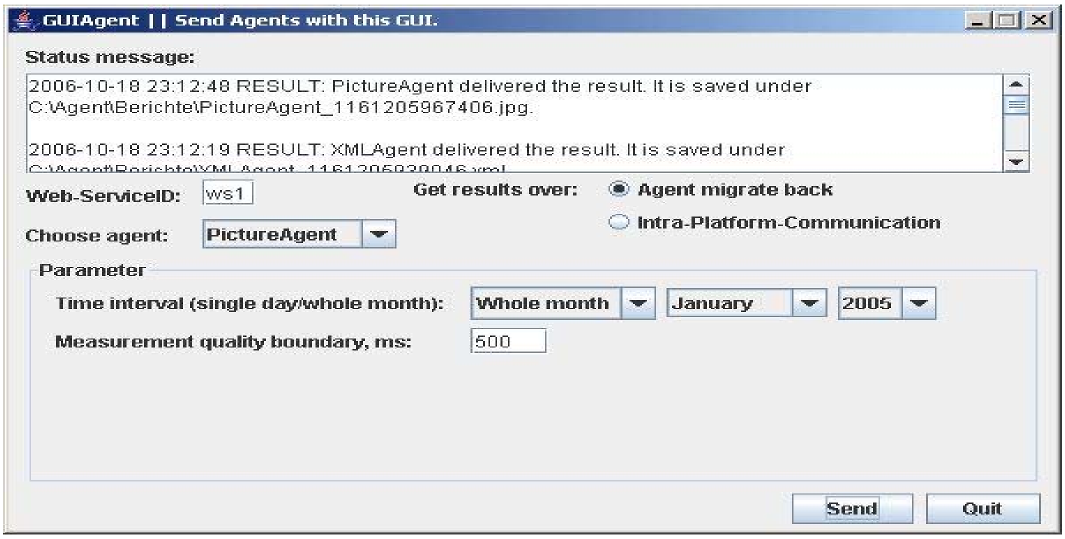


Figure 5.26: Sending a PictureAgent for Remote Processing [Kernchen et al., 2007c]

Parameters common to all agents are the Web Service ID, the result delivering definition and the selection of the processing agent. Therefore, these aspects are integrated into the ClientAgent in addition to a window component presenting status information and locations where the results of the processing agents are locally stored. By the Web Service ID, the data of the Web Service to be analyzed can be clearly identified. This approach takes benefits from the advantages of mobile agents. By this, it is possible to define result delivering by intra-platform communication as well as by agent migration.

The remote container accommodates an additional agent with proxy functionalities. It provides database accessing information for the processing agents and is intended to be extended by safety mechanisms to restrict database access to secure agents. All agent communication acts are ontology-based. Thereby, the possible conversation content is defined and the architecture is extensible towards the usage of additional Web Service measurement services. From a technical point of view, this ontology needs to be mapped to Java. Classes realize the underlying communication in the multiagent system.

The presented PictureAgent needs additional parameters about the time interval to be observed and a measurement quality boundary. As a result it creates a graphical visualization presenting performance statistics of third-party Web Services.

Two more exemplary processing agent types were implemented. The ObserverAgent periodically (e.g. weekly, monthly or self-defined) checks the state of the Web service for a predefined time and sends back alert messages in case of state changes. The XMLAgent creates XML documents containing the performance data of the selected Web Service. They can be used for further processing. Its graphical component is shown in Figure 5.27.

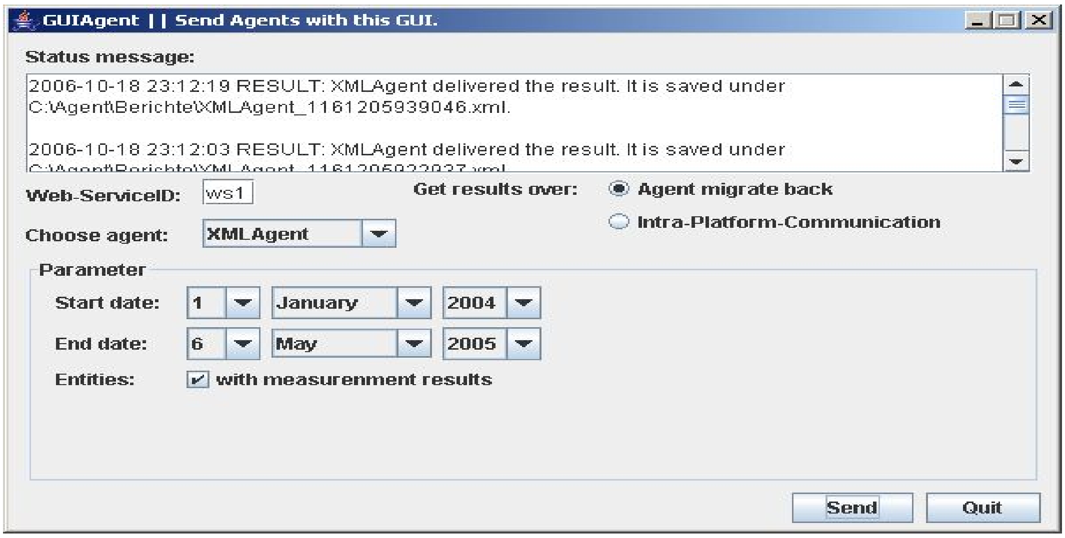


Figure 5.27: Sending an XMLAgent for Remote Processing [Kernchen et al., 2007c]

5.4 Discussion

With this chapter, proactive and semantic-based technologies were introduced to support the initialization phase of the QuaD^2-Framework. Detailed descriptions about ontology-based process models for the domain of e-Learning were one key factor as well as the description of special user models and their lifelong usage. The proactive provision of entities is also targeting initiatory activities. Therefore, a special agent-based approach was described.

At first, an appropriate implementation technology for process models was derived. Additionally, the structure of the Process Model Repository was sketched. Formal, ontology-based representations were used to model those data. Next to the delivered process-expertise and e-Learning course descriptions, a shared vocabulary for teachers, course designers and experts for didactics was presented. Furthermore, features for repository-based search and retrieval mechanisms based on sufficient metadata were used. Thereby, multiple, adaptable accessing and classification schemes become possible according to different starting points for the initial information need as well as according to the differing points of view within the pedagogic science. Additional intentions are the minimization of duplications and the improvement of reuse for the objectives of the $QuaD^2$-Framework.

Another main targeted area were models for the provision of additional information for model selection as well the content aggregation process itself. At this point a special focus was laid on an architecture for proactive and autonomous user model handling for lifelong usage. Therefore, possible solutions were discussed and the most appropriate one further sketched.

The provision of entities is an already established task depending on the special domain of interest. Nevertheless, a proactive extension was proposed. Possible scenarios of accessing content databases using mobile agents were analyzed. Also, different implementation approaches were outlined and discussed in terms of advantages and disadvantages.

The next $QuaD^2$ phases, namely Feasibility Check and Selection were extensively described in Chapter 4. In the next chapter, the focus is laid on proactive and semantics-based improvements of the processing of assembled entities within the Operation phase.

6 An Ontology-Based Approach for Proactive Enrichment and Presentation

Proactive, semantics-based support mechanisms for the Operation Phase of the QuaD^2-Framework are presented in this chapter. Figure 6.1 shows the covered elements. Thereby, especially the provision of related entities and an adaptive presentation technology is presented.

Again, e-Learning is chosen as the major use case to present the developed technologies.

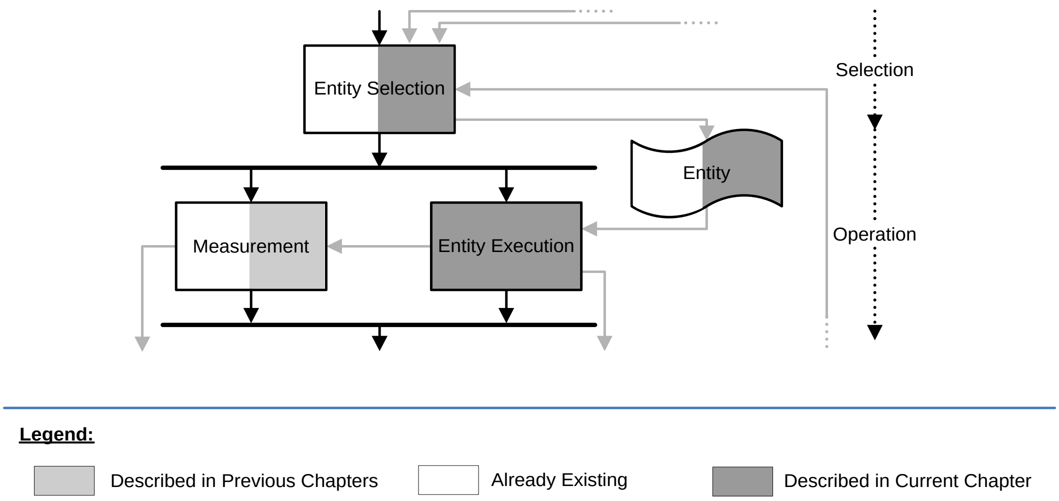


Figure 6.1: Research Questions of the QuaD^2-Framework's Operation Phase

6.1 Enriching Assembled Content

Information consumption is the major use case in almost every domain. But the delivered content is not always sufficient. There may be several reasons for this lack:

- Incomplete content because of content provision design
- Incomplete content due to provider's intention for motivation
- Too difficult content due to missing receiver competencies
- Intended active receiver involvement (e.g. for learning purposes)

From these and other reasons, an additional need for information arises. But not in every case standard search and retrieval mechanisms are capable to satisfy this need.

Application options exist in every domain where information needs to be presented to a user. With the algorithm presented below, the domain of e-Learning is targeted. It is a proposal for the automated enrichment of e-Learning content with ontologically classified resources. The work is also valuably usable for other users of e-Learning systems,for example content creators, learning unit authors or didactical experts.

6.1.1 Content Enrichment Algorithm

E-Learning-related content is any portion of data that can be displayed to a user by the runtime part of an e-Learning system. According to this, content enrichment describes the process of searching and displaying additional information, being semantically related to the information of the e-Learning content [Mencke et al., 2008h].

In the following, a methodology and certain ontology-based algorithms are presented, that are capable to bridge the gap between information need and the provision of appropriate additional content.

6.1.1.1 Integration Variants and Architecture

There may exist different architectural realizations of such an enrichment application. In general, the client's request for new information needs to be satisfied by the content providing server. The resource enrichment must be implemented somewhere on the way of the information flow between server and client. Possible variants are the integration of the enrichment functionality into the server, into the client or into a standalone proxy. Figure 6.2 shows these possibilities.

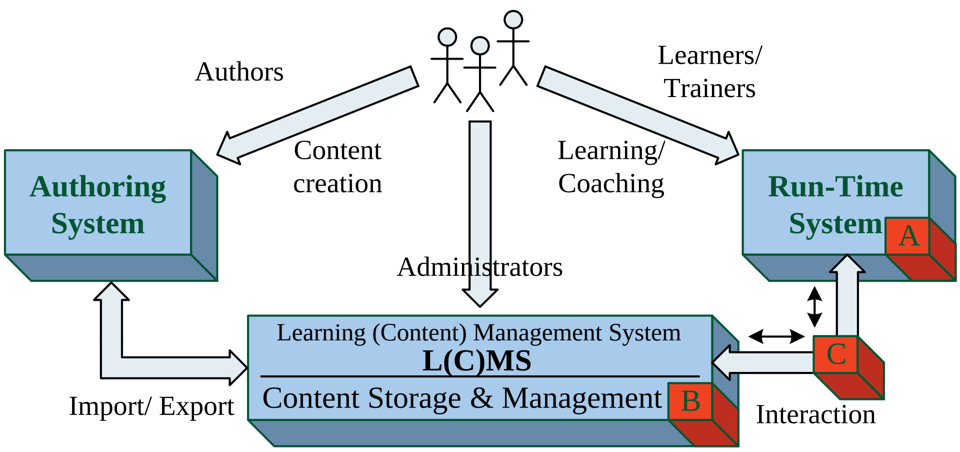


Figure 6.2: Client- (A), Server- (B) and Proxy-Based (C) Realization (adopted from [Pankratius et al., 2004])

Server-side content enrichment causes changes within the existing architecture. The needed redesign can be very cost intensive. Advantages are the possibility to consume full server resources and that there is no need for additional software installation on the client's side. Changing the client results in positive and negative consequences, too. So the nearness to the user leads to better adaptation mechanisms due to more available personal data. But an increased complexity of the communication structure and the installation effort adjust those advantages.

Both solutions can be sufficient, but they suffer from the lack of flexibility. Here, a more adaptive solution is proposed. By the engagement of a proxy, the following aspects become advantageous:

- Easy integration in existing systems (no complete redesign is necessary)
- Improved extensibility and exchangeability of functionality
- Simultaneous applicability for multiple servers and clients
- Proxy location is flexibly changeable

Flexibility is the most important advantage compared to the other approaches. There is no need for a predefinition of the implementation's location. It can be adapted according to special requirements and even changed during operation to be server-sided, client-sided or located somewhere in between.

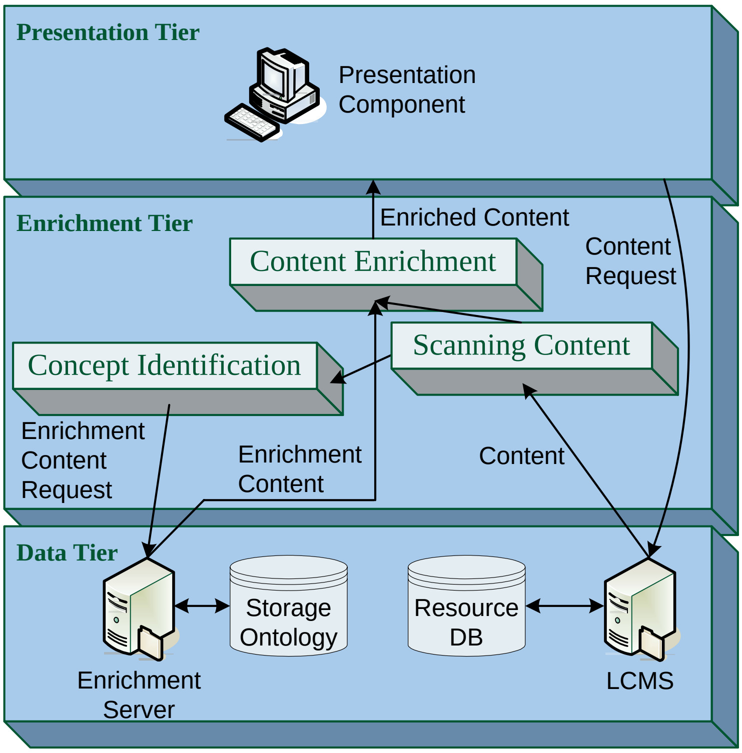


Figure 6.3: Three-Tier Architecture for e-Learning Content Enrichment [Mencke et al., 2008h]

The enrichment component described in [Mencke et al., 2008h] proactively scans the requested e-Learning resources, integrates new semantically-related information and thereby adapts the presented information. A three-tier architecture was developed that is shown in Figure 6.3.

The presentation tier displays the content of a resource, for example e-Learning course material or a web page. It invokes a request for new data from the proxy server of the second tier. Afterwards, they are displayed to the user for further usage. On this level, existing presentation components are used. They are not changed with the proposed solution.

The second tier with the proxy server contains the application logic. It analyzes the request of the presentation front-end and forwards the request to the content provider, too. Additionally, it queries the data tier that contains the ontology. Based on both data sources, the e-Learning resource to be displayed is enriched and sent to the visualization front-end.

The data tier is responsible for the structuring and access of the semantic information within the ontology as well as the storage of enrichment content.

6.1.1.2 Enrichment Process

For the identification of starting points for enrichment in an educational content an 'Enrichment Algorithm' is developed in [Mencke et al., 2008h].

In the first step, an identification of appropriate ontological elements within the ontology $O(C, P)$ with its concepts C and properties P is performed.

The function $f^{naming}(a)$ (Formula 6.1) delivers a human readable name of an ontological element a. The tuples, containing ontology elements a_i and their names determined using $f^{naming}(a_i)$, constitute the set T^O as shown in Equation 6.2.

$$f^{naming} : \text{OntologicalElement} \mapsto \text{String}. \tag{6.1}$$

$$T^O = \{\langle a_i, f^{naming}(a_i)\rangle | a_i \in (C \cup (P \setminus P_{tax}))\}. \tag{6.2}$$

At this point, taxonomic relations within the ontology (P_{tax}) are neglected, because $f^{naming}(a)$ cannot deliver any useful results for them.

A second step is the inflation of T^O with appropriate additional terms, for example taken from the WordNet specifications for the English language ([Princeton University, 2006], [Kruse et al., 2005]). The function $f^{syn}(a)$ delivers additional terms (synonyms) (Formula 6.3). The tuples of the extended set T^{O+SYN} connect ontology elements a_i with their synonyms (Equation 6.4).

$$f^{syn} : \text{String} \mapsto \{\text{String}, \ldots\}. \tag{6.3}$$

$$\begin{aligned} T^{O+SYN} = T^O \cup \{\langle a_i, b_i\rangle \mid a_i \in C \cup P \setminus P_{tax}, \\ b_i \in f^{syn}(f^{naming}(a_i))\}. \end{aligned} \tag{6.4}$$

The function $f^{concept}(x)$ (Formula 6.5) applies to both metadata LO^M and the content LO^C of learning objects LO (Formula 6.6) and extracts names of concepts contained in them. A particular implementation of $f^{concept}$ can use classic mining algorithms. For each learning object LO_i, the initial set T_i^{L+SYN} of concept names and their synonyms, that can serve as starting points of the enrichment, can be determined as shown in the Equation 6.8.

$$f^{concept} : \text{DataObject} \mapsto \{\text{String}, \dots\}. \tag{6.5}$$

$$LO = \{LO_i\} = \{\langle LO_i^M, LO_i^C \rangle\}. \tag{6.6}$$

$$CN_i = f^{concept}(LO_i^M) \cup f^{concept}(LO_i^C). \tag{6.7}$$

$$T_i^{L+SYN} = CN_i \cup \bigcup_{x \in CN_i} f^{syn}(x). \tag{6.8}$$

The next step is to match the identified concepts of the learning objects with the human readable names of ontological elements (Equation 6.9). T_i^S maps ontological elements to possible enrichment points within the learning objects.

$$T_i^S = \{\langle c, d \rangle \mid d \in T_i^{L+SYN}, \langle c, d \rangle \in T^{O+SYN}\}. \tag{6.9}$$

T_i^S is a set of tuples $\langle c, d \rangle$ where d is a concept of the educational content and c is the associated ontological element. The set of all d is D (Equation 6.10).

$$D = \{d \mid \langle c, d \rangle \in T_i^S\}. \tag{6.10}$$

The algorithm's next part is the selection of identified enrichment points $D' \subseteq D$ within the learning object. Possible implementations can limit the set of enrichment points, for example by the selection of the first appearance of the enrichment points. The semantic relevance is proposed as the key factor. For its determination several approaches can be (combined) implemented: (a) choose those enrichment points that are most relevant based on certain mining algorithms, (b) choose those enrichments points that are most relevant based on the semantic relevance according to the metadata of the LO, (c) choose those enrichment points that are most relevant based on the ontological relevance of the associated ontological elements. For the last option, certain ontology metrics can be useful (see Section 6.1.4) [Mencke et al., 2008c].

On the basis of the set RO (Equation 6.12) containing all ontological elements related to the selected enrichment points, and the Semantic Window approaches described in sections 6.1.2 and 6.1.3, an additional set of ontological elements can be computed. It will be referred to as W.

$$f^{onto} : \text{String} \mapsto \{\text{OntologicalElement}, \dots\}. \tag{6.11}$$

$$RO = \bigcup_{d \in D'} f^{onto}(d). \tag{6.12}$$

The next step determines the amount of additional information EC that is used to enrich the educational content (Formula 6.13 and Equation 6.14).

$$f^{enrich} : \text{OntologicalElement} \mapsto \{\text{EnrichmentContent}, \dots\}. \tag{6.13}$$

$$EC = \bigcup_{r \in RO \cup W} f^{enrich}(r). \tag{6.14}$$

Other approaches as well as the cost-based or distance-based 'Semantic Windows' described in the next sections, relate to classic adaptation algorithms for e-Learning and may use additional domain ontologies, specification ontologies and user models.

The presentation is not part of the algorithm above, but results in the highlighting of all selected $d \in D'$ and the selective displaying the prepared enrichment content $EC' \subseteq EC$.

6.1.2 Cost-Based Semantic Windows

For the enrichment algorithm the concept of a 'Semantic Window' is defined. This term describes a set of elements of a given ontology within a certain multi-dimensional distance. Dimensions for its definition are related to the concepts of an ontology as well as to the datatype properties. Furthermore, instances and taxonomic as well as non-taxonomic relations are taken into consideration [Mencke et al., 2008h].

The function f^{cost} returns the "cost" of the transition between two nodes, given their types as well as the sequence of already accepted nodes (Formula 6.15). For the combinations of ontological elements' types, between which no transition is possible, the cost function is assumed to return the positive infinity.

$$f^{cost} : \text{Type}, \text{Type}, \langle \text{Node}, \dots \rangle \mapsto \text{Integer}. \tag{6.15}$$

Function f^{type} returns the type of a given ontological element (a member of the enumeration 6.17). New types of ontological elements can be introduced by splitting the sets of ontological elements of a particular type on the basis of some constraints (subclassing). The domain of f^{cost} for these new types obviously cannot be broader as for the original type.

$$f^{type} : \text{OntologicalElement} \mapsto \text{Type}. \tag{6.16}$$

$$\begin{aligned} Type \in \{&\text{ParentConcept, ParentObjectProperty,} \\ &\text{ChildConcept, ChildObjectProperty,} \\ &\text{Concept, ObjectProperty,} \\ &\text{DatatypeProperty, ConceptInstance,} \\ &\text{ObjectPropertyInstance,} \\ &\text{DatatypePropertyInstance}\}. \end{aligned} \tag{6.17}$$

Elements of a tuple $\langle n_0, \dots, n_m \rangle$, $n_i \in O$, $m \in \mathbb{N}$ are included to the Semantic Window, if n_0 is the enrichment point of the enrichment and inequality 6.18 resolves to true, where A is the cost restricter ("the size of the Semantic Window").

$$\sum_{i=0}^{m-1} f^{cost}(f^{type}(n_i), f^{type}(n_{i+1}), \langle n_0, \ldots, n_i \rangle) \leq A. \tag{6.18}$$

In Figure 6.4, an example for the Semantic Window is given. Concept C_6 is the enrichment point around which the Semantic Window is created. For the sake of simplicity datatype properties are not taken into consideration. The cost function f^{cost} is given in Table 6.1 and the maximum cost is $A = 3$. Filled circles represent concepts, filled squares represent instances and filled diamonds on arrows represent object properties, all being located within the range of the Semantic Window around C_6.

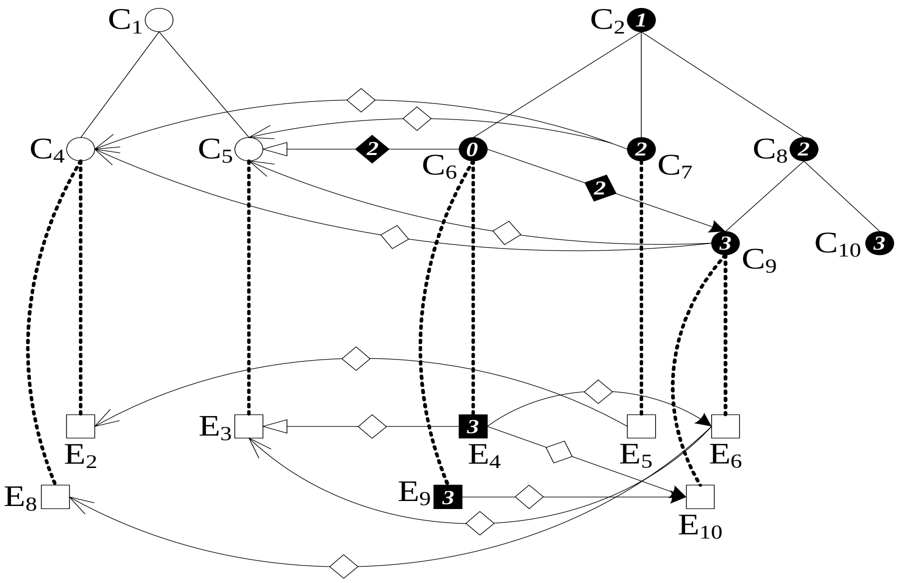


Figure 6.4: Example of a Semantic Window with Enrichment Point C_6, Cost Restricter $A = 3$ and the Transition Costs Given in Table 6.1

	Parent concept / object property	Child concept / object property	Concept	Object property	Datatype property	Concept instance	Object property instance	Datatype property instance
Concept	1	1	∞	2	2	3	∞	∞
Object property	1	1	2	∞	∞	∞	3	∞
Datatype property	∞	∞	2	∞	∞	∞	∞	3
Concept instance	∞	∞	3	∞	∞	∞	2	2
Object property instance	∞	∞	∞	3	∞	2	∞	∞
Datatype property instance	∞	∞	∞	∞	3	2	∞	∞

Table 6.1: Example of Transition Costs Between Ontological Elements [Mencke et al., 2008h].

Based on the developed architecture, a prototype was implemented. To proof the applicability of the proposed approach a Web-based example was chosen for the enrichment of Web pages using semantic information from an ontology (see Figure 6.5).

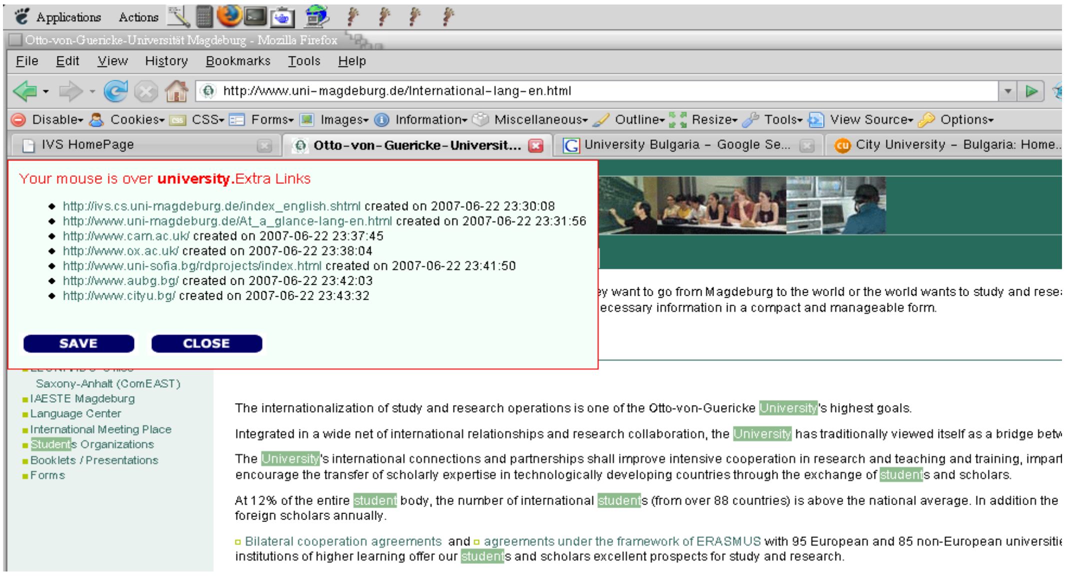


Figure 6.5: Screenshot of an Enriched Web Page [Mencke et al., 2008h]

6.1.3 Distance-Based Semantic Windows

Next to the cost-based approach described above, a distance-based solution for Semantic Windows is possible [Mencke et al., 2008j].

For the further detailing, the description starts with a specialized redefinition of an ontology $O = (C, R, D, I)$, where C is the set of ontological concepts following a taxonomic structure, $R = R_{tax} \cup R_{ntax}$ is the set of object properties/relations taxonomically and non-taxonomically relating two concepts $R_{ij}(C_i, C_j)$ and D is the set of datatype properties/attributes of the ontology. I is the set of instances. An ontological component of each of these types can be the enrichment point for the Semantic Window. This is a generalization of the algorithm of the Section 6.1.1.2 because also taxonomic object properties are taken into account now. From this four different aspects, the dimensions of the Semantic Window, can be derived.

- Concept view
- Datatype property view
- Object property view
- Instance view

For each of the four views, distance measures are defined for the existing dimensions. A help function is $f^{niv}(C_i)$ describing the level of the concept according to its taxonomic level with $f^{niv}(C_{root}) = 0$ (Formula 6.19). Function $f^{parent}(C_i, C_j)$ delivers back the first more abstract concept shared by C_i and C_j, if it exists and is connected to them only via $R \in R_{tax}$(Formula 6.20). $f^{tax}(C_i, C_j)$ (Formula 6.21) and $f^{ntax}(C_i, C_j)$ (Formula 6.22) determine the length of the taxonomic or non-taxonomic path of object properties from C_i to C_j (the result is -1, if there does not exist such a path).

$$f^{niv} : \text{Concept} \mapsto \text{Integer}. \tag{6.19}$$

$$f^{parent} : \langle \text{Concept, Concept} \rangle \mapsto \text{Concept}. \tag{6.20}$$

$$f^{tax} : \langle \text{Concept, Concept} \rangle \mapsto \text{Integer}. \tag{6.21}$$

$$f^{ntax} : \langle \text{Concept, Concept} \rangle \mapsto \text{Integer}. \tag{6.22}$$

f^{tax} and f^{ntax} can be realised as described in the equations 6.23 and 6.24.

$$f^{tax}(C_i, C_j) = \begin{cases} 0 & \text{if } C_i \equiv C_j, \\ 1 & \text{if } |f^{niv}(C_i) - f^{niv}(C_j)| = 1 \\ & \wedge R_{ij}(C_i, C_j) \in R_{tax}, \\ f^{tax}(C_i, C_k) + 1 & \text{if } f^{tax}(C_i, C_k) = n \wedge f^{tax}(C_k, C_j) = 1, \\ -1 & \text{otherwise.} \end{cases} \tag{6.23}$$

$$f^{ntax}(C_i, C_j) = \begin{cases} 0 & \text{if } C_j \equiv C_j, \\ 1 & \text{if } R_{ij}(C_i, C_j) \in R_{ntax}, \\ f^{ntax}(C_i, C_k) + 1 & \text{if } f^{ntax}(C_i, C_k) = n \wedge f^{ntax}(C_k, C_j) = 1, \\ -1 & \text{otherwise.} \end{cases} \tag{6.24}$$

In the following the dimensions of the concept view are described. Detailed formulas for the object property dimension, the datatype property dimension as well as the instance dimension are presented in Appendix A.

6.1.3.1 Concept Dimensions from the Concept View

The dimensions of the distance related to the ontology's concepts having a concept as the focusing point are defined in equations 6.25 to 6.28. The single distance measures relate to the abstraction dimension distance c^{abs}, to the specialization dimension distance c^{spec}, to the sibling dimension distance c^{sib} and to the non-taxonomic dimension distance c^{ntax}. They measure the distance between the focusing point concept C_F and another concept C_j of the ontology.

$$c^{abs}(C_F, C_j) = f^{niv}(C_F) - f^{niv}(C_j). \tag{6.25}$$

$$c^{spec}(C_F, C_j) = f^{niv}(C_j) - f^{niv}(C_F). \tag{6.26}$$

$$c^{sib}(C_F, C_j) = f^{niv}(C_F) - f^{niv}(f^{parent}(C_F, C_j)). \tag{6.27}$$

$$c^{ntax}(C_F, C_j) = f^{ntax}(C_F, C_j). \tag{6.28}$$

The equations above are restricted by: $C_F, C_i, C_j \in C$. Equation 6.25 is restricted by: $f^{niv}(C_F) > f^{niv}(C_j)$ and $f^{tax}(C_F, C_j) \neq -1$. Equation 6.26 is restricted by: $f^{niv}(C_F) < f^{niv}(C_j)$ and $f^{tax}(C_F, C_j) \neq -1$. Equation 6.27 is restricted by: $f^{niv}(C_F) = f^{niv}(C_j)$ and $f^{niv}(f^{parent}(C_F, C_j)) < f^{niv}(C_F)$.

6.1.3.2 Datatype Property Dimensions from the Concept View

The dimensions of the distance related to the ontology's datatype properties having a concept as the focusing point are defined in equations 6.29 to 6.32. The single distance measures relate to the abstraction dimension distance d^{abs}, to the specialization dimension distance d^{spec}, to the sibling dimension distance d^{sib} and to the non-taxonomic dimension distance d^{ntax}. They measure the distance between the focusing point concept C_F and a datatype property D_j of the ontology. $C(D_j)$ is the concept that a datatype property D_j belongs to.

$$d^{abs}(C_F, D_j) = f^{niv}(C_F) - f^{niv}(C(D_j)). \tag{6.29}$$

$$d^{spec}(C_F, D_j) = f^{niv}(C(D_j)) - f^{niv}(C_F). \tag{6.30}$$

$$d^{sib}(C_F, D_j) = f^{niv}(C_F) - f^{niv}(f^{parent}(C_F, C(D_j))). \tag{6.31}$$

$$d^{ntax}(C_F, D_j) = f^{ntax}(C_F, C(D_j)). \tag{6.32}$$

The equations above are restricted by: $C_F, C_i, C_j, C(D_j) \in C$ and $D_j \in D$. Equation 6.29 is restricted by: $f^{niv}(C_F) > f^{niv}(C(D_j))$ and $f^{tax}(C_F, C(D_j)) \neq -1$. Equation 6.30 is restricted by: $f^{niv}(C_F) < f^{niv}(C(D_j))$ and $f^{tax}(C_F, C(D_j)) \neq -1$. Equation 6.31 is restricted by: $f^{niv}(C_F) = f^{niv}(C(D_j))$ and $f^{niv}(f^{parent}(C_F, C(D_j))) < f^{niv}(C_F)$.

6.1.3.3 Object Property Dimensions from the Concept View

The dimensions of the distance related to the ontology's object properties having a concept as the focusing point are defined in equations 6.33 to 6.37. The single distance measures relate to the abstraction dimension distance r^{abs}, to the specialization dimension distance r^{spec} as well as to the (abstraction and specialization) sibling dimension distance $r^{sib^{abs}}$ and $r^{sib^{spec}}$. The non-taxonomic dimension distance is measured by r^{ntax}. They measure the distance between the focusing point concept C_F and an object property R_j of the ontology.

$$r^{abs}(C_F, R_j) = \begin{cases} 0 & \text{if } \nexists R_j(C_F, C_j), \\ 1 & \text{if } \exists R_j = R_{Fj}(C_F, C_j) \\ & \wedge f^{niv}(C_F) > f^{niv}(C_j), \\ r^{abs}(C_F, R_i) & \\ +1 & \text{if } r^{abs}(C_F, R_i) = n \\ & \wedge c^{abs}(C_i, C_j) = 1 \\ & \wedge R_j = R_{ij}(C_i, C_j) \\ & \wedge R_i = R_{hi}(C_h, C_i) \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) \\ & > f^{niv}(C_j). \end{cases} \tag{6.33}$$

$$r^{spec}(C_F, R_j) = \begin{cases} 0 & \text{if } \nexists R_j(C_F, C_j), \\ 1 & \exists R_j = R_{Fj}(C_F, C_j) \\ & \wedge f^{niv}(C_F) < f^{niv}(C_j), \\ r^{spec}(C_F, R_i) & \\ +1 & \text{if } r^{spec}(C_F, R_i) = n \\ & \wedge c^{spec}(C_i, C_j) = 1 \\ & \wedge R_j = R_{ij}(C_i, C_j) \\ & \wedge R_i = R_{hi}(C_h, C_i) \\ & \wedge f^{niv}(C_h) < f^{niv}(C_i) \\ & < f^{niv}(C_j). \end{cases} \tag{6.34}$$

$$\begin{aligned} r^{sib^{abs}}(C_F, R_j) = c^{sib}(C_F, C_h) | c^{abs}(C_h, C_i) = 1 \\ \wedge\, R_j = R_{ij}(C_i, C_j) \\ \wedge\, f^{tax}(C_h, C_i) \neq -1 \\ \wedge\, f^{niv}(C_h) > f^{niv}(C_i) > f^{niv}(C_j). \end{aligned} \tag{6.35}$$

$$\begin{aligned} r^{sib^{spec}}(C_F, R_j) = c^{sib}(C_F, C_h) | c^{spec}(C_h, C_i) = 1 \\ \wedge\, R_j = R_{ij}(C_i, C_j) \\ \wedge\, f^{tax}(C_h, C_i) \neq -1 \\ \wedge\, f^{niv}(C_h) < f^{niv}(C_i) < f^{niv}(C_j). \end{aligned} \tag{6.36}$$

$$r^{ntax}(C_F, R_j) = \begin{cases} 0 & \text{if } \nexists R_j(C_F, C_j), \\ 1 & \text{if } c^{ntax}(C_F, C_j) = 1 \\ & \wedge \exists R_j(C_F, C_j), \\ r^{ntax}(C_F, R_i) & \\ +1 & \text{if } \exists r^{ntax}(C_F, R_i) = \\ & f^{ntax}(C_F, C_i) = n \\ & \wedge R_i = R_{hi}(C_h, C_i) \\ & \wedge R_j = R_{ij}(C_i, C_j). \end{cases} \tag{6.37}$$

The equations above are restricted by $R_j, R_{Fj}, R_{ij} \in R$ and $C_F, C_h, C_i, C_j \in C$. Equations 6.33 to 6.36 are further restricted by $R_{Fj}, R_i, R_j \in R_{tax}$. For equation 6.37 the following restrictions apply: $R_i, R_j \in R_{ntax}$.

6.1.3.4 Instance Dimensions from the Concept View

The dimensions of the distance related to the ontology's instances having a concept as the focusing point are defined in equations 6.38 to 6.41. The single distance measures relate to the abstraction dimension distance i^{abs}, to the specialization dimension distance i^{spec} and to the sibling dimension distance i^{sib} as well as the non-taxonomic dimension distance is measured by i^{ntax}. They measure the distance between the focusing point concept C_F and an instance I_j of the ontology. $C(I_j)$ is the concept that an instance I_j is instantiated of.

$$i^{abs}(C_F, I_j) = f^{niv}(C_F) - f^{niv}(C(I_j)). \tag{6.38}$$

$$i^{spec}(C_F, I_j) = f^{niv}(C(I_j)) - f^{niv}(C_F). \tag{6.39}$$

$$i^{sib}(C_F, I_j) = f^{niv}(C_F) - f^{niv}(f^{parent}(C_F, C(I_j))). \tag{6.40}$$

$$i^{ntax}(C_F, I_j) = f^{ntax}(C_F, C(I_j)). \tag{6.41}$$

The equations above are restricted by: $C_F, C_i, C_j, C(I_j) \in C$ and $I_j \in I$. Equation 6.38 is restricted by: $f^{niv}(C_F) > f^{niv}(C(I_j))$ and $f^{tax}(C_F, C(I_j)) \neq -1$. Equation 6.39 is restricted by: $f^{niv}(C_F) < f^{niv}(C(I_j))$ and $f^{tax}(C_F, C(I_j)) \neq -1$. Equation 6.40 is restricted by: $f^{niv}(C_F) = f^{niv}(C(I_j))$ and $f^{niv}(f^{parent}(C_F, C(I_j))) < f^{niv}(C_F)$.

Every distance measure described above and in Appendix A delivers back -1, if the function's arguments are not appropriate according to the nature of the distance to be measured.

6.1.3.5 Size of the Semantic Window

Within a Semantic Window, from any ontological element's point of view, all distances as well as the ontological element being the focusing point are given and used to determine a set of ontological elements W containing all ontological elements those distance are smaller than the given ones. The distances are summarized in vectors as demonstrated below.

In the following, an example is sketched to show the usage of distances to determine a Semantic Window. A graphical representation of the result is shown in Figure 6.6. Filled circles represent concepts, filled squares represent instances and filled diamonds represent datatype properties, all being located within the Semantic Window.

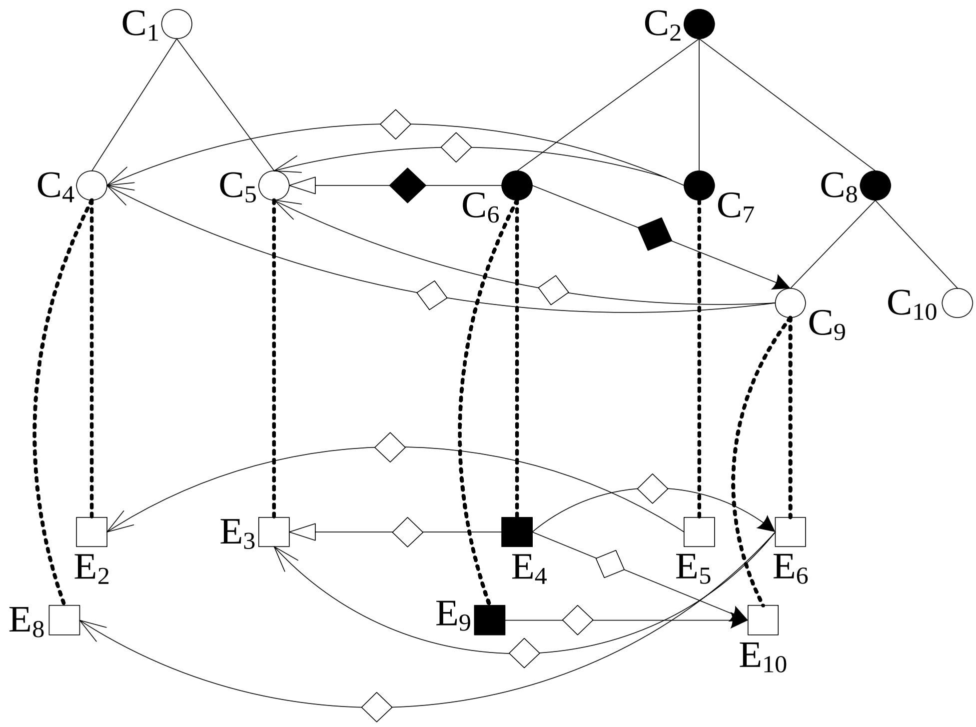


Figure 6.6: Example for a Distance-Based Semantic Window with C_6 as Focusing Point and the Defined Distances in Vectors 6.42 to 6.45 [Mencke et al., 2008j]

The focusing point is a concept and the concept distances are given in vector 6.42, datatype property distances in vector 6.43, object property distances in vector 6.44 and the instance distances are given in vector 6.45.

$$dist^C(C_6) = \begin{pmatrix} c^{abs} \\ c^{spec} \\ c^{sib} \\ c^{ntax} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{6.42}$$

$$dist^D(C_6) = \begin{pmatrix} d^{abs} \\ d^{spec} \\ d^{sib} \\ d^{ntax} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{6.43}$$

$$dist^R(C_6) = \begin{pmatrix} r^{abs} \\ r^{spec} \\ r^{sib^{abs}} \\ r^{sib^{spec}} \\ r^{ntax} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{6.44}$$

$$dist^I(C_6) = \begin{pmatrix} i^{abs} \\ i^{spec} \\ i^{sib} \\ i^{ntax} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{6.45}$$

With these definitions, a detailed overview about a distance-based description of Semantic Windows was presented. It is seen as an alternative to the already presented cost-based approach. One additional advantage is the possibility to define more specific Semantic Windows, like tori instead of spheres, to use an geometrical analogy. Further improvements are possible regarding transitive Semantic Windows. Figure 6.7 presents a screenshot of a tool that uses distance-based Semantic Windows for Web content enrichment (see Figure 6.8). Users can add new enrichment content in order to complete the available data sources and thereby collaborate on quality improvement.

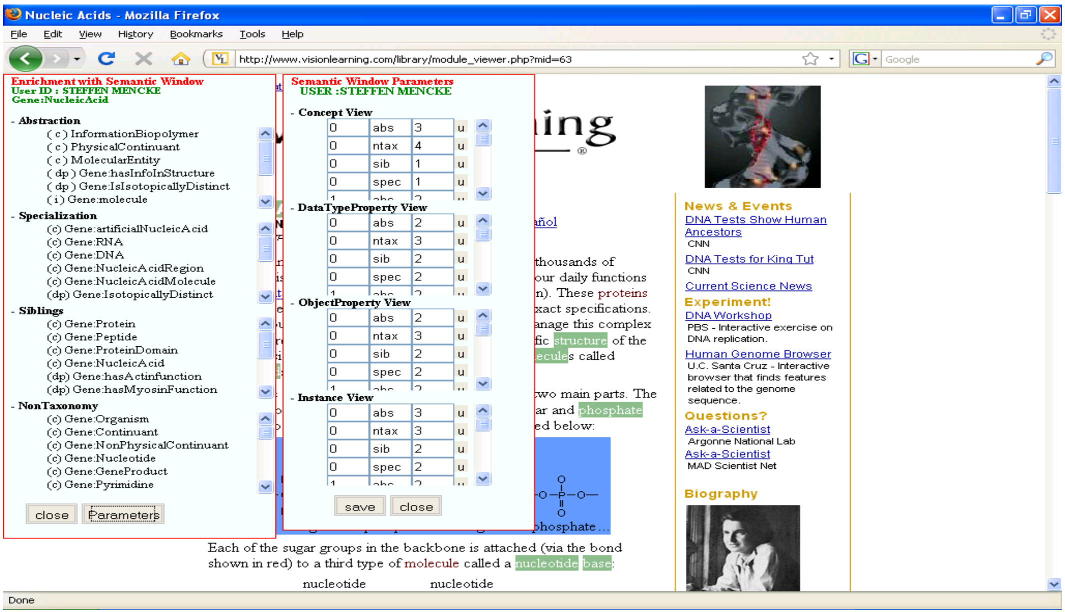


Figure 6.7: Distance-Based Semantic Windows for Content Enrichment

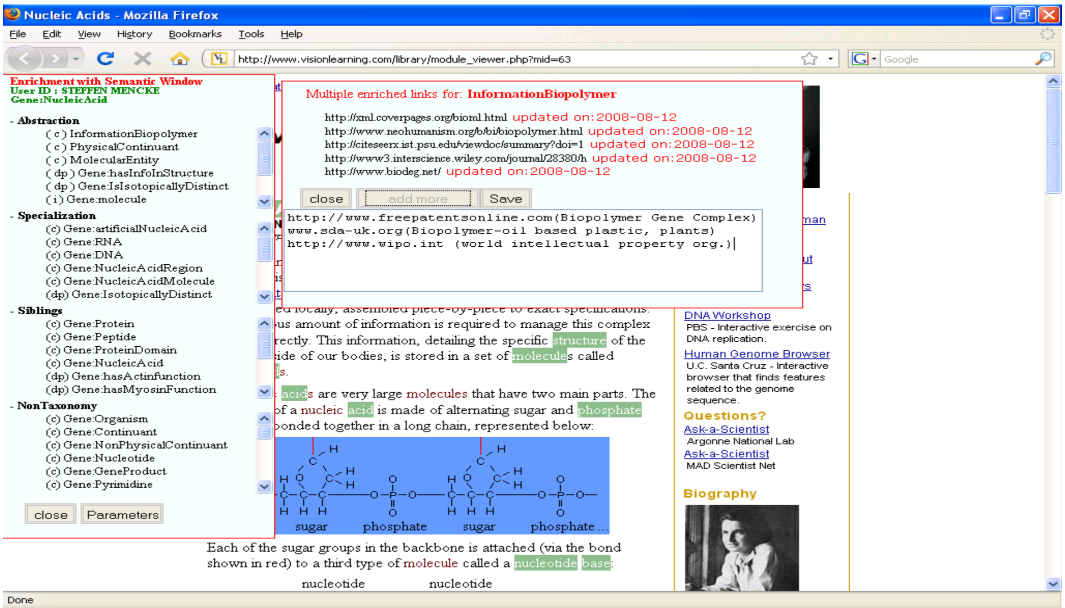


Figure 6.8: Enriched Web Page Based on Semantic Windows

6.1.4 Towards the Balance of Ontologies

Nowadays, measurement and assessment of artifacts within the area of software development are of high concern for industrial organizations as well as for scientific institutions. Ontologies as a fundamental concept of the Semantic Web as envisioned by Tim Berners-Lee. They form are the basis for intelligent next generation applications for the Web and other areas of interest.

The balance of ontology's is on higher interest because the usability and convertibility of ontologies is strongly related to the manner how the elements are arranged. These arrangements have a major impact on the set of ontological elements being within a Semantic Window as described above [Mencke et al., 2008c].

Ontologies cannot only be used to provide metrics for measurement [Kunz et al., 2006a], but they can be measured itself. Therefore, a brief description and categorization of ontology metrics is given with a focus on applicability regarding the balance of ontology's taking into account structure and knowledge related aspects. A Goal-Question-Metric-based procedure was used.

Afterwards, initial approaches for the determination of the ontologies' balance are presented.

6.1.4.1 Classification of Ontology Metrics

For the purpose of measuring, the Goal Question Metric (GQM) approach [Basili and Weiss, 1984] helps in discovering adequate measurement attempts and goals. Initially, it requires the definition of precise goals to form the foundation for the nomination of questions suitable for discussing issues from different viewpoints. Finally, metrics qualified for answering these questions become apparent. Afterwards, a tailored measurement as well as its evaluation concerning goal attainment is possible.

The quantification of metrics attributes is separated into two different areas being divided into four major scopes. These areas are scheme-related and content-related, respectively.

At first, it is analyzed which metrics are used to measure the content of ontologies. One can identify two major goals in this area: the granularity of the enclosed content and the coverage of the content (see Figure 6.9).

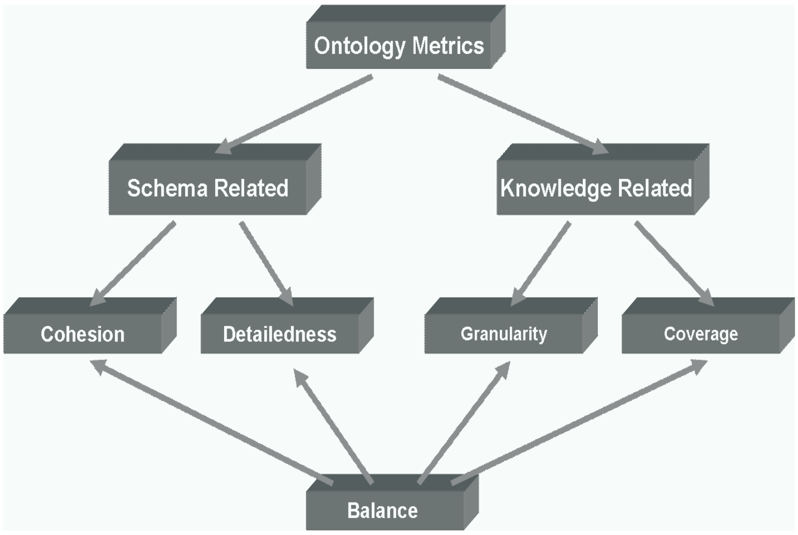


Figure 6.9: Genealogy of Ontology Metrics [Mencke et al., 2008c]

To achieve these goals, the mentioned GQM approach is used to identified the content granularity and content coverage metrics as shown in Table 6.2 and Table 6.3. In the second area (the structures of ontologies) two goals were identified as well.

NAME OF METRIC	FORMULA	DESCRIPTION
Average Population (Pop) [Tartir et al., 2005]	$Pop = \|I\| / \|C\|$ with $\|I\|$ as the number of instances in the knowledge base and $\|C\|$ as the number of classes defined in the ontology.	This metric may serve as an indication of the number of instances compared to the number of classes.
Cohesion (COH) [Tartir et al., 2005]	$COH = \|SCC\|$ as the number of separate connected components	This indicates what areas need more instances in order to enable instances to be more closely connected.

NAME OF METRIC	FORMULA	DESCRIPTION
Connectivity (Cn) [Tartir et al., 2005]	$Cn = \|I_j, P(I_i, I_j) \wedge I_j \in C_i(I)\|$ as the number of instances of other classes that are connected to instances of that class (I_j)	It is an indication of the number of relationships instances of each class to other instances.

Table 6.2: Chosen Content Granularity Related Metrics [Mencke et al., 2008c].

NAME OF METRIC	FORMULA	DESCRIPTION
Class Richness (CR) [Tartir et al., 2005]	$CR = \|C^{'}\|/\|C\|$ with $\|C^{'}\|$ as the number of classes used in the base and $\|C\|$ as the number of classes defined in the ontology.	Describes how instances are distributed across classes.
Density Measure (DEM) [Alani and Brewster, 2005]	$DEM = \frac{1}{n}\sum_{i=1}^{n} w_i\|C_{Sub}\| + w_i\|C_{Sup}\| + w_i\|C_S\| + w_i\|I\| + w_i\|P\|$ with C_{Sub} as the number of a class' subclasses, C_{Sup} as the number of its superclasses, C_S as the number of its siblings, I as the number of its instances, P as the number of its relations, and w_i as a weight factor.	This metric indicates how well a given concept is defined in the ontology.
Relationship Richness (RRC) [Tartir et al., 2005]	$RR_C = \|P(I_i, I_j), I_i \in C_i(I)\|/\|P(C_i, C_j)\|$ with $(P(I_i, I_j))$ as the number of relationships that are being used by instances I_i that belong to C_i, and $(P(C_i, C_j))$ as the number of relationships that are defined for C_i at the schema level.	Identifies how well the extraction process performed in the utilization of information is defined at the schema level.

NAME OF METRIC	FORMULA	DESCRIPTION
Importance (IMP) [Tartir et al., 2005]	$IMP = \|C_i(I)\|/\|I\|$, with $\|C_i(I)\|$ as the number of instances that belong to the subtree rooted at C_i in the knowledge base, and $\|I\|$ as the number of instances in the knowledge base.	It is not an exact measure, but it can give a clear idea on what parts of the ontology are considered focal and what parts are on the edges.
Fullness (F) [Tartir et al., 2005]	$F = \|C_i(I)\|/\|C'_i(I)\|$, with $C_i(I)$ as the actual number of instances that belong to the subtree rooted at C_i, and $C'_i(I)$ as the expected number of instances that belong to the subtree rooted at C_i.	Describes how well was the data extracted with respect to the expected number of instances of each class.

Table 6.3: Chosen Content Coverage Related Metrics [Mencke et al., 2008c].

An aspect which is well described by existing metrics, is the structure of ontology and identified major scopes are the level of detail and cohesion. Especially, a scheme-based level of detail is important to evaluate the ontology, because it is fundamental to achieve content granularity (see Table 6.4). Having introduced this concept as an indicator for information distribution, another one is needed to describe coherence of distinct classes. It quantifies relation-based information within ontologies. Chosen metrics are presented in Table 6.5.

NAME OF METRIC	FORMULA	DESCRIPTION
Attribute Richness (AR) [Tartir et al., 2005], [Buitelaar et al., 2004]	$AR = \|A\|/\|C\|$, with $\|A\|$ as the number of attributes of all classes and $\|C\|$ as the number of classes.	This metric can indicate the quality of ontology design.
Centrality Measure (CEM) [Alani and Brewster, 2005]	$CEM = \frac{1}{n}\sum_{i=1}^{n} 1 - \|\frac{D[C]-\frac{H[C]}{2}}{\frac{H[C]}{2}}\|$, with $H[C]$ as the longest path that contains the class C from root of the branch to its bottom node, and $D[C]$ as the length of the path to C from the root.	For this metric it is assumed that mid-leveled classes tend to be more representative for an ontology due to more details and prototypical character.

NAME OF METRIC	FORMULA	DESCRIPTION
Number of Leaf Nodes (NoL) [Yao et al., 2005]	$NoL = \|C_j\|$, with $1 \leq j \leq n$ and C_j leaf class of the ontology.	A leaf class has no semantic subclass explicitly defined in the ontology.
Number of Root Classes (NoR) [Yao et al., 2005]	$NoR = \|C_j\|$, with $1 \leq j \leq n$ and C_j root class of the ontology.	A root class in an ontology means the class has no semantic super class explicitly defined in the ontology.
Average Depth of Inheritance Tree of Leaf Nodes (ADIT-LN) [Yao et al., 2005]	$ADIT - LN = \|D_j/n\|$, with $1 \leq j \leq n$ and D_j as total number of nodes on j^{th} path.	This metric describes the sum of depths of all paths divided by the total number of paths.

Table 6.4: Chosen Scheme-Based Level of Detail Related Metrics [Mencke et al., 2008c].

NAME OF METRIC	FORMULA	DESCRIPTION
Relationship Strength (RSS^O) [Wu et al., 2006]	$RSS^0(P,Q) = max_{u \in cl(P), v \in cl(Q)}\{RSS(u,v)\}$, with P and Q as the classes of interest and $cl(P)$, $cl(Q)$ as the sets of all concepts assigned to the classes P and Q, and $RSS(C_1,C_2) = \frac{maxDepth}{maxDepth+\gamma} * \frac{\alpha}{\alpha+\beta}$	Describes strength of relationship between two classes.
Relationship Richness (RR) [Tartir et al., 2005]	$RR = \frac{\|P\|}{\|SC\|} + \|P\|$, with P as the number of relationships, and SC as the number of sub-classes (= inheritance relationships).	Describes the diversity of relations and placement of relations in the ontology.
Inheritance Richness (IR_C) [Tartir et al., 2005]	$IR_C = \frac{\sum_{C_i \in C'} \|H^C(C_1,C_i)\|}{\|C'\|}$, with $\|H^C(C_1,C_i)\|$ as the number of subclasses C_1 of a class C_i, and $\|C'\|$ as the number of nodes in the subtree.	Describes the distribution of information in the current class sub-tree per class.

Table 6.5: Chosen Scheme Cohesion Related Metrics [Mencke et al., 2008c].

It is possible to evaluate the structure of ontologies by taking into account these two goals. Other approaches like scheme completeness and scheme granularity are not useful because of different reasons. So, scheme completeness, when creating a completely new ontology, is a semantic question which can not be answered by using metrics. One can target this question by the empirical analysis of ontology usage by taking into account other domain-related ontologies. The question whether an ontology is complete or not can not be finally answered by using the ontology itself. The analysis in this direction depends very much on a subjective point of view.

6.1.4.2 Metrics for the Balance of Ontologies

Having presented four starting points for the evaluation of ontologies, in the following another general aspect concerning the structure and the content of ontologies is introduced: the balance of a distinct ontology (see Figure 6.9). Existing measures in this area (for example Average Depth, Average Breadth) can not completely quantify ontology aspects concerning the balance. The balance of ontology is important because it is to be used as an indicator how good the ontology is built up and one can identify anomalies by analyzing the balance.

However, research efforts in this area are very rare and a complete framework for balancing ontologies is missing. In the following, initial instruments for quantifying ontologies concerning the balance are presented.

For this, different general aspects exist that can be helpful [Mencke et al., 2008c].

- Classes:
  - Equal number of subclass in equal level of abstraction $|C_{Sub}|_L^{C_i} \approx |C_{Sub}|_L^{C_j}$ with $i, j = 1, \dots, n \wedge i \neq j$
  - Equal number of subclass in different subtrees $|C_{Sub_k}|^{C_i} \approx |C_{Sub_l}|^{C_i}$ with $i, k, l = 1, \dots, n$
- Relations:
  - Equal number of relations in equal level of abstraction $|P|_L^{C_i} \approx |P|_L^{C_j}$ with $i, j = 1, \dots, n \wedge i \neq j$
  - Equal number of relations in different subtrees $|P_{C_{Sub_k}}|^{C_i} \approx |P_{C_{Sub_l}}|^{C_i}$ with $i, k, l = 1, \dots, n$
- Attributes:
  - Equal number of attributes in different concepts in equal level of abstraction $|A|_L^{C_i} \approx |A|_L^{C_j}$ with $i, j = 1, \dots, n \wedge i \neq j$
  - Equal number of attributes in different subtrees $|A_{C_{Sub_k}}|^{C_i} \approx |A_{C_{Sub_l}}|^{C_i}$ with $i, k, l = 1, \dots, n$
- Instances:
  - Equal number of instances of different concepts in equal level of abstraction $|I|_L^{C_i} \approx |I|_L^{C_j}$ with $i, j = 1, \dots, n \wedge i \neq j$
  - Equal number of instances in different subtrees $|I_{C_{Sub_k}}|^{C_i} \approx |I_{C_{Sub_l}}|^{C_i}$ with $i, k, l = 1, \dots, n$
- Subtrees:
  - Equal depth of each subtree $|DIT_{C_{Sub_k}}|^{C_i} \approx |DIT_{C_{Sub_l}}|^{C_i}$ with $i, k, l = 1, \dots, n$

With this set of described formulas as well as the ones described in 6.1.3 it is possible to define first knock-out criteria for balanced ontologies:

- An ontology which contains not a single pair of leaf nodes with no sibling distance can not be balanced
- If every subtree of the root node has a different maximal abstraction dimension the root can not be balanced
- Two concepts having a sibling distance must have the same specialization distance to their leafs

The presented approach is a first analysis of the targeted problem of missing balance metrics for ontologies. The mentioned numerous aspects need to be integrated in a set of formulas. Due to manifold characteristics of the described starting points, one has to do fundamental research about the mathematical base to map the existing complexity of the problem to certain metrics formulas. Knock-out criteria can be a first starting point, but it is not sufficient and quality models with distinct measures are desirable.

Related research should follow e.g. the following ideas:

- Gravity-related approach:
  - Identification of a center of gravity
  - Measuring c^{abs}, c^{spec}, c^{sib} and c^{ntax} to the border concepts of the ontology (roots, leafs, ...) 6.1.3
  - Ontology is balanced, if $c^{abs} \approx c^{spec} \approx c^{sib} \approx c^{ntax}$
  - Extension towards multiple centers of gravity
- Weighted graphs approach:
  - Determine a weight W_{c_i} for every node of the ontology's graph representation based on instances' size, instances' number, concept's relations and attributes, etc.
  - Ontology is balanced if (a) every node C_i has a similar weight or (b) all nodes on the same abstraction level have a similar weight.

6.1.4.3 Application Areas

Measuring just because it is possible can not be an intention. In the following, some initial ideas for using ontology metrics in certain application areas are presented. Within e-Learning, there exist many aspects to survey in association with ontologies as there are for example the creation of courses or measuring learning effort, e.g.:

- The determination of the semantic similarity between an ontology describing the domain to be learned and an ontology created by the learner(s) during the learning process is an approach to measure the standard of knowledge at a discrete point in time. By repetition the learning progress of the person/community that built up the second ontology can be analyzed for multipurpose reasons.
- Measuring the complexity of evolving ontologies during a learning effort or an examination can help to identify concepts that were learned very well or were not yet

learned.
- The creation of tests and exercises based on ontologies will lead to automatic determination of the level of difficulty, respectively of the complexity of the question and the expected answer based on the ontology complexity.
- Identifying matching concepts in ontologies to automatically generate courses described by ontologies is another option.
- Another usage for a similarity measure can be the description of course content depending on a domain ontology.

Agent technology is another very interesting application area. Ontology metrics are expected to be extremely useful for several aspects, e.g.:

- An agent's functionality can be characterized by analyzing the used communication ontology.
- It becomes also possible to identify a useful separation of functionalities and evolving communication based on an ontology containing a service description. Such an approach is useful to automatically identify the mapping of functionalities to agents as postulated in [Kernchen et al., 2007d] and [Kernchen and Dumke, 2007d].
- The balancing of workload becomes possible, if the work is effort-driven distributed based on an ontology.

6.1.5 Ontology-Based Generic Learning Path Recommendations

The content of certain presentation systems as well as the instruction to be delivered by e-Learning systems quite often is very static. In the following, a generic approach for individualized paths through such content, exemplified for the domain of e-Learning, is presented [Mencke et al., 2008f].

Instruction consists of multi-medially coded information like for example texts, pictures and video/audio and is delivered by a Learning Management Systems (LMS). Practical considerations as well as research identified the necessity for the further description of the so called Learning Objects (LO) (2.3.3.5). Technically they are reusable, media-independent collections of information used as a modular building block for e-Learning content. The IEEE Learning Objects Metadata standard (LOM) [IEEE Learning Technology Standards Committee, 2003] is one option to provide additional data about the LO [Mencke and Dumke, 2007a].

These Learning Objects and their metadata are the basis for the most modern e-Learning systems. Their sequencing in courses is a process that actually is rarely performed automatically. Technical support exists to describe the structure of the e-Learning courses as described in Section 2.4.3.

They all have one aspect in common: they lack from a consistent support of didactical expertise. The author of the course either simply adapts his former face-to-face course or integrates his personal ideas about the courses' didactical structure. Of course there are didactical experts, but their expertise is not available for every course author. In

[Mencke and Dumke, 2007d] an approach is presented to overcome this gap.

With Semantic Windows presented above and ontology-based process models as described in 5.1.2 it is possible to make individual recommendations about the next steps of a learning path through an e-Learning course. The developed algorithm computes recommendations about what to learn next, based on individual and automatically detected learning style preferences as well as learning strategies of classified peer learners.

6.1.5.1 3-Level Ontology Framework

The generic next-learning-step recommendations for individualized learning paths based on individual learning styles needs a fundament of three ontologies.

The three-level ontology framework consists of a domain ontology describing the domain of interest, a LO ontology describing the structure and metadata of available Learning Objects and a didactical ontology describing an e-Learning course structure.

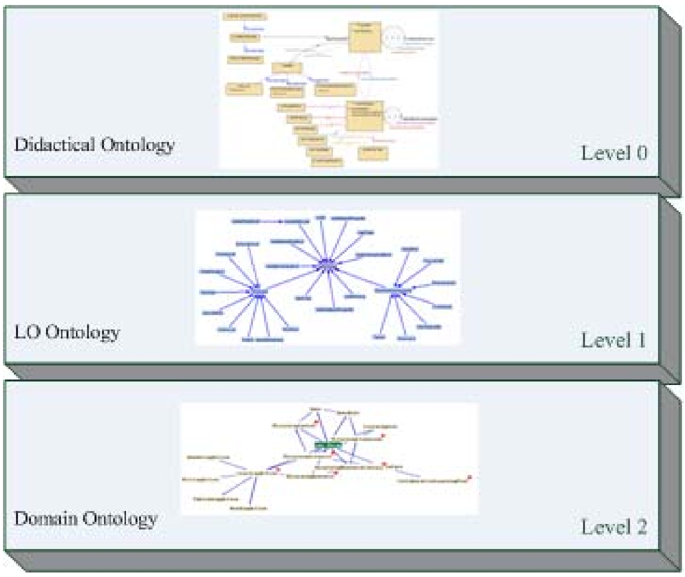


Figure 6.10: 3-Level Ontology Framework [Mencke et al., 2008f]

The basis ontology is a domain ontology and intended to be generated according to classic, taxonomy-based ontology creation methodologies (as e.g. described in [Uschold and Grüninger, 1996], [Swartout et al., 1996], [Staab et al., 2001], [Gómez-Pérez et al., 2004]).

The LO ontology describes the available Learning Objects and their metadata. A major aspect is the description of their semantic related to the domain ontology. So, LOs provide information about a subset of their semantics as defined within the domain ontology. Thereby, the coverage of the LOs' semantic may overlap or even identical for example because of different LOs about the same topic, but varying for example in terms of difficulty, media type or instructional objective. For the targeted purpose an existing ontology ([Hartonas, 2008]) was extended.

The didactical ontology, previously presented in [Mencke and Dumke, 2007d], can be simplified defined as a Graph $D = \{LU, CON\}$ of learning units (LU) and conditions

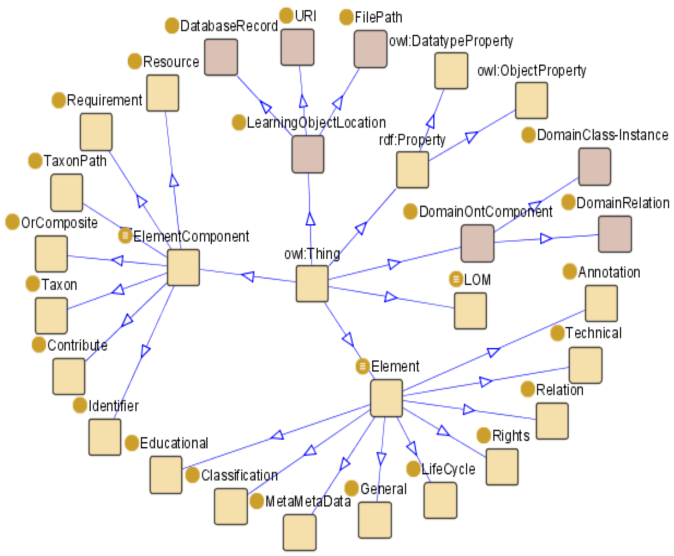


Figure 6.11: Learning Object Ontology (extended from [Hartonas, 2008])

(CON). Each LU contains an ordered list of LO's, but it is no LO itself, because all objects on this complexity level have decreased LO-related characteristics as: reusability, interoperability, atomicity, availability, etc.

The relations between these ontologies can be summarized as follows. The semantic of the domain ontology is covered by a set of Learning Objects that themselves are further described by a LO ontology. These LO are aggregated to more complex learning units, which are the basis for the description of the e-Learning course structure. This structure supports multiple learning paths by the definition of several conditions to decide about the next step.

To support an individual generic recommendation of a suitable next step (so far each next step is predefined according to the didactical expertise that is stored within the didactical ontology), it is necessary to identify appropriate preferences to make automatic decisions. Figure 6.12 presents an example of the 3-Level Ontology Framework containing a didactical ontology about a Verification and Validation course with the related Learning Object metadata ontology and a domain ontology.

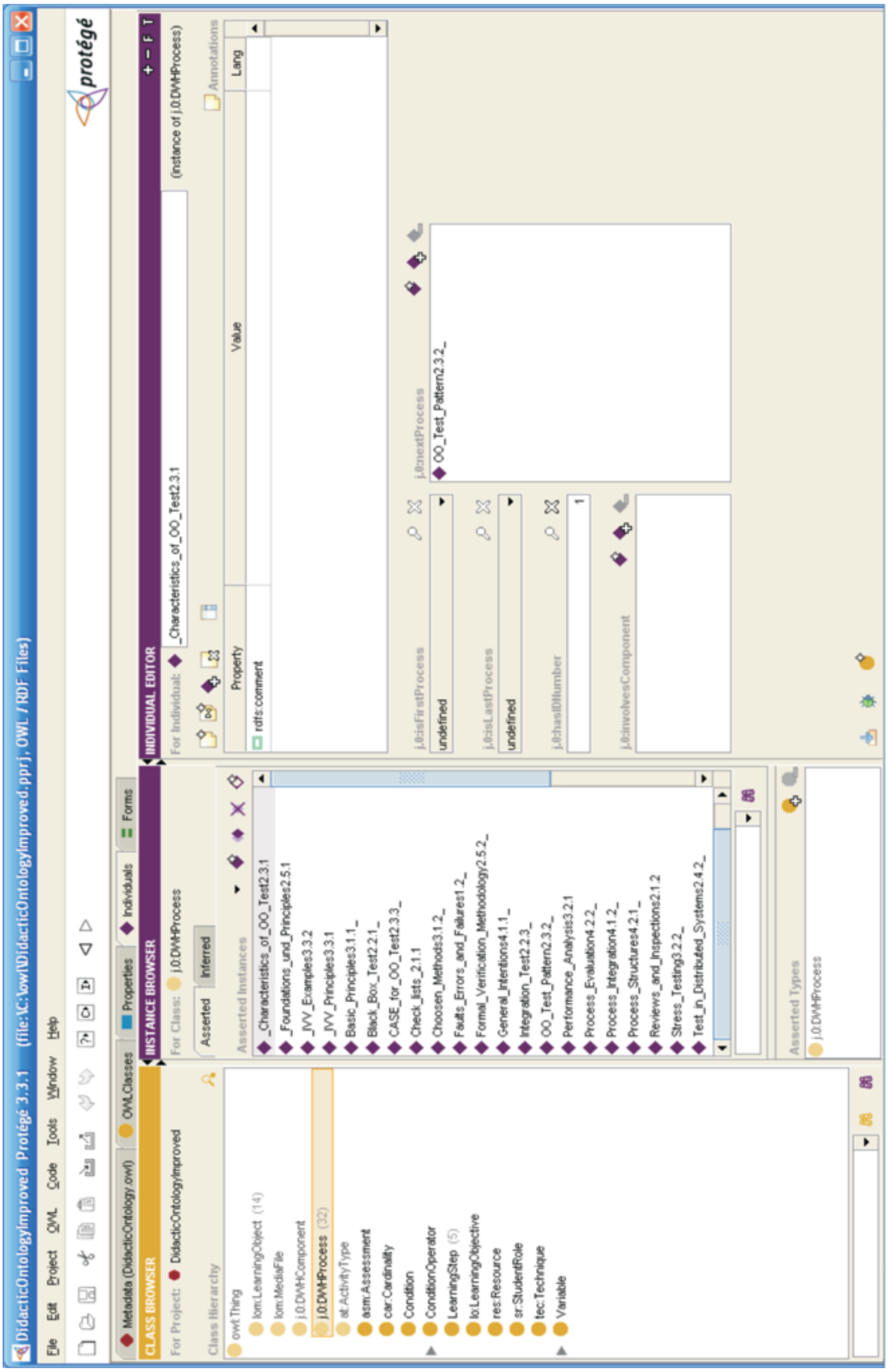


Figure 6.12: Ontology Example of the 3-Level Ontology Framework in Protégé

6.1.5.2 Determination of Individual Preferences

For the determination of individual preferences for this self-directed learning, a course structure is needed, that predefines as less restrictions as possible. Otherwise, those restrictions are measured instead of individual preferences. The ontology-based course structure as defined above is suitable for this purpose after the inclusion of an additional condition type to cover the selection of next-learning-steps according to preferences. This condition type is further explained in the next subsection.

In order to present an algorithm that identifies individual learning style preferences certain required formulas are defined at first. Other learning style preference (methods) can be integrated, but are not focus of the current selection.

A $choice$ is defined as a transition within a course structure from one learning unit to another one based on a user's choice about the condition type mentioned above (see Formula 6.46). Thereby, a learning unit is an instance of a learning unit concept of the didactical ontology consisting of several LO's that depict the knowledge of one or more ontological elements of the domain ontology. $f_i^{LO}(LU)$ determines the LO at position i of the learning unit LU, with $0 \leq i \leq n-1$ and n as the number of LO in LU (see Formula 6.47). In Formula 6.48, $f^{center}(LO)$ identifies the center concept of the LO's domain ontology coverage that is used as a representative of this LO.

$$f^{choice} : \text{LU} \mapsto \text{LU}. \tag{6.46}$$

$$f_i^{LO} : \text{LU} \mapsto \text{LO}. \tag{6.47}$$

$$f^{center} : \text{LO} \mapsto \text{concept}. \tag{6.48}$$

Formulas 6.46 and 6.47 are trivial to realize. For the implementation of Formula 6.48 the following solutions are proposed:

- Always take the most abstract concept
- Always take the most special concept
- Identification of a center concept (as suggested in Section 6.1.4.2) [Mencke et al., 2008c])
  - Measuring c^{abs}, c^{spec}, c^{sib}, and c^{ntax} to the border concepts of the subset of the ontology
  - Identify the concept, where $c^{abs} \approx c^{spec} \approx c^{sib} \approx c^{ntax}$

With these basics, the algorithm is sketched as follows.

Algorithm 1 Identification of Learning Style Preferences

procedure LEARNINGSTYLEIDENTIFICATION($\{choice\}$)
    $W = \{\langle LU_i, f^{choice}(LU_i)\rangle | \exists f^{choice}(LU_i)\}$
        ▷ As the set of all choices the learner made.
    $abs = 0$
    $spec = 0$
    $sib = 0$
    $ntax = 0$
    for all $w_i \in W$ **do**
        $c_a = f^{center}(f_0^{LO}(LU_i))| w_i = \langle LU_i, f^{choice}(LU_i)\rangle$
        $c_b = f^{center}(f_0^{LO}(f^{choice}(LU_i)))| w_i = \langle LU_i, f^{choice}(LU_i)\rangle$
        if $d^{abs}(c_a, c_b) > 0$ **then** $abs++$ **end if**
        if $d^{spec}(c_a, c_b) > 0$ **then** $spec++$ **end if**
        if $d^{sib}(c_a, c_b) = 1$ **then** $sib++$ **end if**
        if $d^{ntax}(c_a, c_b) > 0$ **then** $ntax++$ **end if**
    end for
end procedure

The algorithm focuses on the determination of learning styles but can easily be extended to identify other preference aspects, which can be for example classified based on the additional information from the *conditions*, further qualifications of the *learningsteps* of the didactical course ontology (see Figure 5.5) or based on LO metadata [Mencke et al., 2008f].

- Preferences of any type according to certain LO characteristics (metadata annotations):
  - Media type
  - Instructional object type
  - Learning objectives (Bloom's taxonomy)
  - Other (e.g. from [Mencke and Dumke, 2007d] or [Meder, 2006])
- Preferences identified and classified from the conditions between the learning steps, if a course structure based on the didactical ontologies of [Mencke and Dumke, 2007d] is used (closely relates to user model related as well as LMS runtime system related information)
- Preferences of any type regarding certain domain ontology concepts (semantic preferences) as well as structural elements; for example refinement after each $choice$ towards to learning goal specialized concepts are visited.

The values of abs, $spec$, sib and $ntax$ represent the user's semantic learning style preferences within self-directed learning. An extension of algorithm 1 can identify other preferences as stated above, too.

A high abs-value can be interpreted as a preference to learn from explanations about more abstract concepts rather than from examples or explanations about details. A high

$spec$-value can be interpreted as a preference to learn from detailed examples rather than from abstract explanations (depending on the nature of the ontology: whether it stores examples in instances or in specialized classes). A high sib-value can be interpreted as a preference to learn from explanations on the same level. It is like learning rather from the differences between certain knowledge aspects than from this aspect itself (again, it is depending on the nature of the ontology: what it stores within which elements). A high $ntax$-value can be interpreted as a preference to learn bordering knowledge, too.

6.1.5.3 Generic Learning Path Recommendations

Generic learning path recommendations as described below, base on the already introduced 3-level ontology framework and the didactical ontology-based course structure with a predefined learning unit, whose successful finishing by the learner marks the end of the course. The goal is to decide which learning unit to present in the next learning step [Mencke et al., 2008f]. For the recommendation certain options exist, as e.g.:

- Take information about the current learner
  - Advantage: best results that can be achieved
  - Problem: not always available
  - Problem: different course might cause different preferences which then cannot be available because normally the learner did not attend to them earlier
- Take information about successful other learners (advantage: only successful strategies are taken into account)
  - Of this course (advantage: adoption the current topic)
  - All available courses (advantage: more available data)
- Take information about a subset of other successful learners (advantage: only successful strategies of similar learners are taken into account)
  - With similar general preferences from subsection 6.1.5.2
  - Choice can base on common characteristics (determined from user model, advantage: more data about actual learner by identifying similar learners and classifying them)
    - Attendance to similar courses
    - Attendance to courses of the same author
    - Preferred learning style
    - ...

For the generic next-learning-step algorithm, certain formulas need to be predefined as well. One important is sketched in Formula 6.49, $f^{next}(c_i)$ delivers back the first/most appropriate learning unit that contains the given concept c_i.

$$f^{next} : \text{Concept} \mapsto \text{LearningUnit}. \tag{6.49}$$

$$f^{pref} : \langle \text{Concept, Integer} \rangle \mapsto \{\text{Concept}\}. \tag{6.50}$$

$$
\begin{aligned}
&f_{LU}^{dist}(LU_i, LU_j) = \\
&\quad min(d^{abs}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_0^{LO}(LU_j))) \\
&\quad (d^{spec}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_0^{LO}(LU_j))) \\
&\quad (d^{sib}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_0^{LO}(LU_j))) \\
&\quad (d^{ntax}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_0^{LO}(LU_j))) \\
&\quad (d^{dist}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_0^{LO}(LU_j)))| \\
&\quad d^{abs} > 0 \wedge d^{spec} > 0 \wedge d^{sib} > 0 \wedge d^{ntax} > 0 \\
&\quad \wedge d^{dist} > 0).
\end{aligned}
\tag{6.51}
$$

With these basics the algorithm is sketched as follows.

Algorithm 2 Generation of the Next-Learning-Step-Recommendation

1: **procedure** NEXTLEARNINGSTEP($\{LU_{ac}, CON\}$)
2: $\quad c_i = f^{center}(f_0^{LO}(LU_{ac}))$
3: $\quad$ **case** recommendation option **of**
4: $\quad\quad$ general: $C_j = f^{pref}(c_i, d^{abs})$
5: $\quad\quad$ example: $C_j = f^{pref}(c_i, d^{spec})$
6: $\quad\quad$ ...
7: $\quad$ **end case**
8: $\quad LU_j = f^{next}(c_j)|c_j \in C_j$
$\triangleright$ LU_j as the individual recommendation
9: $\quad LU_A = \{LU|f^{dist}(LU, LU_{ac}) = 1\}$
10: $\quad D_{LU}^{dist} = \{d_i\} = \{f_{LU}^{dist}(LU_a, LU_{ac})|LU_a \in LU_A\}$
11: $\quad LU_k = LU_l|f_{LU}^{dist}(LU_l, LU_{ac} = min(D_{LU}^{dist}))\wedge$
$\quad\quad LU_l \in LU_A \wedge \nexists w_i = \langle LU_k, c^{choice}(LU_k)\rangle$
$\triangleright$ LU_k as the system recommendation
12: **end procedure**

The algorithm above generates two recommendations. The first is based on the learner's identified learning style preferences (from line 2 to line 8). Line 2 is used to identify the location of the current LU within the domain ontology and the *case*-command identifies the next LU containing concepts following the recommendation option. Here, more extensions can be embedded for additional preferences.

The second part recommends a learning unit that is closer to the course's goal learning unit (from line 9 to line 11). Therefore, at first all next possible LU's are identified. Then, the LU_k is chosen that has the smallest distance to the LU_{goal}.

Now the learner can choose between LU_j and LU_k for an appropriate next learning step.

To support this individual preference-based learning, an extension of didactical course ontology with an extra variable type *preference* is proposed. The subclass *preference-Type* can have the following instances: *general* (learning from abstractions), *example* (learning from examples (instances, specializations)), *border* (learning from bordering knowledge (siblings)), *relation* (learning from related knowledge (non-taxonomic)), and *free* (complete free choice without any restriction).

This new variable type supports different learning path creation variants and can be used to, either following determined preferences or to determine possible weaknesses and choose a way that the learner tries to avoid, to make him learning the other ways, too.

Next to the generation of learning style based recommendations, minor guidance should be provided by the presentation of an additional *choice* to move closer to the learning goal. For this purpose, it is important to determine whether a learner is too far away from am optimal learning path and learns something off-topic.

A prototype was developed to show the usefulness of the developed approach. Based on a didactical course ontology, a Learning Object Metadata ontology as well as a domain ontology appropriate next learning steps are suggested according to the algorithms defined above. Figure 6.13 shows a screenshot of the developed tool.

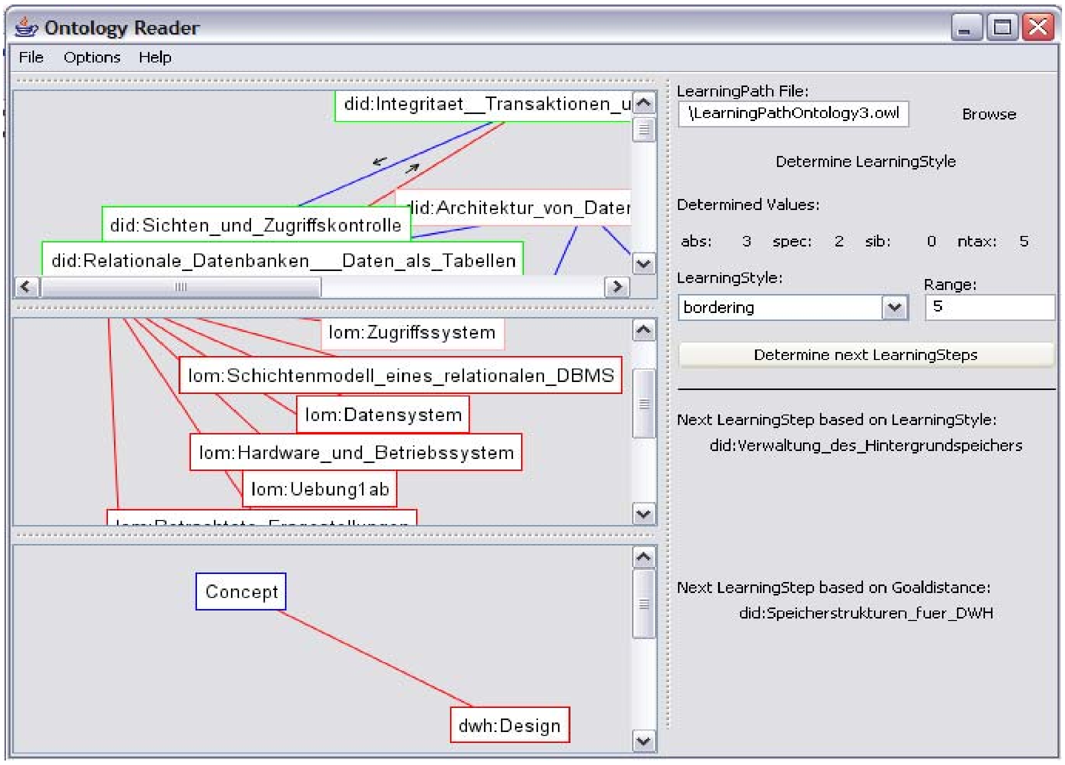


Figure 6.13: Tool for Generic Next Learning Step Recommendations

In conjunction with the already described functionality, graphical hints are used. Thereby, a green node indicates an already visited learning steps, while a red node was not yet visited. A blue edge describes the default learning path and a red edge indicate all learning paths having special conditions (see Section 5.1.2.1).

6.1.6 Supporting the Focus on the Essentials

If a learner tends to easily draw off his attention from his learning goal, then some of the learning units he chooses are too far away from an optimal learning path. That leads to the definition of a distance measure to determine whether a learner easily draws off his attention. In this subsection, such a metric is defined and can be used to define a *condition* within the didactical course ontology to guide him back to the main topics.

Therefore, at first a learning path is defined. According to the introduced 3-level ontology framework (see Figure 6.10) it is the path from the starting learning unit/Learning Object/ontology concept to the goal learning unit/Learning Object/ontology concept on the particular levels. For the didactical ontology, a learning path *LP* is a directed graph $LP = \{LU, W\}$ with *LU's* as nodes and *choices* as edges.

The learning path's length can be defined on every level of the 3-level ontology framework. On the didactical level, the length is determined by the function $f^{LUL}(LU_i, LU_j, \{W\})$ (Formula 6.52).

$$f^{LUL}(LU_i, LU_j, \{W\}) = \begin{cases} 0 & \text{if } LU_i \equiv LU_j, \\ 1 & \text{if } \langle LU_i, LU_j \rangle \in W, \\ f^{LUL}(LU_i, LU_k, \{W\}) + 1 & \text{if } f^{LUL}(LU_i, LU_k, \{W\}) = \\ & \text{n } \wedge \langle LU_k, LU_j \rangle \in W. \end{cases} \tag{6.52}$$

On the LO ontology level, the length of a learning path $f^{LOL}(LU_i, LO_j, \{W\})$ can be computed as the sum of all Learning Objects that exist within the LU of the learning path (Formula 6.54) using Formula 6.53.

$$f^{NOLO} : \text{LearningUnit} \mapsto \text{Integer}. \tag{6.53}$$

$$\begin{aligned} f^{LOL}(LU_i, LU_j, \{W\}) = \sum_{k=0}^{n-1} & f^{NOLO}(LU_k)| \\ & \langle LU_k, f^{choice}(LU_k) \rangle \in W \\ & \wedge n = f^{LUL}(LU_i, LU_j, \{W\}). \end{aligned} \tag{6.54}$$

The length of a learning path can be determined on the domain ontology level, too. It will be defined as the sum of the ontological distances between the concepts of the LOs of the learning units. As a non-specialized distance between concepts (d^{dist} delivers -1 if c_i and c_j are on different, not-connected subgraphs of the ontology). The distances computed by Formula 6.55 and Formula 6.56 are used to determine the length of the learning path $f^{CL}(LU_i, LU_j, \{W\})$ (Formula 6.57).

$$d^{dist}(C_i, C_j) = \begin{cases} 0 & \text{if } C_i \equiv C_j, \\ 1 & \text{if } \exists R(C_i, C_j), \\ d^{dist}(C_j, C_k) + 1 & \text{if } d^{dist}(C_i, C_k) = n \\ & \wedge \exists R(C_k, C_j). \end{cases} \tag{6.55}$$

$$
\begin{aligned}
d_{i,j}^{CL} = min(&d^{abs}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_{n-1}^{LO}(LU_j))) \\
&d^{spec}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_{n-1}^{LO}(LU_j))) \\
&d^{sib}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_{n-1}^{LO}(LU_j))) \\
&d^{ntax}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_{n-1}^{LO}(LU_j))) \\
&d^{dist}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_{n-1}^{LO}(LU_j)))).
\end{aligned} \tag{6.56}
$$

$$
\begin{aligned}
f^{CL}(LU_i, LU_j, \{W\}) = &\sum_{k=i}^{j-1} d_{i,i+1}^{CL} | \\
&(j-k) = f^{LUL}(LU_i, LU_j, \{W\}) \\
&\wedge \langle LU_k, f^{choice}(LU_k) \rangle \in W.
\end{aligned} \tag{6.57}
$$

With this information, it is possible to define a Shortest Learning Path SLP, being the subgraph of LP containing all LU and W from the start LU to the goal LU. The SLP can not directly be determined on LO level, because LO's may be about the same domain concepts and LO's may be reused in different LU and thereby such a SLP could have loops, but this is against the idea of a SLP. A SLP can also not directly be determined on domain ontology level, because the underlying concepts may be reused in different LO/LU and thereby a SLP could again have loops.

The definition of a SLP is based on the set PLP describing possible learning paths. PLP is the set of tuples of learning units determined by W_i describing learning path i of all m possible learning paths (Formula 6.58). This set is the basis for the computation of the SLP (Formula 6.59).

$$
\begin{aligned}
PLP = &\{\langle LU_{start}, \ldots, LU_{goal} \rangle | f^{choice}(LU_i) = LU_{i+1} \\
&\wedge 0 \leq i \leq n-1 \\
&\wedge n = f^{LUL}(LU_{start}, LU_{goal}, W_j) \\
&\wedge 0 \leq j \leq m-1\}.
\end{aligned} \tag{6.58}
$$

$$
\begin{aligned}
SLP = PLP_i | f^{LUL}(LU_{start}, LU_{goal}, W_j) = \\
&min(f^{LUL}(LU_{start}, LU_{goal}, W_0), \ldots, \\
&f^{LUL}(LU_{start}, LU_{goal}, W_{m-1})) \\
&\wedge 0 \leq j \leq m-1 \wedge m = |PLP|.
\end{aligned} \tag{6.59}
$$

To decide, whether a learner learns something off-topic (not being necessary to be successful in the course) or not, it is important to be able to compute distances from learning units, Learning Objects and Concepts to the SLP.

In the following, formulas for their computation are provided, starting with Formulas 6.60 and 6.61 for the measurement of the distance d_{LU}^{min} between a LU and the SLP. Based on the distances' definition within ontologies as described in [Mencke et al., 2008j] as well as in Section 6.1.3, the following formulas are focusing on a concept-distance-related point of view. Thereby, W describes the $choices$ of the current learning path and SLP_i is the LU at position i of the SLP.

In Formula 6.60, $f_{LU}^{drawOff}(LU_i, LU_j)$ determines the distance between two given learning units.

$$\begin{aligned} f_{LU}^{drawOff}&(LU_i, LU_j) = \\ &min(d^{abs}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_0^{LO}(LU_j))) \\ &(d^{spec}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_0^{LO}(LU_j))) \\ &(d^{sib}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_0^{LO}(LU_j))) \\ &(d^{ntax}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_0^{LO}(LU_j))) \\ &(d^{dist}(f^{center}(f_0^{LO}(LU_i)), f^{center}(f_0^{LO}(LU_j)))| \\ &d^{abs} > 0 \wedge d^{spec} > 0 \wedge d^{sib} > 0 \wedge d^{ntax} > 0 \\ &\wedge d^{dist} > 0 \wedge 0 \leq i, j \leq n-1 \\ &\wedge n = f^{LUL}(LU_{start}, LU_{goal}, \{W\}). \end{aligned} \tag{6.60}$$

$$D_{LU} = \{d_i\} = \{d_{LU}^{drawOff}(LU, SLP_i) | SLP_i \in SLP\} \tag{6.61}$$

With this formulas, D_{LU} (Formula 6.61) is the set of distances of LU to each element of SLP. It is used to determine $index = min(\{x | d_x = min(D_{LU})\})$, as the index of the first element of the SLP that has the minimal distance to the given LU as well as to determine $d_{LU}^{min} = d_{index}$, as the shortest distance of LU to the SLP.

The distance between a LO and a learning unit is computed using Formulas 6.62 and 6.63.

$$\begin{aligned} f_{LO}^{drawOff}&(LO, LU_i) = \\ &min(d^{abs}(f^{center}(LO)), f^{center}(f_0^{LO}(LU_i))) \\ &(d^{spec}(f^{center}(LO)), f^{center}(f_0^{LO}(LU_i))) \\ &(d^{sib}(f^{center}(LO)), f^{center}(f_0^{LO}(LU_i))) \\ &(d^{ntax}(f^{center}(LO)), f^{center}(f_0^{LO}(LU_i))) \\ &(d^{dist}((f^{center}(LO)), f^{center}(f_0^{LO}(LU_i))) \\ &d^{abs} > 0 \wedge d^{spec} > 0 \wedge d^{sib} > 0 \wedge d^{ntax} > 0 \\ &\wedge d^{dist} > 0 \wedge 0 \leq i, j \leq n-1 \\ &\wedge n = f^{LUL}(LU_{start}, LU_{goal}, \{W\}). \end{aligned} \tag{6.62}$$

$$D_{LO} = \{d_i\} = \{d_{LO}^{drawOff}(LO, SLP_i) | SLP_i \in SLP\} \tag{6.63}$$

With this formulas, D_{LO} (Formula 6.63) is the set of distances of LO to each element of SLP. It is used to determine $index = min(\{x|d_x = min(D_{LO})\})$, as the index of the first element of the SLP that has the minimal distance to the given LO as well as to determine $d_{LO}^{min} = d_{index}$, as the shortest distance of LO to the SLP.

The distance between a concept and a learning unit is computed using Formulas 6.64 and 6.65.

$$\begin{aligned} f_C^{drawOff}&(C_j, LU_i) = \\ &min(d^{abs}(C_j), f^{center}(f_0^{LO}(LU_i))) \\ &(d^{spec}(C_j), f^{center}(f_0^{LO}(LU_i))) \\ &(d^{sib}(C_j), f^{center}(f_0^{LO}(LU_i))) \\ &(d^{ntax}(C_j), f^{center}(f_0^{LO}(LU_i))) \\ &(d^{dist}(C_j), f^{center}(f_0^{LO}(LU_i))) \\ &d^{abs} > 0 \wedge d^{spec} > 0 \wedge d^{sib} > 0 \wedge d^{ntax} > 0 \\ &\wedge d^{dist} > 0 \wedge 0 \leq i, j \leq n - 1 \\ &\wedge n = f^{LUL}(LU_{start}, LU_{goal}, \{W\}). \end{aligned} \tag{6.64}$$

$$D_C = \{d_i\} = \{d_C^{drawOff}(C_j, SLP_i) | SLP_i \in SLP\} \tag{6.65}$$

With this formulas, D_C (Formula 6.65) is the set of distances of C_j to each element of SLP. It is used to determine $index = min(\{x|d_x = min(D_C)\})$, as the index of the first element of the SLP that has the minimal distance to the given concept as well as to determine $d_C^{min} = d_{index}$, as the shortest distance of C_j to the SLP.

Based on these formulas it is possible to sketch an algorithm that determines where a learner learns something off-topic. Therefore, the following input is necessary: the LU of the learner's learning path and information from a database about the learning paths of successful learners (successful in general, successful and similar learner characteristics, ...). The basic idea is to compute an acceptable range around each LU of the SLP based on the learning paths of the successful learners and then to check if the current LU of the current learner's learning path is within this range (an acceptable range can be for example the maximum distance of all successful learners or the average distance). Figure 6.14 visualizes this idea.

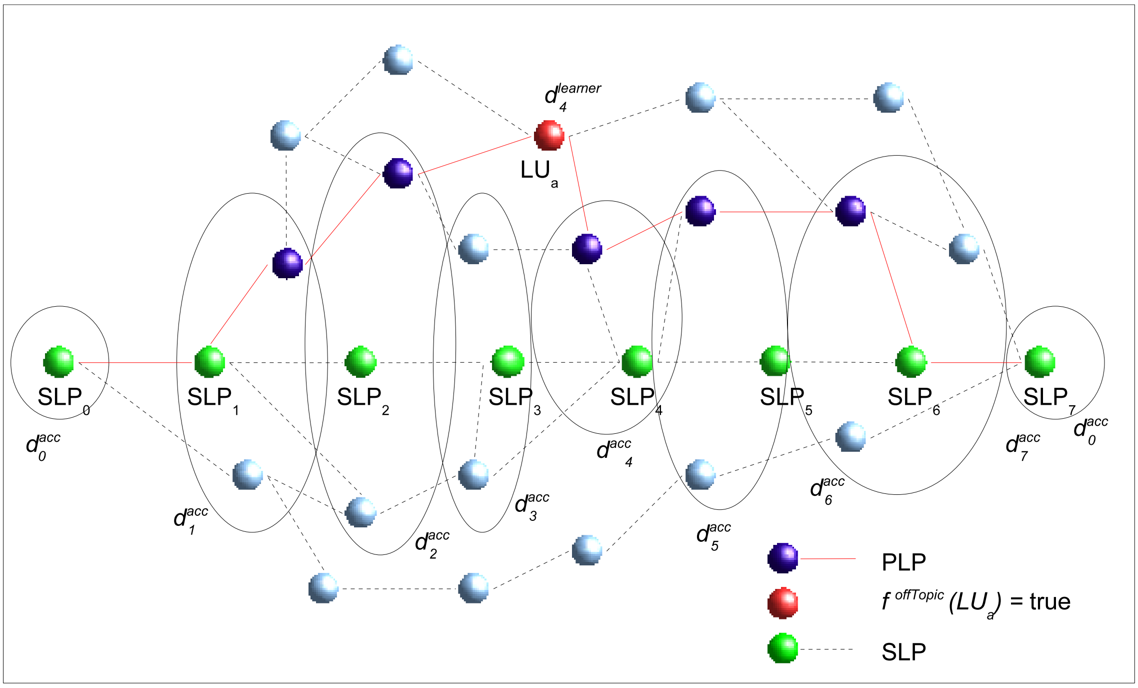


Figure 6.14: Determining When Something Off-Topic Was Learned

The algorithm sketched below, bases on the set of smallest distances d_{LU}^{min} between learning units LU_j of the learning paths of the successful learners having its smallest distance to $LU_i = SLP_i$ of the SLP (Formula 6.66).

$$
\begin{aligned}
P^{learner,i} &= \{p_k^{learner,i}\} \\
&= \{d_{LU}^{min} | \forall LU_j : d_{LU}^{min} \in \{f_{LU}^{drawOff}(LU_j, SLP_i)\} \\
&\wedge SLP_i \in SLP\}
\end{aligned} \tag{6.66}
$$

The algorithm has as input the number of successful learners n, the observed LU of the current learner LU_a, the shortest learning path SLP and database entries about the learning paths of the successful learners $SLLP$. With this information, the algorithm decides whether the current LU of the current learner is off-topic compared to the learning paths of successful learners.

The first case-block defines the acceptable distance from the learning path according to a predefined option. Possibilities are for example a maximum distance or an average distance. The second case-block determines the distance of the user's current LU from the SLP. The algorithm's decision bases on the difference of these two distances. The variable t is a predefined variable describing some kind of acceptance margin. A large distance of the current user's current LU to the direct learning path indicates that his learning efficiency might be reduced, because he learns something off-topic. The set of *learner's* LU's beyond the acceptable range is computed with Formula 6.67.

$$
LU^{offtopic} = \{LU_a | f^{offTopic}(LU_a, SLP, SLLP, n) = true\}. \tag{6.67}
$$

Algorithm 3 Identification of off-topic LUs

function OFFTOPIC($LU_a, SLP, SLLP, n$)
  $i = min(\{x|d_x = f_{LU}^{drawOff}(LU_a, SLP_r)| \forall r : SLP_r \in SLP\})$
  case range option **of**
    maximum: $d_i^{acc} = max(p_k^{b,i})$
    average: $d_i^{acc} = \frac{1}{n*o}\sum_{j=0}^{n-1}\sum_{k=0}^{m-1} p_k^{j,i}|o = |P^{j,i}|$
  end case
  case range option **of**
    maximum: $d_i^{learner} = max(p_k^{learner,i})$
    average: $d_i^{learner} = \frac{1}{o}\sum_{k=0}^{n-1} p_k^{learner,i}$
  end case
  $decision = \begin{cases} true & \text{if } d_i^{learner} - (d_i^{acc} + t) > 0, \\ false & \text{otherwise.} \end{cases}$
  return decision
end function

These formulas and algorithms can be used to evaluate the learner's learning performance and preferences. Another usage is to make recommendations about appropriate next learning steps or to reject learner choices.

Above, important approaches of enriching existing content based on ontologies were presented. Thereby, a proactive provision of content is enabled.

6.2 Proactive Presentation with Agent Technology

Not only the provision of content can be proactively supported. Technical aspects of the presentation of content can be enhanced, too. In the following, an agent-based methodology for the creation of adaptive and self-managed Graphical User Interfaces (GUIs) is presented.

6.2.1 Adaptive and Self-Managed Graphical User Interfaces

User interfaces are the basis for human-computer interaction (HCI). Their careful design is indispensable for mastering requirements of the modern society.

"Every designer wants to build high-quality interfaces" [Shneiderman and Plaisant, 2005]. Therefore, quality does not only mean style, but usability, universality, and usefulness. High-quality interfaces take into account diversities. They can have cultural reasons as well as they can focus on special populations, like adults and children.

Adaptive characteristics for increased flexibility are essential for their effectiveness and usability, because context, tasks, environment as well as user attributes may change.

Creating interfaces based on agent technology does not only follow the new paradigm of agent-based software development but leads to several striven advantages, too. Kernchen and Dumke list several guidelines whether an agent-oriented approach is useful [Kernchen and Dumke, 2007c] or not. According to them, agent-based graphical user interfaces can be necessary, because:

- GUI's may include complex/diverse types of interaction between components as well as to external distributed heterogeneous resources.
- Negotiation, cooperation and competition may occur among different entities.
- Some aspects of a GUI can have autonomous characteristics.
- A modification or expansion of the system can be anticipated.

Actually, specialized pieces of software can be designed and developed to work together in a logical goal hierarchy ([Beer, 1995]) for the fulfillment of the global goal: the support of interaction and information visualization. Their autonomous characteristics in combination with learning and adaptation capabilities provide the necessary basics for the realization of the mentioned and required flexibility aspects. Security issues are not part of this work, one possibility for secure access is the docking agent principle as described in [Kargl et al., 1999].

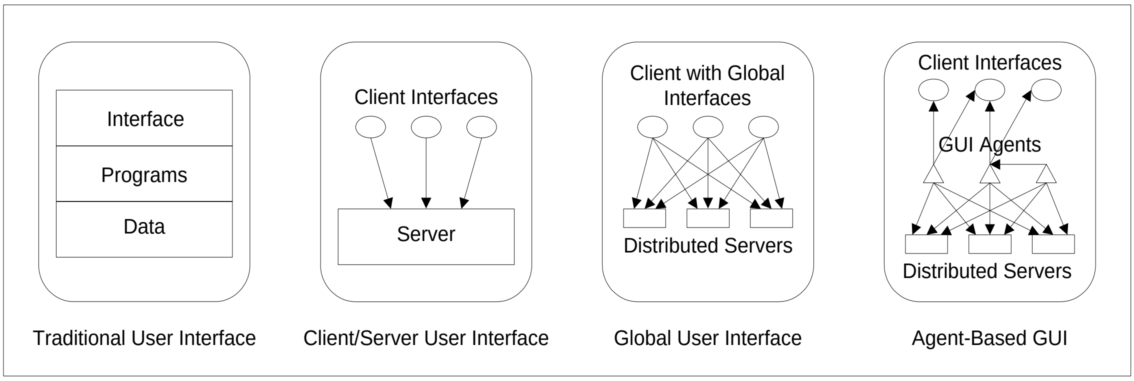


Figure 6.15: Paradigms of Interface Design (extended from [Tsou and Buttonfield, 1998])

Figure 6.15 shows existing paradigms of interface design, extended by the technology presented below.

By agent technology, a framework-based development becomes possible. It can provide basic functions to fulfill minimal requirements. Additionally, specialized agents can be added during runtime to extend the GUI. Thereby, lightweight software provision and functional adaptation are new features of the system. Agent-based GUIs, as described here, relate to up-to-date approaches for user interfaces integration as shown in Table 6.6.

	UI Component Model and External Specification	Composition Language	Communication Style	Discovery and Binding	Component Visualization
Desktop UI Components	Published, programmable API	General-purpose programming language	Centrally mediated and component-to-component communication could be supported	Static and dynamic binding	Component rendered
Browser Plug-In components	Published, basic interface (startup configuration parameters)	Document markup code and JavaScript	Centrally mediated; very limited intercomponent communication via ad-hoc JavaScript	Static binding	Component rendered
Web Mashups	Hidden interface, published API	General-purpose programming language	Centrally mediated	Static binding	Typically markup based
Web Portals and Portlets	Standard interface base don public API; interface wrapped as a Web service	General-purpose programming language	Centrally mediated (interportlet communication under development)	Static and dynamic binding	Markup based
Agent-based	Prototypical, programmable API	General-purpose programming language	Markup based	Standardized communication languages	Static and dynamic binding

Table 6.6: Comparison of Current User Interface Integration Approaches (extended from [Daniel et al., 2007])

6.2.1.1 Development Methodology

For the sketched goals, a structured approach for the analysis, specification and design of agent-based graphical user interfaces is presented. A development process using design patterns as well as creative techniques is described. Performing interface design with this approach leads to a logical goal/task hierarchy that can be easily depicted by a society of agents. Furthermore, an implicit partition of the GUI hidden in the given problem takes shape and gets connected with the particular agents. In this section, an approach respecting this nature is presented.

For developing agent-based graphical user interfaces, an iterative goal-directed methodology is proposed that includes design patterns and creative techniques. The first steps of the software life cycle are addressed – mainly specification and design [Dumke, 2003]. Figure 6.16 visualizes the stages of the approach that are described more detailed in the following.

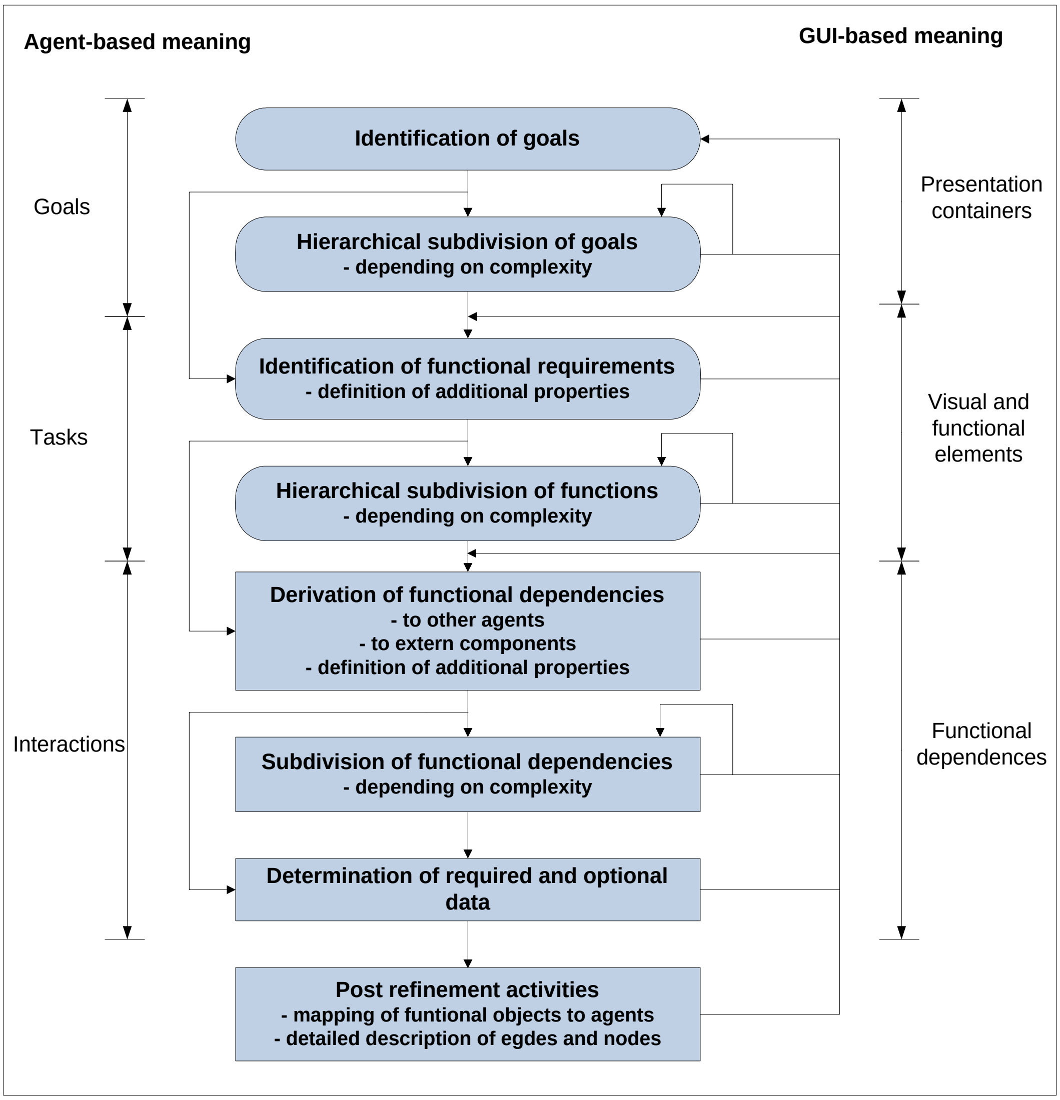


Figure 6.16: Stages of Developing Agent-Based Interfaces [Kernchen and Dumke, 2007c]

6.2.1.2 Specification Step

At the first stage, the goals that should be solved are defined and refined. The starting point is the given problem in a special application domain. The goals and sub-goals have general character and form a tree. No special functions will be defined yet.

To identify subgoals, creative techniques like brainstorming and mind-mapping can be used. Especially mind-mapping is useful, because of the hierarchical nature of this technique. By this, not only complete but innovative results can be achieved. It is possible to identify goals that where initially not intended and the possibility to create a more complete software product is given.

This distinction of required and optional nodes of the created tree helps the customer and the contractor to determine the possible and the intended functional range of the product as well as the expenditure for its implementation. All sub-edges of an optional edge are also optional for the current model. If a change – from optional to required – is made for an edge, then the special subtree needs to be iteratively analyzed for possible changes in structure and type of the edges. Thr refinement of goals results in their subdivision into sub-goals depending on the complexity without defining special functionalities.

Figure 6.17 presents a possible manifestation of the described procedure. The usage of different shapes to visualize different hierarchical levels of nodes is recommended within a stage, but required for the distinction of the first two stages. Dotted and solid lines represent optional and required edges, respectively. This distinction is necessary.

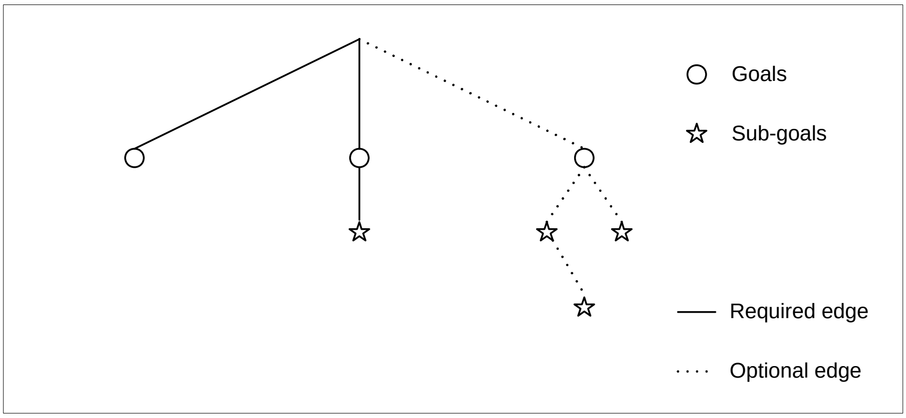


Figure 6.17: Hierarchical Subdivision of Goals and Sub-Goals [Kernchen and Dumke, 2007c]

The nodes of the created tree represent agents that should be included in the multi-agent system. They solve the given problem in a general way without taking into account necessary functionalities this this point. They concern about the solution of the given problem. Therefore, they need functionalities that will be provided by the agents whose specification will be described in the following section.

In the context of the graphical user interface, the already specified agents represent the container and organizational elements for the visualization of the current presentation and interaction mechanisms. For this strategy, Composite – a structural pattern – is recommended for usage. It bonds objects to tree-structures to represent part-whole-hierarchies and allows a uniform treatment of objects as well as compositions of objects

[Gamma et al., 2001].

While analyzing and specifying the subdivision, new goals or subgoals may become clear, and then an iterative process of refinement needs to be instantiated.

In a next step, the necessary functional requirements and their hierarchical subelements will be identified. By this, current leafs of the tree, goal leafs as well as sub-goal leafs, will be extended.

6.2.1.3 Identification and Subdivision of Functional Artifacts

The reason for the distinction of goals and functions is the distinction of container and visualization elements that is now possible. From an agent-based point, of view the upper nodes of the tree represent agents that use the functional "working" agents to reach their goals.

At this point, functional objects are defined as objects with some intended functionality; the identification of agents is a later stage. Thereby, the main focus is on those functional objects to specify agent-oriented tasks. The other qualitative, system-based and process-oriented requirements known from standard IEEE Std. 830 are defined as additional properties at this point [Institute of Electrical and Electronics Engineers, Inc., 1998].

Further analysis of the given requirements and designing an appropriate software architecture is the intention of this and the next stage. Special tasks are data processing, data allocation, and the visualization of interaction or information, for example.

The special functionalities that are needed to reach the defined goals are identified at this stage. As described in Section 6.2.1.2, creative techniques may lead to the already described advantages. Due to the non-linear nature of creativity it may occur that new goals become clear and need to be analyzed as well as specified.

Another special visualization artifact for supporting the design is introduced in Figure 6.18. Arrows visualize an "is-a" dependence between functional objects. Those child nodes inherit all constraints of the parent node.

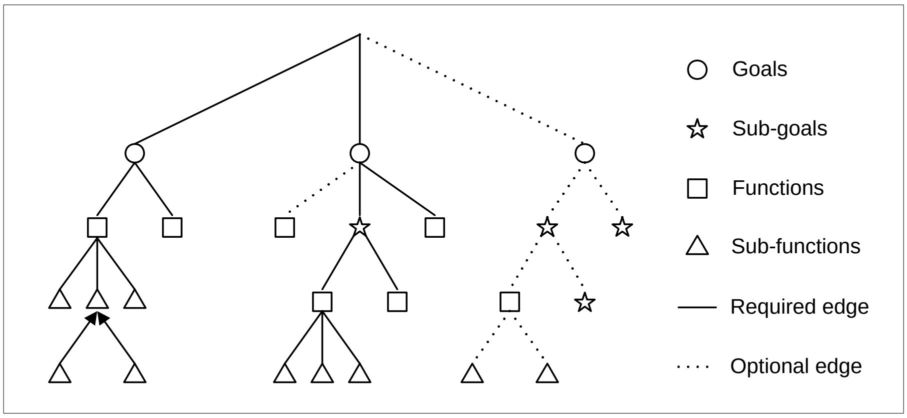


Figure 6.18: Advanced Hierarchical Subdivision Including Functions [Kernchen and Dumke, 2007c]

The nodes created at this stage of the proposed approach, represent functionalities of working agents and their tasks. They are organized in the already defined containers and provide processing and/or presentation capabilities for example. These graphical user interface functionalities directly refer to graphical aspects for information and interaction visualization.

If a function is too complex for an easy implementation, then its subdivision in smaller nodes is recommended. The complexity does not only refer to the functionality but also to the amount of dependencies to other nodes or external components. The later aspect is described in the following section.

6.2.1.4 Derivation and Subdivision of Functional Dependencies

Supporting overall communication of all elements is neither necessary nor useful. The modeling of required interactions is recommended. This is justified by security, performance as well as implementation reasons.

Up to this point, a tree-shaped structure is defined based on the subdivision of goals and tasks. Now, this structure is extended by defining additional internal and external connections. Next to the connections of the specified goal and task hierarchy – the edges of the tree – more internal functional dependencies may exist and need to be specified: a graph structure develops.

Furthermore, it may be necessary to add external connections at this stage. They represent dependencies on external databases, applications or other distributed resources.

At this point, creative techniques as well as approaches to examine Formal Completeness are useful again to identify more required or optional dependencies. Therefore, Formal Completeness in general means that all aspects, connections or artifacts are evaluated even if they do not seem to have any necessity or practical reasons for existence.

Mainly, interaction between agents is not limited to single informing messages. Sequence diagrams are appropriate for modeling more complex interaction acts. To facility interaction design in the created graph structure, special items for the clarification of connections from or to all other agents are proposed as shown at the first function of the second goal in Figure 6.19. That can be useful for management purposes.

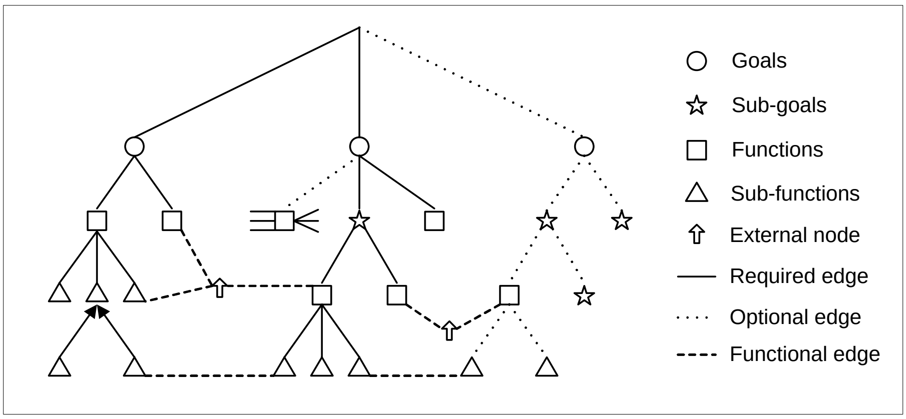


Figure 6.19: Graph Presenting Additional Internal and External Connections [Kernchen and Dumke, 2007c]

The data identified to be required or optional are the parameters of communication – actually the communication content. As already described, iterative repetitions of previous stages may be necessary in order to describe a model that should be as complete as possible.

For the inclusion of external resources, the usage of proxy-agents is proposed. They follow the structural pattern of a proxy [Gamma et al., 2001]. Several benefits can be derived from using such a surrogate, e.g.:

◦ Integration of external objects
◦ Encapsulation of safety critical objects and access
◦ Encapsulation of objects relevant for several optimization approaches
◦ Easy update of access
◦ Parallel existence of accessing objects
◦ Reduced network load by caching

For the agents, the completion of this stage results in the definition of their communication acts as well as their communication content. Now they can collaborate to solve the given problem. Modeling special node and edge-related issues is described in the following section as the next stage of the methodology.

6.2.1.5 Post Refinement Activities

Up to this point, the general structure of the multi-agent system and the graphical user interface is modeled. The next steps are the mapping of functional objects respectively sets of those objects to agents and the detailed description of edges and nodes.

Complexity measures like the number of sub-functional objects or the number of edges can indicate an appropriate mapping. Sometimes, aspects of the later implementation need to be taken into account, too. By recombining functionalities all sub-functional objects as well as all internal and external connections are compressed into one agent disregarding the detailed functional design. To handle this, an ontology-based characterization of the modeled architecture is proposed as well as the usage of ontology-metrics to identify a useful segmentation as proposed in [Mencke et al., 2008c].

The identified agents, their functionality and interactions can be further analyzed using classic (object-oriented) techniques. A node-based definition can result in several aspects, e.g.:

◦ Agent identifier
◦ Role-based access
◦ Related ontologies
◦ Description
◦ ...

This analyzing of the agents allows the combination of the reactive agent type with other types. Therefore, explicit models from this other types may be included. That can be the assumption, goal, plan or interaction model of the Belief-Desire-Intention agents

(BDI) ([Rao and Georgeff, 1991], [Mencke and Dumke, 2007a]). New design iterations may become necessary.

Similar definitions can be proposed for the edges/interaction acts as well, e.g.:

- Message parameters of the FIPA ACL Message Structure Specification of the Foundation for Intelligent Physical Agents (FIPA), e.g. type of communicative act, participants, content [Foundation for Intelligent Physical Agents (FIPA), 2002b]
- Communication act identifier
- Conditions
- Related ontologies
- Parameters
- ...

Depending on the precision and depth of following the proposed methodology, all necessary functionalities of the agents, their interactions and the structure of the interface can be modeled and designed.

6.2.1.6 Advantages

Designing agent-based interfaces with the proposed approach has several advantages. A first one is the similarity of goal the hierarchy intended by the problem definition and the created hierarchy of agents/interface components. That is an important advantage, because it supports the user to learn the usage of the interface.

Another one is the definition of optional and required nodes/agents and edges/communication acts. By this, possible manifestations become clear for customers as well as for the contractor. The implementation effort of additional elements can be estimated, too. Implementing interfaces for optional and not yet realized agents supports the later extension of the multi-agent system. Easy, fast and dynamic update of agents is also supported; for example because of the clear distinction of functional and presentation components. By this development methodology, all possible affected components can be identified by following the edges of the particular node of the graph.

Another advantage is the easy extension of the system during runtime by simple addition and registration of new agents. Agents also allow the separation of content, presentation, and content delivery protocol. Using the proxy pattern the presentation of the content is not directly coupled to the type of the external resource or the type of its transmission anymore.

Failure tolerance is another major advantage of agent-based systems that rub on interfaces. The failure of one component does not cause a complete system crash anymore due to possible redundancy and loosely coupling of the components. Performance control of the components becomes easier, too. Generalized approaches for observing and reporting failures, performance and utilization can be easily and uniquely embedded.

With this approach, one more important characteristic of agents – pro-activity – is directly available for interfaces now. That might for example result in motivational avatars, the automatic highlighting of content or other proactive adaptation patterns.

The integration of creative techniques leads to more complete and innovative solutions; for the general modeling as well as for the detailed technical design. With the presented approach, the usage of UML diagrams for object oriented modeling, for example sequence diagrams, is possible for specification and design of the agents and their functionality. Using a tree-shaped modeling approach allows the detailed inspection of sub-trees as well as a detailed and exact modeling of the system, respectively.

There are several advantages related to graphical user interface design, too. At first an implicit interface partition hidden in the given problem takes shape and gets connected with the particular agents. By this, a very natural design of the interface is possible.

The special visualization components are defined very deep in the tree and at the leaf nodes, respectively. Higher nodes/agents serve as containers for the visualization components. By this, a level-specific uniform as well as a personalized presentation is possible. That supports the consideration of diversities, which can be depicted by the usage of agent technology.

Furthermore, depending on the precision and depth of following the proposed methodology the creation of either frameworks or detailed graphical user interfaces is possible.

6.2.2 Agent-Based Graphical User Interfaces for e-Learning

As already stated out flexibility, adaptation and personalization are key features of a GUI for e-Learning. Using the previously described methodology developed a graphical user interface based on agent technology is developed (ABEL-GUI). By turning into account its heterogeneous and autonomous characteristics the intended goals can be achieved.

But why so much flexibility for e-Learning? Learning is one of the most important aspects of our culture. Its effective implementation and support is essential for human beings in all phases of their life. Lifelong learning is the corresponding slogan. There are many changes in that long period of time. That refers to internal human being aspects as well as to external environmental ones. Biological, psychological changes and changes because of new knowledge require highly flexible mechanisms for the adaptation of supporting systems. That obtains to learning in a particular manner. External reasons are new scientific knowledge about didactics, psychology, interface design and new information about the topic to be learned. Versatile problems require adaptive and extensible solutions [Kernchen et al., 2007d].

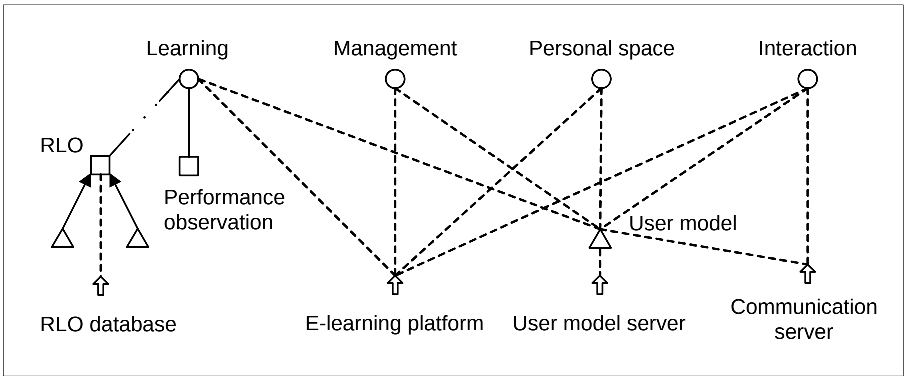


Figure 6.20: Core Components of ABEL-GUI and Selected Transitive Connections to External Distributed Elements [Kernchen et al., 2007d]

In the following, the architecture of the developed system is described from a student's point of view. Other roles are tutor, author and administrator and lead to a distinct usage of existing elements as well as to completely new features.

6.2.2.1 Learning Components

The first and obviously most important parts of the GUI to be developed are the learning components. That refers to all goals and functions that are directly coupled with the learning tasks.

Following the already described methodology, four direct sub-goals and one direct functionality are identified.

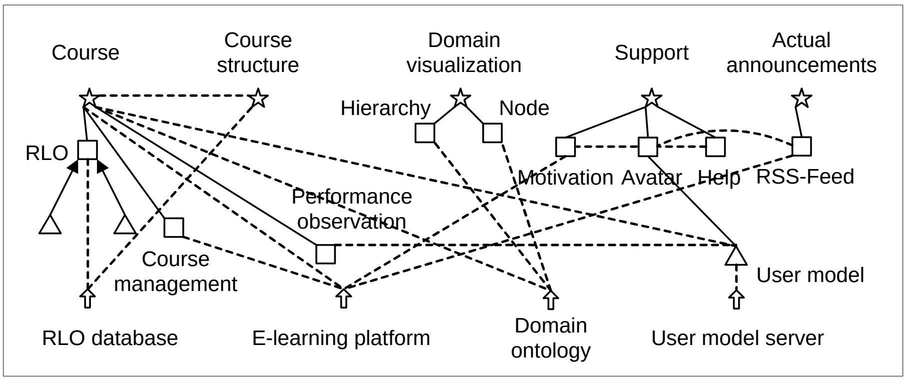


Figure 6.21: Graph Presenting Additional Internal and External Connections [Kernchen et al., 2007d]

Course: The task of this container element is the presentation of the course content. Following a given course structure, it organizes the presentation of the content of its sub-elements. This separation of content and function supports the usage of different sources. Learning Objects may be read ahead or completely loaded at the time of subscription. A kind of version management for guaranteed provision of the current content is needed. A performance measuring component may provide additional functionality. Functional edges to other internal or external elements are targeting the content database, the domain ontology, the user model and the user-model server.

Course structure: This visualization goal supplies support for intra-course navigation. Therefore, it provides personalization as well as adaptation support. An overview about possible adaptation techniques can be found in [Kernchen, 2005]. Additional connections are heading for the course element and the e-Learning platform.

Domain visualization: The presentation of overall context information is an effective way for student motivation and for sensitizing them for the context of the current topic they are learning. Therefore, a domain visualization component with sub-functions for hierarchical shrinkable visualization and node visualization is proposed. The domain ontology is functionally connected with this modeled goal. Other possible connected and later described elements are e.g. wiki, forum, search, current announcements and the personal space.

Support: The support element refers to help as well as motivation. Another key feature is the avatar, a reference object for the student which individually provides individual help and support. One sub-functionality which is also necessary for many other parts is the user model that can be stored in an external user model database. It is for example used for the storing of technical or presentation preferences, personal information as well as learning states as a basement for adaptation and personalization. Entries can be classified as private or public for restricted access of other agents.

Current announcements: This identified functionality provides up-to-date information for the student about various topics he might be interested in. Therefore, it is functionally connected with the e-Learning platform.

6.2.2.2 Management Components

The adaptive functionality of this system is based on technical as well as organizational management. That refers to the management of agents, courses, user model and to logging mechanisms.

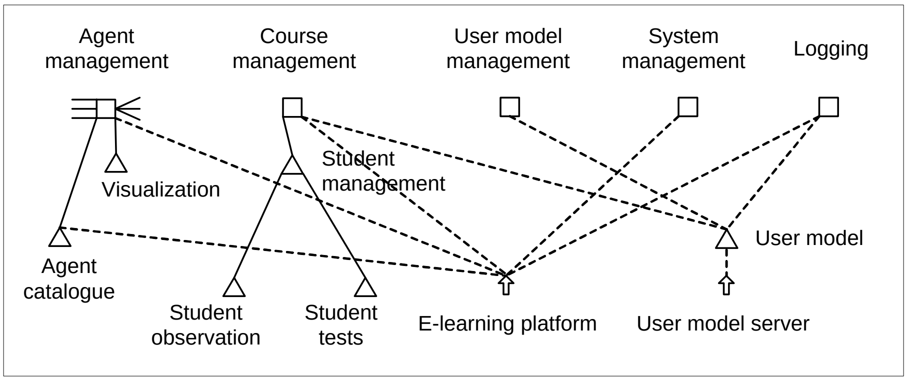


Figure 6.22: Graph Presenting Management Components, Chosen Subcomponents and Functional Edges [Kernchen et al., 2007d]

Agent management: For provision of maximal functional flexibility, the GUI is designed as a framework. It consists of the necessary parts to provide minimal support for the learning tasks. Additional agents can be integrated for more features. They have a recommended docking station but may be placed individually, too. The agent management element provides support for agent integration, management and disintegration. That includes the necessity of saving the current architectural configuration. A graphical component increases the usability of this aspect.

Course management: Booking courses to learn is another essential management issue. Therefore, connections to the e-Learning platform and the user model are important.

User model management: The user model collects and models information about the student. After some manual initialization, most data are added automatically. It is the information base of personalization and adaptation. Role-based access, control and management purposes are remits of this element.

Logging mechanisms: Authentication is needed in the context of e-Learning, too. Users need to be identified to allow personalization, adaptation and accounting. Functional connections for example exist to the user model and the e-Learning platform.

6.2.2.3 Management Components

Personal parts that might be implemented in an e-Learning GUI are e.g. a personal homepage, a course scheduler, personal space for storing of files and information, individualized search mechanisms and a kind of diary. A user server might be a technological base for these elements.

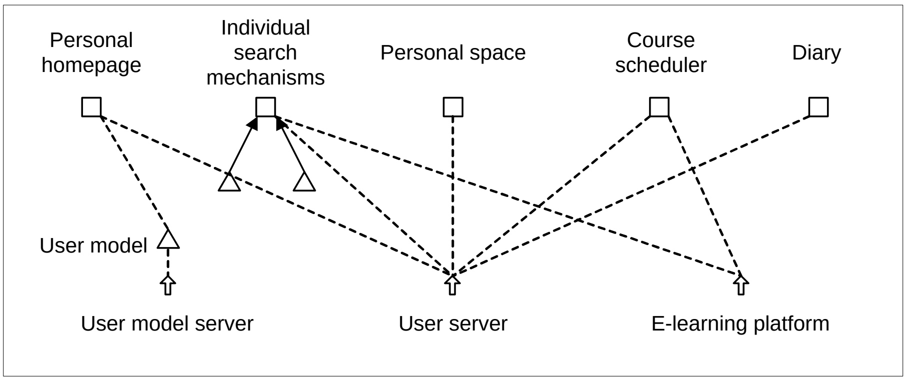


Figure 6.23: Graph Presenting Personal Components, Chosen Subcomponents and Functional Edges [Kernchen et al., 2007d]

Personal homepage: This element is an important aspect for the learning community. Everybody introduces oneself. Gaining information about each other is one of the first steps of the group development process. Functionally it is connected with the already described user model.

Course Scheduler: Users can book several courses at the same time. Their organization is the intended usage of this scheduler. Therefore, connections to the user model as well as to the e-Learning platform are necessary.

Personal space: The integration of an internal configurable file system is useful for the students. It provides a time, location and computer-independent access to needed resources. It can be media-based or domain-dependently structured.

Individual search mechanisms: Some repeating search tasks can be automated or at least stored for reuse. Therefore, connections to the e-Learning platform and the user server are needed.

Diary: This element contains automatic logfile mechanisms as well as functions for individual user input. A connection to the user server is needed for permanent storing.

6.2.2.4 Interaction Components

Human interaction is essential for learning and learning-related motivation ([Walther, 1992], [Dimitrova et al., 2003]). The presented approach offers various core synchronous and asynchronous communication supporting elements and the easy usage, update and extension of this set.

All interaction components imply functional connections to the user model, the user server and the communication server. Preferences and logfiles need to be stored for every communication tool for personalization and adaptation.

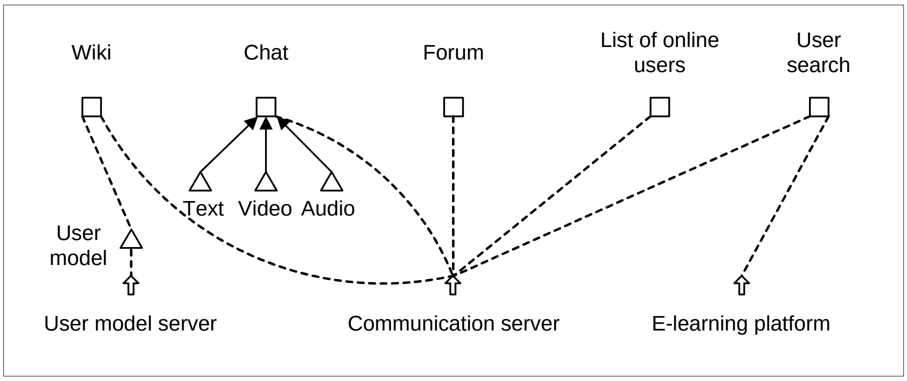


Figure 6.24: Graph Presenting Interaction Components, Chosen Subcomponents and Functional Edges [Kernchen et al., 2007d]

Wiki: That is mostly a collection of Web pages forming a kind of content management systems. The main difference is the possibility for online users to change the content, following the spirit of Web 2.0. There can be different views on a wiki: a course-based view, an e-Learning view or a domain-dependent view.

Chats: Video, audio and text-based chats are very important interaction channels of actual Web users. Their synchronous nature of message transport is the base for establishing and maintaining an online (learning) community.

Forum: This interaction medium provides asynchronous discussion functionalities. It may include sub-forums and may be organized based on the course structure or the domain ontology including technical or organizational issues. The chronology of the argumentation is saved and can be tracked.

List of online users: Such a list can be a useful tool in the context of a learning community. Connections to friends or colleagues are explicitly visible, avoid the feeling of loneliness and always offer the possibility of help, advice or at least somebody listening. A list of online user can be individually, course-based or domain-depended grouped.

User search: The grouping mechanisms of the previously described list of online users provide basic search characteristics, too. Additional search constraints may be the role, current or previous course-memberships, aspects of the domain ontology, public features of the user model (e.g.: personal interests, spatial location). Also, a keyword-based search is imaginable. The intended benefits were already described, too.

6.2.2.5 Implementation Remarks

This GUI is intended to be implemented as a framework. That is an important aspect for the provision of the flexibility, the necessity and advantages of which was proven in the introduction. Therefore, minimal required elements as well as necessary infrastructural aspects must be defined.

The non-ambiguous identification of agents and information about them is the basis for all other infrastructural aspects. That refers to the agent type, its functionality as well as the concrete instantiation. At this point, standardized approaches like [Foundation for Intelligent Physical Agents (FIPA), 2002b] or [Foundation for Intelligent Physical Agents (FIPA), 2002a] are not directly applicable due to the multitopic nature of the required solution. To close this gap, machine-readable meta descriptions for the identification of agents were developed. Two implementations were performed: an XML schema definition file (XSD) and an OWL file. They model the same area of interest but due to their different implementation technology the provide appropriate support for different requirements. They include existing standards, extend them and make them available in a larger context.

For example, the FIPA standard specifying message structures was adopted and integrated within the developed specifications [Foundation for Intelligent Physical Agents (FIPA), 2002b]. Figure 6.25 shows a related excerpt from the XSD file.

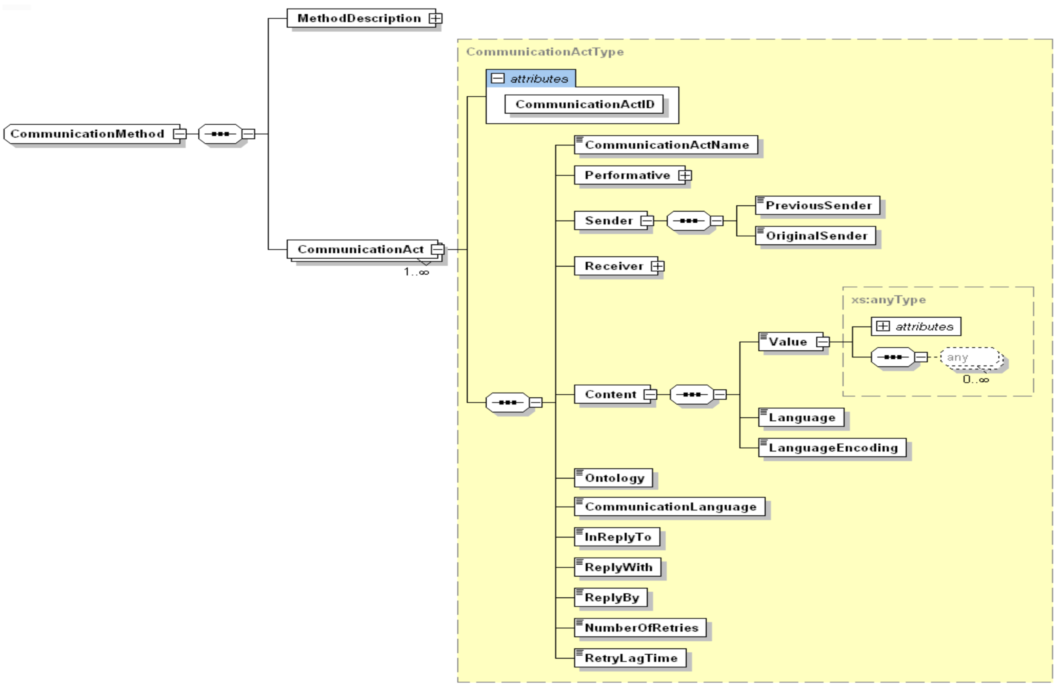


Figure 6.25: XSD-File Excerpt for Agents's Communication Method Description

6.3 Discussion

As shown in this chapter, proactive and semantic-based technologies are useful to enhance the operation phase of the QuaD^2-Framework.

A very detailed focus was laid on the enrichment of e-Learning content. Therefore, algorithms were introduced to enhance the learning and teaching process with the help of semantic information. The concept of Semantic Windows as a core approach to identify semantically-related information were described in detail. A cost-based approach defines costs for different types of semantic relationships and up to a certain threshold every element is within the range.

The distance-based Semantic Window approach is seen as an alternative to the cost-based one. An additional advantage is the possibility to define more specific semantic windows, like tori instead of spheres, to use an geometrical analogy. Further improvements are possible regarding transitive semantic windows.

Additionally, an overview of existing metric ontology measurement was presented following a structured approached based on the concept of GQM. During research, a lack of metrics for balance-measuring for ontologies was observed. To close this gap, different criteria for a balance measuring framework were identified and future steps towards a balance-metrics set were outlined.

Presenting additional information is not always advantageous. Sometimes user tend to loose the focus on the essential content. To close this gap, further solution were presented for the semantic-supported, proactive creation and provision of learning path recommendations as well as for the coverage of the focus on the essentials.

A 3-level ontology framework was introduced and used as a basis to automatically determine learner's learning preferences. With this information and prior work, an algorithm was presented to generate individual recommendations about appropriate next learning steps. The developed that computes recommendations about possible next learning steps, based on individual and automatically-detected learning style preferences as well as learning strategies of classified peer learners. Finally, formulas and an algorithm were defined to determine whether a special learning unit may be off-topic for a special e-Learning course.

For the proactive presentation of content, a tree-based approach for developing agent-based graphical user interfaces was outlined. Therefore, several stages of specification and design were presented. Furthermore, the usage of several strategies like creative techniques and design patterns for innovative and complete architectures was found to be helpful. To examine the usefulness of the approach, the ABEL-GUI was designed – an agent-based interface for the e-Learning domain. High flexibility, organized interaction mechanisms as well as adaptation support are key benefits of this technique.

Summarized, this chapter presented a methodology for the enhancement of the QuaD^2-Framework's operation phase that not only just provides content, but improves/extends entity provision and presentation in a semantic and proactive way.

7 Chosen Proactive Applications

7.1 Proactive Educational Courses

The technology of proactive educational courses proposes a scenario of the integrated application of agent technology, user models and classification techniques in e-Learning.

One current problem is about coupling learners and distributed educational courses. The common approach is to try to locate the course and its content by (metadata-based) search algorithms or directories. Here, the other way is investigated. Based on existing metadata-labeled educational courses as well as clients enriched by models about the user, the coupling between course and learner can be initialized on behalf of the course itself. [Kernchen and Dumke, 2007e]

7.1.1 Architecture

A proactive educational course is an entity containing learning information that starts to search for possible learners on behalf of its own. Using appropriate algorithms based on user models and meta-data enriched learning content, a classification of users is initialized to identify possible candidates. Based on different criteria, a kind of map of possible students can be generated. The motivated environment is shown in Figure 7.1.

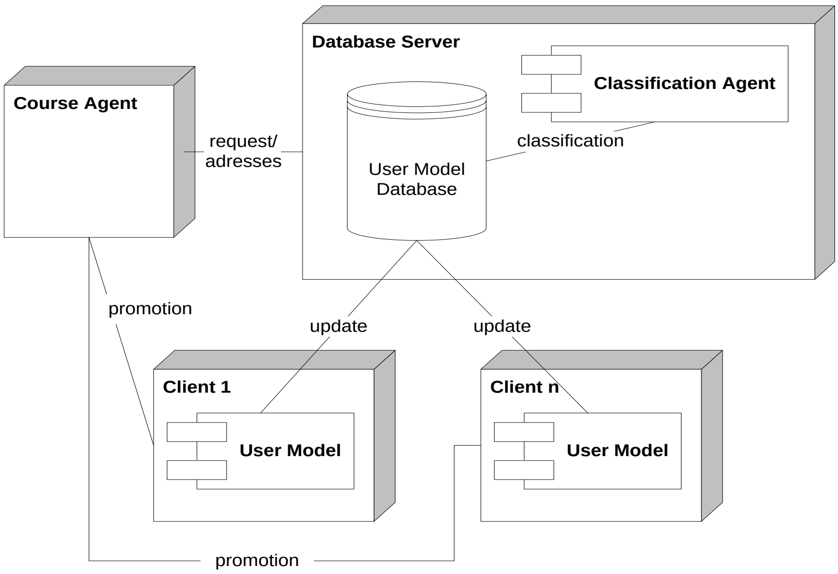


Figure 7.1: Possible Architecture with Proactive Educational Courses [Kernchen and Dumke, 2007e]

The presented main parts are the database server, the client interfaces to the learning environment and the various educational courses. The database centrally stores the user models. Furthermore, it provides access for the classification agent. It is not necessary to define any special content of the educational course at this point. Only appropriate meta-data need to be provided.

7.1.2 User Model Database

The user model database centrally stores the periodically updated user models. This approach provides several advantages.

Centralized access for classification algorithms is given. So, the classification is independent from the uses' online times.

Reduced network traffic can be assumed while performing classification. There are no network capacities to the client needed. Remitting the use of the network to times with lower network load is possible. With appropriate algorithms, updating the centralized user model can be proceeded when the user is idle and no other tasks are harmed while updating.

An **independent access** to the user model can be realized. So, it is not necessary to take the user model data physically with you.

These possible advantages of centralized storing the user model data lead to new requirements. Next to providing an appropriate server, update mechanisms are needed. These algorithms must consider information about user online times, user interaction times and general network traffic.

For the definition of the user model structure, the usage of appropriate ontologies is proposed. They can be used for database design and some metadata descriptions of the educational course, too.

There are three different interactions to be modelled.

At first, the updating of the user models need to be supported. Therefore, the structure of the database needs to fit the interesting user information that is collected. Scheduling the updating process can vary. Due to unpredictable online times of the students no static schedule is possible. As long as there is an established network connection a dynamic update can be realized either on login or logoff of the user. At this point the update during login is proposed, because no correct logoff process can be guaranteed. In the case of offline learning updating can be delayed and leads to deprecated centralized models. But the quality of the motivated scenario will not suffer, because no online course can be suggested to an offline student.

The second interaction is the one between the educational course agent and the database server. The course agent creates requests for addresses of possible students. A possible parameter for the message transfer might be the criteria to be used for user classification. As an answer, the generated map with addresses of possible students is sent back.

As a third interaction the coupling of the classification agent and the usage model database is analyzed. The classification agent tries to find relations between data of the usage model and some predefined groups. If the difference is small enough, then the usage model is added to this special group. The model will not be added to the group or it will be removed if the difference is too big. Of course, newly added models can be used for updating the group properties. It is thinkable to use only one or more criterion in combination for classification. The criteria for those algorithms are submitted in the request of the educational course agent. There are many possible parameters.

- Spatial location
- Requirements for new courses are fulfilled
- Topics of interest
- Little knowledge about a topic
- Externally initiated participation (by tutor or teacher)

In the case of conflicting or lack of information, the time of classification is re-scheduled to a later point in time. The process of classification can be started due to several reasons. So, it is possible to schedule predefined times or to start because of the request of the educational course agent. Updating the centrally-stored user model can lead to a classification, too. All these parameters can be categorized in hard or weak to start a new classification.

Hard:
- Passed or failed exam
- Tutor-initiated participation
- ...

Weak:
- New, changed or extended interests
- Aging of students
- ...

At this point, an initialization based on these hard/weak categories is proposed. While changes of hard properties lead to instant classification, updated weak parameters only start the algorithm if a certain threshold is passed. Together with a periodically started classification schedule instant up-to-date results can be presented to the requesting course agent.

This classification agent can be part of the server infrastructure or dynamically leased from special providers [Kernchen et al., 2007c].

7.1.3 Client

The client side is the interface for the user's access to the distributed learning environment for online learning. It is supposed to contain a user model that is as complete as possible which is used as the base of the user classification. Of course, the needed criteria must exist and be observed. Due to the assumed distributed properties of this scenario, a software agent is proposed as an observer. Target of its work is the logging of the interactions of the user with the client interface to the e-learning environment. It is mainly used to get information to fill the user model and to support adaptive techniques for information presentation. There are two possible scenarios for the architectural integration of the observer agent to get the needed data.

Proxy: The agent can be implemented as an interceptor between the learning content distribution unit and the graphical user interface. There it can sniff and forward the exchange of data with a fine granularity.

Log-file: Another possibility is to process logged information after the in many interaction sequences resulting learning process in a coarse granularity for increased performance.

7.1.4 Educational Course Agent

The educational course agent contains the content as well as appropriate metadata. The content does not matter as long as the needed metadata are included. These data are needed to describe the intended consumer. This information is used as a parameter in the request that is sent to the user model database server. Another interface needs to be implemented for interaction with potential students. Information about the course itself is intended data to be sent. By this, the student can decide to take part in the course or not.

The initialization of the proposed pro-activity can base on scheduling algorithms or a certain percentage of intended usage.

7.1.5 Extension for Mobile Environments

By using additional information about the spatial location of the learner, even more adaptive courses with increased motivation as a fundamental basis for successful learning can be expected [Dumke et al., 2007b].

An innovative feature is the coupling of meta-data about the user and the course to be learned with spatial information. By this, the extension of existing customization approaches like the following one becomes visional.

Tourists in museums or in towns can already get further information about several objects. With this approach, this information delivery can be modeled more adaptively. Currently only that information can be presented that is potentially interesting them. A professor for architecture does not need information about baroque era. A hint about this special building and highlights may be sufficient, meanwhile “normal” tourists might be interested in more fundamental information. Another aspect to be investigated is the extension of existing courses by meta-data for the course agent to request potential user addresses from the classification agent. Based on this information and appropriate classification algorithms, new interaction schemes can be established, namely the update of distributed user models to a central database, the course request for new users and the promotion of the course to identified potential users.

The motivation in order to implement proactive courseware is based on:

(P1) The necessity of lifelong learning requiring a disciplinary continuous motivation for course participation
(P2) The missing information about new conditions of (certified) qualification in special work areas

(P3) The necessity of managing many (personal) processes where qualification/learning is only one of these processes that could sometimes be placed in the "background"
(P4) The missing information about new courses based on new research results in a special area of professionals

Hence, the course should be autonomous or intelligent in itself in order to provide the qualification or learning activities in different fields of education. On the other hand, more and more courses can be created on an e-Learning-based technology and strategy. An intelligent course might be able to locate its customers itself.

That is a typical offline support for users/students which is "waiting for using". The current situation can be described as:

(E1) Existing materials of course supports such as exercises and information as Web applications
(E2) Existing support in order to install software infrastructures and to keep the exercises in software development and implementation
(E3) Existing portals of course information, (Web-based) examination registration, and course scheduling
(E4) Existing platforms of discussion addressing the study that considers all aspects of courses, solutions of exercises and examinations and course rankings by the students

The evolution of this situation using the presented proactive approach should provide the following
(P5) supporting the students in order to manage the course scheduling more efficiently as a "background planner"
(P6) addressing more students/learners for the lifelong learning in a postgraduate kind of study

Considering the Sharable Content Object Reference Model (SCORM) [Advanced Distributed Learning (ADL), 2006b], the following types of adaptations can be identified in order to keep the intended pro-activities P_i (see Figure 7.2).

Considering the mobile techniques, the following activities can be established in order to provide the mobility of e-learning:

(M1) It is necessary to choose the mobile technologies which should be supported in the e-leaning courses (as protocols (TCP/IP, WAP, HTTP, etc.), communication networks (GSM, HSCSD, GPRS, UMTS etc.) and platforms (Palm OS, Windows CE etc.).

(M2) The course contents should be made compatible with different chosen mobile technologies (HTML, WML, etc.).

(M3) Context-related situation of organizational and policy standards must be taken into consideration.

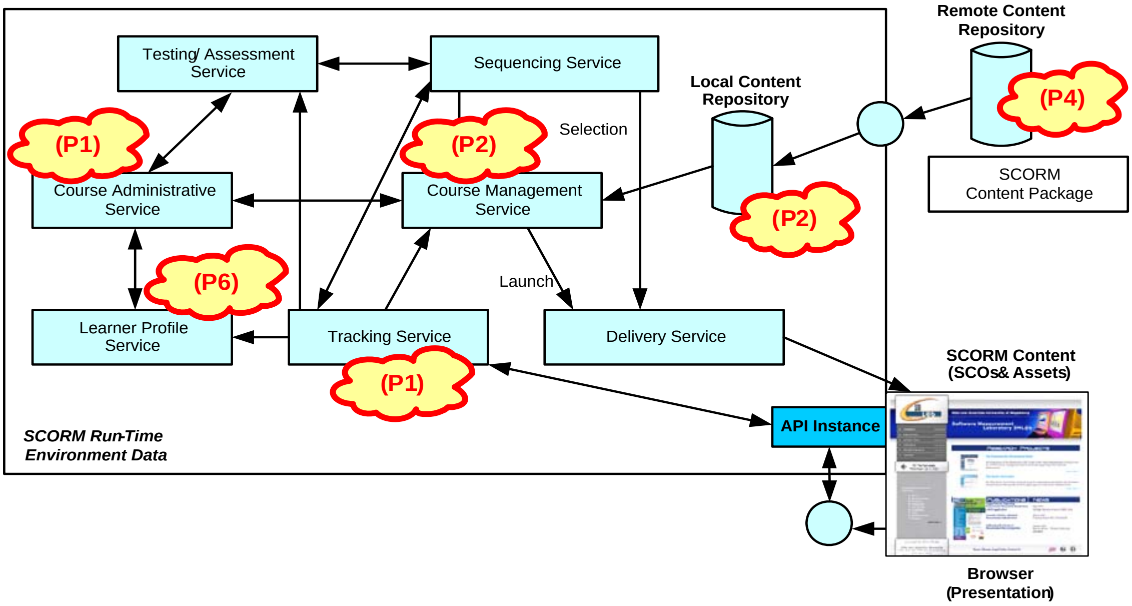


Figure 7.2: The SCORM Learning Management System and Adaptations [Dumke et al., 2007b]

7.2 Proactive Class Schedule

For students, the creation of their own class schedule sometimes is a complex and confusing task. That relies to the fact that they have to plan learning activities for up to 2 or 3 years without explicit knowledge about available courses or seminars. Another fact is the distribution and diversity of needed information across multiple sources. The proposed application does not want to automatically plan the whole scholastics for the students, but to provide them a tool that compromises all needed and available information into one consistent presentation with the possibility to make sound decisions. Based on a user model and available external information, the application forms a kind of a proactive, adaptable class schedule for every student.

Based on the presented architecture (see Figure 7.3), some of the expected advantages are:

- Consistent presentation of study requirements and course offers
- Proactive notification of needed and available courses
- Alternative course suggestions
- And in the end, reduced time effort for the students' study is espected.

7.2.1 Current Situation

Studying at German universities is a process that is fundamentally based on initiative of one's own. Mainly that is because of individual learning. Another potential key fact is the absence of pre-defined class schedules at German universities. There are existing frameworks defining required as well as optional lectures. The later ones need to meet

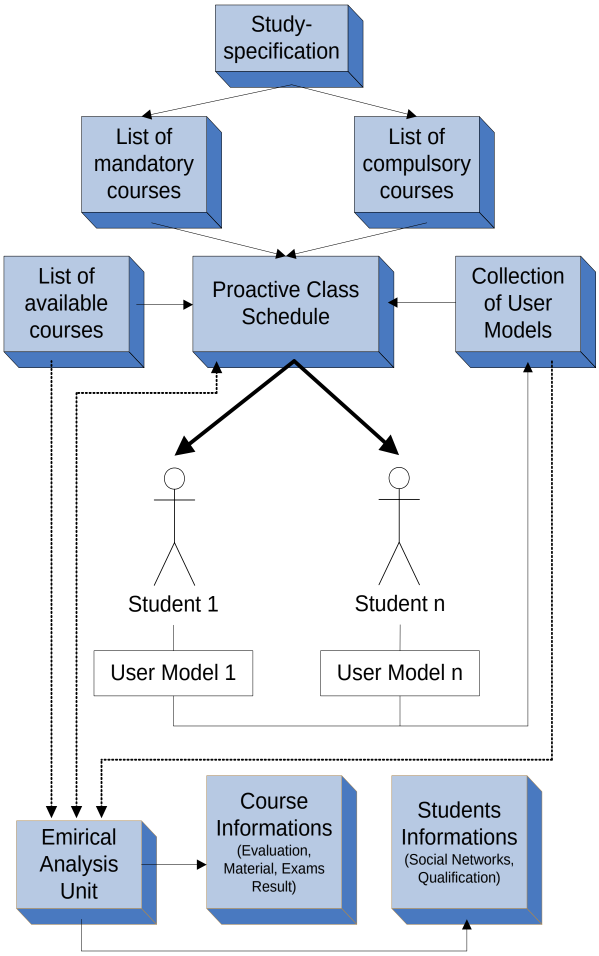


Figure 7.3: Architectural Components of a Proactive Class Schedule [Kernchen et al., 2007a]

certain requirements like a theoretical, practical or application-oriented nature. But unfortunately not every lecture can be available every semester. Because of this, planning activities arise that are supported by the Proactive Class Schedule. Another reason for those activities is the existence of distinct frameworks describing the requirements for different vintages of students.

The repertoire of degrees to be awarded at the University of Magdeburg ranges from Bachelor/Master of Science, Engineering and even Arts. In the computer science profession the University of Magdeburg concentrates to Bachelor and Master of Science [Otto-von-Guericke-University Magdeburg, 2008].

The specification of Bachelor/Master Programs is standardized by using Credit Points according to the European Credit Transaction System: Per semester 30 credit points have to be achieved on average with a related study effort of 900 hours, the compulsory thesis for Bachelors programs is worth 6-12, and the Master Thesis 15-30 credit points. Bachelor programs can span over a period of three to four years and necessitate 180-200 credit points, Master programs should last between one and two years and require 60-120 credits [Braungarten et al., 2006b]. A more detailed overview is for example given in [Kernchen et al., 2007b].

7.2.2 Prototype

In this section, a first implemented prototype is presented. As outlined above, the goal of the planner is to create personal schedule by using semantic knowledge about study specifications to identify mandatory courses and to propose the most suitable lectures.

The proposed key features are:

- Central uniform access to various user related information
- User model that includes completed courses and the overall scholastics progress
- Integration of external course descriptions
- Central storage and management of course information and teaching material
- Collection of personal credit points and inclusion in course suggestions
- Support of teaching evaluation and inclusion in course suggestions

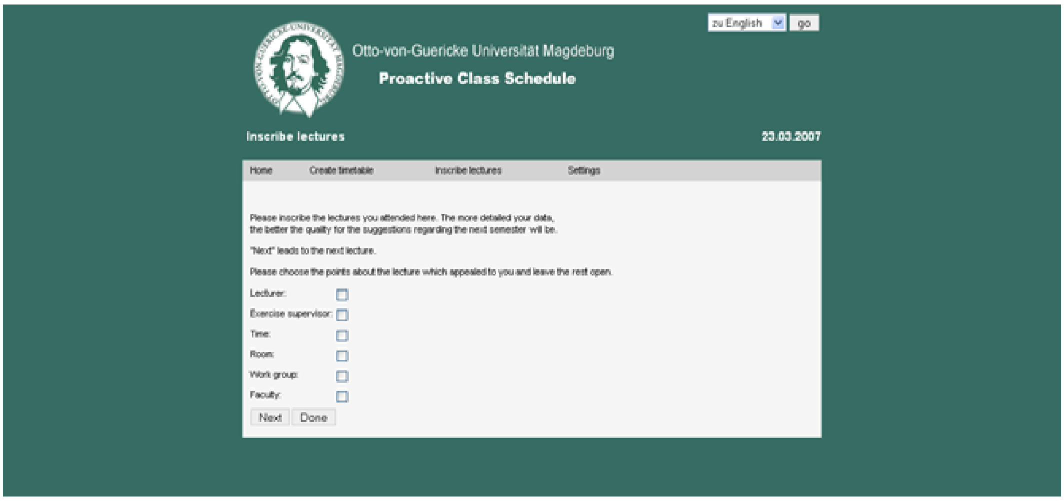


Figure 7.4: Individualizations within the Proactive Class Schedule [Kernchen et al., 2007a]

Manually added and automatically learned parameters (see Figure 7.4) are the basis of a proposed schedule (see Figure 7.5. Courses that are highlighted red are necessary by study specifications. Green highlighted courses are optional

7.2.3 Steps to an Empirical Analysis

To identify system improvement starting points by using empirical analysis, an approach to combine a bottom-up and top-down procedure was developed. Due to existing preliminary work in both directions, the combination of both approaches seems to be useful (see Figure 7.6).

This section describes existing data sources as well as possible investigations and possible outcomes. The main focus is to enhance the usefulness of the proactive class schedule in the long run [Mencke et al., 2008b].

The technical realization component of the later described improvement approaches is an Empirical Analysis Unit (EAU) to be integrated into the existing tool.

According to Figure 7.6, the bottom-up approach contains three steps: the identification of existing data sources, the selection of appropriate data and the empirical analysis

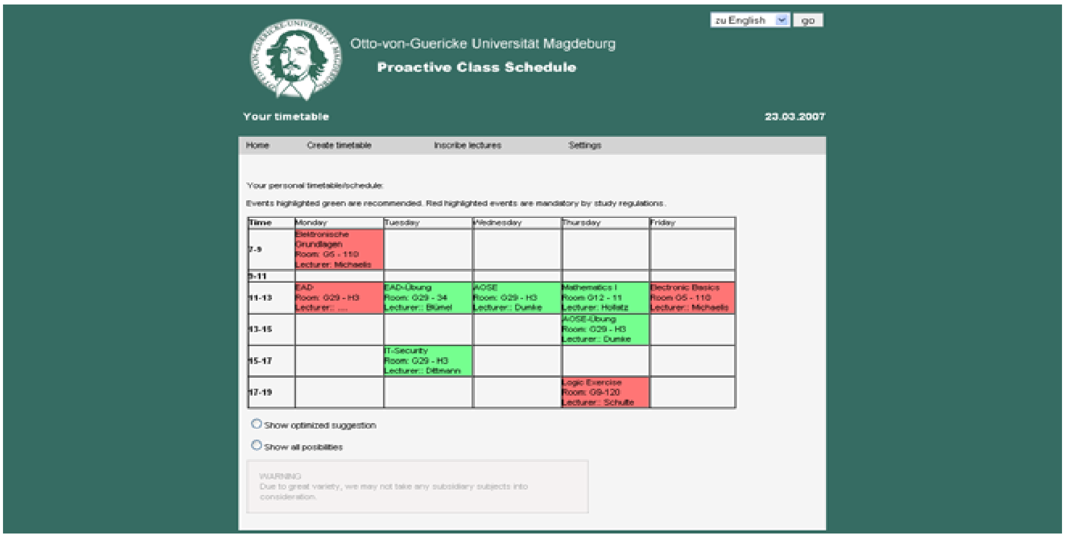


Figure 7.5: Personal Schedule in the Proactive Class Schedule [Kernchen et al., 2007a]

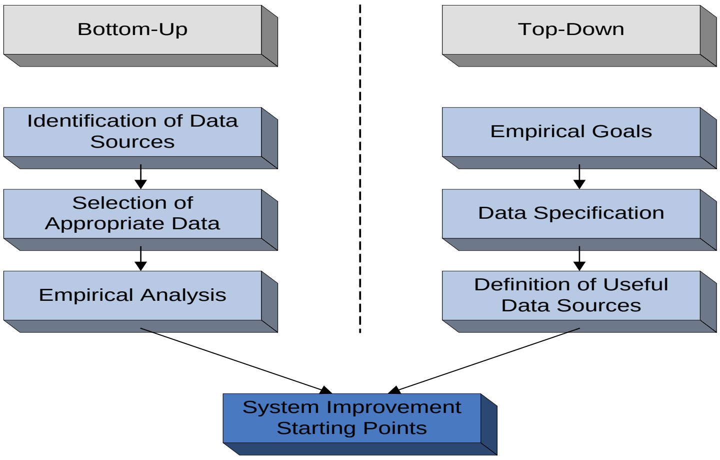


Figure 7.6: Y-Approach for Improvement Starting Point Identification [Mencke et al., 2008d]

of the selected data.

In general, the proactive class schedule contains a lot of data sources concerning different directions.

The first important type of lecture-related information covers the evaluation of distinct lectures. The Empirical Analysis Unit will take into account the following main data sources for the evaluation of lectures: the student's evaluation of lectures, the exams result and external sources.

The student's evaluation is an anonymous questionnaire-based procedure. The student should evaluate every course, exercise and seminar at the end of the semester. It is an already established procedure at the University of Magdeburg to evaluate and to improve the quality of teaching. The evaluation results for each teaching unit are available and can easily be analyzed by the EAU.

The exams' results can be integrated by using the average for each lecture. For more sophisticated results, anonymized results for each student can be stored in virtual study process descriptions.

The combination of aggregated student's evaluation and exam result can be an objective benchmark for every lecture. One empirical enhancement could be to privilege better ranked courses.

A more complex empirical enhancement could be to check if preliminary exams results in specific lectures are eligible for the successful attendance of another course. But that leads to numerous questions that still need to be discussed and answered, like: Which percentage of students with the same preliminary courses gains good results? Which failure rate is to expect by specific previous results?

According to the top-down approach, the goal of creating social networks among different groups of student was specified by the use of empirical data.

The empirical data can be efficiently modelled as an extension of the already existing user models. They are already stored inside the Proactive Class Schedule. Based on the previously described user model in [Kernchen and Dumke, 2007a], the OWL-based user model is enhanced with information concerning lectures and social networks to provide the means for appropriate data storage. Figure 7.7 provides an excerpt of basic information that is useful for our empirical analysis and thereby for the intended recommendation enhancement. The structure of the user model bases on categories from IEEE PAPI as well as IMS LIP (see Section 2.5.2).

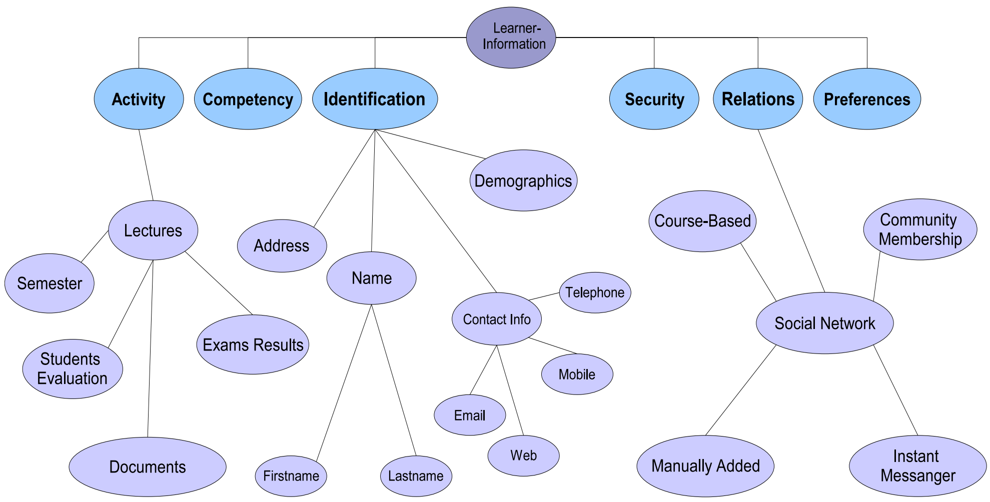


Figure 7.7: Extended User Model for EAU Data Storage [Mencke et al., 2008d]

By using this information, social networks among students as well as interest groups can be identified. The course selection of single users belonging to this interests group can be an important benchmark for other members of the same group. To create this social network for each student, the user of the proactive class schedule should be able to import contacts from existing community Web sites, instant messengers, etc. Additionally, the Empirical Analysis Unit can autonomously create connections between students which attended at the same course or seminar.

A second goal is, to take effort estimation for specific lectures into consideration. Here, an extension of the already performed lecture evaluation is proposed. Additional information of estimated efforts for learning materials, learning for the exam, preparation of the exercises, etc. can complete the effort estimation. Thereby, a better planning of the next semester based on these effort estimations can be performed. Of course, such estimations are subjective and can only provide an impression for the student. A more objective point of view can be derived from comparisons between the efforts for different courses. Thereby, the possibility increases that the current student already took part in a related course and can compare the efforts based on this shared information.

After describing the major possibilities and advantages some pitfalls regarding the collection an analysis of the mentioned data need to be taken into account. In learning environments, the data belonging to distinct students, courses and lectures should be stored in an anonymized manner. In general, the proactive class schedule should only have access to anonymized data but the merge of different data sources can annihilate the anonymous procedure. A personalization of student's evaluation is devastating in respect to psychological means and should be avoided. Especially the suspiciousness of students in the usage of the Proactive Class Schedule is destructive.

Another pitfall for empirical analysis can arise, if the Proactive Class Schedule prioritize one course high above average. In this case, the differentiated higher education programs can become uniform studies because all students attend at the same course.

Other data can have their origin in external sources like job profiles or job offers, political or industrial surveys about the future development of industry sectors or the current situation on the job market. Such information can be used to change course recommendations. By this, a smooth and short-term transition of the study plan towards current situations and requirements on the job market can be performed. A possible problem is the quality and up-to-dateness of the data sources itself.

Table 7.1 summarizes the main starting points of the empirical-analysis-based enhancement of the Proactive Class Schedule.

7.3 Discussion

In this chapter, two chosen proactive application scenarios are sketched.

For example, proactive educational courses in a distributed agent-based environment are introduced. Expected benefits for course distributors are a selective assortment of potential customers as well as justified arguments to convince them. Furthermore, their reputation will increase due a qualified presentation on the market. The learner benefits from reduced spam, adapted suggestions and offers as well as more effective learning for substantial advantages on the job market for their whole life.

As an additional application example, an approach for an adaptive information management system by using modern technologies for the Web 2.0 was described. The Proactive Class Schedule provides the opportunity to create personal schedules according to distinct user models and regarding study specifications, mandatory courses and course schedules.

Additionally, improvement approaches of the recommendation capabilities of the Proactive Class Schedule are proposed. The tool adapts to existing processes and does not determine them as for example demanded in [Bartels, 2007]. Based on several new and existing data sources, certain schemes are developed and described with their advantages and pitfalls. Additionally, new questions are presented for discussion. As shown, empirical data could be useful for further improvements of this Proactive Class

DATA SOURCE	EMPIRICAL ENHANCEMENT	POSSIBLE DRAWBACKS
Course evaluation	Selection of appropriated courses	Data integrity concerns
Course evaluation	Objective assessment	Anonymisation concerns
Course evaluation	Effort-based planning	Comparability of individual evaluations
User model	Social network	
Possible repetition of inappropriate decision		
User model	Identification of useful learning material	Lack of resources
Student exams	Objective assessment	Anonymisation concerns
External resources	Adaptation to actual job requirements	Future up-to-dateness

Table 7.1: Summary of Discussed Analysis Approaches [Mencke et al., 2008d]

Schedule. Thereby, the already existing tool for students will be improved to a new quality level.

The ideas and prototypes described in this chapter have a more visionaire character. But the integration of combined solutions as proposed in the previous chapters may lead to even more effective and innovative results.

8 Concluding Remarks and Further Work

The provision of content, especially information, is one of the most important lifelines in the current information society. Data, information and knowledge increase in importance as well as in their amount. It is necessary to pre-select, prepare and provide them in adequate ways to the user. Due to his individualism, an adaptive presentation is necessary to support his information consumption.

It is the aim of this dissertation to propose semantic technology approaches in order to enable proactive applications for quality-driven content provision. Semantic Web technologies, especially ontologies, are seen as the key approach with tremendous potential to future applications [Hendler, 2008]. This work focussed on the presentation of methodologies, application scenarios, architectures and prototypical tool implementations in order to improve selected aspects of the novel, quality-driven content provision framework "QuaD^2". It is the core of this work and presents the abstract frame as demanded as a goal of this thesis. There, the other solutions are integrated in order to improvement certain aspects in proactive way, based on ontologies.

8.1 Summary of the Dissertation

The results of a literature research presented in Chapter 3 revealed a lot of approaches of proactive approaches for the chosen domain of e-Learning. Another outcome of this research was the absence of an adequate framework for their classification. As a result, a novel framework with a special focus on proactiveness was defined based on existing standards.

Content and especially information provision is not very detailed in existing (content/information) lifecycle processes. Furthermore, a throughout focus on quality was missing. To close this gap the novel QuaD^2-Framework was introduced in Chapter 4. It enables a holistic view on quality criteria, not only for e-Learning content provision but for content in general, too.

Beginning with Chapter 5, the focus shifted to e-Learning as the chosen use case scenario. To enable education content provision following the QuaD^2-Framework, e-Learning processes are identified and adequate process description are defined. This categorization and argumentation is also a contribution of this work. The core of this chapter describes the applicability of ontologies to model didactical expertise as well as course structures. Together with the Ontology Framework of Chapter 6, they form the basis for the ontology-based content provision. This strong semantic basis throughout this complete process is seen as a further contribution of this work to the current research within the domain of e-Learning. Also, some new ontology-based quality attributes to evaluate Learning Objects, were part of this chapter. For the adapted information provision user models were identified as being important. The described analysis and suggested process solution for their their lifelong usage was not yet proposed in literature.

Chapter 6 focused on the presentation of content itself. Semantic technologies were introduced to improve this QuaD^2 subprocess. New ideas resulted in a number of metrics and prototypical implementations in order to present more appropriate and more adapted content to a user. Some existing ideas, like the individual learning paths were solved with a more detailed focus on ontologies. The other key area of this chapter was the presentation of a novel methodology for the modeling of inherently proactive graphical user interfaces. So far, agent technology was rarely used for this purpose, except for ideas like personal avatars.

Selected presentation areas for the introduced proactive and semantic approaches were part of Chapter 7. Especially, the concept of a proactive e-Learning course and the Proactive Class Schedule were described. Especially the second idea has not been a research focus so far, but is seen to have the potential for a commercial application.

Ontologies are not a new concept for e-Learning. But with this dissertation a throughout usage for the provision process is proposed and described. So far, only selected application scenarios were developed. Within the domain of e-Learning, especially-focused quality criteria are important. Following [Devedžić, 2006] and [Vouk et al., 1999], they can be differentiated into quality of educational content, quality of pedagogy and quality of the technical framework.

With this dissertation, improvement contributions to all above quality criteria dimensions were performed and are summarized as follows:

- Quality of educational content
  - Semantic relation to domain ontologies
  - Relation of educational content to didactical expertise
  - Provision of educational enrichment content
  - Course-driven, proactive recommendation framework
  - Study-specification-driven, proactive recommendation framework
  - Proactive, autonomous content provision/processing scenario
- Quality of pedagogy
  - Didactical ontologies
  - Modelling and classification of didactical expertise
  - Central provision framework for didactical expertise
  - Individualized learning path recommendations
  - Detection of off-topic learning steps
- Quality of the technical framework
  - Quality-driven content provision framework
  - Architectural framework with a focus on proactivity
  - Proactive, autonomous presentation framework

8.2 Suggestions for Further Work

The most important aspect being dedicated to future work is the complete instantiation of the $QuaD^2$-Framework. This dissertation presented advances for the initialization and presentation subprocesses.

A key area for an education-oriented implementation of the $QuaD^2$-Framework is the specification of more product-related quality attributes for Learning Objects. So far, mainly metrics exist that focus on the technical realization of content (e.g. metrics about text/homepage, figures/homepage). By this, an additional substantial contribution to quality-oriented e-Learning content provision can be achieved.

After the implementation of the entire framework, an empirical study should be performed to evaluate the performance of the proposed approach against existing e-Learning course presentation applications.

A sketched approach was the agent-based processing of central user models for life-long learning. This idea should be further developed and evaluated in terms of the quality of the generated unified user models. In fact, several future projects are planned and started to research about agent-based ontology processing on theoretical basis as well as from an application's point of view. Expected results will show the usefulness of the combination of both technologies.

Ontologies are the main semantic technology used in this dissertation. While analyzing it, the necessity of balance metrics became obvious. In this dissertation, introductory notes about knock-out criteria, gravity- and weighted-graphs-related approaches are described that may serve as starting points for more detailed research. Future work, being also related to ontologies, is for example extension of the developed Semantic Window approach towards a combination of the weight-based and the distance-based solutions.

Some of the presented approaches are architecturally sketched or only prototypically implemented so far. Their complete realization is dedicated to future work, too.

A Distance Definitions for the Semantic Window Approach

A.1 Concept Dimensions from the Datatype Property View

The dimensions of the distance related to the ontology's concepts having a datatype property as the focusing point are defined in equations A.1 to A.4. The single distance measures relate to the abstraction dimension distance c^{abs}, to the specialization dimension distance c^{spec}, to the sibling dimension distance c^{sib} and to the non-taxonomic dimension distance c^{ntax}. They measure the distance between the focusing point datatype property D_F and another concept C_j of the ontology. $C(D_j)$ determines the concept that a datatype property D_j belongs to.

$$c^{abs}(D_F, C_j) = f^{niv}(C(D_F)) - f^{niv}(C_j). \tag{A.1}$$

$$c^{spec}(D_F, C_j) = f^{niv}(C_j) - f^{niv}(C(D_F)). \tag{A.2}$$

$$c^{sib}(D_F, C_j) = f^{niv}(C(D_F)) - f^{niv}(f^{parent}(C(D_F), C_j)). \tag{A.3}$$

$$c^{ntax}(D_F, C_j) = \begin{cases} 0 & \text{if } C(D_F) \equiv C_j, \\ 1 & \text{if } \exists R_{Fj}(C(D_F), C_j), \\ c^{ntax}(D_F, C_i) & \\ +1 & \text{if } \exists c^{ntax}(D_F, C_i) = n \\ & \wedge \exists R_{ij}(C_i, C_j). \end{cases} \tag{A.4}$$

The equations above are restricted by: $C_i, C_j, C(D_F) \in C$ and $D_F \in D$. Equation A.1 is restricted by: $f^{niv}(C(D_F)) > f^{niv}(C_j)$ and $f^{tax}(C(D_F), C_j) \neq -1$. Equation A.2 is restricted by: $f^{niv}(C(D_F)) < f^{niv}(C_j)$ and $f^{tax}(C(D_F), C_j) \neq -1$. Equation A.3 is restricted by: $f^{niv}(C(D_F)) = f^{niv}(C_j)$ and $f^{niv}(f^{parent}(C(D_F), C_j)) < f^{niv}(C(D_F))$. Equation A.4 is further restricted by: $R_{Fj}, R_{ij} \in R_{ntax}$.

A.2 Datatype Property Dimensions from the Datatype Property View

The dimensions of the distance related to the ontology's datatype properties having a datatype property as the focusing point are defined in equations A.5 to A.8. The single distance measures relate to the abstraction dimension distance d^{abs}, to the specialization dimension distance d^{spec}, to the sibling dimension distance d^{sib} and to the non-taxonomic dimension distance d^{ntax}. They measure the distance between the focusing point datatype property D_F and a datatype property D_j of the ontology. $C(D_j)$ determines the concept that the datatype property D_j belongs to.

$$d^{abs}(D_F, D_j) = f^{niv}(C(D_F)) - f^{niv}(C(D_j)). \tag{A.5}$$

$$d^{spec}(D_F, D_j) = f^{niv}(C(D_j)) - f^{niv}(C(D_F)). \tag{A.6}$$

$$\begin{aligned} d^{sib}(D_F, D_j) = & f^{niv}(C(D_F)) - \\ & f^{niv}(f^{parent}(C(D_F), C(D_j))). \end{aligned} \tag{A.7}$$

$$d^{ntax}(D_F, D_j) = \begin{cases} 0 & \text{if } C(D_F) \equiv C(D_j), \\ 1 & \text{if } \exists R_{Fj}(C_F, C(D_j)), \\ d^{ntax}(D_F, D_i) & \\ +1 & \text{if } \exists d^{ntax}(D_F, D_i) = n \\ & \wedge \exists R_{ij}(C(D_i), C(D_j)). \end{cases} \tag{A.8}$$

The equations above are restricted by: $C_i, C_j, C(D_j), C(D_F) \in C$ and $D_F, D_j \in D$. Equation A.5 is restricted by: $f^{niv}(C(D_F)) > f^{niv}(C(D_j))$ and $f^{tax}(C(D_F), C(D_j)) \neq -1$. Equation A.6 is restricted by: $f^{niv}(C(D_F)) < f^{niv}(C(D_j))$ and $f^{tax}(C(D_F), C(D_j)) \neq -1$. Equation A.7 is restricted by: $R_{Fj}, R_{ij} \in R_{ntax}$.

A.3 Object Property Dimensions from the Datatype Property View

The dimensions of the distance related to the ontology's object properties having a datatype property as the focusing point are defined in equations A.9 to A.13. The single distance measures relate to the abstraction dimension distance r^{abs}, to the specialization dimension distance r^{spec} as well as to the (abstraction and specialization) sibling dimension distance $r^{sib^{abs}}$ and $r^{sib^{spec}}$. The non-taxonomic dimension distance is determined by r^{ntax}. They measure the distance between the focusing point datatype property D_F and an object property R_j of the ontology. $C(D_j)$ determines the concept that a datatype property D_j belongs to.

$$r^{abs}(D_F, R_j) = \begin{cases} 0 & \text{if } \nexists R_j(C(D_F), C_j), \\ 1 & \exists R_{Fj}(C(D_F), C_j) = 1 \\ & \wedge f^{niv}(C(D_F)) > f^{niv}(C_j), \\ r^{abs}(D_F, R_i) & \\ +1 & \text{if } r^{abs}(D_F, R_i) = n \\ & \wedge c^{abs}(C_i, C_j) = 1 \\ & \wedge R_j = R_{ij}(C_i, C_j) \\ & \wedge R_i = R_{hi}(C_h, C_i) \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) \\ & > f^{niv}(C_j). \end{cases} \tag{A.9}$$

$$r^{spec}(D_F, R_j) = \begin{cases} 0 & \text{if } \nexists R_j(C(D_F), C_j), \\ 1 & \text{if } \exists R_j(C(D_F), C_j) \\ & \wedge f^{niv}(C(D_F)) < f^{niv}(C_j), \\ r^{spec}(D_F, R_i) & \\ +1 & \text{if } r^{spec}(C(D_F), R_i) = n \\ & \wedge c^{spec}(C_i, C_j) = 1 \\ & \wedge R_j = R_{ij}(C_i, C_j) \\ & \wedge R_i = R_{hi}(C_h, C_i) \\ & \wedge f^{niv}(C_h) < f^{niv}(C_i) \\ & < f^{niv}(C_j). \end{cases} \tag{A.10}$$

$$\begin{aligned} r^{sib^{abs}}(D_F, R_j) = c^{sib}(C(D_F), C_h) | & c^{abs}(C_h, C_i) = 1 \\ & \wedge R_j = R_{ij}(C_i, C_j) \\ & \wedge f^{tax}(C_h, C_i) \neq -1 \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) > f^{niv}(C_j). \end{aligned} \tag{A.11}$$

$$\begin{aligned} r^{sib^{spec}}(D_F, R_j) = c^{sib}(C(D_F), C_h) | & c^{spec}(C_h, C_i) = 1 \\ & \wedge R_j = R_{ij}(C_i, C_j) \\ & \wedge f^{tax}(C_h, C_i) \neq -1 \\ & \wedge f^{niv}(C_h) < f^{niv}(C_i) < f^{niv}(C_j). \end{aligned} \tag{A.12}$$

$$r^{ntax}(D_F, R_j) = \begin{cases} 0 & \text{if } \nexists R_j(C(D_F), C_j) \\ 1 & \text{if } \exists R_j(C(D_F), C_j) \\ & \wedge \exists R_j = (C(D_F), C_j), \\ r^{ntax}(D_F, R_i) & \\ +1 & \text{if } \exists r^{ntax}(C_F, R_i) = \\ & f^{ntax}(C(D_F), C_i) = n \\ & \wedge R_i = R_{hi}(C_h, C_i) \\ & \wedge R_j = R_{ij}(C_i, C_j). \end{cases} \tag{A.13}$$

Equations A.9 to A.13 are restricted by $C(D_F) = C_F$, $D_F \in D$, $R_j, R_{Fj}, R_{ij} \in R$ and $C_F, C_i, C_j \in C$. Equations A.9 to A.12 are further restricted by $R_j \in R_{tax}$. For equation A.13 the following restriction applies: $R_i, R_j \in R_{ntax}$.

A.4 Instance Dimensions from the Datatype Property View

The dimensions of the distance related to the ontology's instances having a datatype property as the focusing point are defined in equations A.14 to A.17. The single distance measures relate to the abstraction dimension distance i^{abs}, to the specialization dimension distance i^{spec} and to the sibling dimension distance i^{sib} as well as the non-taxonomic dimension distance is measured by i^{ntax}. They measure the distance between the focusing point datatype property D_F and an instance I_j of the ontology. $C(I_j)$ is the concept that an instance I_j is instantiated of and $C(D_j)$ is the concept that a datatype property D_j belongs to.

$$i^{abs}(D_F, I_j) = f^{niv}(C(D_F)) - f^{niv}(C(I_j)). \tag{A.14}$$

$$i^{spec}(D_F, I_j) = f^{niv}(C(I_j)) - f^{niv}(C(D_F)). \tag{A.15}$$

$$\begin{aligned} i^{sib}(D_F, I_j) = & f^{niv}(C(D_F)) \\ & - f^{niv}(f^{parent}(C(D_F), C(I_j))). \end{aligned} \tag{A.16}$$

$$i^{ntax}(D_F, I_j) = f^{ntax}(C(D_F), C(I_j)). \tag{A.17}$$

The equations above are restricted by: $C(D_F), C(I_j) \in C$ and $I_j \in I$. Equation A.14 is restricted by: $f^{niv}(C(D_F)) > f^{niv}(C(I_j))$ and $f^{tax}(C(D_F), C(I_j)) \neq -1$. Equation A.15 is restricted by: $f^{niv}(C(D_F)) < f^{niv}(C(I_j))$ and $f^{tax}(C(D_F), C(I_j)) \neq -1$. Equation A.16 is restricted by: $f^{niv}(C(D_F)) = f^{niv}(C(I_j))$ and $f^{niv}(f^{parent}(C(D_F), C(I_j))) < f^{niv}(C(D_F))$.

A.5 Concept Dimensions from the Object Property View

The dimensions of the distance related to the ontology's concepts having an object property as the focusing point are defined in equations A.18 to A.22. The single distance measures relate to the abstraction dimension distance c^{abs}, to the specialization dimension distance c^{spec} as well as to the (abstraction and specification) sibling dimension distance $c^{sib^{abs}}$ and $c^{sib^{spec}}$. The non-taxonomic dimension distance is determined with c^{ntax}. They measure the distance between the focusing point object property R_F and a concept C_j of the ontology.

$$\begin{aligned} c^{abs}(R_F, C_j) =& f^{niv}(C_i) - f^{niv}(C_j) \mid R_F(C_h, C_i) \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) > f^{niv}(C_j). \end{aligned} \tag{A.18}$$

$$\begin{aligned} c^{spec}(R_F, C_j) =& f^{niv}(C_j) - f^{niv}(C_i) \mid R_F(C_h, C_i) \\ & \wedge f^{niv}(C_h) < f^{niv}(C_i) < f^{niv}(C_j). \end{aligned} \tag{A.19}$$

$$\begin{aligned} c^{sib^{abs}}(R_F, C_j) = c^{sib}(C_i, C_k) | & c^{abs}(C_j, C_k) = 1 \\ & \wedge R_F = R_{hi}(C_h, C_i) \\ & \wedge f^{tax}(C_k, C_j) \neq -1 \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) \\ & \wedge f^{niv}(C_i) = f^{niv}(C_k) \\ & \wedge f^{niv}(C_k) > f^{niv}(C_j). \end{aligned} \tag{A.20}$$

$$\begin{aligned} c^{sib^{spec}}(R_F, C_j) = c^{sib}(C_h, C_k) | & c^{spec}(C_j, C_k) = 1 \\ & \wedge R_F = R_{hi}(C_h, C_i) \\ & \wedge f^{tax}(C_j, C_k) \neq -1 \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) \\ & \wedge f^{niv}(C_h) = f^{niv}(C_k) \\ & \wedge f^{niv}(C_k) < f^{niv}(C_j). \end{aligned} \tag{A.21}$$

$$\begin{aligned} c^{ntax}(R_F, C_j) = & min(f^{ntax}(C_h, C_j), f^{ntax}(C_i, C_j))| \\ & R_F = R_{hi}(C_h, C_i). \end{aligned} \tag{A.22}$$

The equations above are restricted by: $R_F \in R$ and $C_h, C_i, C_j, C_k \in C$. Equation A.22 is further restricted by $R_F \in R_{ntax}$.

A.6 Datatype Property Dimensions from the Object Property View

The dimensions of the distance related to the ontology's datatype properties having an object property as the focusing point are defined in equations A.23 to A.27. The single distance measures relate to the abstraction dimension distance d^{abs}, to the specialization dimension distance d^{spec} as well as to the (abstraction and specialization) sibling dimension distance $d^{sib^{abs}}$ and $d^{sib^{spec}}$. The non-taxonomic dimension distance is determined by d_{ntax}. They measure the distance between the focusing point object property R_F and a datatype property D_j of the ontology. $C(D_j)$ determines the concept that a datatype property D_j belongs to.

$$\begin{aligned} d^{abs}(R_F, D_j) =& f^{niv}(C_i) - f^{niv}(C(D_j)) \mid R_F(C_h, C_i) \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) > f^{niv}(C_j). \end{aligned} \tag{A.23}$$

$$\begin{aligned} d^{spec}(R_F, D_j) =& f^{niv}(C(D_j)) - f^{niv}(C_i) \mid R_F(C_h, C_i) \\ & \wedge f^{niv}(C_h) < f^{niv}(C_i) < f^{niv}(C_j). \end{aligned} \tag{A.24}$$

$$\begin{aligned} d^{sib^{abs}}(R_F, D_j) = c^{sib}(C_i, C_k) | & c^{abs}(C(D_j), C_k) = 1 \\ & \wedge R_F = R_{hi}(C_h, C_i) \\ & \wedge f^{tax}(C_k, C(D_j)) \neq -1 \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) \\ & \wedge f^{niv}(C_i) = f^{niv}(C_k) \\ & \wedge f^{niv}(C_k) > f^{niv}(C(D_j)). \end{aligned} \tag{A.25}$$

$$\begin{aligned} d^{sib^{spec}}(R_F, D_j) = c^{sib}(C_h, C_k) | & c^{spec}(C(D_j), C_k) = 1 \\ & \wedge R_F = R_{hi}(C_h, C_i) \\ & \wedge f^{tax}(C(D_j), C_k) \neq -1 \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) \\ & \wedge f^{niv}(C_h) = f^{niv}(C_k) \\ & \wedge f^{niv}(C_k) < f^{niv}(C(D_j)). \end{aligned} \tag{A.26}$$

$$\begin{aligned} d^{ntax}(R_F, D_j) =& min(f^{ntax}(C_h, C(D_j)), \\ & f^{ntax}(C_i, C(D_j))) | R_F = R_{hi}(C_h, C_i). \end{aligned} \tag{A.27}$$

The equations above are restricted by: $R_F \in R$, $D_j \in D$ and $C(D_j), C_h, C_i, C_k \in C$. Equation A.27 is further restricted by $R_F \in R_{ntax}$.

A.7 Object Property Dimensions from the Object Property View

The dimensions of the distance related to the ontology's object properties having an object property as the focusing point are defined in equations A.28 to A.31. The single distance measures relate to the abstraction dimension distance r^{abs}, to the specialization dimension distance r^{spec}, to the sibling dimension distance r^{sib} as well as to the non-taxonomic dimension distance r^{ntax}. They measure the distance between the focusing point object property R_F and a object property R_j of the ontology.

$$\begin{aligned} r^{abs}(R_F, R_j) = f^{niv}(C_h) - f^{niv}(C_j) | R_F = R_{gh}(C_g, C_h) \\ \wedge R_j = R_{ij}(C_i, C_j) \\ \wedge f^{niv}(C_g) > f^{niv}(C_h) \\ \wedge f^{niv}(C_i) > f^{niv}(C_j) \\ \wedge f^{niv}(C_h) \geq f^{niv}(C_j). \end{aligned} \tag{A.28}$$

$$\begin{aligned} r^{spec}(R_F, R_j) = f^{niv}(C_j) - f^{niv}(C_h) | R_F = R_{gh}(C_g, C_h) \\ \wedge R_j = R_{ij}(C_i, C_j) \\ \wedge f^{niv}(C_g) > f^{niv}(C_h) \\ \wedge f^{niv}(C_i) > f^{niv}(C_j) \\ \wedge f^{niv}(C_h) \leq f^{niv}(C_j). \end{aligned} \tag{A.29}$$

$$\begin{aligned} r_{sib}(R_F, R_j) = c_{sib}(C_h, C_j) | R_F = R_{gh}(C_g, C_h) \\ \wedge R_j = R_{ij}(C_i, C_j) \\ \wedge f^{niv}(C_g) > f^{niv}(C_h) \\ \wedge f^{niv}(C_i) > f^{niv}(C_j) \\ \wedge f^{niv}(C_h) = f^{niv}(C_j). \end{aligned} \tag{A.30}$$

$$r^{ntax}(R_F, R_j) = \begin{cases} 0 & \text{if } R_F \equiv R_j, \\ 1 & \text{if } R_F = R_{gh}(C_g, C_h) \\ & \wedge R_j = R_{ij}(C_i, C_j) \\ & \wedge((C_h \equiv C_i) \vee \\ & (C_g \equiv C_j)), \\ r^{ntax}(R_F, R_i) & \\ +1 & if r^{ntax}(R_F, R_i) = n \\ & \wedge R_j = R_{ij}(C_i, C_j) \\ & \wedge R_i = R_{gh}(C_g, C_h) \\ & \wedge((C_g \equiv C_i) \vee \\ & (C_h \equiv C_j)), \\ -1 & \text{otherwise.} \end{cases} \tag{A.31}$$

Equations A.28 to A.30 are restricted by: $R_F, R_j \in R$ and $C_g, C_h, C_i, C_j \in C$. Equation A.31 is restricted by: $R_F, R_i, R_j \in R_{ntax}$, $f^{niv}(C_g)g > f^{niv}(C_h)$ and $f^{niv}(C_i) > f^{niv}(C_j)$.

A.8 Instance Dimensions from the Object Property View

The dimensions of the distance related to the ontology's instances having an object property as the focusing point are defined in equations A.32 to A.36. The single distance measures relate to the abstraction dimension distance i^{abs}, to the specialization dimension distance i^{spec} as well as to the (abstraction and specification) sibling dimension distance $i^{sib^{abs}}$ and $i^{sib^{spec}}$. The non-taxonomic dimension distance is determined with i^{ntax}. They measure the distance between the focusing point object property R_F and an instance I_j of the ontology. $C(I_j)$ is the concept that an instance I_j is instantiated of.

$$\begin{aligned} i^{abs}(R_F, I_j) =& f^{niv}(C_i) - f^{niv}(C(I_j))| \\ & R_F = R_{hi}(C_h, C_i) \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) \geq f^{niv}(C(I_j)). \end{aligned} \tag{A.32}$$

$$\begin{aligned} i^{spec}(R_F, I_j) =& f^{niv}(C(I_j)) - f^{niv}(C_h)| \\ & R_F = R_{hi}(C_h, C_i) \\ & \wedge f^{niv}(C(I_j)) \geq f^{niv}(C_h) > f^{niv}(C_i). \end{aligned} \tag{A.33}$$

$$\begin{aligned} i^{sib^{abs}}(R_F, I_j) = c^{sib}(C_i, C_k)| & c^{abs}(C(I_j), C_k) = 1 \\ & \wedge R_F = R_{hi}(C_h, C_i) \\ & \wedge f^{tax}(C_k, C(I_j)) \neq -1 \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) \\ & \wedge f^{niv}(C_i) = f^{niv}(C_k) \\ & \wedge f^{niv}(C_k) > f^{niv}(C(I_j)). \end{aligned} \tag{A.34}$$

$$\begin{aligned} i^{sib^{spec}}(R_F, I_j) = c^{sib}(C_h, C_k)| & c^{spec}(C(I_j), C_k) = 1 \\ & \wedge R_F = R_{hi}(C_h, C_i) \\ & \wedge f^{tax}(C(I_j), C_k) \neq -1 \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) \\ & \wedge f^{niv}(C_h) = f^{niv}(C_k) \\ & \wedge f^{niv}(C_k) < f^{niv}(C(I_j)). \end{aligned} \tag{A.35}$$

$$\begin{aligned} i^{ntax}(R_F, I_j) = & min(f^{ntax}(C_h, C(I_j)), f^{ntax}(C_i, C(I_j)))| \\ & R_F = R_{hi}(C_h, C_i). \end{aligned} \tag{A.36}$$

The equations above are restricted by: $R_F \in R$, $C_h, C_i, C_k, C(I_j) \in C$ and $I_j \in I$. Equations A.32 to A.35 are further restricted by $R_F \in R_{tax}$. Equation A.36 is further restricted by $R_F \in R_{ntax}$.

A.9 Concept Dimensions from the Instance View

The dimensions of the distance related to the ontology's concepts having an instance as the focusing point are defined in equations A.37 to A.40. The single distance measures relate to the abstraction dimension distance c^{abs}, to the specialization dimension distance c^{spec}, to the sibling dimension distance c^{sib} and to the non-taxonomic dimension distance c^{ntax}. They measure the distance between the focusing point instance I_F and a concept C_j of the ontology. $C(I_F)$ determines the concept, that the instance I_F is instantiated of.

$$c^{abs}(I_F, C_j) = f^{niv}(C(I_F)) - f^{niv}(C_j). \tag{A.37}$$

$$c^{spec}(I_F, C_j) = f^{niv}(C_j) - f^{niv}(C(I_F)). \tag{A.38}$$

$$c^{sib}(I_F, C_j) = f^{niv}(C(I_F)) - f^{niv}(f^{parent}(C(I_F), C_j)). \tag{A.39}$$

$$c^{ntax}(I_F, C_j) = f^{ntax}(C(I_F), C_j). \tag{A.40}$$

The equations above are restricted by: $C(I_F), C_i, C_j \in C$ and $I_F \in I$. Equation A.37 is restricted by: $f^{niv}(C(I_F)) > f^{niv}(C_j)$ and $f^{tax}(C(I_F), C_j) \neq -1$. Equation A.38 is restricted by: $f^{niv}(C(I_F)) < f^{niv}(C_j)$ and $f^{tax}(C(I_F), C_j) \neq -1$. Equation A.39 is restricted by: $f^{niv}(C(I_F)) = f^{niv}(C_j)$ and $f^{niv}(f^{parent}(C(I_F), C_j)) < f^{niv}(C(I_F))$.

A.10 Datatype Property Dimensions from the Instance View

The dimensions of the distance related to the ontology's datatype properties having an instance as the focusing point are defined in equations A.41 to A.44. The single distance measures relate to the abstraction dimension distance d^{abs}, to the specialization dimension distance d^{spec}, to the sibling dimension distance d^{sib} and to the non-taxonomic dimension distance d^{ntax}. They measure the distance between the focusing point instance I_F and a datatype property D_j of the ontology. $C(D_j)$ is the concept that a datatype property D_j belongs to and $C(I_F)$ is the concept that the instance I_F is instantiated of.

$$d^{abs}(I_F, D_j) = f^{niv}(C(I_F)) - f^{niv}(C(D_j)). \tag{A.41}$$

$$d^{spec}(I_F, D_j) = f^{niv}(C(D_j)) - f^{niv}(C(I_F)). \tag{A.42}$$

$$d^{sib}(I_F, D_j) = f^{niv}(C(I_F)) - f^{niv}(f^{parent}(C(I_F), C(D_j))). \tag{A.43}$$

$$d^{ntax}(I_F, D_j) = f^{ntax}(C(I_F), C(D_j)). \tag{A.44}$$

The equations above are restricted by: $C(I_F), C_i, C_j, C(D_j) \in C$, $I_F \in I$ and $D_j \in D$. Equation A.41 is restricted by: $f^{niv}(C(I_F)) > f^{niv}(C(D_j))$ and $f^{tax}(C(I_F), C(D_j)) \neq -1$. Equation A.42 is restricted by: $f^{niv}(C(I_F)) < f^{niv}(C(D_j))$ and $f^{tax}(C(I_F), C(D_j)) \neq -1$. Equation A.43 is restricted by: $f^{niv}(C(I_F)) = f^{niv}(C(D_j))$ and $f^{niv}(f^{parent}(C(I_F), C(D_j))) < f^{niv}(C(I_F))$.

A.11 Object Property Dimensions from the Instance View

The dimensions of the distance related to the ontology's object properties having an instance as the focusing point are defined in equations A.45 to A.49. The single distance measures relate to the abstraction dimension distance r^{abs}, to the specialization dimension distance r^{spec} as well as to the (abstraction and specialization) sibling dimension distance $r^{sib^{abs}}$ and $r^{sib^{spec}}$. The non-taxonomic dimension distance is measured by r^{ntax}. They measure the distance between the focusing point instance I_F and an object property R_j of the ontology. $C(I_F)$ is the concept that the instance I_F is instantiated of.

$$r^{abs}(I_F, R_j) = \begin{cases} 0 & \text{if } \nexists R_j(C(I_F), C_j), \\ 1 & \text{if } \exists R_j = R_{Fj}(C(I_F), C_j) \\ & \wedge f^{niv}(C(I_F)) > f^{niv}(C_j), \\ r^{abs}(I_F, R_i) & \\ +1 & \text{if } r^{abs}(I_F, R_i) = n \\ & \wedge c^{abs}(C_i, C_j) = 1 \\ & \wedge R_j = R_{ij}(C_i, C_j) \\ & \wedge R_i = R_{hi}(C_h, C_i) \\ & \wedge f^{niv}(C_h) > f^{niv}(C_i) \\ & > f^{niv}(C_j). \end{cases} \tag{A.45}$$

$$r^{spec}(I_F, R_j) = \begin{cases} 0 & \text{if } \nexists R_j(C(I_F), C_j), \\ 1 & \exists R_j = R_{Fj}(C(I_F), C_j) \\ & \wedge f^{niv}(C(I_F)) < f^{niv}(C_j), \\ r^{spec}(I_F, R_i) & \\ +1 & \text{if } r^{spec}(I_F, R_i) = n \\ & \wedge c^{spec}(C_i, C_j) = 1 \\ & \wedge R_j = R_{ij}(C_i, C_j) \\ & \wedge R_i = R_{hi}(C_h, C_i) \\ & \wedge f^{niv}(C_h) < f^{niv}(C_i) \\ & < f^{niv}(C_j). \end{cases} \tag{A.46}$$

$$\begin{aligned} r^{sib^{abs}}(I_F, R_j) = c^{sib}(C(I_F), C_h) | c^{abs}(C_h, C_i) = 1 \\ \wedge\ R_j = R_{ij}(C_i, C_j) \\ \wedge\ f^{tax}(C_h, C_i) \neq -1 \\ \wedge\ f^{niv}(C_h) > f^{niv}(C_i) > f^{niv}(C_j). \end{aligned} \tag{A.47}$$

$$\begin{aligned} r^{sib^{spec}}(I_F, R_j) = c^{sib}(C(I_F), C_h) | c^{spec}(C_h, C_i) = 1 \\ \wedge\ R_j = R_{ij}(C_i, C_j) \\ \wedge\ f^{tax}(C_h, C_i) \neq -1 \\ \wedge\ f^{niv}(C_h) < f^{niv}(C_i) < f^{niv}(C_j). \end{aligned} \tag{A.48}$$

$$r^{ntax}(I_F, R_j) = \begin{cases} 0 & \text{if } \nexists R_j(C(I_F), C_j), \\ 1 & \text{if } c^{ntax}(C(I_F), C_j) = 1 \\ & \wedge \exists R_j(C(I_F), C_j), \\ r^{ntax}(I_F, R_i) & \\ +1 & \text{if } \exists r^{ntax}(I_F, R_i) = \\ & f^{ntax}(C(I_F), C_i) = n \\ & \wedge R_i = R_{hi}(C_h, C_i) \\ & \wedge R_j = R_{ij}(C_i, C_j). \end{cases} \tag{A.49}$$

The equations above are restricted by $R_j, R_{Fj}, R_{ij} \in R$, $I_F \in I$ and $C(I_F), C_h, C_i, C_j \in C$. Equations A.45 to A.48 are further restricted by $R_{Fj}, R_i, R_j \in R_{tax}$. For equation A.49 the following restriction applies: $R_i, R_j \in R_{ntax}$.

A.12 Instance Dimensions from the Instance View

The dimensions of the distance related to the ontology's instances having an instance as the focusing point are defined in equations A.50 to A.53. The single distance measures relate to the abstraction dimension distance i^{abs}, to the specialization dimension distance

i^{spec} and to the sibling dimension distance i^{sib} as well as the non-taxonomic dimension distance is measured by i^{ntax}. They measure the distance between the focusing point instance I_F and another instance I_j of the ontology. $C(I_j)$ is the concept that an instance I_j is instantiated of.

$$i^{abs}(I_F, I_j) = f^{niv}(C(I_F)) - f^{niv}(C(I_j)). \tag{A.50}$$

$$i^{spec}(I_F, I_j) = f^{niv}(C(I_j)) - f^{niv}(C(I_F)). \tag{A.51}$$

$$i^{sib}(I_F, I_j) = f^{niv}(C(I_F)) - f^{niv}(f^{parent}(C(I_F), C(I_j))). \tag{A.52}$$

$$i^{ntax}(I_F, I_j) = f^{ntax}(C(I_F), C(I_j)). \tag{A.53}$$

The equations above are restricted by: $C(I_F), C_i, C_j, C(I_j) \in C$ and $I_F, I_j \in I$. Equation A.50 is restricted by: $f^{niv}(C(I_F)) > f^{niv}(C(I_j))$ and $f^{tax}(C(I_F), C(I_j)) \neq -1$. Equation A.51 is restricted by: $f^{niv}(C(I_F)) < f^{niv}(C(I_j))$ and $f^{tax}(C(I_F), C(I_j)) \neq -1$. Equation A.52 is restricted by: $f^{niv}(C(I_F)) = f^{niv}(C(I_j))$ and $f^{niv}(f^{parent}(C(I_F), C(I_j))) < f^{niv}(C(I_F))$.

List of Tables

List of Figures

List of Abbreviations

3GPP	3rd Generation Partnership Project
A-SPICE	Automotive SPICE
AACR2	Anglo-American Cataloguing Rules
ABEL	Agent-Based E-Learning
ACL	Agent Communication Language
ADL	Advanced Distributed Learning
AE	Administration Environment
AI	Artificial Intelligence
AMOL	Agent-Mediated Online Learning
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ASTD	American Society for Training & Development
BDI	Belief, Desire, Intention
BPEL	Business Process Execution Language
BPMN	Business Process Modeling Notation
CAI	Computer Assisted Instruction
CAM	Content Aggregation Model
CBSE	Component-Based Software Engineering
CBT	Computer-Based Training
CCO	Chief Content Officer
CD-ROM	Compact Disc Read-Only Memory
CE	Content Environment
CLO	Chief Learning Officer
CMMI	Capability Maturity Model Integration
CMS	Content Management System
COA	Content Agent
COBIT	Control Objectives for Information and Related Technology
CORBA	Object Request Broker Architecture
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
DAML	DARPA Agent Markup Language
DARPA	Defense Advanced Research Projects Agency
DB	Database
DBMS	Database Management System
DC	Dublin Core
DCMI	Dublin Core Metadata Initiative
DDI	Data Documentation Initiative
DE	Delivery Environment
DeLC	Distributed e-Learning Center

DICOM	Imaging and Communications in Medicine
DIN	Deutsches Institut für Normung e. V.
DIN DOM	DIN Didactical Object Model
DL	Description Logic
DoD	Department of Defense
DVD	Digital Versatile Disc
EAD	Encoded Archival Description
EAU	Empirical Analysis Unit
EER	Extended Entity Relationship
EIA	Electronic Industries Alliance
EJB	Enterprise Java Beans
ELF	E-Learning Framework
EML	Educational Modeling Language
eMM	E-Learning Maturity Model
ER	Entity Relationship
F-SMILE	File-Store Manipulation Intelligent Learning Environment
FIPA	Foundation for Intelligent Physical Agents
GIF	Graphics Interchange Format
GLOB	Learning Object of the educational Grid
GPRS	General Packet Radio Service
GQM	Goal, Question, Metric
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HSCSD	High Speed Circuit Switched Data
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDL	Interface Definition Language
IE	Interaction Environment
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IGES	Initial Graphics Exchange Specification
IIS	Fraunhofer-Institut für Integrierte Schaltungen
ILMDA	Intelligent Learning Materials Delivery Agents
IMDL	Instructional Material Description Language
I-MINDS	Intelligent Multiagent Infrastructure for Distributed Systems in Education
IRS-II	Internet Reasoning Service
ISAD (G)	International Standard Archival Description (General)
ISO	International Organization for Standardization
IT	Information Technology
ITIL	IT Infrastructure Library
JADE	Java Agent DEvelopment Framework
JPEG	Joint Photographic Experts Group
K-InCA	Knowledge Intelligent Conversational Agents
KOD	Knowledge On Demand
LAN	Local Area Network

LCMS	Learning Content Management System
LD	Learning Design
LIP	Learner Information Package
LMML	Learning Material Markup Language
LMS	Learning Management System
LO	Learning Object
LOM	Learning Object Metadata
LOML	Learning Object Markup Language
LPA	Learning Path Agent
LTSA	IEEE Learning Technology Systems Architecture
LTSC	IEEE Learning Technology Standards Committee
LUE	Learning Unit Environment
MARC	MAchine-Readable Cataloging
MAS	Multiagent System
MASEL	Multi-Agent System for e-Learning and Skill Management
MIL	Military
MIME	Multipurpose Internet Mail Extensions
MP3	MPEG-1 Audio Layer 3
MP4	MPEG 4
MPEG	Moving Picture Experts Group
NS	Namespace
OIL	Ontology Inference Layer
OMS	Ontology Management System
OS	Operating System
OWL	Web Ontology Language
OWL-S	OWL-Schema
QA	Quality Attribute
QuaD^2	Quality Driven Design
PAPI	IEEE Personal and Private Information Project
PBL	Problem-Based Learning
PC	Personal Computer
PE	Presentation Environment
PERT	Program Evaluation and Review Technique
PELO	Production of E-Learning Offerings
PLATO	Programmed Logic for Automatic Teaching Operations
PMBOK	Project Management Body Of Knowledge
PNG	Portable Network Graphics
RDF	Resource Description Framework
RDF/S	RDF Schema
RFC	Requests for Comments
RLO	Reusable Learning Object
RMI	Remote Method Invocation
RSLP	Research Support Libraries Programme
RTF	Rich Text Format
SGML	Standard Generalized Markup Language
SCAMPI	Standard CMMI Assessment Method for Process Improvement

SCO	Shareable Content Object
SCORM	Shareable Content Object Reference Model
SIF	Schools Interoperability Framework
SLO	Smart Learning Object
SMA	Skill Manager Agent
SMIL	Synchronized Multimedia Integration Language
SMS	Short Message Service
SOA	Service Oriented Architecture
SOX	Sarbanes-Oxley Act
SPICE	Software Process Improvement and Capability dEtermination
SQL	Structured Query Language
SSA	Student Assistant Agent
SWEBOK	Software Engineering Body Of Knowledge
SW-CMM	Software Maturity Model
TCP/IP	Transmission Control Protocol/Internet Protocol
TEI	Text Encoding Initiative
TICCIT	Time-shared Interactive Computer Controlled Information Television
TIFF	Tagged Image File Format
TL	Telecom Leadership
TML	Tutorial Markup Language
TS	Technical Specification
UDDI	Universal Description, Discovery and Integration
UI	User Interface
UML	Unified Modeling Language
UMTS	Universal Mobile Telecommunications System
UPA	User Profile Agent
URI	Uniform Resource Identifier
VIRGE	Virtual Reality Game for English
VM-XT	V-Model XT
VRA	Visual Resources Association
VRML	Virtual Reality Modeling Language
VTT	Video Tele-Training
W3C	World Wide Web Consortium
WAP	Wireless Application Protocol
WBT	Web-Based Training
WE	Working Environment
WML	Wireless Markup Language
WSDL	Web Service Description Language
WSMF	Web Service Modeling Framework
WWW	World Wide Web
X3D	Extensible 3D
XHTML	Extensible Hypertext Markup Language
XMF	Extensible Music Format
XML	Extensible Markup Language
XSD	XML Schema Definition
XSLT	Extensible Stylesheet Language Transformation

Bibliography

[Abran et al., 2001] Abran, A., Moore, J. W., Bourque, P., and Dupuis, R., editors (2001). *Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society Press.

[Advanced Distributed Learning (ADL), 2006a] Advanced Distributed Learning (ADL) (2006a). Sharable Content Object Reference Model (SCORM) 2004 3rd Edition Content Aggregation Model.

[Advanced Distributed Learning (ADL), 2006b] Advanced Distributed Learning (ADL) (2006b). Sharable Content Object Reference Model (SCORM) 2004 3rd Edition Overview.

[Ahern et al., 2008] Ahern, D. M., Clouse, A., and Turner, R. (2008). *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*. The SEI Series in Software Engineering. Addison-Wesley Professional, 3 edition.

[Alani and Brewster, 2005] Alani, H. and Brewster, C. (2005). Ontology Ranking Based on the Analysis of Concept Structures. In *K-CAP '05: Proceedings of the 3rd International Conference on Knowledge Capture*, pages 51–58, New York, NY, USA. ACM Press.

[Amelung et al., 2008] Amelung, M., Forbig, P., and Rösner, D. (2008). Towards Generic and Flexible Web Services for E-Assessment. In *Proceedings of the 13th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE '08)*, Madrid, Spain.

[American Heritage, 2003a] American Heritage (2003a). The American Heritage® Dictionary of the English Language: Definition of Knowledge. Link: http://www.thefreedictionary.com/knowledge.

[American Heritage, 2003b] American Heritage (2003b). The American Heritage® Dictionary of the English Language: Definition of Proactivity. Link: http://www.thefreedictionary.com/proactivity.

[American Society for Training & Development (ASTD), 2008] American Society for Training & Development (ASTD) (2008). E-learning Glossary. Link: http://www.learningcircuits.org/glossary.html.

[Angehrn et al., 2001] Angehrn, A., Nabeth, T., and Roda, C. (2001). Towards Personalised, Socially Aware and Active e-Learning Systems - Illustrated with the agent-based system K-InCA. White Paper of Centre for Advanced Learning Technologies (CALT) - INSEAD.

[Anghel and Salomie, 2003] Anghel, C. and Salomie, I. (2003). JADE Based Solutions for Knowledge Assessment in eLearning Environments. *EXP - in search of innovation (Special Issue on JADE)*.

[Antoniou and van Harmelen, 2004] Antoniou, G. and van Harmelen, F. (2004). *A Semantic Web Primer*. MIT Press.

[Aroyo and Dicheva, 2004] Aroyo, L. and Dicheva, D. (2004). The New Challenges for E-learning: The Educational Semantic Web. *Educational Technology & Society*, 7(4):59–69.

[Aroyo et al., 2004] Aroyo, L., Dicheva, D., Brusilovsky, P., Díaz, P., Dimitrova, V., Duval, E., Greer, J. E., Hirashima, T., Hoppe, H. U., Houben, G.-J., Ikeda, M., Kay, J., Kinshuk, Melis, E., Mitrovic, T., Naeve, A., Nykänen, O., Paquette, G., Retalis, S., Sampson, D. G., Sinitsa, K. M., Soller, A., Staab, S., Vassileva, J., Verdejo, F., and Wagner, G. (2004). Workshop on Applications of Semantic Web Technologies for E-learning. In *Intelligent Tutoring Systems*, page 912.

[Aumüller and Rahm, 2007] Aumüller, D. and Rahm, E. (2007). Caravela: Semantic Content Management with Automatic Information Integration and Categorization (System Description). In *Proceedings of the 4th European Semantic Web Conference: The Semantic Web: Research and Applications (ESWC 2007)*, pages 729–738.

[Barrows, 1986] Barrows, H. (1986). A Taxonomy of Problem-Based Learning Methods. *Medical Education*, 20(6):481–486.

[Bartels, 2007] Bartels, C. (2007). Automation of exam and student management for the Bologna Process (in German). *Praxis der Informationsverarbeitung und Kommunikation*, 3:134–138.

[Basili, 1999] Basili, V. R. (1999). The Experience Factory: Packaging Software Experience. In *Proceedings of the Fourteenth Annual Software Engineering Workshop*. NASA Goddard Space Flight Center, Greenbelt MD 20771.

[Basili et al., 1994] Basili, V. R., Caldiera, G., and Rombach, H. D. (1994). The Experience Factory. In Marciniak, J. J., editor, *Encyclopedia of SE*, volume 1, pages 511–519. John Wiley & Sons.

[Basili and Weiss, 1984] Basili, V. R. and Weiss, D. M. (1984). A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering*, 10(6):728–738.

[Bauer and Müller, 2004] Bauer, B. and Müller, J. P. (2004). Methodologies and Languages. In Luck, M., Ashri, R., and d'Inverno, M., editors, *Agent-based Software Development*, pages 77–131. Artech House, Boston, USA.

[Beall, 2006] Beall, J. (2006). Metadata Schemes Points of Comparison. Link: http://eprints.rclis.org/archive/00005544/01/comparingschemes.pdf. The open archive for Library and Information Science.

[Beer, 1995] Beer, S. (1995). *Diagnosing the System for Organizations*. John Wiley & Sons.

[Belanger and Jordan, 2000] Belanger, F. and Jordan, D. H. (2000). *Evaluation and Implementation of Distance Learning: Technologies, Tools and Techniques*. Idea Group Publishing.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5):34–44.

[Bügel, 2004] Bügel, U. (2004). Knowledge Representation in Content Management System (In German). *visIT Wissensrepräsentation*, 2:4–5.

[Boehm, 1978] Boehm, B. W., editor (1978). *Characteristics of Software Quality*. Elsevier.

[Boiko, 2005] Boiko, B. (2005). *Content Management Bible*. Wiley, 2 edition.

[Boley et al., 2001] Boley, H., Tabet, S., and Wagner, G. (2001). Design Rationale of RuleML: A Markup Language for Semantic Web Rules. In Cruz, I. F., Decker, S., Euzenat, J., and McGuinness, D. L., editors, *Proceedings of the first Semantic Web Working Symposium (SWWS 2001)*, pages 381–401, Stanford, USA.

[Borst, 2006] Borst, T. (2006). Ontologies for Semantic Annotation of Digital Learning Material (in German). Technical report, Fernuniversität Hagen.

[Braungarten et al., 2006a] Braungarten, R., Farooq, A., Kunz, M., Schmietendorf, A., and Dumke, R. (2006a). Applying Service-Oriented Software Measurement to Derive Quality Indicators of Open Source Components. *UPGRADE - The European Journal for the Informatics Professional*, VII(1).

[Braungarten et al., 2006b] Braungarten, R., Kunz, M., and Dumke, R. (2006b). Historical Evolution of Courses of Study in Computer Science: A German Experience Report. *UPGRADE – The European Journal for the Informatics Professional*, 7(4).

[Braungarten et al., 2005] Braungarten, R., Kunz, M., Farooq, A., and Dumke, R. (2005). Towards Meaningful Metrics Data Bases. In Abran, A. and Dumke, R., editors, *Proceedings of the 15th Workshop on Software Measurement (IWSM05)*, pages 1–34, Montreal, Canada. Shaker Publ.

[Brenner et al., 1998] Brenner, W., Zarnikow, R., and Wittig, H. (1998). *Intelligent Software Agents (in German)*. Springer Verlag.

[Brooks, 1986] Brooks, R. A. (1986). A Robust Layered Control System For A Mobile Robot. *IEEE Journal Of Robotics And Automation*, RA-2(1):14–23.

[Brooks, 1991] Brooks, R. A. (1991). Intelligence without Representation. In Myopoulos, J. and Reiter, R., editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 569–595, Sydney, Australia. Morgan Kaufmann Publishers Inc.: San Mateo, CA, USA.

[Brown and Duguid, 2000] Brown, J. S. and Duguid, P. (2000). *The Social Life of Information*. Harvard Business School Press, Boston, USA.

[Brusilovsky, 1996] Brusilovsky, P. (1996). Methods and Techniques of Adaptive Hypermedia. In *Adaptive Hypertext and Hypermedia*, pages 1–43.

[Brusilovsky, 1999] Brusilovsky, P. (1999). Adaptive and Intelligent Technologies for Web-Based Education. *Künstliche Intelligenz*, 4:19–25.

[Brusilovsky et al., 1998] Brusilovsky, P., Kobsa, A., and Vassileva, J., editors (1998). *Adaptive Hypertext and Hypermedia*. Kluwer Academic Publishers.

[Brusilovsky and Nejdl, 2004] Brusilovsky, P. and Nejdl, W. (2004). Adaptive Hypermedia and Adaptive Web. In Singh, M. P., editor, *The Practical Handbook of Internet Computing*. CHAPMAN & HALL/CRC.

[Buitelaar et al., 2004] Buitelaar, P., Eigner, T., and Declerck, T. (2004). OntoSelect: A Dynamic Ontology Library with Support for Ontology Selection. In *Proceedings of the Demo Session at the International Semantic Web Conference*.

[Buraga, 2003] Buraga, S.-C. (2003). Developing Agent-Oriented E-Learning Systems. In Ioan Dumitrache, C. B., editor, *Proceedings of The 14th International Conference on Control Systems And Computer Science*, volume 2, Bucharest, Romania. Politehnica Press.

[Buzzell and Gale, 1987] Buzzell, R. D. and Gale, B. T. (1987). *The PIMS Principles: Linking Strategy to Performance*. The Free Press.

[Cabral et al., 2004] Cabral, L., Domingue, J., Motta, E., Payne, T. R., and Hakimpour, F. (2004). Approaches to Semantic Web Services: An Overview and Comparison. In *Proceedings of the European Semantic Web Conference*.

[Caire et al., 2001] Caire, G., Garijo, F., Leal, F., Chainho, P., and Massonet, P. (2001). PIR4.3: Case Study Development. EURESCOM Project P907 Technical Information.

[Cannataro and Pugliese, 2004] Cannataro, M. and Pugliese, A. (2004). A Survey of Architectures for Adaptive Hypermedia. In *Web Dynamics - Adapting to Change in Content, Size, Topology and Use*, pages 357–386.

[Casey and Pahl, 2003] Casey, M. and Pahl, C. (2003). Web Components and the Semantic Web. *Electronic Notes in Theoretical Computer Science*, 82(5):156–163.

[Castelfranchi, 1996] Castelfranchi, C. (1996). To Be or Not To Be an Agent. In Müller, J. P., Wooldridge, M., and Jennings, N. R., editors, *Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, 1996, Proceedings*, pages 37–39, Budapest, Hungary.

[Cerri, 2002] Cerri, S. A. (2002). Human and Artificial Agent's Conversations on the GRID. In *Proceedings of 1st LEGE-WG International Workshop on e-Learning and GRID Technologies: Educational Models for GRID Based Services*.

[Chainho et al., 2000] Chainho, P., Massonet, P., Garijo, F., Sanz, J. G., Pavon, J., Caire, G., Leal, F., and Stark, J. (2000). Analysis of the UPA Case Study (PIR4.2 Case Study 1 B1). Eurescom project p907 technical information, EURESCOM.

[Chandrasekaran et al., 1999] Chandrasekaran, B., Josephson, J. R., and Benjamins, V. R. (1999). What Are Ontologies, and Why Do We Need Them? *IEEE Intelligent Systems*, 14(1):20–26.

[Cheng and Chang, 2007] Cheng, T. and Chang, K. C.-C. (2007). Entity Search Engine: Towards Agile Best-Effort Information Integration over the Web. In *Proceedings of the Third Biennial Conference on Innovative Data Systems Research (CIDR 2007)*, pages 108–113, Asilomar, USA.

[Cost et al., 2002] Cost, R. S., Finin, T. W., Joshi, A., Peng, Y., Nicholas, C. K., Soboroff, I., Chen, H., Kagal, L., Perich, F., Zou, Y., and Tolia, S. (2002). ITtalks: A Case Study in the Semantic Web and DAML+OIL. *IEEE Intelligent Systems*, 17(1):40–47.

[Cristea, 2004] Cristea, A. I. (2004). What can the Semantic Web do for Adaptive Educational Hypermedia? *Educational Technology & Society*, 7(4):40–58.

[Crnkovic and Larsson, 2002] Crnkovic, I. and Larsson, M., editors (2002). *Building Reliable Component-Based Software Systems*. Artech House Publishers.

[Daconta et al., 2003] Daconta, M. C., Smith, K. T., and Obrst, L. J. (2003). *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. John Wiley & Sons, Inc.

[Daniel et al., 2007] Daniel, F., Yu, J., Benatallah, B., Casati, F., Matera, M., and Saint-Paul, R. (2007). Understanding UI Integration: A Survey of Problems, Technologies, and Opportunities. *IEEE Internet Computing*, 11(3):59–66.

[Darbyshire and Lowry, 2000] Darbyshire, P. and Lowry, G. (2000). An Overview of Agent Technology and its Application to Subject Management. In *Proceedings of the 2000 Information Resources Management Association International Conference on Challenges of Information Technology Management in the 21st Century*, pages 509–512, Hershey, PA, USA. Idea Group Publishing.

[DARPA Agent Markup Language (DAML) Program, 2008] DARPA Agent Markup Language (DAML) Program (2008). DAML Ontology Library. Link: http://www.daml.org/ontologies/.

[Davies et al., 2006] Davies, J., Studer, R., and Warren, P., editors (2006). *Semantic Web Technologies: Trends and Research in Ontology-based Systems*. John Wiley & Sons.

[de Bra, 2002] de Bra, P. (2002). Adaptive Educational Hypermedia on the Web. *Communications of the ACM*, 45(5):60–61.

[de Rosis et al., 1993] de Rosis, F., De Carolis, N., and Pizzutilo, S. (1993). User Tailored Hypermedia Explanations. In *Adjunct Proceedings of the Conference on Human Factors in Computing Systems (INTERACT'93)*, pages 169–170, Amsterdam, Niederlande.

[de Vrieze and van Bommel, 2004] de Vrieze, P. and van Bommel, P. (2004). A Generic Engine for User Model Based Adaptation. In *Proceedings of the 8th ERCIM Workshop - User Interfaces for All*.

[DeTina and Poehlman, 2002] DeTina, P. and Poehlman, W. F. S. (2002). DISTRIBUTED SYSTEMS II – A Survey of Software Agent Attributes. Technical Report CAS 2002-02-SP, McMaster University, Hamilton, Ontario, Canada, Hamilton, Ontario, Canada.

[Devedžić, 1999] Devedžić, V. (1999). Ontologies: Borrowing from Software Patterns. *ACM Intelligence*, 10(3):14–24.

[Devedžić, 2006] Devedžić, V. (2006). *Semantic Web and Education*. Springer's Integrated Series in Information Systems. Springer.

[Dias and Bidarra, 2007] Dias, A. and Bidarra, J. (2007). Designing e-Content: A Challenge for Open Educational Resources. In *Proceedings of the EADTU's 20th Anniversary Conference*, Lisbon, Portugal.

[Dimitrova et al., 2003] Dimitrova, M., Stadler, C., Hatzipanagos, S., and Murphy, A. (2003). Addressing Learner Diversity by Promoting Flexibility in e-Learning Environments. In *Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03)*.

[Dolog et al., 2003a] Dolog, P., Gavriloaie, R., Nejdl, W., and Brase, J. (2003a). Integrating Adaptive Hypermedia Techniques and Open RDF-Based Environments. In *Proceedings of the Twelfth International World Wide Web Conference*, Budapest, Hungary.

[Dolog et al., 2003b] Dolog, P., Henze, N., Nejdl, W., and Sintek, M. (2003b). Towards the Adaptive Semantic Web. In François Bry, Nicola Henze, J. M., editor, *Proceedings of the International Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR 2003)*, volume 2901 of *Lecture Notes in Computer Science*, pages 51–68, Mumbai, India. Springer.

[Dolog and Nejdl, 2003] Dolog, P. and Nejdl, W. (2003). Using UML and XMI for Generating Adaptive Navigation Sequences in Web-Based Systems. In Stevens, P., Whittle, J., and Booch, G., editors, *Proceedings of the 6th International Conference on The Unified Modeling Language, Modeling Languages and Applications (UML 2003)*, volume 2863 of *Lecture Notes in Computer Science*, pages 205–219, USA. Springer.

[Domjan, 1998] Domjan, M. (1998). *Principles of Learning and Behaviour*. Pacific Grove: Brooks/Cole.

[Doyle et al., 2003] Doyle, B., Rothfuss, G., and Gilbane (2003). The Classification & Evaluation of Content Management Systems. *The Gilbane Report*, 11(2). Link: http://www.cmsreview.com/Resources/GilbaneReport.html.

[Dublin Core Metadata Initiative, 2008] Dublin Core Metadata Initiative (2008). DCMI Type Vocabulary. Link: http://dublincore.org/documents/dcmi-type-vocabulary/.

[Dumke, 2003] Dumke, R. (2003). *Software Engineering*. Vieweg Verlag.

[Dumke, 2005] Dumke, R. (2005). Software Measurement Frameworks. In *Proceedings of the 3rd World Congress for Software Quality*, volume III, pages 75–84, Munich, Germany. Invited talk.

[Dumke et al., 2006a] Dumke, R., Abran, A., and Bulgione, L. (2006a). Suggestion for Improving Measurement Plans: First Results from a BMP Application. In *Proceedings of the 3rd Software Measurement European Forum (Smef 2006)*, pages 209–224, Rome, Italy.

[Dumke et al., 2006b] Dumke, R., Blazey, M., Hegewald, H., Reitz, D., and Richter, K. (2006b). Causalities in Software Process Measurement and Improvement. In Abran, A., Dumke, R., and Ruiz, M., editors, *Proceedings of the International Conference on Software Process and Product Measurement (MENSURA 2006)*, pages 42–52, Cádiz, Spain. Servicio de Publicaciones de la Universidad de Cádiz, Spanien.

[Dumke et al., 2005a] Dumke, R., Braungarten, R., Kunz, M., and Hegewald, H. (2005a). An ISO 15939-Based Infrastructure Supporting the IT Software Measurement. In *Praxis der Software-Messung –Tagungsband des DASMA Software Metrik Kongresses (MetriKon 2005)*, pages 87–106. Shaker Publ.

[Dumke et al., 2006c] Dumke, R., Braungarten, R., Kunz, M., Schmietendorf, A., and Wille, C. (2006c). Strategies and Appropriateness of Software Measurement Frameworks. In Abran, A., Dumke, R., and Ruiz, M., editors, *Proceedings of the International Conference on Software Process and Product Measurement (MENSURA 2006)*, pages 150–170, Cádiz, Spain. Servicio de Publicaciones de la Universidad de Cádiz, Spanien.

[Dumke et al., 2007a] Dumke, R., Braungarten, R., Mencke, S., Richter, K., and Yazbek, H. (2007a). Experience-Based Software Measurement and Evaluation Considering Paradigm Evolution. In Büren, G., Bundschuh, M., and Dumke, R., editors, *Proceedings of the DASMA Metric Congress (Metrikon 2007)*, pages 47–62, Kaiserlautern, Germany. Shaker Publ.

[Dumke et al., 2007b] Dumke, R., Kernchen, S., and Blazey, U. (2007b). Proactive Educational Courses for Mobile Environments. In *Proceedings of the Second International Conference on Interactive Mobile and Computer Aided Learning (IMCL2007)*, pages 539–550, Amman, Jordan.

[Dumke et al., 2005b] Dumke, R., Kunz, M., Hegewald, H., and Yazbek, H. (2005b). An Agent-based Measurement Infrastructure. In Abran, A. and Dumke, R., editors, *Proceedings of the 15th Workshop on Software Measurement (IWSM05)*, pages 415–434, Montreal, Canada. Shaker Publ.

[Dumke et al., 2003] Dumke, R., Lother, M., Wille, C., and Zbrog, F. (2003). *Web Engineering*. Pearson Education Publ.

[Dumke et al., 2009] Dumke, R., Mencke, S., and Wille, C. (2009). *Quality Assurance of Agent-Based and Self-Managing Software Systems*. Auerbach Publ. accepted.

[Dumke et al., 2006d] Dumke, R., Schmietendorf, A., and Zuse, H. (2006d). Formal Descriptions of Software Measurement and Evaluation - A Short Overview and Evaluation - Part I. *Metrics News, Journal of GI-Interest Group on Software Metrics*, 11(1):35–72.

[Dumke et al., 2007c] Dumke, R., Schmietendorf, A., and Zuse, H. (2007c). Formal Descriptions of Software Measurement and Evaluation - A Short Overview and Evaluation - Part II. *Metrics News, Journal of GI-Interest Group on Software Metrics*, 12(2):11–56.

[Dumke et al., 2000] Dumke, R. R., Koeppe, R., and Wille, C. (2000). Software Agent Measurement and Self-Measuring Agent-Based Systems. Preprint, Otto-von-Guericke University of Magdeburg.

[e Cunha and Rana, 2006] e Cunha, J. C. and Rana, O., editors (2006). *Grid Computing: Software Environments and Tools*. Springer Programming, SWE & Operating Systems. Springer-Verlag.

[Ebert and Dumke, 2007] Ebert, C. and Dumke, R. (2007). *Software Measurement*. Springer Verlag.

[Eklund and Zeiliger, 1996] Eklund, J. and Zeiliger, R. (1996). Navigating the Web: Possibilities and Practicalities for Adaptive Navigational Support. In *Proceedings of Ausweb'96: The Second Australian World Wide Web Conference*. Southern Cross University Press.

[Elio et al., 2000] Elio, R., Haddadi, A., and Singh, A. (2000). Task Models, Intentions, and Agent Conversation Policies. In *Proceedings of the Sixth Pacific Rim International Conference on AI (PRICAI2000)*, number 1886 in Lecture Notes in Artificial Intelligence, pages 394–403, Melbourne, Australia. Springer Verlag.

[Encyclopædia Britannica - Online, 2008] Encyclopædia Britannica - Online (2008). Semantics. Link: http://www.britannica.com/eb/article-9110293/semantics.

[Endres, 2004] Endres, A. (2004). Forschungs- und Entwicklungsprojekte: Der Informationsbegriff - eine informatikorientierte Annäherung. *Informatik, Forschung und Entwicklung*, 18(2):88–93.

[Farooq, 2005] Farooq, A. (2005). Conception and Prototypical Implementation of a Web Service as an Empirical-based Consulting about Java Technologies. Master's thesis, University of Madgeburg.

[Farooq et al., 2005] Farooq, A., Braungarten, R., and Dumke, R. (2005). An Empirical Analysis of Object-Oriented Metrics for Java Technologies. In *Proceedings of the 9th IEEE International Multitopic Conference (INMIC2005)*, Karachi, Pakistan.

[Farooq et al., 2006a] Farooq, A., Braungarten, R., Kunz, M., and Dumke, R. (2006a). Towards SOA-Based Approaches for IT Quality Assurance. In *Proceedings of Conference on Quality Engineering in Software Technology (CONQUEST 2006)*, Berlin, Germany. Shaker Verlag Gmbh.

[Farooq et al., 2006b] Farooq, A., Kernchen, S., Dumke, R., and Wille, C. (2006b). Web Services based Measurement for IT Quality Assurance. In Abran, A., Dumke, R., and Ruiz, M., editors, *Proceedings of the International Conference on Software Process and Product Measurement (MENSURA 2006)*, pages 241–251, Cádiz, Spain. Servicio de Publicaciones de la Universidad de Cádiz.

[Farooq et al., 2006c] Farooq, A., Kernchen, S., Kunz, M., Dumke, R., and Wille, C. (2006c). Complexity and Quality Evaluation of Basic Java Technologies. In Abran, A., Bundschuh, M., Büren, G., and Dumke, R., editors, *Proceedings of the International Workshop on Software Measurement and DASMA Software Metrik Kongress (IWSM/MetriKon 2006)*, pages 107–118, Potsdam, Germany. Shaker Publ.

[Feng and Heffernan, 2007] Feng, M. and Heffernan, N. T. (2007). Towards Live Informing and Automatic Analying of Student Learning: Reporting in the Assistment System. *Journal of Interactive Learning Research*, 28(2):207–230.

[Fensel and Bussler, 2002] Fensel, D. and Bussler, C. (2002). The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*, 1(2):113–137.

[Fensel et al., 2001] Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D. L., and Patel-Schneider, P. F. (2001). OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2):38–45.

[Ferber, 1999] Ferber, J. (1999). *Multi-Agent Systems*. Addison Wesley, Harlow, UK.

[Fernández-Caballero et al., 2003] Fernández-Caballero, A., López-Jaquero, V., Montero, F., and González, P. (2003). Adaptive Interaction Multi-Agent Systems in E-Learning / E-Teaching on the Web. In Lovelle, J. M. C., Rodríguez, B. M. G., Aguilar, L. J., Gayo, J. E. L., and del Puerto Paule Ruíz, M., editors, *Proceedings of the International Conference on Web Engineering (ICWE 2003)*, volume 2722 of *Lecture Notes in Computer Science*, pages 144–153, Oviedo, Spain. Springer.

[Fikes, 1998] Fikes, R. (1998). Multi-Use Ontologies. Link: http://www-ksl.stanford.edu/people/fikes/cs222/1998/Ontologies/sld001.htm. Stanford University (February 07, 2007).

[Flechsig, 1996] Flechsig, K.-H. (1996). *Little Handbook of Didactical Models (in German)*. Neuland.

[Foster and Kesselman, 1998] Foster, I. and Kesselman, C., editors (1998). *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann.

[Foundation for Intelligent Physical Agents (FIPA), 2002a] Foundation for Intelligent Physical Agents (FIPA) (2002a). FIPA Abstract Architecture Specification.

[Foundation for Intelligent Physical Agents (FIPA), 2002b] Foundation for Intelligent Physical Agents (FIPA) (2002b). FIPA ACL Message Structure Specification. Link: http://www.fipa.org/specs/fipa00061/SC00061G.html.

[Foundation for Intelligent Physical Agents (FIPA), 2006] Foundation for Intelligent Physical Agents (FIPA) (2006). FIPA Agent Management Specification. Available from http://www.fipa.org/.

[Franklin and Graesser, 1996] Franklin, S. and Graesser, A. C. (1996). Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. In *Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, 1996, Proceedings*, pages 21–35, Budapest, Hungary.

[Fraunhofer-Institut für Integrierte Schaltungen IIS, 2008] Fraunhofer-Institut für Integrierte Schaltungen IIS (2008). MPEG Layer-3. Link: http://www.iis.fraunhofer.de/bf/amm/.

[Gaede, 2000] Gaede, B. (2000). The Instructional Material Description Language and the production system for interactive learning software ProfiL (in German). In *Proceedings of the 10. Arbeitstreffen der GI-Fachgruppe Intelligente Lehr-/ Lernsysteme*, Hamburg, Germany.

[Gamma et al., 2001] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (2001). *Design Patterns – Elements of Reusable Object-Oriented Software (in German)*. Addison-Wesley Professional Computing Series. Addison-Wesley.

[Ganchev et al., 2007] Ganchev, I., Stojanov, S., O'Droma, M., and Meere, D. (2007). An InfoStation-Based Multi-Agent System Supporting Intelligent Mobile Services Across a University Campus. *Journal of Computers*, 2(3):21–33.

[Garro et al., 2003] Garro, A., Leone, N., and Ricca, F. (2003). Logic Based Agents for E-learning. In *Proceedings of the Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems (18th International Joint Conference on Artificial Intelligence (IJCIA 2003))*, pages 36–45, Acapulco, Mexico.

[Garro and Palopoli, 2002] Garro, A. and Palopoli, L. (2002). An XML Multi-Agent System for e-Learning and Skill Management. In Kowalczyk, R., Müller, J. P., Tianfield, H., and Unland, R., editors, *Agent Technologies, Infrastructures, Tools, and Applications for E-Services*, volume 2592 of *Lecture Notes in Computer Science*, pages 283–294. Springer. Third International Symposium on Multi-Agent Systems, Large Complex Systems, and E-Businesses (MALCEB'2002), Erfurt, Thuringia (Germany).

[Garro and Palopoli, 2003] Garro, A. and Palopoli, L. (2003). MASeL: A Multi-Agent System for e-Learning and Skill Management. *EXP - in search of innovation (Special Issue on JADE)*, 3(3):126–136.

[Garvin, 1984] Garvin, D. A. (1984). What Does 'Product Quality' Really Mean? *MIT Sloan Management Review*, 26(1):25–43.

[Geroimenko and Chen, 2002] Geroimenko, V. and Chen, C., editors (2002). *Visualizing the Semantic Web*. Springer.

[Giotopoulos et al., 2005] Giotopoulos, K. C., Alexakos, C. E., Beligiannis, G. N., and Likothanassis, S. D. (2005). Integrating Agents and Computational Intelligence Techniques in E-learning Environments. In *International Enformatika Conference (IEC'05)*, pages 231–238, Prague, Czech Republic.

[Goldberg, 1996] Goldberg, M. W. (1996). Student Participation and Progress Tracking for Web-Based Courses Using WebCT. In *Proceedings of the 2nd North American Web Conference*, Fredericton, Canada.

[Gómez-Pérez and Corcho, 2002] Gómez-Pérez, A. and Corcho, O. (2002). Ontology Specification Languages for the Semantic Web. *IEEE Intellgient Systems*, 17(2):54–60.

[Gómez-Pérez et al., 2004] Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2004). *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer.

[Government of Victoria, 2003] Government of Victoria (2003). Web Content Lifecycle and Content Management Roles. Technical report, Government of Victoria. Link: http://www.egov.vic.gov.au/pdfs/WebContentLifecycleReport-v1-Oct2003.pdf.

[Gruber, 1993] Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2):199–220.

[Grunenberg, 2008] Grunenberg, J. (2008). *IASON Semantic Content Management System*. Vdm Verlag Dr. Müller.

[Grütter, 2005] Grütter, R. (2005). Software Agents in Semantic Web. (in German). *Informatik Spektrum*, (6):1–11.

[Hahn, 1971] Hahn, D. (1971). *Decision Cases in Business Practice (in German)*, chapter Decision Process and Case Method (in German). Gabler Verlag.

[Hametner et al., 2006] Hametner, K., Jarz, T., Moriz, W., Pauschenschwein, J., Sandtner, H., Schinnerl, I., Sfiri, A., and Teufel, M. (2006). Quality Criteria for e-Learning – A Guide for Teachers, Lecturers and Content Creators (In German). Link: http://e-teaching-austria.at/download_mat/Qualitaetskriterien.pdf.

[Hartonas, 2008] Hartonas, T. (2008). Learning Objects Metadata Ontology. Link: http://www.teilar.gr/ hartonas/lom.owl. Retrieved (January 2008).

[Hayzelden and Bigham, 1999a] Hayzelden, A. L. and Bigham, J., editors (1999a). *Software Agents for Future Communication Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Hayzelden and Bigham, 1999b] Hayzelden, A. L. G. and Bigham, J. (1999b). Agent Technology in Communications Systems: An Overview. *Knowledge Engineering Review*, 14(4):341–375.

[Heflin and Hendler, 2001] Heflin, J. and Hendler, J. (2001). A Portrait of the Semantic Web in Action. *IEEE Intelligent Systems*, 16(2):54–59.

[Helic, 2005] Helic, D. (2005). An Ontology-Based Approach to Supporting Didactics in E-Learning Systems. In *Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies*, pages 174–176, Kaohsiung, Taiwan.

[Hendler, 2001] Hendler, J. (2001). Agents and the Semantic Web. *IEEE Intelligent Systems*, 16(2):30–37.

[Hendler, 2008] Hendler, J. (2008). Web 3.0: Chicken Farms on the Semantic Web. *IEEE Computer*, 41(1):106–108.

[Hendler and McGuinness, 2000] Hendler, J. and McGuinness, D. L. (2000). The DARPA Agent Markup Language. *IEEE Intelligent Systems*, 16(6):67–73.

[Henze et al., 2004] Henze, N., Dolog, P., and Nejdl, W. (2004). Reasoning and Ontologies for Personalized E-Learning in the Semantic Web. *Educational Technology & Society*, 7(4):82–97.

[Holst, 2003] Holst, S. (2003). An Integrated Infrastructure for Content Services. Whitepaper, Link: http://www.sun.com/third-party/global/emc/docs/gilbane_whitepaper.pdf.

[Höök et al., 1996] Höök, K., Karlgren, J., Wærn, A., Dahlbäck, N., Jansson, C. G., Karlgren, K., and Lemaire, B. (1996). A Glass Box Approach to Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):157–184.

[Hyatt and Rosenberg, 1996] Hyatt, L. E. and Rosenberg, L. H. (1996). A Software Quality Model and Metrics for Identifying Project Risks and Assessing Software Quality. In *Proceedings of the 8th Annual Software Technology Conference*, Salt Lake City, USA.

[IBM, 2002] IBM (2002). Aglets. Link: http://www.trl.ibm.com/aglets/.

[IEEE Learning Technology Standards Committee, 2002a] IEEE Learning Technology Standards Committee (2002a). 1484.12.1-2002 IEEE Standard for Learning Object Metadata.

[IEEE Learning Technology Standards Committee, 2002b] IEEE Learning Technology Standards Committee (2002b). IEEE P1484.1/D9 Draft Standard for Learning Technology: Learning Technology Systems Architecture (LTSA).

[IEEE Learning Technology Standards Committee, 2002c] IEEE Learning Technology Standards Committee (2002c). IEEE P1484.2, PAPI Learner, Draft 8 Specification. available from URL: http://edutool.com/papi/.

[IEEE Learning Technology Standards Committee, 2003] IEEE Learning Technology Standards Committee (2003). IEEE 1484.12:2002, Standard for Learning Object Metadata.

[IMS Global Learning Consortium, 2003] IMS Global Learning Consortium (2003). IMS Learning Design Specification. Link: http://www.imsglobal.org/learningdesign/index.html.

[IMS Global Learning Consortium, 2006] IMS Global Learning Consortium (2006). IMS Learning Resource Meta-data Specification Version 1.3. Link: http://www.imsglobal.org/metadata/index.html.

[IMS Global Learning Consortium, Inc., 2001] IMS Global Learning Consortium, Inc. (2001). IMS Learner Information Package Specification.

[IMS Global Learning Consortium, Inc., 2003a] IMS Global Learning Consortium, Inc. (2003a). IMS Abstract Framework: White Paper.

[IMS Global Learning Consortium, Inc., 2003b] IMS Global Learning Consortium, Inc. (2003b). IMS Content Packaging Best Practice Guide.

[IMS Global Learning Consortium, Inc., 2003c] IMS Global Learning Consortium, Inc. (2003c). IMS Learning Design Specification. Link: http://www.imsglobal.org/learningdesign.

[IMS Global Learning Consortium, Inc., 2005] IMS Global Learning Consortium, Inc. (2005). IMS Learner Information Package Specification v.1.0.1. Link: http://www.imsglobal.org/profiles/index.html.

[IMS Global Learning Consortium, Inc., 2007] IMS Global Learning Consortium, Inc. (2007). IMS Learner Content Packaging Specification.

[Institute of Electrical and Electronics Engineers, Inc., 1998] Institute of Electrical and Electronics Engineers, Inc. (1998). IEEE Std 830-1998.

[International Organisation for Standardisation, 2002] International Organisation for Standardisation (2002). ISO/IEC JTC1/SC29/WG11: Overview of the MPEG-4 Standard. Link: http://www.chiariglione.org/mpeg/standards/mpeg-4/mpeg-4.htm.

[ISO/IEC, 1995] ISO/IEC (1995). ISO/IEC 12207: Information technology – Software life cycle processes. Link: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=21208.

[ISO/IEC, 2000] ISO/IEC (2000). ISO/IEC 9001: Quality management systems – Requirements. Link: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=21823.

[ISO/IEC, 2001] ISO/IEC (2001). ISO/IEC 9126-1:2001: Software Engineering – Product Quality – Part 1: Quality Model. Link: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=22749.

[ISO/IEC, 2002] ISO/IEC (2002). ISO/IEC 15939: Software Engineering – Software Measurement Process. Link: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=29572.

[ISO/IEC, 2004a] ISO/IEC (2004a). ISO/IEC 15504: Information technology – Process assessment. Link: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=38932.

[ISO/IEC, 2004b] ISO/IEC (2004b). ISO/IEC 9000: Quality Management Standards. Link: http://www.iso.org/iso/iso_catalogue/management_standards/iso_9000_iso_14000.htm.

[ISO/IEC, 2008] ISO/IEC (2008). ISO/IEC 15288: Systems and software engineering – System life cycle processes. Link: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=43564.

[Jennings and Wooldridge, 1998a] Jennings, N. R. and Wooldridge, M. J. (1998a). *Agent Technology - Foundation, Application, and Markets*. Springer Publ.

[Jennings and Wooldridge, 1998b] Jennings, N. R. and Wooldridge, M. J. (1998b). Applications of Intelligent Agents. In *Agent Technology: Foundations, Applications, and Markets*, pages 3–28. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Jeon et al., 2000] Jeon, H., Petrie, C. J., and Cutkosky, M. R. (2000). JATLite: A Java Agent Infrastructure with Message Routing. *IEEE Internet Computing*, 4(2):87–96.

[Jovanović et al., 2007] Jovanović, J., Gašević, D., Brooks, C., Devedžić, V., Hatala, M., Eap, T., and Richards, G. (2007). Using Semantic Web Technologies to Analyze Learning Content. *IEEE Internet Computing*, 11(5):45–53.

[Juneidi and Vouros, 2005] Juneidi, S. J. and Vouros, G. A. (2005). Engineering an E-learning Application using the ARL Theory for Agent Oriented Software Engineering. In *2005 AAAI Fall Symposium*, Arlington, Virginia, USA. MIT press.

[Kabassi and Virvou, 2003] Kabassi, K. and Virvou, M. (2003). Learner Modelling in a Multi-Agent System through Web Services. In Devedzic, V., Spector, J. M., Sampson, D. G., and Kinshuk, editors, *Proceedings of 2003 IEEE International Conference on Advanced Learning Technologies: Technology Enhanced Learning - ICALT 2003*, pages 115–119, Athens, Greece.

[Kargl et al., 1999] Kargl, F., Illmann, T., and Weber, M. (1999). CIA - a Collaboration and Coordination Infrastructure for Personal Agents. In *Proceedings of the IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems II (DAIS 1999)*, pages 213–218, Helsinki, Finland. Kluwer, B.V.

[Kay, 2000] Kay, J. (2000). Learner Know Thyself: Student Models to Give Learner Control and Responsibility. In *Proceedings of International Conference on Computers in Education*.

[Kearney et al., 2000] Kearney, P., Stark, J., Leal, F., and Rodrigues, J. P. (2000). Analysis of the ACSOSS Case Study (PIR 4.2 B.0). Eurescom project p907 technical information, EURESCOM.

[Kelly, 2006] Kelly, B. (2006). Choosing a Metadata Standard For Resource Discovery. Link: http://www.ukoln.ac.uk/qa-focus/documents/briefings/briefing-63/html/.

[Kernchen, 2004] Kernchen, S. (2004). Evaluation of ALIVE - Implemented with the Grasshopper Technology (Case study in German).

[Kernchen, 2005] Kernchen, S. (2005). Adaptive Web-Content in the Context of Lecture Notes (in German). Master's thesis, Otto-von-Guericke University of Magdeburg.

[Kernchen, 2007] Kernchen, S. (2007). *Software Measurement*, chapter Measurement of Agent-Based Systems, pages 317–326. Springer Publ.

[Kernchen and Dumke, 2007a] Kernchen, S. and Dumke, R. (2007a). Autonomous Management of User Models for Lifelong Learning. *IEEE Multidisciplinary Engineering Education Magazine*, 2(2):44–50. ISSN 1558-7908.

[Kernchen and Dumke, 2007b] Kernchen, S. and Dumke, R. (2007b). Autonomous Management of User Models for Lifelong Learning. In *Proceedings of the Second International Conference on Interactive Mobile and Computer Aided Learning (IMCL2007)*, pages 381–392, Amman, Jordan.

[Kernchen and Dumke, 2007c] Kernchen, S. and Dumke, R. (2007c). Developing Adaptive and Self-Managed Graphical User Interfaces. In *Proceedings of the Second International Conference on Interactive Mobile and Computer Aided Learning (IMCL2007)*, pages 939–948, Amman, Jordan.

[Kernchen and Dumke, 2007d] Kernchen, S. and Dumke, R. (2007d). Developing Adaptive and Self-Managed Graphical User Interfaces. *IEEE Multidisciplinary Engineering Education Magazine*, 2(2):38–43. ISSN 1558-7908.

[Kernchen and Dumke, 2007e] Kernchen, S. and Dumke, R. (2007e). Proactive Educational Courses in a Distributed Environment. In *Proceedings of the International Technology, Education and Development Conference (INTED2007)*, Valencia, Spain.

[Kernchen et al., 2006] Kernchen, S., Farooq, A., Dumke, R., and Wille, C. (2006). Evaluation of Java-Based Agent Technologies. In Abran, A., Bundschuh, M., Büren, G., and Dumke, R., editors, *Proceedings of the International Workshop on Software Measurement and DASMA Software Metrik Kongress (IWSM/MetriKon 2006)*, Potsdam, Germany. Shaker Publ.

[Kernchen et al., 2007a] Kernchen, S., Kunz, M., and Dumke, R. (2007a). Proactive Class Schedule. In *Proceedings of the International Conference on Interactive Computer Aided Blended Learning (ICBL2007)*, Florianópolis, Brazil.

[Kernchen et al., 2007b] Kernchen, S., Kunz, M., and Dumke, R. (2007b). Proactive Class Schedule. *IEEE Multidisciplinary Engineering Education Magazine*, 2(3):24–28. ISSN 1558-7908.

[Kernchen et al., 2007c] Kernchen, S., Rud, D., Zbrog, F., and Dumke, R. (2007c). Processing Remote Measurement Databases by the Means of Mobile Agents. In Filipe, J., Cordeiro, J., Encarnação, B., and Pedrosa, V., editors, *In Proceedings of the 3rd International Conference on Web Information Systems and Technologies - Volume IT (WEBIST2007)*, pages 184–188, Barcelona, Spain. INSTICC Press.

[Kernchen and Vornholt, 2004] Kernchen, S. and Vornholt, S. (2004). The Agent System ALIVE (in German). Internship Report at the University of Magdeburg.

[Kernchen et al., 2007d] Kernchen, S., Zbrog, F., and Dumke, R. (2007d). ABEL-GUI: An Agent-Based Graphical User Interface for E-Learning. In Filipe, J., Cordeiro, J., Encarnação, B., and Pedrosa, V., editors, *Proceedings of the 3rd International Conference on Web Information Systems and Technologies - Volume SeBeG/eL (WEBIST2007)*, pages 491–494, Barcelona, Spain. INSTICC Press.

[Khan, 2004] Khan, B. H. (2004). People, Process and Product Continuum in E-learning: The E-learning P3 Model. *Educational Technology*, 44(5):33–40.

[Kirkpatrick and McLaughlan, 2000] Kirkpatrick, D. and McLaughlan, R. (2000). Flexible Lifelong Learning in Professional Education. *Educational Technology and Society*, 3(1).

[Klobas and Renzi, 2000] Klobas, J. and Renzi, S. (2000). Selecting Software and Services for Web-Based Teaching and Learning. In Aggarwal, A. K., editor, *Web-Based Learning and Teaching Technologies: Opportunities and Challenges*, pages 43–59. Idea Group Publishing, Hershey, PA, USA.

[Koper, 2001] Koper, R. (2001). Modeling Units of Study from a Pedagogical Perspective the Pedagogical Meta-Model behind EML. Link: http://dspace.ou.nl/bitstream/1820/36/1/Pedagogical+metamodel+behind+EMLv2.pdf.

[Koper and Olivier, 2004] Koper, R. and Olivier, B. (2004). Representing the Learning Design of Units of Learning. *Educational Technology & Society*, 7(3):97–111.

[Koyama et al., 2001] Koyama, A., Barolli, L., Tsuda, A., and Cheng, Z. (2001). An Agent-Based Personalized Distance Learning System. In *Proceedings of the 15th International Conference on Information Networking (ICOIN'01)*, pages 895–899, Beppu City, Japan.

[Kron and Sofos, 2003] Kron, F. W. and Sofos, A. (2003). *Media Didactic (in German)*. Ernst Reinhardt Verlag.

[Kruse et al., 2005] Kruse, P. M., Naujoks, A., Rösner, D., and Kunze, M. (2005). Clever Search: A WordNet Based Wrapper for Internet Search Engines. In Fisseni, B., Schmitz, H.-C., Schröder, B., and Wagner, P., editors, *Proceedings of GLDV-Tagung 2005*, pages 367–380, Bonn, Germany.

[Kunz et al., 2005] Kunz, M., Braungarten, R., and Dumke, R. (2005). Measurement eLearning - A Classification Approach for eLearning Systems. In *Proceedings of the 15th International Workshop on Software Measurement (IWSM2005)*, pages 79–94, Montreal, Canada.

[Kunz et al., 2006a] Kunz, M., Kernchen, S., Dumke, R., and Schmietendorf, A. (2006a). Ontology-Based Web Service for Object-Oriented Metrics. In Abran, A., Bundschuh, M., Büren, G., and Dumke, R., editors, *Proceedings of the International Workshop on Software Measurement and DASMA Software Metrik Kongress (IWSM/MetriKon 2006)*, Potsdam, Germany. Shaker Publ.

[Kunz et al., 2008a] Kunz, M., Mencke, S., Rud, D., Braungarten, R., and Dumke, R. R. (2008a). From QoS towards Quality-Driven Orchestration. In *Proceedings of the 3rd Workshop Bewertungsaspekte serviceorientierter Architekturen (BSOA 2008)*, Leinfelden, Germany.

[Kunz et al., 2008b] Kunz, M., Mencke, S., Rud, D., and Dumke, R. (2008b). Empirical-Based Design – Quality-Driven Assembly of Components. In *Proceedings of the IEEE International Conference on Information Reuse and Integration (IRI 2008)*, Las Vegas, Nevada, USA.

[Kunz et al., 2008c] Kunz, M., Mencke, S., Zenker, N., Braungarten, R., and Dumke, R. R. (2008c). Empirical-Based, Quality-Driven Orchestration of Services. In *Proceedings of the 18th International Workshop in Software Measurement in conjunction with the Mensura and the DASMA Software Metric Kongress (ISWM/Mensura/Metrikon 2008)*, volume 5338 of *Lecture Notes in Computer Science*, Munich, Germany. Springer Publ.

[Kunz et al., 2006b] Kunz, M., Schmietendorf, A., Dumke, R., and Wille, C. (2006b). Towards a Service-Oriented Measurement Infrastructure. In *Proceedings of the 3rd Software Measurement European Forum (SMEF 2006)*, pages 197–207, Rome, Italy.

[Laakko and Hiltunen, 2005] Laakko, T. and Hiltunen, T. (2005). Adapting Web Content to Mobile User Agents. *IEEE Internet Computing*, 9(2):46–53.

[Leutner, 1992] Leutner, D. (1992). *Adaptive Teaching Systems. Instruction-Psychological Foundations and Experimental Analyses*. Beltz - Psychologie Verlags Union, Weinheim.

[Lin et al., 2005] Lin, F., Esmahi, L., and Poon, L. (2005). A Human Collaborative Online Learning Environment Using Intelligent Agents. In Lin, F., editor, *Designing Distributed Learning Environments with Intelligent Software Agents*. Idea Group Publishing.

[Lin et al., 2008] Lin, N., Kuter, U., and Sirin, E. (2008). Web Service Composition with User Preferences. In Bechhofer, S., Hauswirth, M., Hoffmann, J., and Koubarakis, M., editors, *Proceedings of 5th European Semantic Web Conference – The Semantic Web: Research and Applications (ESWC 2008)*, pages 629–643, Tenerife, Canary Islands, Spain.

[López et al., 1999] López, M. F., Gómez-Pérez, A., Sierra, J. P., and Sierra, A. P. (1999). Building a Chemical Ontology Using Methontology and the Ontology Design Environment. *IEEE Intelligent Systems*, 14(1):37–46.

[Maddocks et al., 2000] Maddocks, A. P., Sher, W. D., and Wilson, A. (2000). A Web-Based Personal and Professional Development Tool to Promote Lifelong Learning Within the Construction Industry. *Educational Technology and Society*, 3(1):77–83.

[Maedche and Staab, 2001] Maedche, A. and Staab, S. (2001). Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2):72–79.

[Maes, 1997] Maes, P. (1997). Agents that Reduce Work and Information Overload. In *Software Agents*, pages 145–164. MIT Press, Cambridge, MA, USA.

[Malack et al., 2008] Malack, K., Dumke, R., Mencke, S., and Wille, C. (2008). Semi-Automated Test Environment For Web Content Accessibility Based On WCAG 2.0. In *Proceedings of the 4th International Conference on Web Information Systems and Technologies (WEBIST 2008)*, volume 2, pages 304–307, Funchal, Madeira, Portugal. INSTICC Press.

[Marchionini, 2006] Marchionini, G. (2006). Exploratory Search: From Finding to Understanding. *Communications of the ACM*, 49(4):41–46.

[Marshall, 2007] Marshall, S. (2007). E-Learning Maturity Model - eMM 2.3. Link: http://www.utdc.vuw.ac.nz/research/emm/publications.shtml, University Teaching Development Centre, Victoria University of Wellington.

[Masuoka and Ohtani, 1999] Masuoka, R. and Ohtani, T. (1999). Agent Description Ontology. Proposal to the 13th meeting of FIPA.

[McCall et al., 1977] McCall, J. A., Richards, P. K., and Walters, G. F. (1977). Factors in Software Quality. Technical report, Rome Air Development Center Air Force Systems Command.

[McCallum, 2005] McCallum, A. (2005). Information Extraction: Distilling Structured Data from Unstructured Text. *ACM Queue*, 3(9):48–57.

[McGreal, 2004] McGreal, R. (2004). *Online Learning Using Learning Objects*. Routledge Falmer.

[McIlraith et al., 2001] McIlraith, S. A., Son, T. C., and Zeng, H. (2001). Semantic Web Services. *IEEE Intelligent Systems*, pages 46–53.

[Meder, 2001] Meder, N. (2001). Didactical Ontologies (in German). Link: http://cweb.uni-bielefeld.de/agbi/cgi-bin-noauth/cache/VAL_BLOB/167/167/63/did.pdf.

[Meder, 2006] Meder, N. (2006). *Web-Didactic*. Bertelsmann Verlag.

[Mencke et al., 2008a] Mencke, N., Mencke, S., and Höpfner, A. (2008a). An Online Virtual Vocational Training System. In *Proceedings of the 2008 International Technology, Education and Development Conference (INTED 2008)*, Valencia, Spain.

[Mencke and Dumke, 2007a] Mencke, S. and Dumke, R. (2007a). Agent-Supported e-Learning. Technical Report 07-08, Otto-von-Guericke University of Magdeburg.

[Mencke and Dumke, 2007b] Mencke, S. and Dumke, R. (2007b). Agents in e-Learning. *Emerging Technologies in e-Learning (iJET)*, 2(4):20–31.

[Mencke and Dumke, 2008] Mencke, S. and Dumke, R. (2008). Didactical Ontologies. *Emerging Technologies in e-Learning (iJET)*, 3(1):65–73.

[Mencke and Dumke, 2007c] Mencke, S. and Dumke, R. R. (2007c). A Framework for Agent-Supported E-learning. In Auer, M. E., editor, *Proceedings of the International Conference on Interactive Computer aided Learning (ICL2007)*, Villach, Austria.

[Mencke and Dumke, 2007d] Mencke, S. and Dumke, R. R. (2007d). A Hierarchy of Ontologies for Didactics-Enhanced E-learning. In Auer, M. E., editor, *Proceedings of the International Conference on Interactive Computer aided Learning (ICL2007)*, Villach, Austria.

[Mencke et al., 2008b] Mencke, S., Kunz, M., and Dumke, R. (2008b). Steps to an Empirical Analysis of the Proactive Class Schedule. *IEEE Multidisciplinary Engineering Education Magazine*, 3(3):100–103.

[Mencke et al., 2008c] Mencke, S., Kunz, M., and Dumke, R. (2008c). Towards Metrics for Ontology Balance. In *Proceedings of the Twentieth International Conference on Software Engineering and Knowledge Engineering (SEKE 2008)*, Redwood City, USA.

[Mencke et al., 2008d] Mencke, S., Kunz, M., and Dumke, R. R. (2008d). Steps to an Empirical Analysis of the Proactive Class Schedule. In *Proceedings of the Third International Conference on Interaction Mobile and Computer Aided Learning (IMCL 2008)*, Amman, Jordan.

[Mencke et al., 2008e] Mencke, S., Kunz, M., and Pukall, M. (2008e). Runtime Adaptations within the QuaD^2-Framework. In *Proceedings of the 5th ECOOP'2008 Workshop on Reflection, AOP and Meta-Data for Software Evolution (RAM-SE 2008)*, Paphos, Cyprus.

[Mencke et al., 2008f] Mencke, S., Kunz, M., Zenker, N., and Dumke, R. R. (2008f). Ontology-Based Generic Learning Path Recommendations. In *Proceedings of the 2008 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE 2008)*, Las Vegas, Nevada, USA.

[Mencke et al., 2008g] Mencke, S., Mencke, N., and Dumke, R. (2008g). Granularity of Learning Objects in Virtual Environments. In *Proceedings of the 2008 International Technology, Education and Development Conference (INTED 2008)*, Valencia, Spain.

[Mencke et al., 2008h] Mencke, S., Rud, D., Zbrog, F., and Dumke, R. (2008h). Proactive Autonomous Resource Enrichment for e-Learning. In *Proceedings of the 4th International Conference on Web Information Systems and Technologies (WEBIST 2008)*, volume 1, pages 464–467, Funchal, Madeira, Portugal. INSTICC Press.

[Mencke et al., 2008i] Mencke, S., Vornholt, S., and Dumke, R. (2008i). The Role of Ontologies in Virtual Engineering. In *Proceedings of the 1st International Workshop on Data Management in Virtual Engineering (DMVE 2008) as part of the 19th International Conference on Database and Expert Systems Applications (DEXA 2008)*, Turin, Italy. IEEE Computer Society.

[Mencke et al., 2008j] Mencke, S., Wille, C., and Dumke, R. (2008j). Measuring Distances for Ontology-Based Systems. In *Proceedings of the 18th International Workshop in Software Measurement in conjunction with the Mensura and the DASMA Software Metric Kongress (ISWM/Mensura/Metrikon 2008)*, volume 5338 of *Lecture Notes in Computer Science*, Munich, Germany. Springer Publ.

[Mencke et al., 2008k] Mencke, S., Zbrog, F., and Dumke, R. (2008k). Useful e-Learning Process Descriptions. In *Proceedings of the 4th International Conference on Web Information Systems and Technologies (WEBIST 2008)*, volume 1, pages 460–463, Funchal, Madeira, Portugal. INSTICC Press.

[Merceron and Yacef, 2005] Merceron, A. and Yacef, K. (2005). TADA-ED for Educational Data Mining. *Interactive Multimedia Journal of Computer-Enhanced Learning*, 7(1):267–287.

[Mertens, 2002] Mertens, H. (2002). *Conception and Development of a Problem-oriented Interaction Component for the Depiction of Economic Case Studies in Hypermedia Teaching/Learning Systems (in German)*. PhD thesis, University of Würzburg.

[Milgrom et al., 2001] Milgrom, E., Chainho, P., Deville, Y., Evans, R., Kearney, P., and Massonet, P. (2001). MESSAGE - Methodology for Engineering Systems of Software Agents (Final Guidelines for the Identification of Relevant Problem Areas where Agent Technology is Appropriate). Technical report, EURESCOM.

[Müller et al., 2005] Müller, C., Trier, M., and Herzog, M. A. (2005). Process-oriented Production of Learning Units for sustainable E-Learning Offerings. In Breitner, M. H. and Hoppe, G., editors, *E-Learning - Einsatzkonzepte und Geschäftsmodelle*, pages 331–348. Physica-Verlag, Heidelberg, Germany.

[Müller, 1996] Müller, J. P. (1996). *The Design of Intelligent Agents: A Layered Approach*, volume 1177 of *Lecture Notes in Computer Science*. Springer.

[MOBIlearn Project Consortium, 2005] MOBIlearn Project Consortium (2005). MOBIlearn - Final Project Report. Technical report, MOBIlearn Project Consortium.

[Mohammed and Mohan, 2005] Mohammed, P. and Mohan, P. (2005). Agent Based Learning Objects on the Educational Semantic Web. In *Proceedings of the 2nd Annual e-Learning Conference on Intelligent Interactive Learning Object Repositories (I2LOR-2005)*, Vancouver, Canada.

[Morik, 1993] Morik, K. (1993). Balanced Cooperative Modeling. *Machine Learning*, 11:217–235.

[Motta et al., 2003] Motta, E., Domingue, J., and Cabral, L. (2003). IRS-II: A Framework and Infrastructure for Semantic Web Services. In *Proceedings of the 2nd International Semantic Web Conference*.

[Narayanan and McIlrath, 2001] Narayanan, S. and McIlrath, S. (2001). Semantic Web Services. *IEEE Intelligent Systems*, 16(2):46–53.

[Nelson, 2000] Nelson, M. (2000). *Buckets: Smart Objects in Digital Libraries*. PhD thesis, Old Dominion University.

[Nentwich et al., 2002] Nentwich, C., Capra, L., Emmerich, W., and Finkelstein, A. (2002). xlinkit: A Consistency Checking and Smart Link Generation Service. *ACM Transactions on Internet Technology*, 2(2):151–185.

[Netquest, 1998] Netquest (1998). Tutorial Markup Language (TML). Link: http://www.ilrt.bris.ac.uk/netquest/about/lang/.

[Newell and Simon, 1976] Newell, A. and Simon, H. A. (1976). Computer Science as Empirical Enquiry. *Communication of the ACM*, 19:113–126.

[Norman and Porter, 2007] Norman, S. and Porter, D. (2007). Designing Learning Objects for Online Learning. *The Knowledge Series*.

[Noy and Mcguinness, 2001] Noy, N. F. and Mcguinness, D. L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Technical report, Stanford Medical Informatics, Palo Alto.

[Nwana, 1996] Nwana, H. S. (1996). Software Agents: An Overview. *Knowledge Engineering Review*, 11(3):205–244.

[Nwana and Ndumu, 1998] Nwana, H. S. and Ndumu, D. T. (1998). A Brief Introduction to Software Agent Technology. In *Agent Technology: Foundations, Applications, and Markets*, pages 29–47. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[OASIS, 2005] OASIS (2005). UDDI: Universal Description, Discovery and Integration Protocol . Link: http://www.uddi.org/.

[Object Management Group, 2004] Object Management Group (2004). Unified Modeling Language. Link: http://www.uml.org/.

[Object Management Group, 2005] Object Management Group (2005). Business Process Modelling Notation (BPMN).

[O'Donnell, 1997] O'Donnell, M. (1997). Variable Length On-Line Document Generation. In *Proceedings of the 6th European Workshop on Natural Language Generation*, Duisburg, Germany.

[O'Malley and DeLoach, 2002] O'Malley, S. A. and DeLoach, S. (2002). Determining When to Use an Agent-Oriented Software Engineering Paradigm. In *AOSE '01: Revised Papers and Invited Contributions from the Second International Workshop on Agent-Oriented Software Engineering II*, pages 188–205, London, UK. Springer-Verlag.

[Open Knowledge Initiative, 2003] Open Knowledge Initiative (2003). O.K.I. Architectural Concepts. Technical report, Massachusetts Institute of Technology.

[Otto-von-Guericke-University Magdeburg, 2008] Otto-von-Guericke-University Magdeburg (2008). Full specification for Bachelor and Master Programs in Computer Science studies at the Otto-von-Guericke-University Magdeburg. Link: http://www.sim-md.de/management/.

[Pankratius et al., 2004] Pankratius, V., Sandel, O., and Stucky, W. (2004). Retrieving Content with Agents in Web Service E-Learning Systems. In *The Symposium on Professional Practice in AI, IFIP WG12.5 - First IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI)*, Toulouse, France.

[Pankratius and Vossen, 2003] Pankratius, V. and Vossen, G. (2003). Towards E-Learning Grids: Using Grid Computing in Electronic Learning. In *Proceedings of the IEEE Workshop on Knowledge Grid and Grid Intelligence (in conjunction with 2003 IEEE/WIC International Conference on Web Intelligence / Intelligent Agent Technology)*, pages 4–15, Halifax, Canada.

[Patel et al., 2000] Patel, A., Kinshuk, and Russell, D. (2000). Intelligent Tutoring Tools for Cognitive Skill Acquisition in Life Long Learning. *Educational Technology and Society*, 3(1).

[Pawlowski, 2000] Pawlowski, J. M. (2000). The Essen Learning Model a Multi-Level Development Model. *Staff and Education Development International*, 4.

[Pawlowski, 2001] Pawlowski, J. M. (2001). *The Essen Learning Model: A Model for the Development of Computer-Supported Learning Environments*. PhD thesis, University of Essen.

[Pawlowski, 2005] Pawlowski, J. M. (2005). The Didactical Object Model: Managing Didactical Expertise. In *Proceedings of The 5th IEEE International Conference on Advanced Learning Technologies*, pages 788–792, Kaohsiung, Taiwan.

[Pentland et al., 1999] Pentland, B. T., Osborn, C. S., Wyner, G., and Luconi, F. (1999). Useful Descriptions of Organizational Processes: Collecting Data for the Process Handbook. Working Paper Series 208, MIT Center for Coordination Science.

[Pierrakos et al., 2003] Pierrakos, D., Paliouras, G., Papatheodorou, C., Karkaletsis, V., and Dikaiakos, M. D. (2003). Web Community Directories: A New Approach to Web Personalization. In Berendt, B., Hotho, A., Mladenic, D., van Someren, M., Spiliopoulou, M., and Stumme, G., editors, *Web Mining: From Web to Semantic Web*, volume 3209 of *Lecture Notes in Computer Science*, pages 113–129, Cavtat-Dubrovnik, Croatia. Springer.

[Piotrowski et al., 2007] Piotrowski, M., Amelung, M., and Rösner, D. (2007). EduComponents: Educational Components for Plone. In Lytras, M. and Naeve, A., editors, *Open Source for Knowledge and Learning Management: Strategies Beyond Tools*. Idea Group, Hershey, PA, USA.

[Porter et al., 2002] Porter, D., Curry, J., Muirhead, B., and Galan, N. (2002). A Report on Learning Object Repositories – Review and Recommendations for a Pan-Canadian Approach to Repository Implemention in Canada. Technical report, CANARIE Inc.

[Preece et al., 1994] Preece, J., Rogers, Y., Sharp, H., and Benyon, D. (1994). *Human-Computer Interaction*. Addison-Wesley Longman Ltd., Essex, UK.

[Princeton University, 2006] Princeton University (2006). WORDNET - A Lexical Database for the English Language. Link: http://wordnet.princeton.edu/.

[Project Management Institute, 2004] Project Management Institute (2004). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. Project Management Institute, 3 edition.

[Qiu et al., 2006] Qiu, L., Lin, F., Wan, C., and Shi, Z. (2006). Semantic Web Services Composition Using AI Planning of Description Logics. In *Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC 2006)*, pages 340–347, Washington, DC, USA. IEEE Computer Society.

[Rahkila, 2001] Rahkila, M. (2001). A Double Agent Architecture for eLearning Applications. In *Proceedings of the ICEE 2001 Conference*, volume 1, Oslo, Norway.

[Rao and Georgeff, 1991] Rao, A. S. and Georgeff, M. P. (1991). Modelling Rational Agents within a BDI-Architecture. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufman Publishers.

[Richards et al., 2002] Richards, G., McGreal, R., Hatala, M., and Friesen, N. (2002). The Evolution of Learning Object Repository Technologies: Portals for On-line Objects for Learning. *Journal of Distance Education*, 17(3):67–79.

[Roda et al., 2003] Roda, C., Angehrn, A. A., Nabeth, T., and Razmerita, L. (2003). Using Conversational Agents to Support the Adoption of Knowledge Sharing Practices. *Interacting with Computers*, 15(1):57–89.

[Rodríguez-Artacho and Maíllo, 2004] Rodríguez-Artacho, M. and Maíllo, M. F. V. (2004). Modeling Educational Content: The Cognitive Approach of the PALO Language. *Educational Technology & Society*, 7(3):124–137.

[Rodríguez-Artacho et al., 1999] Rodríguez-Artacho, M., Maíllo, M. F. V., Mayorga, J. I., and Calero, M. Y. (1999). Using a High-Level Language to Describe and Create Web-Based Learning Scenarios. In *Proceedings of the 29th ASEE/IEEE Frontiers in Education Conference*, San Juan, Puerto Rico.

[Roure et al., 2001] Roure, D. D., Jennings, N., and Shadbolt, N. (2001). Research Agenda for the Semantic Grid: A Future e-Science Infrastructure. Link: http://www.semanticgrid.org/v1.9/semgrid.pdf.

[Rösner et al., 2007] Rösner, D., Piotrowski, M., and Amelung, M. (2007). A Sustainable Learning Environment based on an Open Source Content Management System. In Bühler, W., editor, *Proceedings of the German e-Science Conference (GES 2007)*, Baden-Baden, Germany. Max-Planck-Gesellschaft.

[Rud, 2006] Rud, D. (2006). *Qualität von Web Services: Messung und Sicherung der Performance*. VDM Verlag Dr. Müller, Saarbrücken.

[Rud et al., 2007a] Rud, D., Kunz, M., Schmietendorf, A., and Dumke, R. (2007a). Performance Analysis in WS-BPEL-Based Infrastructures. In *Proceedings of the 23rd Annual UK Performance Engineering Workshop (UKPEW 2007)*, pages 130–141, Ormskirk, Lancashire, UK. Edge Hill University.

[Rud et al., 2007b] Rud, D., Mencke, S., Schmietendorf, A., and Dumke, R. (2007b). Granularity Metrics for Service-Oriented Architectures (In german). In Büren, G., Bundschuh, M., and Dumke, R., editors, *Proceedings of the DASMA Metric Congress (Metrikon 2007)*, pages 297–308, Kaiserlautern, Germany. Shaker Publ.

[Rud et al., 2006a] Rud, D., Schmietendorf, A., and Dumke, R. (2006a). Performance Modeling of WS-BPEL-Based Web Service Compositions. In *Proceedings of the 2006 IEEE Services Computing Workshops (SCW 2006)*, pages 140–147, Chicago, Illinois, USA. IEEE Computer Society.

[Rud et al., 2006b] Rud, D., Schmietendorf, A., and Dumke, R. (2006b). Product Metrics for Service-Oriented Infrastructures. In Abran, A., Bundschuh, M., Büren, G., and Dumke, R., editors, *Proceedings of the International Workshop on Software Metrics and DASMA Software Metrik Kongress (IWSM/MetriKon 2006)*, Magdeburger Schriften zum Empirischen Software Engineering, pages 161–174, Potsdam, Germany. Hasso-Plattner-Institut, Shaker Verlag.

[Rud et al., 2007c] Rud, D., Schmietendorf, A., and Dumke, R. (2007c). Performance Annotated Business Processes in Service-Oriented Architectures. *JS3T: International Journal of Simulation: Systems, Science & Technology*, 8(3):61–71. Special Issue on Performance Modelling of Computer Networks, Systems and Services.

[Rud et al., 2007d] Rud, D., Schmietendorf, A., and Dumke, R. (2007d). Resource Metrics for Service-Oriented Infrastructures. In Lübke, D., editor, *Proceedings of the Workshop on Software Engineering Methods for Service-oriented Architecture 2007 (SEMSOA 2007)*, pages 90–98. Leibniz Universität Hannover, FG Software Engineering.

[Russell and Norvig, 1995] Russell, S. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

[Russell, 2001] Russell, T. L. (2001). *The No Significant Difference Phenomenon*. The International Distance Education Certification Center, 5th edition.

[Sadiig, 2005] Sadiig, I. A. M. J. (2005). An Autonomous Mobile Agent-Based Distributed Learning Architecture-A Proposal and Analytical Analysis. *The Turkish Online Journal of Distance Education*, 6(4):106–117.

[Sampson and Karagiannidis, 2002] Sampson, D. and Karagiannidis, C. (2002). Knowledge-on-Demand in e-Learning and e-Working Settings. *Educational Technology & Society*, 5(2).

[Sampson et al., 2002] Sampson, D., Karagiannidis, C., and Cardinali, F. (2002). An Architecture for Web-based e-Learning Promoting Re-usable Adaptive Educational e-Content. *Educational Technology & Society*, 5(4).

[Sampson et al., 2004] Sampson, D. G., Lytras, M. D., Wagner, G., and Díaz, P. (2004). Ontologies and the Semantic Web for E-learning (Guest editorial). *Educational Technology & Society*, 7(4):26–28.

[Saussure, 1915] Saussure, F. (1915). *Basic Questions of General Language Science (In german)*. Walter de Gruyter & Co, Berlin. 2nd Edition 1967.

[SchemaWeb, 2008] SchemaWeb (2008). SchemaWeb. Link: http://www.schemaweb.info/.

[Scheuer and Zinn, 2007] Scheuer, O. and Zinn, C. (2007). How Did the e-Learning Session Go? The Student Inspector. In Luckin, R., koedinger, K. R., and Greer, J., editors, *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED-2007)*, volume 158, pages 487–494. IOS Press.

[Schmidt, 1983] Schmidt, H. G. (1983). Problem-Based Learning: Problem and Definition. *Medical Education*, 17:11–16.

[Schmidtner, 1996] Schmidtner, R. (1996). Adaptivity and Adaptability (In German). Link: http://www.iim.uni-giessen.de/osinet/paedagog/instrukt/cul/ADAPINT1.HTM. OsiNet-Project: The Open Study- and Information Net.

[Schmietendorf and Dumke, 2004] Schmietendorf, A. and Dumke, R. (2004). A Measurement Service for Monitoring the Quality Behaviour of Web Services Offered within the Internet. In *Proceedings of the International Workshop on Software Measurement and DASMA Software Metrik Kongress (IWSM/MetriKon 2004)*, pages 381–390. Shaker Publ.

[Schmietendorf et al., 2004] Schmietendorf, A., Dumke, R., and Reitz, D. (2004). SLA Management - Challenges in the Context of Web-Service-Based Infrastructures. In *Proceedings of the IEEE International Conference on Web Services (ICWS 2004)*, pages 606–613, San Diego, USA.

[Schmietendorf and Mencke, 2008] Schmietendorf, A. and Mencke, S. (2008). Evaluation Aspects for the Sustainable Integration of e-Learning in the Domain of Software Engineering (In German). In *Proceedings of the 18th International Workshop in Software Measurement in conjunction with the Mensura and the DASMA Software Metric Kongress (ISWM/Mensura/Metrikon 2008)*, volume 5338 of *Lecture Notes in Computer Science*, Munich, Germany. Springer Publ.

[Schools Interoperability Framework (SIF) Association, 2007] Schools Interoperability Framework (SIF) Association (2007). Schools Interoperability Framework Implementation Specification 2.0r1. Technical report, Schools Interoperability Framework (SIF) Association.

[Self, 1994] Self, J. (1994). Formal Approaches to Student Modelling. In Greer, J. E. and McCalla, G. I., editors, *Student Modelling: The Key to Individualized Knowledge-Based Instruction*, volume 25 of *NATO ASI Series F: Computer and Systems Sciences*. Springer.

[Shannon and Weaver, 1949] Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois, USA.

[Sheth et al., 2002a] Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., and Warke, Y. (2002a). Managing Semantic Content for the Web. *IEEE Internet Computing*, 06(4):80–87.

[Sheth et al., 2002b] Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., and Warke, Y. (2002b). Semantic Content Management for Enterprises and the Web. *IEEE Internet Computing*, pages 80–87.

[Shi et al., 2000] Shi, H., Shang, Y., and Chen, S.-S. (2000). A Multi-Agent System for Computer Science Education. In *Proceedings of the 5th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2000)*, pages 1–4, Helsinki, Finland.

[Shih et al., 2008] Shih, M., Feng, J., and Tsai, C.-C. (2008). Research and Trends in the Field of e-Learning from 2001 to 2005: A Content Analysis of Cognitive Studies in Selected Journals. *Computers & Education*, 51(2):955–967.

[Shneiderman and Plaisant, 2005] Shneiderman, B. and Plaisant, C. (2005). *Designing the User Interface – Strategies for Effective Human-Computer Interaction*. Pearson Education, 4th edition.

[Simon, 2002] Simon, B. (2002). Do E-learning Standards Meet their Challanges? In *Proceedings of the Workshop Standardisation in E-learning (In German)*, pages 14–22, Frankfurt/Main, Germany.

[Sinitsa, 2000] Sinitsa, K. (2000). Learning Individually: A Life-Long Perspective. Introduction to the Special Issue. *Educational Technology and Society*, 3(1).

[Sipser, 1996] Sipser, M. (1996). *Introduction to the Theory of Computation*. Course Technology.

[SIVECO Romania SA., 2005] SIVECO Romania SA. (2005). Advanced eLearning Objects.

[Smith, 1999] Smith, A. S. G. (1999). *Application of Machine Learning Algorithms in Adaptive Web-based Information Systems*. PhD thesis, School of Computing Science, Middlesex University, UK.

[Smolle and Sure, 2002] Smolle, P. and Sure, Y. (2002). FRED: Ontology-based Agents for Enabling E-Coaching Support in a Large Company. In *Second International Workshop on Ontologies in Agent Systems (OAS 2002), held at the 1st International Conference on Autonomous Agents & Multiagent Systems*, Bologna, Italy.

[Sánchez, 1997] Sánchez, J. A. (1997). A Taxonomy of Agents. Technical Report ICT-97-1, Universidad de las Américas-Puebla, México.

[Soh et al., 2005] Soh, L.-K., Blank, T., Miller, L. D., and Person, S. (2005). ILMDA: An Intelligent Learning Materials Delivery Agent and Simulation. In *Proceedings of the 2005 IEEE International Conference on Electro Information Technology (EIT'2005)*, Lincoln, USA.

[Soh et al., 2004] Soh, L.-K., Jiang, H., and Ansorge, C. (2004). Agent-Based Cooperative Learning: A Proof-of-Concept Experiment. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*, pages 368–372, Norfolk, USA.

[Sosnovsky et al., 2007] Sosnovsky, S., Dolog, P., Henze, N., Brusilovsky, P., and Nejdl, W. (2007). Translation of Overlay Models of Student Knowledge for Relative Domains Based on Domain Ontology Mapping. In Luckin, R., koedinger, K. R., and Greer, J., editors, *Proceedings of the 13th International Conference on Artificial Intelligence in Education (AIED-2007)*, volume 158, pages 289–296. IOS Press.

[Sowa, 2000] Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co.

[Spitta, 2006] Spitta, T., editor (2006). *Informationswirtschaft*. BWL im Bachelor-Studiengang. Springer.

[Süss, 2000] Süss, C. (2000). Adaptive Knowledge Management: A Meta-Modeling Approach and its Binding to XML. In Klein, H.-J., editor, *12th GI-Workshop Grundlagen von Datenbanken*.

[Staab et al., 2001] Staab, S., Studer, R., Schnurr, H.-P., and York, S. (2001). Knowledge Processes and Ontologies. *IEEE Intelligent Systems*, 16(1):26–34.

[Stojanov et al., 2005] Stojanov, S., Ganchev, I., Popchev, I., O'Droma, M., and Doychev, E. (2005). An Approach for the Development of Agent-Oriented Distributed eLearning Center. In *Proceedings of the International Conference on Computer Systems and Technologies (CompSysTech' 2005)*, Varna, Bulgaria.

[SUN Microsystems, Inc., 2003] SUN Microsystems, Inc. (2003). E-learning Framework. Technical report, SUN Microsystems, Inc.

[Swartout et al., 1996] Swartout, B., Patil, R., Knight, K., and Russ, T. (1996). Towards Distributed Use of Large-Scale Ontologies. In *Proceedings of the 10 th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96)*, Banff, Canada.

[Sycara et al., 1996] Sycara, K., Pannu, A., Williamson, M., Zeng, D., and Decker, K. (1996). Distributed Intelligent Agents. *IEEE Expert: Intelligent Systems and Their Applications*, 11(6):36–46.

[Tallon and Scannell, 2007] Tallon, P. P. and Scannell, R. (2007). Information Life Cycle Management. *Commununications of the ACM*, 50(11):65–69.

[Tao et al., 2003] Tao, F., Chen, L., Cox, S., Shadbolt, N., Puleston, C., and Goble, C. (2003). Semantic Support for Grid-Enabled Design Search in Engineering. In *Proceedings of the 1st Semantic Grid Workshop*.

[Tartir et al., 2005] Tartir, S., Arpinar, I. B., Moore, M., Sheth, A. P., and Aleman-Meza, B. (2005). OntoQA: Metric-Based Ontology Quality Analysis. In *Proceedings of IEEE ICDM 2005 Workshop on Knowledge Acquisition from Distributed, Autonomous, Semantically Heterogeneous Data and Knowledge Sources*.

[Telecom Italia, 2007] Telecom Italia (2007). JADE. Link: http://jade.tilab.com/.

[The Internet Corporation for Assigned Names and Numbers, 2007] The Internet Corporation for Assigned Names and Numbers (2007). MIME Media Types. Link: http://www.iana.org/assignments/media-types/.

[Trabucchi, 2001] Trabucchi, R. (2001). The KOD (Knowledge on Demand) Packaging Toolkit. In *3rd SCHEMAS Workshop*, Budapest, Hungary.

[Trevena, 2007] Trevena, L. J. (2007). Problem-Based Learning in Public Health Workforce Training: A Discussion of Educational Principles and Evidence. *N S W Public Health Bulletin*, 18(1-2):4–8.

[Tsou and Buttonfield, 1998] Tsou, M.-H. and Buttonfield, B. (1998). An Agent-Based, Global User Interface for Distributed Geographic Information Services. In *Proceedings of the 7th International Symposium on Spatial Data Handling*, pages 602–612, Vancouver, Canada.

[Ueno, 2005] Ueno, M. (2005). Intelligent LMS with an Agent that Learns from Log Data. In Richards, G., editor, *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN) 2005*, pages 3169–3176, Vancouver, Canada.

[Ullrich, 2004] Ullrich, C. (2004). Description of an Instructional Ontology and its Application in Web Services for Education. In *Proceedings of Workshop on Applications of Semantic Web Technologies for E-learning (SW-EL'04)*, pages 17–23, Hiroshima, Japan.

[UNESCO, 1987] UNESCO (1987). *UNESCO, Distance Learning Systems and Structures: Training Manual, Report of a Sub-Regional Training Workshop*. Bangkok : UNESCO Regional Office for Education in Asia and the Pacific, Colombo, Sri Lanka.

[Unicode Consortium, 2008] Unicode Consortium (2008). The Unicode Standard. Link: http://www.unicode.org/.

[Uschold and Grüninger, 1996] Uschold, M. and Grüninger, M. (1996). Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, 11(2):93–155.

[van der Vet and Mars, 1998] van der Vet, P. E. and Mars, N. J. I. (1998). Bottom-Up Construction of Ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 10(4):513–526.

[van Rosmalen et al., 2005] van Rosmalen, P., Brouns, F., Tattersall, C., Vogten, H., van Bruggen, J., Sloep, P., and Koper, R. (2005). Towards an Open Framework for Adaptive, Agent-Supported E-learning. *International Journal of Continuing Engineering Education and Lifelong Learning*, 15(3-6):261–275.

[Vidgen et al., 2001] Vidgen, R., Goodwin, S., and Barnes, S. (2001). Web Content Management. In *Proceedings of the 14th Bled Electronic Commerce Conference*, Bled, Slovenia.

[Virvou and Kabassi, 2002] Virvou, M. and Kabassi, K. (2002). F-SMILE: An Intelligent Multi-Agent Learning Environment. In *Proceedings of 2002 IEEE International Conference on Advanced Learning Technologies - ICALT 2002*.

[Virvou and Katsionis, 2003] Virvou, M. and Katsionis, G. (2003). Web Services for an Intelligent Tutoring System that Operates as a Virtual Reality Game. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 2003*, volume 1, pages 378–383, Washington, D.C., USA.

[Vornholt and Mencke, 2008] Vornholt, S. and Mencke, S. (2008). Ontologies for the Virtual Engineering Process. In *Proceedings of the 11. IFF-Wissenschaftstage*, Magdeburg, Germany. Fraunhofer IFF.

[Vouk et al., 1999] Vouk, M. A., Bitzer, D. L., and Klevans, R. L. (1999). Work-Flow and End-User Quality of Service Issues in Web-Based Education. *IEEE Transactions on Knowledge and Data Engineering*, 11(4):673–687.

[W3C Glossary, 1999a] W3C Glossary (1999a). Content Definition. Link: http://www.w3.org/2003/glossary/keyword/All/?keywords=content.

[W3C Glossary, 1999b] W3C Glossary (1999b). Metadata Definition. Link: http://www.w3.org/2003/glossary/keyword/All/?keywords=metadata.

[Walther, 1992] Walther, J. B. (1992). Interpersonal Effects in Computer-Mediated Interaction: A Relational Perspective. *Communication Research*, 19:52–90.

[Wang and King, 2000] Wang, Y. and King, G. (2000). *Software Engineering Processes: Principles and Applications*. CRC Press, Inc., Boca Raton, FL, USA.

[Web 3D Consortium, 2004] Web 3D Consortium (2004). ISO/IEC 19775:2004: X3D International Specification Standards. Link: http://www.web3d.org/x3d/specifications/x3d_specification.html.

[Webster and Hackley, 1997] Webster, J. and Hackley, P. (1997). Teaching Effectiveness in Technology-Mediated Distance Learning. *Academy of Management Review*, 40(6):1282–1309.

[Weglarz, 2004] Weglarz, G. (2004). Two Worlds of Data: Unstructured and Structured. *DM Review Magazine*.

[Welte, 1974] Welte, W. (1974). *Modern Linguistic: Terminology (In german)*. Max Hueber Verlag, München.

[Weske, 2007] Weske, M. (2007). *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag Berlin Heidelberg.

[Wiley, 2000] Wiley, D. A. (2000). *Learning Object Design and Sequencing Theory*. PhD thesis, Brigham Young University.

[Wille, 2005] Wille, C. (2005). *Agent Measurement Framework*. PhD thesis, Otto-von-Guericke University of Magdeburg.

[Wille et al., 2006] Wille, C., Braungarten, R., and Dumke, R. (2006). Addressing Drawbacks of Software Measurement Data Integration. In *Proceedings of the 3rd Software Measurement European Forum (Smef 2006)*, pages 209–224, Rome, Italy.

[Wille and Dumke, 2005] Wille, C. and Dumke, R. (2005). A Framework for Measuring and Evaluation of the Development Process of Multi Agent Systems. In *Proceedings of the 2nd International Workshop on eServices and eLearning*, pages 47–62, Plovdiv, Bulgaria. Eigenverlag Otto-von-Guericke-Universität Magdeburg.

[Wille and Dumke, 2007] Wille, C. and Dumke, R. (2007). Recording and Evaluation of Metrics about Web Pages (In German). In Büren, G. and Bundschuh, M., editors, *Proceedings of the DASMA Software Metrik Kongress (MetriKon 2007)*, pages 241–252, Kaiserslautern, Germany. Shaker Verlag.

[Wille et al., 2004] Wille, C., Dumke, R., Abran, A., and Desharnais, J.-M. (2004). E-Learning Infrastructure for Software Engineering Education: Steps in Ontology Modeling for SWEBOK. In *Proceedings of the IASTED Conference on Web-Based Education (IASTED 2004)*, Innsbruck, Austria.

[Wille et al., 2002] Wille, C., Dumke, R., and Stojanow, S. (2002). Quality Assurance in Agent-Based Systems - Current State and Open Problems. Technical report, Otto-von-Guericke University of Magdeburg.

[Wilson et al., 2004] Wilson, S., Blinco, K., and Rehak, D. (2004). An e-Learning Framework - A Summary. Technical report, DEST and JISC-CETIS and Industry Canada.

[Wooldridge and Jennings, 1995] Wooldridge, M. and Jennings, N. R. (1995). Intelligent Agents: Theory and Practice. *Knowledge Engineering Review*, 10(2):115–152.

[Wooldridge, 1996] Wooldridge, M. J. (1996). Agents as a Rorschach Test: A Response to Franklin and Graesser. In Müller, J. P., Wooldridge, M., and Jennings, N. R., editors, *Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, 1996, Proceedings*, pages 47–48, Budapest, Hungary.

[Wooldridge and Jennings, 1994] Wooldridge, M. J. and Jennings, N. R. (1994). Agent Theories, Architectures, and Languages: A Survey. In Wooldridge, M. J. and Jennings, N. R., editors, *Intelligent Agents: ECAI-94 Workshop on Agent Theories, Architectures, and Languages*, pages 1–39. Springer, Berlin,.

[World Wide Web Consortium (W3C), 1999] World Wide Web Consortium (W3C) (1999). HTML 4.01 Specification. Link: http://www.w3.org/TR/html4/.

[World Wide Web Consortium (W3C), 2003a] World Wide Web Consortium (W3C) (2003a). Extensible Markup Language (XML). Link: http://www.w3.org/XML/.

[World Wide Web Consortium (W3C), 2003b] World Wide Web Consortium (W3C) (2003b). SOAP: Simple Object Access Protocol version 1.2. Link: http://www.w3.org/TR/soap/.

[World Wide Web Consortium (W3C), 2004a] World Wide Web Consortium (W3C) (2004a). Extensible Markup Language Schema (XML Schema). Link: http://www.w3.org/TR/xmlschema-0/.

[World Wide Web Consortium (W3C), 2004b] World Wide Web Consortium (W3C) (2004b). Resource Description Framework (RDF) Primer. Link: http://www.w3.org/RDF-primer/.

[World Wide Web Consortium (W3C), 2004c] World Wide Web Consortium (W3C) (2004c). Semantic Markup for Web Services (OWL-S). Link: http://www.w3.org/Submission/OWL-S/.

[World Wide Web Consortium (W3C), 2004d] World Wide Web Consortium (W3C) (2004d). Web Ontology Language (OWL). Link: http://www.w3.org/2004/OWL/.

[World Wide Web Consortium (W3C), 2004e] World Wide Web Consortium (W3C) (2004e). Web Service Architecture Requirements (W3C Working Group Note 11). Link: http://www.w3.org/TR/wsa-reqs/.

[World Wide Web Consortium (W3C), 2005] World Wide Web Consortium (W3C) (2005). Glossary of Terms for Device Independence. Link: http://www.w3.org/TR/2005/WD-di-gloss-20050118/.

[World Wide Web Consortium (W3C), 2007] World Wide Web Consortium (W3C) (2007). Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. Link: http://www.w3.org/TR/wsdl20/.

[Wu, 2002] Wu, H. (2002). Designing a Reusable and Adaptive E-learning System. Master's thesis, University of Saskatchewan, Saskatoon, Canada.

[Wu et al., 1998] Wu, H., Houben, G.-J., and de Bra, P. (1998). AHAM: A Reference Model to Support Adaptive Hypermedia Authoring. In *Proceedings of the "Zesde Interdisciplinaire Conferentie Informatiewetenschap"*, pages 77–88, Antwerp, Belgium.

[Wu et al., 2006] Wu, X., Zhu, L., Guo, J., Zhang, D.-Y., and Lin, K. (2006). Prediction of Yeast Protein-Protein Interaction Network: Insights from the Gene Ontology and Annotations. *Nucleic Acids Research*, 34(7):2137–2150.

[Yao et al., 2005] Yao, H., Orme, A. M., and Etzkorn, L. (2005). Cohesion Metrics for Ontology Design and Application. *Journal of Computer Science*, 1(1):107–113.

[Yeaxlee, 1929] Yeaxlee, B. A. (1929). *Lifelong Education*. Cassel, London, UK.

[Yi et al., 2001] Yi, X., Siew, C. K., and Syed, M. R. (2001). Agent-Mediated Online Learning - AMOL. In *Workshop on Agent Supported Cooperative Work*, Montreal, Canada.

[Zeng et al., 2003] Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., and Sheng, Q. Z. (2003). Quality Driven Web Services Composition. In *Proceedings of the Twelfth International World Wide Web Conference (WWW 2003)*, pages 411–421, Budapest, Hungary.

[Zenker et al., 2008] Zenker, N., Kunz, M., and Mencke, S. (2008). Resource Consumption in Heterogeneous Environments. In *Proceedings of the 2008 International Conference on Grid Computing and Applications (GCA 2008)*, Las Vegas, Nevada, USA.

[Zinn and Scheuer, 2006] Zinn, C. and Scheuer, O. (2006). Getting to Know your Student in Distance-Learning Contexts. In *Proceedings of the European Conference on Technology Enhanced Learning (EC-TEL 2006)*, volume 4227 of *Lecture Notes in Computer Science*, pages 437–451, Crete, Greece. Springer Publ.

Curriculum Vitae

1978, August 12	Born in Brandenburg (Havel), Germany
1983 - 1990	Johann-Wolfgang-von-Goethe Oberschule Pritzerbe, Germany
1990 - 1998	Gymnasium Premnitz, Premnitz, Germany; graduated with German "Abitur"
1998 - 1999	Military Service, Brandenburg (Havel), Germany
1999 - 2005	Studies in Computer Science, Otto-von-Guericke-Universität Magdeburg, Germany; graduated with degree "Diplom-Informatiker" (equivalent to MSc in Computer Science)
2003 - 2004	Student assistant, Logistics and Factory Systems Business Unit, Fraunhofer Institute for Factory Operation and Automation IFF, Magdeburg, Germany
2004 - 2005	Student assistant, Department of Distributed Systems, Faculty of Computer Science, Otto-von-Guericke-Universität Magdeburg, Germany
2005 - 2008	Ph.D. student, Department of Distributed Systems, Otto-von-Guericke-Universität Magdeburg, Germany

Magdeburg, January 25, 2009