

WISSENSREPRÄSENTATION ZUR INTEGRIERTEN STEUERUNG VON
EXPERIMENTEN IN DER ECHTZEIT-FMRT: NEUROÖKONOMISCHE
UNTERSUCHUNG SOZIALER INTERAKTION

Dissertation

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing)

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

von: Dipl. Ing. Maurice Hollmann
geb. am 01.12.1977 in Magdeburg

Gutachterinnen/Gutachter:

Prof. Dr. Dr. Johannes Bernarding

Prof. Dr. Heinz Handels

PD Dr. Jürgen Braun

Eingereicht am: 17. August 2009

Verteidigt am: 25. November 2009

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, 17.08.2009

Maurice Hollmann

Danksagung

An dieser Stelle möchte ich mich ganz herzlich bei all jenen Menschen bedanken, die es mir mit ihrer Hilfe ermöglicht haben diese Arbeit zu verfassen. Mein Dank geht besonders an meinen Doktorvater Prof. Johannes Bernarding, der mich für das Arbeitsgebiet der Neurowissenschaft begeistert hat und selbst in stressigen Zeiten immer ein offenes Ohr für meine Probleme hat. Des Weiteren bedanke ich mich bei PD Jürgen Braun und Prof. Heinz Handels für die Begutachtung dieser Arbeit und die fruchtbaren Diskussionen und Hinweise im Vorfeld der Dissertation. Der Austausch mit meinem Betreuer und den Gutachtern ermöglichte mir von ihren Erfahrungen zu profitieren, was für jeden jungen Wissenschaftler unersetzlich ist.

Mein Dank geht auch an die Mitarbeiter unseres Institutes, die mir jederzeit mit Anregungen und Rat zur Seite standen. Besonders mein Kollege Charles Müller hat mit der im Rahmen seiner Diplomarbeit entstandenen Schnittstelle einen wesentlichen Teil zu dieser Arbeit beigetragen. Ich möchte mich auch bei Daniela Adolf bedanken, die mir mit ihrer Expertise im Bereich der Statistik aus mancher Klemme herausgeholfen hat. Für den regen Erfahrungsaustausch und hilfreiche Diskussionen zur multivariaten Analyse funktioneller MRT-Daten bedanke ich mich herzlich bei PD Jochem Rieger und Christoph Reichert. Ihre Ideen und Anregungen haben entscheidende Impulse für die Auswertung der Studie zu sozialen Interaktionen geliefert.

Sehr wichtig für das Gelingen dieser Arbeit war auch die Unterstützung der Fakultät für Informatik. Insbesondere das Prüfungsamt und das Institut für Simulation und Grafik haben mit ihrer professionellen Organisation und Hilfsbereitschaft geholfen die eine oder andere Hürde zu überwinden. Weiterhin danke ich Prof. Bernhard Preim und Prof. Dietmar Rösner für ihre Hilfe bezüglich praktischer und fachlicher Probleme. Es ist ein besonderes Merkmal der Fakultät für Informatik, dass Professoren sich ganz selbstverständlich die Zeit nehmen junge Wissenschaftler zu unterstützen, ein Umstand, von dem ich stark profitiert habe.

Ganz besonders möchte ich mich natürlich bei meiner Familie, meinen Freunden und meiner Freundin bedanken. Die Unterstützung im privaten Bereich hat mir stets den nötigen Rückhalt gegeben, wenn schwierige Phasen der Dissertation anstanden.

Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
Abkürzungsverzeichnis	IX
1. Einführung	1
1.1. Funktionelle Bildgebung	2
1.2. Echtzeit-fMRT	2
1.3. Herausforderungen der Echtzeit-fMRT	3
1.4. Zielstellung der Arbeit	4
1.5. Gliederung der Arbeit	5
2. Funktionelle MRT: Messtechnik und Datenverarbeitung	7
2.1. Magnetresonanztomographie	7
2.1.1. Physik der Magnetresonanztomographie	8
2.1.2. Relaxation	10
2.1.3. Ortskodierung	13
2.1.4. Kontrastwichtung	15
2.1.5. Messsequenzen	16
2.2. Funktionelle Magnetresonanztomographie	18
2.2.1. Physiologische Grundlagen	18
2.2.2. Modellannahmen der HRF	20
2.2.3. Aufnahmeverfahren und Design von fMRT-Experimenten	21
2.2.4. Vorverarbeitung funktioneller MRT-Daten	23
2.2.5. Auswertung funktioneller MRT-Daten: Univariate Verfahren und Klassifikation	26
3. Hirn-Computer Schnittstellen auf der Basis von Echtzeit-fMRT	35
3.1. Technische Lösungen und Auswerteverfahren der Echtzeit-fMRT	35

3.1.1.	Vorverarbeitung	37
3.1.2.	Auswertung	37
3.2.	Anwendungsgebiete der Echtzeit-fMRT	38
3.3.	Zeitverlauf der HRF und Feedbackstrategien	39
4.	Problembeschreibung	43
4.1.	Informationstheoretische Anforderungen an die Echtzeit-fMRT	43
4.2.	Einführung eines Beispiexperimentes	45
4.3.	Verwandte Arbeiten	46
5.	Wissensrepräsentation zur Beschreibung von Experimenten in der Echtzeit-fMRT	49
5.1.	Strukturierte Informationsrepräsentation mit XML	50
5.2.	Wissensrepräsentation in regelbasierten Systemen	51
5.3.	Anforderungen	53
5.4.	Entwurf	54
5.5.	EDL und EDL-Regeln als Wissensrepräsentation	55
5.5.1.	EDL-Spezifikation	55
5.5.2.	Beschreibung der EDL-Regelbasis	66
5.6.	Anwendung auf das Beispiexperiment	67
5.7.	Diskussion	72
6.	Eine Softwareumgebung für die Echtzeit-fMRT auf Basis von EDL	77
6.1.	Eine Echtzeit-fMRT-Applikation auf Basis einer Experimentbeschreibung mit EDL: rtExplorer	77
6.1.1.	Anforderungen und Entwurf	78
6.1.2.	Implementierung	81
6.1.3.	Untersuchungen zu Leistungsmerkmalen	84
6.2.	Stimuluspräsentation auf Basis einer Experimentbeschreibung mit EDL .	85
6.3.	Die EDL-Schnittstelle zur Datenakquisition	86
6.4.	Der EDL-Editor	89
6.5.	Anwendung auf das Beispiexperiment	91
6.6.	Untersuchungen zur Nutzerfreundlichkeit der Softwareumgebung	92
6.7.	Ergebnisse und Diskussion	94
7.	Vorhersage von Entscheidungen in sozialen Interaktionsszenarien mit Hilfe der Echtzeit-fMRT	99
7.1.	Problembeschreibung	99

7.2.	Soziale Kognition	100
7.3.	Interaktionsszenarien in Neuroökonomischen Experimenten: Das Ultimatum Game	103
7.4.	Methodik und Materialien	105
7.4.1.	Probanden und Paradigma	106
7.4.2.	Technischer Aufbau und Datenakquisition	108
7.4.3.	Integration in das EDL-Framework	110
7.4.4.	Echtzeit-Verarbeitung und Klassifikation	110
7.4.5.	Ermittlung des Rateniveaus	113
7.4.6.	Offline-Klassifikation	114
7.5.	Ergebnisse und Diskussion	115
7.5.1.	Klassifikationsergebnisse	115
7.5.2.	Verhaltensbezogene Ergebnisse	117
7.5.3.	Diskussion und Ausblick	119
8.	Zusammenfassung und Ausblick	123
8.1.	Zusammenfassung	123
8.2.	Ausblick	125
8.2.1.	Erstellungsmöglichkeiten für Experimente	126
8.2.2.	Neuroökonomische Experimente und Echtzeit-fMRT	126
8.2.3.	Echtzeit-Klassifikation und Neurofeedback	127
	Literaturverzeichnis	i
	Anhang	xvii
A.	Auszüge EDL-Spezifikation	xvii
A.1.	EDL-Codeauszüge	xvii
A.1.1.	Element <i>paradigm</i>	xvii
A.1.2.	Element <i>mediaObject: contentComplex</i>	xviii
A.1.3.	Element <i>timeTable</i>	xix
A.1.4.	Ultimatum Game: <i>mriParams</i>	xix
A.1.5.	Ultimatum Game: <i>analysisRuntimeModule</i>	xix

Abbildungsverzeichnis

2.1. Orientierung des Eigendrehimpulses	9
2.2. Darstellung der longitudinalen Relaxation	10
2.3. Verlust der Transversalmagnetisierung	11
2.4. Kombinierte Darstellung der Entwicklung der Magnetisierung	12
2.5. Schichtsektion	13
2.6. Kodierung in X-Richtung (Readout)	14
2.7. Kodierung in Y-Richtung (Phasenkodierung)	15
2.8. Kontraste in der MRT	16
2.9. Sequenzdiagramm einer Spin-Echo Sequenz	17
2.10. Lokale Sauerstoffanreicherung aufgrund neuronaler Aktivität	19
2.11. Verlauf einer hämodynamischen Antwortfunktion (HRF)	20
2.12. Block-Design und Ereigniskorreliertes Design	22
2.13. Signalverlauf eines Voxels	24
2.14. Darstellung einer Korrelationsfunktion	27
2.15. Darstellung eines zweidimensionalen Klassifikationsproblems	31
2.16. Darstellung eines nicht-linear trennbaren Klassifikationsproblems	34
3.1. Allgemeiner Aufbau eines Echtzeit-fMRT-Systems	36
3.2. Aktivierungsvisualisierung	41
4.1. Subsysteme einer Echtzeit-fMRT-Implementierung	44
4.2. Stimulusdesign des Beispielerperimentes	46
5.1. Informationsstruktur des EDL-Frameworks	56
5.2. EDL-Grundstruktur	57
6.1. Komponenten der Applikation	80
6.2. Benutzeroberfläche des <i>rtExplorers</i>	82
6.3. Einbindung der Schnittstelle in die Scannerumgebung	87
6.4. Erweiterung der grafischen Oberfläche der Sequenz	89

6.5. Die grafische Benutzerschnittstelle des EDL-Editors	90
7.1. Darstellung neuronaler Korrelate der sozialen Interaktion	101
7.2. Darstellung der Ergebnisse von Sanfey et al.	106
7.3. Ablauf eines Trials im Ultimatum Game	107
7.4. Aufbau des Echtzeit-fMRT-Systems und interne Datenströme	109
7.5. Aktivierte Regionen Kontrast: Fair vs. Unfair	111
7.6. Aktivierte Regionen Kontrast: Unfair vs. Fair	112
7.7. Ablaufschema der Echtzeit-Verarbeitung für jeden Trial	112
7.8. Schematische Darstellung der Lokalisation der vordefinierten ROI's . . .	114
7.9. Ablaufschema für die Offline-Verarbeitung	114
7.10. Korrekt vorhergesagte Entscheidungen (Echtzeit-Klassifikation)	116
7.11. Korrekt klassifizierte Entscheidungen (offline-Klassifikation)	117
7.12. Prozentuale Akzeptanzrate der Angebote im Ultimatum Game	118

Tabellenverzeichnis

3.1. Wichtige Echtzeit-fMRT-Studien	40
6.1. Durchschnittliche Verarbeitungszeit für 3D-Datensätze	84
6.2. Bearbeitungszeit der Teilaufgaben der Nutzbarkeitsanalyse	93
6.3. Bewertungen der Nutzbarkeitsanalyse	94
7.1. Mittelpunkte der verwendeten ROI's	113

Abkürzungsverzeichnis

ACC	Anterior Cingulate Cortex
BCI	Brain Computer Interface
BOLD	Blood Oxygen Level Dependent
CBF	Cerebral Blood Flow
CBV	Cerebral Blood Volume
CT	Computertomographie
DACC	Dorsales Anteriores Cingulum
DOM	Document Object Model
DPA	Direct Protocol Access
EDL	Experiment Description Language
EPI	Echo Planar Imaging
fMRT	funktionelle Magnetresonanztomographie
FDR	False Discovery Rate
FWE	Family Wise Error
GLM	General Linear Model
GUI	Graphical User Interface
HBI	Human Brain Interface
HRF	Hemodynamic Response Function
HTML	Hypertext Markup Language
ICA	Independent Component Analysis
ICE	Image Calculation Environment
IDEA	Integrated Development Environment for Applications
LOOCV	Leave One Out Cross Validation
LPFC	Lateraler Präfrontaler Kortex
MPFC	Medialer Präfrontaler Kortex
MRT	Magnetresonanztomographie
MTL	Medialer Temporallappen
PCA	Principal Component Analysis

ROI	Region of Interest
rACC	Rostrales Anteriores Cingulum
RVM	Relevance Vector Machine
rtfMRI	real-time functional Magnetic Resonance Imaging
SDL	Specification and Description Language
SGML	Standard Generalized Markup Language
SNR	Signal-Rausch-Verhältnis
SOM	Schema Object Model
SPECT	Single-Photon-Emissionscomputertomographie
SVM	Support Vector Machine
TE	Echozeit
ToM	Theory of Mind
TR	Repetitionszeit
UG	Ultimatum Game
URI	Uniform Resource Identifier
VLPFC	Ventro-Lateraler Präfrontaler Kortex
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XSD	XML Schema Definition
ZNS	Zentrales Nervensystem

1. Einführung

Die Funktionsweise des menschlichen Gehirns ist schon seit langem Gegenstand intensiver Forschung. Die Suche nach neuronalen Korrelaten kognitiver Funktionen wird in der klassischen Neurowissenschaft mit der Zuordnung von Hirnarealen zu ihren Aufgaben bewerkstelligt, wobei sich eine Ausweitung der Methodik auch auf die Analyse verteilter Netzwerke von funktionalen Einheiten abzeichnet (Frackowiak et al. [2004]; Haynes u. Rees [2006]; Lange [1999]). Einen großen Anteil am rasanten Wissenszuwachs im Bereich der Neurowissenschaft verdankt man der Entwicklung moderner medizinischer Bildgebungsverfahren. Zu diesen Verfahren gehört die *Magnetresonanztomographie* (MRT¹), ein hochauflösendes Schnittbildverfahren, welches seit den 1980er Jahren etabliert wurde. Die MRT zeichnet sich durch einen sehr guten Weichteilkontrast und das Fehlen von ionisierender Strahlung aus, womit eine präzise nicht-invasive Bildgebung von Hirngewebe ermöglicht wird (Dössel [2000]). Somit spielt die MRT auch in der funktionellen Analyse des menschlichen Gehirnes eine sehr wichtige Rolle. Die dabei zugrunde liegende Technik nennt man *funktionelle Magnetresonanztomographie* (fMRT).

Die Anwendungsmöglichkeiten der fMRT werden durch ein modernes Verfahren erweitert, bei dem die funktionellen Daten bereits während der Messung ausgewertet werden. Dieses Gebiet der fMRT nennt man *Echtzeit-fMRT* (DeCharms [2008]; Weiskopf et al. [2007]). Die Implementierung von Experimenten in der fMRT und Echtzeit-fMRT ist aufgrund des technischen Aufbaus der beteiligten Systeme und der anspruchsvollen Auswerteverfahren sehr komplex. Diese Arbeit soll dazu beitragen, Experimente in der Echtzeit-fMRT für Nutzer zu vereinfachen. Die Vereinfachung wird erreicht, indem komplexe Zusammenhänge von Experimentparametern zentral abgebildet und der Verarbeitung durch Software zugänglich gemacht werden. Die folgenden Abschnitte werden einen näheren Einblick in die Problematik geben und die praktische Anwendung der Echtzeit-fMRT für Experimente in der Neuroökonomie einführen.

¹englisch: MRI, magnetic resonance imaging

1.1. Funktionelle Bildgebung

Während man früher für die Erforschung der Funktionsweise einzelner Hirnareale auf die Analyse von Funktionsstörungen aufgrund von Hirnschädigungen angewiesen war, stehen heute eine Vielzahl von invasiven und nicht-invasiven Methoden zur funktionellen Untersuchung des menschlichen Gehirns zur Verfügung (Dössel [2000]; Frackowiak et al. [2004]). Im Gegensatz zu Verfahren, welche direkt elektrische oder magnetische Potenziale bei neuronaler Aktivität ableiten, wie die Elektroencephalographie (EEG) oder Magnetencephalographie (MEG), bildet die fMRT lokale Stoffwechselprozesse bei Hirnaktivität ab. Erstgenannte Verfahren verfügen über eine zeitliche Auflösung im Millisekundenbereich, erzielen jedoch nur eine geringe räumliche Auflösung. Die fMRT hat gerade wegen der hohen räumlichen Auflösung der verwendeten Daten an Bedeutung gewonnen (Baert et al. [1999]; Huettel et al. [2004]). Mit dieser nicht-invasiven funktionellen Methodik wird die lokale hämodynamische Reaktion auf neuronale Aktivität dargestellt. Dieser hämodynamische Effekt wird als Blood-Oxygen-Level-Dependent-Effekt (BOLD-Effekt) bezeichnet (Ogawa et al. [1990]). Da das Signal-Rausch-Verhältnis (SNR) in der fMRT vergleichsweise gering ist, werden funktionelle MRT-Daten als Zeitreihen im Allgemeinen vorverarbeitet und signifikant aktivierte Volumeneinheiten mit Hilfe statistischer Verfahren bestimmt. Diese Verarbeitungsschritte sind bei der herkömmlichen fMRT als post-experimentelle Verarbeitung implementiert und können bis zu mehreren Stunden zur Auswertung benötigen (Frackowiak et al. [2004]; Friston et al. [1995]; Huettel et al. [2004]).

1.2. Echtzeit-fMRT

In den letzten Jahren gewinnt eine Technik an Bedeutung, bei der die akquirierten fMRT-Daten schon zur Laufzeit der Messung ausgewertet werden. Mit diesem Verfahren ist man in der Lage, die Hirnaktivität des Probanden im laufenden Experiment abzuschätzen. Dieses Teilgebiet der fMRT wird als *Echtzeit-fMRT*² bezeichnet (DeCharms [2008]; Weiskopf et al. [2007]). Die Echtzeit-fMRT ermöglicht neben der Laufzeitkontrolle von fMRT-Experimenten auch weiterführende Anwendungen wie die Implementierung von Hirn-Computer-Schnittstellen³ (Weiskopf et al. [2004a]; Wolpaw et al. [2002]). Somit können *Neurofeedback-Systeme* erstellt werden, bei denen Probanden Informationen „rückgemeldet“ werden, die zuvor aus dem Aktivierungszustand ihres Gehirns extrahiert wurden (DeCharms [2008]; DeCharms et al. [2004]; Weiskopf et al. [2007]). Mit dieser

²Echtzeit-fMRT englisch: real-time fMRI (rtfMRI)

³Hirn-Computer-Schnittstelle englisch: Brain-Computer Interface (BCI)

Technik ist es möglich, die willentliche Beeinflussbarkeit verschiedener neuronaler Systeme zu untersuchen und ggf. therapeutisch zu nutzen. Zu Arbeiten in diesem Bereich zählen Versuche, bei denen Probanden durch Neurofeedback unterstützt werden, bei sich selbst Traurigkeit zu induzieren (Posse et al. [2003]) oder willentlich die eigene Schmerzempfindlichkeit zu regulieren (DeCharms et al. [2005]).

Zu Kommunikations- und Steuerungssystemen zählen BCI's, die es einem Probanden erlauben, über seine Hirnaktivität mit seiner Umwelt in Verbindung zu treten oder externe Systeme zu steuern (DeCharms [2008]; Owen et al. [2006]; Yoo et al. [2004]). Gerade für die Kommunikation mit Patienten, die sich aufgrund von Hirn- oder Nervenschädigungen⁴ nicht über Laute oder Gebärden mitteilen können, sind diese Systeme hilfreich.

Potenzial birgt die Echtzeit-fMRT auch für die Untersuchung von sozialen Interaktionsszenarien, wobei eine Vielzahl von neuronalen Prozessen eine Rolle spielen (Critchley [2005]; Lieberman [2007]). Die Untersuchung menschlichen Verhaltens in interaktiven Situationen ist sowohl aus verhaltensökonomischer als auch aus neurowissenschaftlicher Sicht interessant. Die Verbindung dieser Wissenschaftsgebiete erfolgt im Bereich in der *Neuroökonomie* (Camerer et al. [2003]; Glimcher et al. [2008]; Kenning et al. [2007]; Sanfey et al. [2006]). Eine neuroökonomische Studie im Rahmen dieser Arbeit wird eine praktische Anwendung der Echtzeit-fMRT bilden.

1.3. Herausforderungen der Echtzeit-fMRT

Wie bereits ausgeführt wurde, erfolgt bei der Echtzeit-fMRT die Verarbeitung der funktionellen Daten während der laufenden Datenakquisition. Die funktionellen Bilddaten werden oftmals zur Laufzeit auf ein vom Tomographen getrenntes System übertragen, welches die Auswertung übernimmt. Die Ergebnisse der Auswertung können wiederum verwendet werden, um die Kommunikation mit dem Probanden im Scanner (z.B. über auditive oder visuelle Stimulationssysteme) oder die Steuerung externer Geräte zu implementieren (Weiskopf et al. [2007]; Wolpaw et al. [2002]).

Neben den Anforderungen an die Performanz der Auswerteverfahren müssen die Parameter der eingebundenen Systeme valide und kohärent sein. Des Weiteren müssen die Systeme in der Lage sein miteinander zu kommunizieren. Beispielsweise muss ein Ergebnis der Auswerteroutinen eine definierte Reaktion des Stimulationssystems auslösen (Weiskopf et al. [2004a, 2007]). Die Verknüpfung getrennter und mitunter sehr komplexer Parameterräume der verschiedenen Systeme, welche in ein Echtzeit-fMRT-Experiment eingebunden sind, stellt eine erhebliche Herausforderung dar (Hollmann et al. [2008b]).

⁴Zu diesen Krankheitsbildern gehört beispielsweise das Locked-in-Syndrom.

Bei herkömmlichen Echtzeit-fMRT Implementierungen obliegt die Sicherstellung der Parameterkohärenz und der korrekten Kommunikation der Systeme allein der manuellen Überprüfung durch den Nutzer. Diese Konstellation führt neben der erschwerten Bedienung der Systeme zu einer hohen Anfälligkeit für systematische Fehler und Flüchtigkeitsfehler bei der Experimentdefinition. Bestehende Implementierungen von Echtzeit-fMRT Anwendungen sind noch immer Insellösungen, die weitreichende Möglichkeiten zur automatisierten Kontrolle von Parameterkohärenzen und zur Unterstützung der Anwender bei der Experimentdefinition ungenutzt lassen (Hollmann et al. [2008b]).

Die verwendeten Methodiken in der Echtzeit-fMRT werden durch neue Auswerteverfahren wie z.B. Klassifikationsansätze erweitert (vgl. Hollmann et al. [2009]; LaConte et al. [2007] und Kapitel 7 dieser Arbeit). Neben der Verbesserung der Nutzbarkeit muss also auch die Flexibilität einer Applikation zur Echtzeit-fMRT berücksichtigt werden. Um neue Techniken aufnehmen zu können, muss eine Softwarelösung in diesem Bereich also skalierbar sein und ein Nutzer muss neben vorhandener Standardfunktionalität auch eigene, komplexe Funktionalität integrieren können.

1.4. Zielstellung der Arbeit

Ausgehend von den bisher formulierten Überlegungen wurde eine Wissensrepräsentation entwickelt, welche einen praktischen Lösungsansatz für die angesprochenen Probleme in der Echtzeit-fMRT darstellt. Zusammen mit Softwarelösungen, welche diese Wissensrepräsentation verarbeiten, sollen die folgenden Ziele erreicht werden:

1. Zusammenführung von getrennten Parameterräumen der an einem Echtzeit-fMRT-Experiment beteiligten Hard- und Softwarekomponenten
2. Implementierung der softwaregestützten Validierung von systemübergreifenden Abhängigkeiten von Parametern zur Sicherstellung der Parameterkonsistenz
3. Entwicklung von Methoden für eine schnelle und zuverlässige Auswertung von fMRT-Daten auf Basis einer integrierten Lösung zur Parameterrepräsentation
4. Evaluierung und Beurteilung des Systems mit Hilfe von Beispielexperimenten und einer Nutzbarkeitsstudie

Neben der praktischen Implementierung des Systems wird ein zweiter Bestandteil den interdisziplinären Ansatz dieser Arbeit betonen. Bei neurowissenschaftlichen Fragestellungen soll das o.g. Konzept eingesetzt werden. Insbesondere sollen menschliche Interaktionsszenarien untersucht werden und innovative methodische Techniken mit der

Echtzeit-fMRT verknüpft werden, wobei gleichzeitig die Vorteile des o.g. Frameworks genutzt werden sollen. In einer Echtzeit-fMRT-Studie wird die Vorhersagbarkeit von Probandenentscheidungen in einem Interaktionsszenario aus der Verhaltensökonomie, dem sogenannten Ultimatum Game (UG), untersucht. Es wird eine Technik vorgestellt, die Echtzeit-fMRT und multivariate Analysemethoden verbindet und somit die Vorhersage von Entscheidungen der Probanden ermöglicht. Die Teilziele dieses Bereiches sind:

1. Auswahl und Implementierung von Echtzeit-Methodiken zur Vorhersage von Probandenentscheidungen auf Basis der multivariaten Analyse von funktionellen Bild Datensätzen
2. Implementierung eines neuroökonomischen Echtzeit-fMRT-Experimentes (Ultimatum Game)
3. Auswertung und Interpretation der Ergebnisse

1.5. Gliederung der Arbeit

Im Anschluss an diesen Abschnitt werden in Kapitel 2 die messtechnischen Grundlagen der MRT betrachtet, da diese die Basis der Echtzeit-fMRT bildet. Es werden die Grundlagen der Magnetresonanztomographie selbst aber auch die Grundlagen und Auswerteverfahren funktioneller Messungen mit MRT vorgestellt.

In Kapitel 3 wird auf die funktionelle MRT in Echtzeit eingegangen, indem technische Lösungsmöglichkeiten und Auswerteverfahren skizziert werden. Weiterhin werden die Anwendungsgebiete der Echtzeit-fMRT auf Grundlage einer Übersicht über Studien der letzten Jahre erläutert.

Die Herausforderungen, die sich aus dem methodischen Aufbau von Echtzeit-fMRT Systemen ergeben, werden in Kapitel 4 erläutert. Es werden Arbeiten präsentiert, welche sich mit der Problematik integrierter Lösungen auf diesem Gebiet beschäftigt haben. Des Weiteren wird in diesem Kapitel ein Beispielszenario vorgestellt, um die entwickelte Problemstellung zu verdeutlichen. Auf dieses Beispielszenario wird in den folgenden Kapiteln Bezug genommen, um exemplarisch die hier erzielten Lösungen darzustellen.

In den Kapiteln 5 und 6 wird eine integrierte Lösung zur Echtzeit-fMRT vorgestellt. Die Grundlage bildet dabei eine Wissensrepräsentation zu Parametern von Echtzeit-fMRT-Experimenten auf der Basis eines regelbasierten Systems. Diese Wissensrepräsentation wird in Kapitel 5 erläutert, wobei auch informationstheoretische Hintergründe zur strukturierten Informationsdarstellung und zur Wissensdarstellung in regelbasierten Systemen

vermittelt werden. Die hier entwickelte Wissensrepräsentation erfolgt dabei in Form einer Beschreibungssprache⁵ für Experimente der Echtzeit-fMRT.

In Kapitel 6 wird eine Softwareumgebung vorgestellt, welche die Wissensrepräsentation nutzt, um Experimente in der Echtzeit-fMRT zu implementieren. Die an diesem Prozess beteiligten Applikationen werden beschrieben und eine Studie zur Nutzbarkeit des Frameworks wird vorgestellt.

Die Echtzeit-fMRT bietet weitreichende Möglichkeiten neuronale Aktivität während einer Messung zu analysieren. Die praktische Anwendung der Echtzeit-fMRT in einer Implementierung des zuvor dargestellten Ansatzes wird in Kapitel 7 vorgestellt. In einer neuroökonomischen Studie werden neuronale Korrelate der sozialen Interaktion untersucht. Dabei steht die Frage im Vordergrund, ob es mit Hilfe der Echtzeit-fMRT möglich ist, Entscheidungen von Probanden in einem verhaltensökonomischen Szenario vorherzusagen.

Eine Zusammenfassung und der Ausblick auf zukünftige Entwicklungsmöglichkeiten in den vorgestellten Problemfeldern schließen die Arbeit in Kapitel 8 ab.

⁵Experiment Description Language (EDL)

2. Funktionelle MRT: Messtechnik und Datenverarbeitung

Dieses Kapitel beschäftigt sich mit den theoretischen Aspekten der funktionellen MRT. Neben physikalischen Grundlagen der Magnetresonanztomographie und der Methodik zur Bildgebung mit MRT (Abschnitt 2.1) werden die Grundzüge der funktionellen MRT in Abschnitt 2.2 behandelt. Beim zweitgenanntem Abschnitt stehen statistische Verfahren und die technische Umsetzung der benötigten Funktionalitäten im Vordergrund.

2.1. Magnetresonanztomographie

Die Kernspinresonanz hat seit ihrer Entdeckung in den 1940er Jahren in der Wissenschaft immer mehr an Bedeutung gewonnen (Dössel [2000]; Haake et al. [1999]; Huettel et al. [2004]). Seit den 1980er Jahren wird sie in der Magnetresonanztomographie für die medizinische Bildgebung verwendet und unterliegt seither einer ständigen Weiterentwicklung. Diese besteht zum einen aus der technischen Entwicklung in Form von leistungsfähigeren Hardwarekomponenten und höheren Magnetfeldstärken. Zum anderen spielt jedoch die Entwicklung von Software, welche die Steuerung der Aufnahme und die Weiterverarbeitung der Daten übernimmt eine essenzielle Rolle in der MRT. Die Vermeidung ionisierender Strahlung und der gute Weichteilkontrast, der erzielt werden kann, sind die Stärken dieses Bildgebungsverfahrens. Dem stehen sehr hohe Kosten bei der Anschaffung und eine komplexe technische Umsetzung als Nachteile gegenüber.

Dieses Kapitel wird sich mit den Grundlagen der Magnetresonanztomographie befassen und dabei auf die physikalischen Prinzipien, die Methodik der Bildgebung sowie auf das Design von Messsequenzen eingehen. Dieser Abschnitt soll lediglich die zum Verständnis der Arbeit notwendigen Grundlagen vermitteln. Für eine umfassende Behandlung dieses komplexen Themenbereiches wird auf weiterführende Literatur wie z.B. Canet [1994] oder Haake et al. [1999] verwiesen.

2.1.1. Physik der Magnetresonanztomographie

Die Magnetresonanztomographie basiert auf der kernmagnetischen Resonanz (NMR). Physikalische Grundlage ist der Spin der Nukleonen, d.h., der Protonen und der Neutronen, aus welchen die Atomkerne aller Elemente zusammengesetzt sind. Nur Kerne mit ungerader Nukleonenzahl wie z.B. der des Wasserstoff- Isotops (1H), des Kohlenstoff-Isotops (^{13}C) oder des Fluor-Isotops (^{19}F) sind NMR-detektierbar (Canet [1994]). Die klinischen MR-Methoden und die überwiegende Anzahl der MR-Grundlagenforschungen basieren auf der Detektion der NMR von Protonen des Wasserstoffisotopes 1H . Da Protonen sowohl eine hohe NMR-Empfindlichkeit als auch eine hohe Konzentration in wässrigen und fetthaltigen Anteilen des Gewebes aufzeigen, konnte sich die MRT zu einem der erfolgreichsten diagnostischen Verfahren in der Medizin entwickeln.

Die genannten Isotope besitzen einen Eigendrehimpuls, den Spin. Der Spin des Wasserstoffprotons ist halbzahlig. Mit jedem Eigendrehimpuls ungleich null ist ein magnetisches Dipolmoment $\vec{\mu}$ verbunden, das mit anderen Magnetfeldern wechselwirken kann:

$$\vec{\mu} = \gamma \cdot \vec{J}. \quad (2.1)$$

Die Proportionalitätskonstante γ (gyromagnetisches Verhältnis) ist spezifisch für jeden Atomkern. Legt man ein äußeres Magnetfeld B_0 an, das vektoriell entlang der z-Achse ausgerichtet ist, so folgt daraus eine Wechselwirkungsenergie des magnetischen Momentes mit einem Magnetfeld \vec{B}_0 :

$$E = -\vec{\mu} \cdot \vec{B}_0 = -\gamma \cdot J_z \cdot B_0. \quad (2.2)$$

Diese Wechselwirkung führt zu einer Energiedifferenz, je nachdem ob der Spin parallel (niedrigere Energie) oder antiparallel (höhere Energie) zum äußeren Magnetfeld ausgerichtet ist (Haake et al. [1999]). Bei halbzahligem Kernspin erhält man für die Differenz zweier Energiestufen:

$$\Delta E = \gamma \cdot (h/2\pi) \cdot B_0. \quad (2.3)$$

Hierbei stellt h das Planck'sche Wirkungsquantum dar. Abbildung 2.1 verdeutlicht den Sachverhalt. Bevorzugt wird von den Spins der energetisch niedrigere Zustand (parallele Ausrichtung) eingenommen. Abhängig von der Energie der beiden Grundzustände im Verhältnis zur thermischen Energie ergeben sich die Besetzungszahlen n für die Zustände nach der Boltzmann-Statistik (Horowitz [1995]):

$$\frac{n \uparrow}{n \downarrow} = e^{-\frac{\Delta E}{kT}}. \quad (2.4)$$

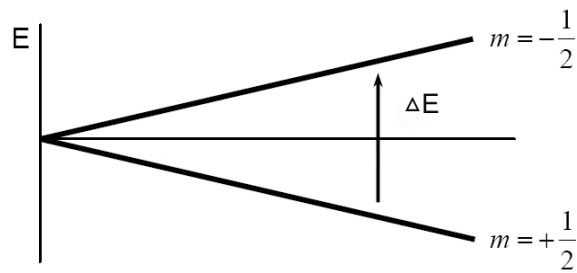


Abbildung 2.1.: Parallele und antiparallele Ausrichtung des Eigendrehimpulses zum äußeren Magnetfeld und die zwei resultierenden Energieniveaus ($m = +1/2$ und $m = -1/2$). Der Pfeil markiert die Möglichkeit des Überganges zwischen den Zuständen unter Zufuhr der Energiedifferenz ΔE .

Die Konstante k bezeichnet dabei die Boltzmann-Konstante und T die Temperatur. Bei Raumtemperatur ergibt sich für Protonen ein sehr geringer Unterschied zwischen den Besetzungszahlen von circa 10^{-6} . Eine messbare makroskopische Magnetisierung wird durch die Aufsummierung der magnetischen Momente derjenigen Spins eines bestimmten Volumens erzeugt, welche sich aus dieser Besetzungszahldifferenz ergeben (Horowitz [1995]; Siemens [2003]). Quantenmechanisch erfordert der Übergang vom Grundzustand (paralleler Spin) zum angeregten Zustand (antiparalleler Spin) die exakte Zufuhr der Energiedifferenz ΔE , was durch Einstrahlung von Energiequanten mit der sogenannten Larmorfrequenz ω erreicht werden kann:

$$\omega = \gamma \cdot |\vec{B}_0|. \quad (2.5)$$

Da hier immer eine große Anzahl von Teilchen betrachtet wird, lassen sich die meisten Phänomene in einer guten Näherung klassisch beschreiben. In der klassischen Elektrodynamik bewirkt ein äußeres Magnetfeld B_0 ein Drehmoment auf das magnetische Moment μ . Dieses hat eine Präzessionsbewegung des magnetischen Momentes mit der sogenannten Larmorfrequenz um das Feld B_0 zur Folge. Bei einer Magnetfeldstärke von 1,5 Tesla präzedieren die Spins von Wasserstoffprotonen mit einer Larmorfrequenz von ca. 63 MHz (Siemens [2003]). Die beschriebene Kreisbewegung hat bei phasenkohärenter Anregung eine relative Phasenlage der Spins untereinander zur Folge. Die Bedeutung der Phasenkohärenz eines Spin-Ensembles wird sich im Folgenden zeigen.

2.1.2. Relaxation

Um ein messbares Signal zu erhalten, welches zur Bildgebung verwendet werden kann, ist es notwendig, die makroskopische Magnetisierung mit einem magnetischen Feld der Larmorfrequenz aus ihrer „Ruhelage“ (Gleichgewichtsmagnetisierung) zu lenken. Die Anregung wird in der Regel über einen hochfrequenten elektromagnetischen Impuls (Hochfrequenz [HF]-Puls) realisiert, die magnetische Komponente wird als \vec{B}_1 -Feld bezeichnet. Obwohl theoretisch das äußere Magnetfeld \vec{B}_0 jede beliebige Richtung definieren kann, wird seine Richtung im Folgenden als Z-Achse bezeichnet und dementsprechend orthogonal dazu die XY-Ebene definiert. Damit übereinstimmend wird das zeitlich veränderliche Magnetfeld \vec{B}_1 , dessen transversale Komponente in der XY-Ebene mit der Larmorfrequenz ω rotiert, die Magnetisierung in Richtung der XY-Ebene auslenken.

Je nach Einstrahldauer des HF-Pulses wird die Magnetisierung zu einem bestimmten Teil in die XY-Ebene gelenkt. Der Winkel dieser Auslenkung wird als *Flip-Winkel* bezeichnet. Ein perfekter 90° -Puls wandelt die Z-Magnetisierung (M_z) vollständig in eine *Transversalmagnetisierung* M_{xy} um, die um die Z-Achse präzediert, wobei die verbleibende Z-Magnetisierung M_z (*Longitudinalmagnetisierung*) null wird (Haake et al. [1999]). Die präzedierende Transversalmagnetisierung erzeugt in der Empfangsspule durch Induktion eine zeitlich veränderliche Spannung, die als Signal gemessen wird.

Das Abschalten des eingestrahlt HF-Pulses führt zu zwei parallel verlaufenden Prozessen. Zum einen nimmt die longitudinale Komponente der Magnetisierung mit der Zeitkonstanten T_1 wieder ihren ursprünglichen Wert an (siehe Abbildung 2.2). Zum anderen sinkt die transversale Komponente auf null ab. Beide Prozesse erfolgen exponen-

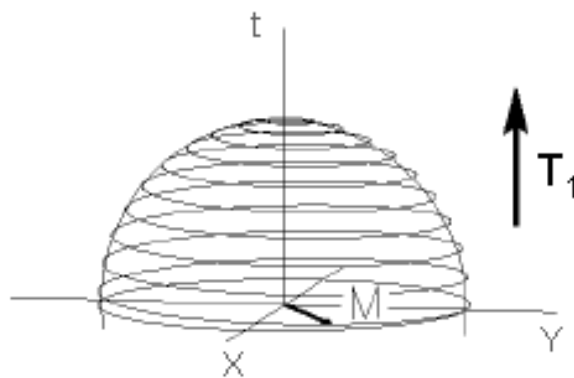


Abbildung 2.2.: Die Zeitkonstante T_1 beschreibt die longitudinale Relaxation (nach Bernarding [1997]).

tiell, wobei die Relaxation der Transversalmagnetisierung (über die Zeitkonstante T_2 beschrieben) in der Regel deutlich schneller erfolgt als die longitudinale Relaxation. Die durch die Anregung aufgenommene Energie wird dabei an die Umgebung abgegeben. Die zeitliche Entwicklung der Magnetisierung in den drei Raumkomponenten wurde von Bloch über Bewegungsgleichungen beschrieben (Bloch et al. [1946]):

$$\frac{dM_x}{dt} = \gamma \cdot (\vec{M} \times \vec{B})_x - \frac{M_x}{T_2} \quad (2.6)$$

$$\frac{dM_y}{dt} = \gamma \cdot (\vec{M} \times \vec{B})_y - \frac{M_y}{T_2} \quad (2.7)$$

$$\frac{dM_z}{dt} = \gamma \cdot (\vec{M} \times \vec{B})_z - \frac{M_z - M_0}{T_1}. \quad (2.8)$$

Die Relaxationsprozesse sind nach ihrer Ursache eingeteilt in: Spin-Gitter Relaxation und Spin-Spin Relaxation. Die Spin-Gitter Relaxation ist für die Geschwindigkeit des Wiederaufbaues der Z-Magnetisierung verantwortlich (*Longitudinale Relaxation*). Sie ergibt sich durch Wechselwirkungen der Spins mit ihrer molekularen Umgebung (dem Gitter) und ist somit spezifisch für verschiedene Moleküle, Materialien und damit auch Gewebearten (Rooney et al. [2007]). Für die zeitliche Entwicklung der Longitudinalmagnetisierung

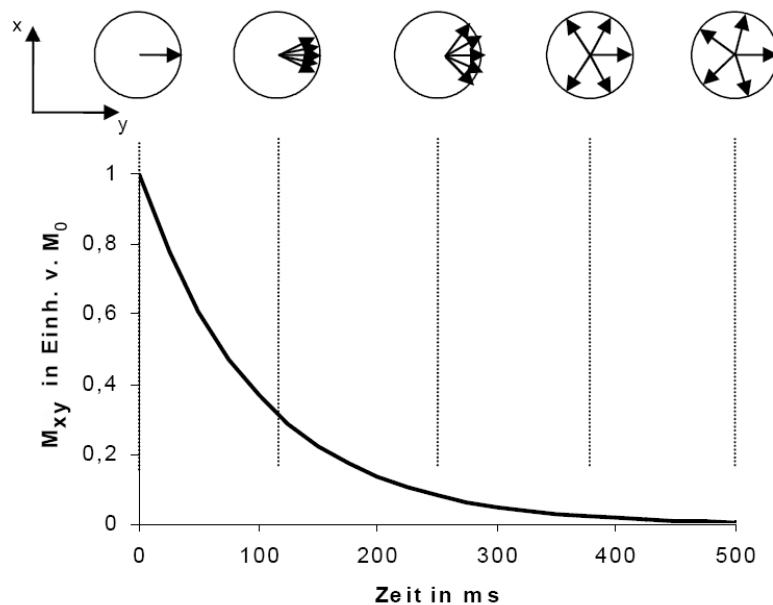


Abbildung 2.3.: Die Transversalmagnetisierung zerfällt auf Grund des Verlustes der Phasenkohärenz durch Spin-Spin Wechselwirkungen. Modifiziert nach Warmuth [2002], mit freundlicher Genehmigung des Autors.

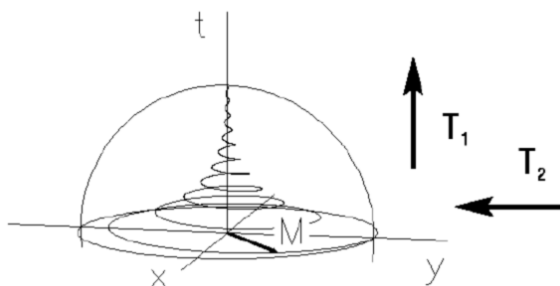


Abbildung 2.4.: Kombinierte Darstellung der Entwicklung der Magnetisierung nach Anregung mit einem 90°-Puls (nach Bernarding [1997]).

M_z nach einer Anregung mit einem Flipwinkel α und der Ausgangsmagnetisierung M_0 ergibt sich aus Gleichung 2.8:

$$M_z(t) = M_0 \cdot \cos \alpha + M_0 \cdot (1 - \cos \alpha) \cdot (1 - e^{-\frac{t}{T_1}}). \quad (2.9)$$

Hierbei ist T_1 für einen Gewebetyp als die Zeit definiert, nach der 63% der Z-Magnetisierung wieder aufgebaut sind.

Der Zerfall der Transversalmagnetisierung ist Spin-Spin Wechselwirkungen geschuldet. Dies bedeutet, dass die Phasenkohärenz der Spins nach dem HF-Puls wieder verloren geht. Dargestellt ist der Vorgang in Abbildung 2.3. Der Verlust der Phasenkohärenz geschieht um so schneller, je stärker die Kopplung benachbarter Spins ist. Genau wie T_1 ist T_2 molekül-, material- bzw. gewebespezifisch, wobei T_2 als die Zeit definiert ist, nach der 63% der XY-Magnetisierung wieder zerfallen sind (Dössel [2000]; Siemens [2004c]). Das kombinierte Verhalten der Magnetisierung ist in Abbildung 2.4 verdeutlicht. In diesem Zusammenhang ist zu bemerken, dass die Phasenkohärenz auch durch lokale Inhomogenitäten des äußeren Magnetfeldes und Suszeptibilitätsunterschiede der betrachteten Gewebe abnimmt und deshalb die effektive transversale Relaxationszeit T_2^* abhängig von diesen Einflüssen (zusammengefasst in T_2') wie folgt definiert wird:

$$\frac{1}{T_2^*} = \frac{1}{T_2} + \frac{1}{T_2'}. \quad (2.10)$$

Es gilt in der Regel:

$$T_2^* < T_2 < T_1. \quad (2.11)$$

2.1.3. Ortskodierung

Mit welchen Techniken ist es nun möglich, das Signal eines beliebigen Spin-Ensembles räumlich zuzuordnen? In den folgenden Abschnitten werden Techniken verdeutlicht, die diese Aufgabe lösen.

Schichtselektion

Im Folgenden wird der Vorgang der Schichtselektion bei sogenannten 2D-Sequenzen betrachtet. Wie im Abschnitt 2.1.1 festgestellt wurde, ist die Larmorfrequenz, mit der die Spins um die Richtung des äußeren Magnetfeldes präzedieren, direkt von der Feldstärke dieses Feldes abhängig. Durch das Schalten eines zusätzlichen Gradienten, des *Schichtselektionsgradienten*, kann nun die lokale Resonanzfrequenz entlang der Z-Richtung eingestellt werden (Abbildung 2.5) (Horowitz [1995]). Ein HF-Puls der Frequenz ω_0 wird dann nur diejenigen Spins anregen, welche durch ihre Lage im sich räumlich veränderlichen Magnetfeld ($B_0 + G_z$) eben mit der Frequenz ω_0 präzedieren. Um eine Schicht mit einer definierten Ausdehnung in Z-Richtung anzuregen, wird der HF-Puls mit einer Bandbreite von $\Delta\omega_0$ eingestrahlt. Die Verwendung eines Z-Gradienten zur Schichtselektion ist

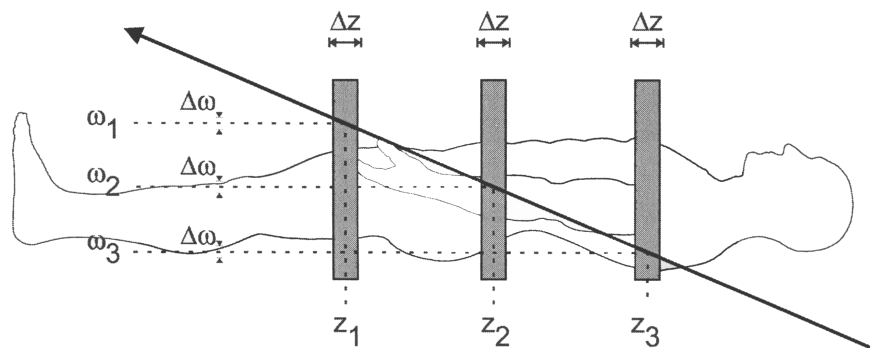


Abbildung 2.5.: Die Schichtselektion erfolgt mit Hilfe eines Schichtselektionsgradienten G_z . Dieser Gradient beeinflusst die lokale Resonanzfrequenz (hier ω_1 , ω_2 und ω_3) entlang der Z-Richtung. Der Pfeil in der Abbildung verdeutlicht die Wirkung des Gradienten G_z (modifiziert nach Brown u. Semelka [2003]).

hier nur exemplarisch, genauso kann man X-,Y-Gradienten oder Überlagerungen dieser Gradienten benutzen, um beliebig orientierte Schichten anzuregen.

Bei sogenannten 3D-Sequenzen wird das gesamte Volumen durch den HF-Puls angeregt, wobei die räumliche Zuordnung entlang der Z-Achse durch eine zusätzliche Phasenkodierung

rung, wie sie im nächsten Abschnitt für die Y-Komponente geschildert ist, erreicht wird (Reiser u. Semmler [1997]).

Frequenz und Phasenkodierung

Nach der Schichtselektion wird nun die Kodierung der Lokalisation der einzelnen Volumeneinheiten in XY-Richtung der selektierten Schicht beschrieben. Legt man während des Auslesens des Signals einen zeitlich konstanten Gradienten an, der orthogonal zum Schichtselektionsgradienten ausgerichtet ist, präzedieren die Spins entlang dieses *Frequenzkodiergradienten* oder auch *Readoutgradienten* mit steigender bzw. fallender Frequenz (vgl. Abbildung 2.6). Das gemessene komplexe Signal besteht nun aus mehreren Frequenzen, welche später durch die Fouriertransformation analysiert werden und damit eine räumliche Zuordnung des Signals in X-Richtung erlauben (Haake et al. [1999]; Reiser u. Semmler [1997]). Um die fehlende dritte Dimension (hier Y) zu kodieren,

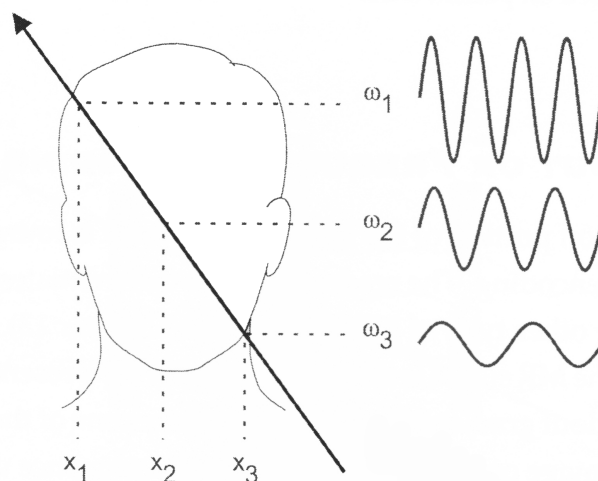


Abbildung 2.6.: Mit Hilfe eines Frequenzkodiergradienten G_{RO} wird die Position von Spin-Ensembles in X-Richtung kodiert. Die Wirkung von G_{RO} auf die lokale Frequenz wird mit dem Pfeil verdeutlicht. Die Grafik wurde modifiziert nach Brown u. Semelka [2003].

wird nach dem Schichtselektionsgradienten aber vor dem Frequenzkodiergradienten ein Gradient geschaltet, welcher orthogonal zu diesen beiden Gradienten ausgerichtet ist. Dieser *Phasenkodiergradient* verschiebt entlang seiner Richtung die Phase der selektierten Spins, mit welcher diese präzedieren. Abbildung 2.7 verdeutlicht diesen Mechanismus. Um also ein zweidimensionales Bild der Auflösung 256×256 einer Schicht aufzunehmen,

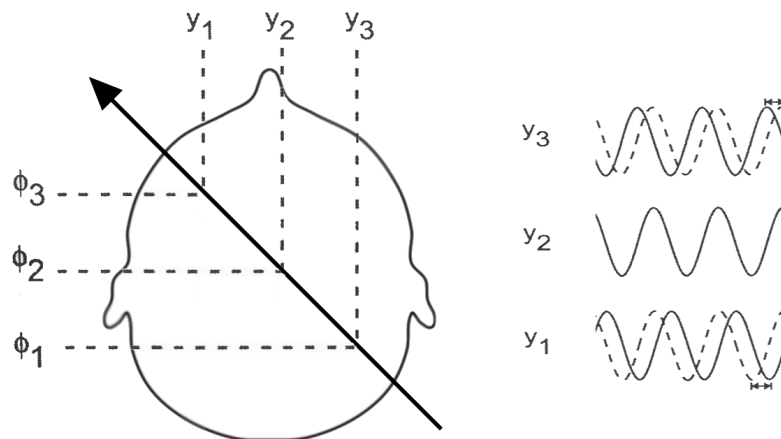


Abbildung 2.7.: Die Kodierung der Position von Spin-Ensembles in Y-Richtung erfolgt mit Hilfe der Phasenverschiebung (über den Pfeil verdeutlicht) durch einen Phasenkodiergradienten. Die Grafik wurde modifiziert nach Brown u. Semelka [2003].

muss im Standardfall die Phasenkodierung mit unterschiedlicher Amplitude des Phasenkodiergradienten 256 mal wiederholt werden. Auch die Phasenlage kann mit Hilfe der Fouriertransformation analysiert werden.

Die Rohdatenmatrix, die somit Zeile für Zeile gefüllt wird, nennt man den *K-Raum*. Jedes Element des K-Raumes ist nun eindeutig durch die Phasenausrichtung und die Frequenz kodiert. Mit Hilfe der zweidimensionalen Fouriertransformation werden die durch Spulen empfangenen Signale den Pixeln einer Schicht zugeordnet und es entsteht ein zweidimensionales Bild. Neben dem beschriebenen Verfahren existieren weitere Techniken zum Füllen des K-Raumes wie z.B. die Spiralbildgebung. Vertiefende Informationen hierzu finden sich in Haake et al. [1999] und Reiser u. Semmler [1997].

2.1.4. Kontrastwichtung

Die Kontrastwichtung in einer MRT-Messesequenz wird durch verschiedene Parameter bestimmt, von denen der Flip-Winkel, die Echozeit (TE) und die Repetitionszeit (TR) die wichtigsten sind. Dabei bestimmt TE, zu welchem Zeitpunkt nach dem HF-Puls das entstehende Signal ausgelesen werden soll, und TR, welche Zeit bis zum nächsten HF-Puls vergeht. Die lokalen Relaxationszeiten (T_1 , T_2 und T_2^*) von Geweben im menschlichen Körper variieren stark. Somit ist es möglich, durch die Wahl der oben genannten Parameter den Bildkontrast zu beeinflussen. Ein kurzes TR kombiniert mit einem kurzen TE

erzeugt einen T_1 -gewichteten Kontrast, da sich in Geweben mit kurzen T_1 -Zeiten die Longitudinalmagnetisierung am schnellsten erholt. Dementsprechend erzeugt ein langes TR in Verbindung mit einem langen TE einen T_2 -gewichteten Kontrast. Abbildung 2.8 zeigt verschiedene Kontraste, welche mit unterschiedlichen Kombinationen von TE und TR erzielt werden können. Verschiedene Messtechniken, mit deren Hilfe man trotz schneller Dephasierung ein für die Bildgebung ausreichendes Signal erzeugen kann, werden im Abschnitt 2.1.5 betrachtet.

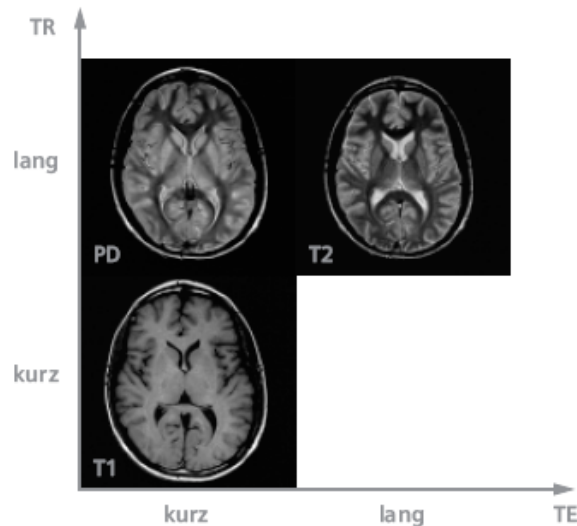


Abbildung 2.8.: Verschiedene Kontraste (Wichtungen) können durch die Variation von TE und TR erzeugt werden (Siemens [2003]) (mit freundlicher Genehmigung von Siemens Medical, Erlangen).

2.1.5. Messesequenzen

Um die komplexen Vorgänge bei einer Magnetresonanztomographiemessung zu steuern, verwendet man sogenannte *Messesequenzen*. Diese sind Programme, welche automatisiert Arbeitsschritte, wie z.B. das Schalten von Gradienten oder die Einstrahlung eines HF-Pulses, ausführen. Um den Ablauf solch einer Sequenz zu verdeutlichen, eignen sich Diagramme wie beispielhaft in Abbildung 2.9 dargestellt. In Abbildung 2.9 ist eine Sequenz dargestellt, die den Signalverlust durch das schnelle Dephasieren der Spins nach dem 90° -Puls kompensiert. Dies wird realisiert, indem ein refokussierender 180° Puls folgt, der die Spins zum Umkehren „zwingt“. Somit laufen langsamer und schneller präzedierende Spins wieder zusammen und es ergibt sich ein Signal, wenn sie sich wieder in kohärenter Phasenlage befinden. Dieses Signal wird als Spinecho bezeichnet.

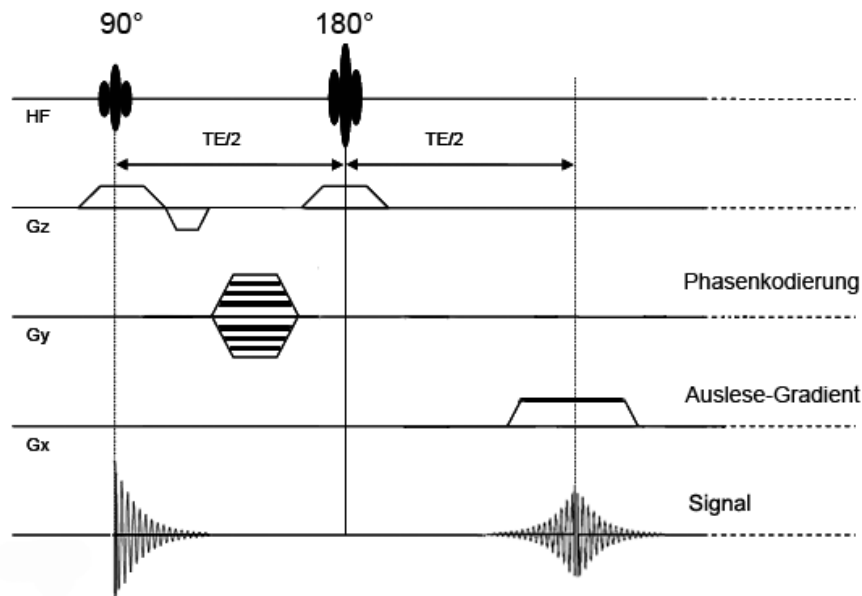


Abbildung 2.9.: Sequenzdiagramm einer Spin-Echo Sequenz (beginnend von oben): HF-Pulse (90° und 180°), Schichtselektionsgradient G_z , Frequenzkodiergradient G_y , Phasenkodiergradient G_x , entstehendes Signal

Die aktuelle Forschung beschäftigt sich intensiv damit, die Bildgebung am MRT artefaktärmer und schneller zu machen. Dazu wurden zahlreiche Sequenzen entwickelt, wie die im Folgenden beschriebene EPI-Sequenz¹, welche für die funktionellen Messungen verwendet wurde, die dieser Arbeit zugrunde liegen. Bei einer EPI-Sequenz werden nach der einmaligen Anregung alle Zeilen einer Schicht aufgenommen. Die Zeilen werden dabei durch kurze Gradienten („Blips“) in der Phase fourierkodiert (Haake et al. [1999]; Reiser u. Semmler [1997]). Die Transversalmagnetisierung wird dabei durch bipolare Gradienten (G_x) immer wieder refokussiert. Bei leistungsfähigen Gradientensystemen kann so je nach Auflösung ein Bild in unter 100 Millisekunden aufgenommen werden (Warmuth [2002]). Ein Nachteil dieser Sequenzart ist, dass die Zeilen des Bildes zu unterschiedlichen Zeiten nach dem HF-Puls aufgenommen werden. Hierdurch entstehen typische Verzerrungen in EPI-Bildern, vor allem an Grenzflächen mit stark unterschiedlichen magnetischen Eigenschaften (Suszeptibilitäten). Möglichkeiten zur Korrektur dieser Effekte finden sich z.B. in Jezzard u. Balaban [1995]. Des Weiteren stellt die EPI-Sequenz sehr hohe Anforderungen an das Gradientensystem eines Tomographen und durch die hohen Schaltgeschwindigkeiten ergibt sich eine starke Lärmbelastung.

¹EPI - Echo Planar Imaging

2.2. Funktionelle Magnetresonanztomographie

Das Hauptziel funktioneller Messungen in der Neurobildgebung ist die Darstellung von Hirnaktivität als Funktion von Ort und Zeit. Neben elektrophysiologischen Methoden wie Elektroenzephalografie (EEG) und Magnetenzephalografie (MEG) existieren Methoden, die sich auf metabolische Veränderungen bei Hirnaktivität stützen (Villringer [1999]). Zu den Letzteren zählt neben Positronen-Emissions-Tomographie (PET) auch die *Funktionelle Magnetresonanztomographie* (fMRT). Während elektrophysiologische Verfahren eine Zeitaufösung im Millisekundenbereich aufweisen, ist ihr räumliches Auflösungsvermögen stark begrenzt. Die fMRT hingegen bietet eine sehr gute räumliche Auflösung, die zeitliche Auflösung ist jedoch eingeschränkt, wie die weiteren Ausführungen zeigen werden. Das wichtigste Verfahren bei der fMRT stützt sich auf den so genannten *BOLD-Effekt* (Blood-Oxygen-Level-Dependent-Effekt), welcher erstmals von Ogawa im Jahr 1990 beschrieben wurde (Ogawa et al. [1990]). Wenn nachfolgend von funktioneller Magnetresonanztomographie gesprochen wird, wird daher immer BOLD-Bildgebung gemeint sein. Neben den physiologischen Grundlagen der BOLD-fMRT werden in diesem Abschnitt auch grundlegende Verfahren vorgestellt, mit denen signifikante Signalunterschiede in den Daten gefunden werden können bzw. komplexe Aktivierungsmuster analysiert werden können.

2.2.1. Physiologische Grundlagen

Um ihre Funktion aufrecht erhalten zu können, sind Nervenzellen auf die arterielle Versorgung mit Sauerstoff und Glukose angewiesen. Lokale neuronale Aktivität führt dabei zu lokal erhöhtem Sauerstoffverbrauch und mittelbar zu einer lokalen Erhöhung des Blutflusses (CBF - Cerebral Blood Flow) sowie zu einer lokalen Steigerung des Blutvolumens (CBV - Cerebral Blood Volume) (Huettel et al. [2004]; Iadecola et al. [1997]). Diesen Zusammenhang bezeichnet man als *Neurovaskuläre Kopplung*. Es wurde gezeigt, dass der lokal gesteigerte Sauerstoffmetabolismus durch die Erhöhung des CBF überkompensiert wird, so dass während neuronaler Aktivität ein erhöhter Oxygenierungsgrad des Blutes in den aktivierten Regionen festgestellt werden kann. Dieser Mechanismus ist Grundlage des *BOLD-Effektes* (Ogawa et al. [1990]), welcher für die Messung von Hirnaktivität in der BOLD-fMRT verwendet wird.

Da der für die beschriebenen Vorgänge nötige Sauerstoff über Hämoglobin transportiert wird, führt neuronale Aktivität also zu einer lokalen Verschiebung des Anteils von Hämoglobin und Deoxyhämoglobin in aktivierten Regionen (vgl. Abbildung 2.10). Im Gegensatz zu Oxyhämoglobin ist Deoxyhämoglobin paramagnetisch und bewirkt durch lokale

Magnetfeldänderungen eine Verkürzung der Transversalrelaxationszeit T_2^* und somit eine Signalabnahme bei T_2^* -gewichteten Pulssequenzen (Baert et al. [1999]; Dössel [2000]). In aktivierten Hirnarealen zeigt sich die relative Abnahme deoxygenierten Hämoglobins durch einen Signalanstieg, wie im oberen Teil von Abbildung 2.10 verdeutlicht wird. Dieser Signalanstieg macht nur wenige Prozent des Gesamtsignals aus. Abbildung 2.11 zeigt

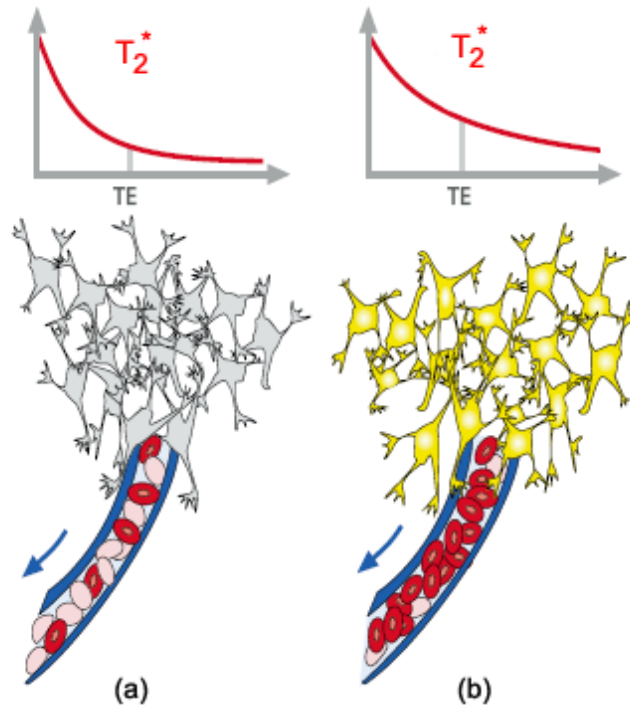


Abbildung 2.10.: Dargestellt ist die lokale Sauerstoffanreicherung im Blut bei Neuronenverbänden in Ruhe (a) und im aktiven Zustand (b) (Siemens [2003]). Die lokale Abnahme deoxygenierten Hämoglobins bewirkt einen Signalanstieg in T_2^* -gewichteten Pulssequenzen (mit freundlicher Genehmigung von Siemens Medical, Erlangen).

beispielhaft den Verlauf der hämodynamischen Antwort als Signalanstieg in fMRT Aufnahmen während neuronaler Aktivierung in Form der HRF^2 . Wie man erkennen kann, zeigt die HRF in den ersten zwei Sekunden nach einem Stimulus einen Abfall unter das Grundniveau (Baseline). Dieser *Initial Dip* wird auf die erhöhte Sauerstoffextraktion im Kapillarbett zurückgeführt (Huettel et al. [2004]). Durch die angesprochene Überkompensation des Sauerstoffverbrauches kommt es im weiteren Verlauf der HRF zu einem Signalanstieg. Die Zeit zwischen Stimulusbeginn und dem Maximum der BOLD-Antwort beträgt je nach Physiologie zwischen vier und sechs Sekunden, wodurch die Zeitauflösung der fMRT begrenzt wird. Nachdem das Signal sein Maximum erreicht hat, fällt es wieder

²HRF - Hemodynamic Response Function

unter das Ausgangsniveau ab und es kommt zu einem sogenannten *Undershoot*. Die physiologischen Hintergründe der BOLD-Antwort sind noch immer Gegenstand intensiver Forschung. Es wurde aber bereits gezeigt, dass es starke Variationen der BOLD-Antwort zwischen verschiedenen Individuen aber auch zwischen verschiedenen Hirnregionen gibt (Aguirre et al. [1998]; Handwerker et al. [2004]).

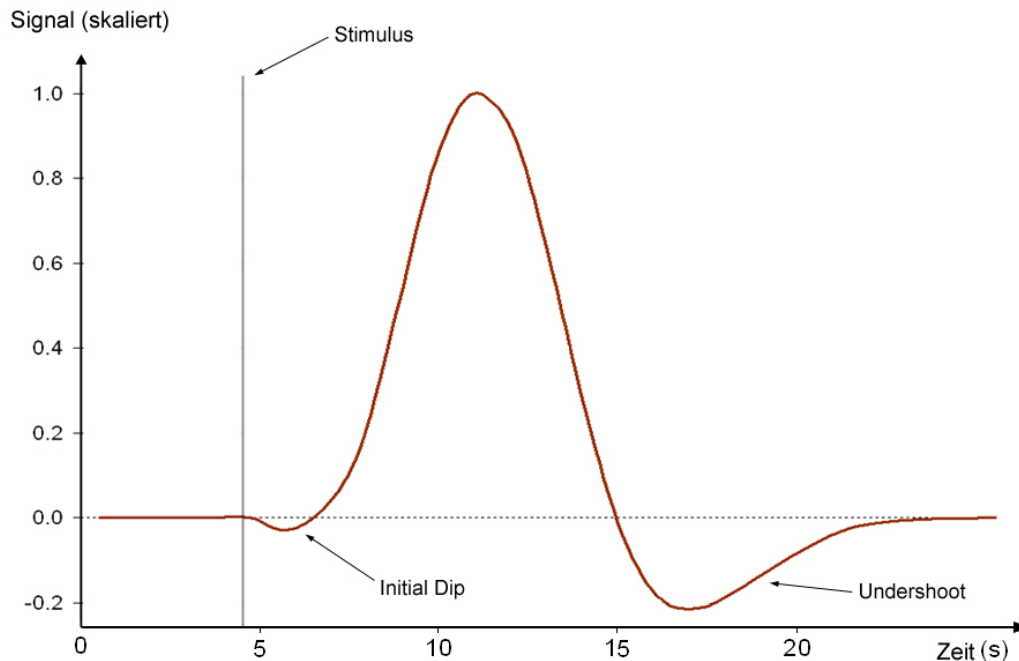


Abbildung 2.11.: Beispielhafter Verlauf des Signals aufgrund einer hämodynamischen Antwort (HRF): Mit dem Anstieg von neuronaler Aktivität im Hirn aufgrund eines Stimulus kommt es zum vermehrten Verbrauch von Sauerstoff. Als Folge steigt kurzzeitig der Anteil von Deoxyhämoglobin in aktivierten Arealen, was sich im *Initial Dip* ausdrückt. Im weiteren Verlauf kommt es zu einem erhöhten Signal und einem Abfall bis unter das Ausgangsniveau (*Undershoot*).

2.2.2. Modellannahmen der HRF

Eine weit verbreitete Methode zur Modellierung der HRF geht auf Friston et al. zurück (Friston et al. [1994]). In diesem Ansatz wird die hämodynamische Reaktion als lineares System betrachtet und der Signalverlauf mit Hilfe einer Stimulusfunktion modelliert. Die finale Antwortfunktion ergibt sich aus der Summe der sich überlappenden Antworten auf zeitlich nahe beieinander liegender Stimuli. Demnach kann man die erwartete hämodynamische Antwort auf einen Stimulus aus einer Faltung der Stimulusfunktion und einer angenommenen HRF berechnen. Die erwartete hämodynamische Antwort kann nun wie-

derum als Hypothese in statistische Verfahren einfließen. Unter Annahme eines linearen Systems verfolgten Boynton et al. den Ansatz, die HRF als Gamma-Variate Funktion zu modellieren (Boynton et al. [1996]). Dieses Modell ist ein weit verbreiteter Ansatz in der BOLD-fMRT.

Ein nicht-linearer Ansatz zur Modellierung der HRF ist das sogenannte *Ballon-Modell* (Buxton et al. [1998]; Friston et al. [2000]). In diesem Input/Output-Modell bildet der Blutfluss (CBF) den Input und das zu erwartende BOLD-Signal den Output. Das Signal wird in einen intra- und einen extravaskulären Anteil unterteilt und mit den Variablen Blutvolumen (CBV) und Deoxyhämoglobingehalt moduliert. Eine Studie von Stephan et al. hat gezeigt, dass das Ballon-Modell einen zuverlässigen Prädiktor für die HRF darstellt (Stephan et al. [2007]).

2.2.3. Aufnahmeverfahren und Design von fMRT-Experimenten

Wie bereits ausgeführt wurde, ist das Ziel funktioneller Hirnbildgebung in einem Kernspintomographen die Lokalisierung der Hirnaktivität. Die spezifische Aktivierung wird durch Stimuli erreicht, welche motorischer, sensomotorischer, visueller, auditiver, emotionaler oder anderer kognitiver Art sein können. Da der Signalunterschied in aktivierten Arealen nur wenige Prozent ausmacht, hat sich eine strukturierte Reizpräsentation und eine rechnergestützte statistische Bearbeitung von fMRT Daten durchgesetzt. An dieser Stelle werden kurz die möglichen Verfahren der Datenakquisition und des Experimentdesigns vorgestellt, wobei die Verfahren der Datenverarbeitung im nächsten Abschnitt beschrieben werden.

Die Aktivierung von Hirnarealen ist eine Funktion der Zeit. Somit ergeben sich bei einer funktionellen MRT-Messung Zeitreihen dreidimensionaler Daten. Diese Serien können mit verschiedenen Pulssequenzen aufgenommen werden. Hier wird jedoch nur die bereits beschriebene EPI Sequenz einbezogen, da diese auch für die praktische Umsetzung der durchgeführten Experimente verwendet wurde. Mit dieser Sequenz ist man in der Lage, das gesamte Gehirn in einer Repetitionszeit von weniger als 2 Sekunden aufzunehmen. Abhängig von der verwendeten Feldstärke werden im Allgemeinen Auflösungen von 32×32 bis 256×256 Voxeln pro Schicht verwendet, wobei die Schichtdicke meist auf circa ein bis fünf Millimeter festgelegt wird. Da das messbare Signal abhängig von der Auflösung abnimmt, werden größere Bildmatrizen tendenziell bei höheren B_0 Feldern verwendet. Als Richtwert für die Echozeit wird bei drei Tesla Feldstärke ein Wert von ca. 25 bis 40 Millisekunden empfohlen (siehe Kennan [1999] bzw. Clare [1997]). Es gibt auch Verfahren mit kurzen Repetitionszeiten, welche benutzt werden, um die HRF „abzutasten“. Somit ist es z.B. möglich, Aussagen über die zeitliche Entwicklung der

hämodynamischen Antwort zu erhalten. Die Speicherung funktioneller MRT-Datensätze erfolgt meist in standardisierten Formaten wie DICOM³ (NEMA [2008]), ANALYZE (Mayo-Research [2008]) oder als Rohdaten.

Da das Signal eines Voxels zu einem bestimmten Zeitpunkt nicht qualitativ beurteilt werden kann, ist man auf den Vergleich von Bedingungen während eines Experimentes angewiesen. Dabei wird versucht, signifikante Änderungen der neuronalen Aktivierung unter Beachtung der zeitlichen Verzögerung der HRF hervorzurufen. Es haben sich zwei Grundkonzepte des experimentellen Designs durchgesetzt, wobei auch Mischformen dieser Konzepte existieren (Aguirre u. D’Esposito [1999]):

1. Block-Design
2. Ereigniskorreliertes Design

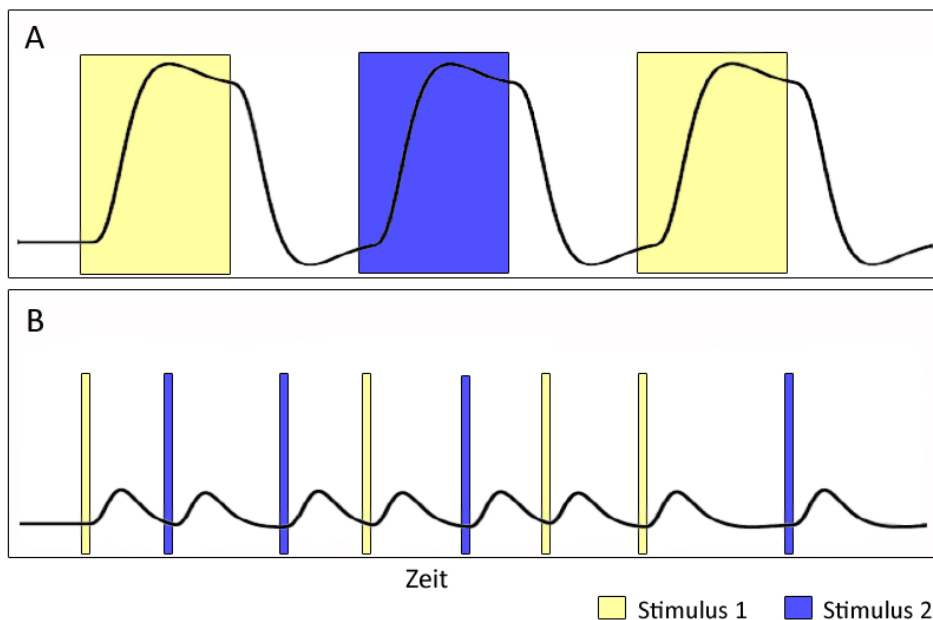


Abbildung 2.12.: Beispielhaft sind die erwarteten zeitlichen Signalverläufe für ein *Block-Design* (A) und ein *Ereigniskorreliertes Design* (B) dargestellt. Das Signal ergibt sich dabei aus der Faltung des Designs (Stimulus 1 und 2) mit der HRF.

Abbildung 2.12 stellt die erwarteten Signalverläufe für diese Designtypen dar. Je nach Signal-Rausch-Verhältnis werden mehrere Mittelungen eines Aktivierungsverlaufs benutzt um eine signifikante Aktivierung zu detektieren. Bei einem klassischen Blockdesign wird dabei ein fester Ablauf von Ruhe- und Stimulationsperioden implementiert.

³DICOM - Digital Imaging and Communications in Medicine

Das Ergebnis ist eine repetitive Ereignissequenz.

Bei dem zweiten angesprochenen Konzept, dem ereigniskorrelierten Design, werden randomisiert einzelne kurze Stimuli präsentiert und die akquirierten Signale mit der prädiktierten hämodynamischen Antwort verglichen. Es handelt sich also um eine variable Ereignissequenz. Dieses Verfahren vermeidet Fehler (z.B. durch Adaption), welche durch feste Frequenzen bei der Stimulation entstehen (Aguirre u. D'Esposito [1999]). Des Weiteren ist es mit diesem Verfahren möglich, die Zeitauflösung der Messung über die Repetitionszeit hinaus zu steigern, indem man die Stimuluspräsentation so wählt, dass zu verschiedenen Stimulationszeitpunkten verschiedene Zeitpunkte der HRF aufgenommen werden. In beiden Designtypen kann es Stimuli verschiedener Arten geben, um auch Kontraste zwischen diesen erzeugen zu können.

Für eine Echtzeitauswertung funktioneller Daten können die beiden beschriebenen Designtypen verwendet werden, jedoch sind die speziellen Gegebenheiten der sofortigen Auswertung zu beachten. Näheres zu den speziellen Anforderungen der Echtzeit-fMRT findet sich im Abschnitt 3. Die Vorverarbeitungsschritte und die statistischen Verfahren zur Auswertung der akquirierten Daten werden in den folgenden Abschnitten dargestellt.

2.2.4. Vorverarbeitung funktioneller MRT-Daten

Es wurde bereits angesprochen, dass der BOLD-Signalunterschied in aktivierten bzw. nicht aktivierten Gebieten nur wenige Prozent des Gesamtsignals ausmacht. Daraus folgt ein sehr geringes Signal-Rausch-Verhältnis in fMRT-Daten. Abbildung 2.13 zeigt beispielhaft den unbearbeiteten Signalverlauf in einem einzelnen Voxel des motorischen Kortex während eines einfachen Finger-Tapping⁴ Experimentes. Wie man in der Abbildung erkennen kann, unterliegt das Signal hoch- und niederfrequenten Störungen sowie linearen Drifts. Ursachen solcher Störungen liegen in der Messtechnik aber z.B. auch in Probandenbewegungen und physiologischem Rauschen. Es wurden einige Methoden entwickelt, welche schon vor der statistischen Auswertung den Problemen entgegenwirken und Probandendaten vergleichbar machen. Die Wichtigsten davon sind:

- Bewegungskorrektur
- Räumliche Normalisierung
- Lineares Detrending
- Räumliche Filterung

⁴Unter Finger-Tapping versteht man die gezielte Bewegung der Finger auf ein Signal hin.

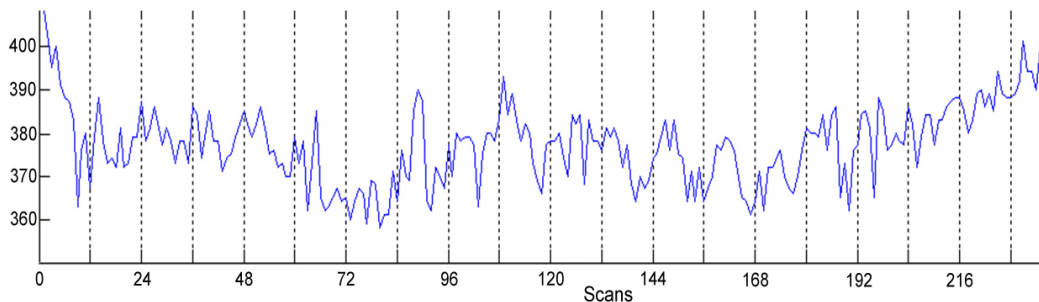


Abbildung 2.13.: Signalverlauf in einem einzelnen Voxel des motorischen Kortex während eines Finger-Tapping Experimentes. Die kurze Fingerbewegung erfolgte dabei in einem festen Abstand von 12 Scans (beginnend bei Scan 12). Neben linearen Drifts sind auch hoch- und niederfrequente Schwankungen dem stimulusinduzierten Signal überlagert. Die Repetitionszeit (TR) beträgt in diesem Fall 2 Sekunden.

- Zeitliche Filterung
- Slice Time Correction

Obige Prozeduren können in die statistische Verarbeitung integriert werden. Des Weiteren können Verfahren wie z.B. die Segmentierung von Hirngewebe hinzukommen (vgl. hierzu Abschnitt 2.2.5).

Bewegungskorrektur und Normalisierung

Unter Bewegungskorrektur versteht man die rechnerische Korrektur von Kopfbewegungen, die der Proband während der funktionellen Messung trotz Fixierung vollführt hat. Eine räumliche Normalisierung transformiert die Daten in einen vordefinierten Raum, um z.B. die Lokalisierung von aktivierten Arealen mit Daten von anderen Probanden vergleichen zu können oder die Verwendung anatomischer Atlanten zu ermöglichen. Solch einen Standard-Raum stellt z.B. der Talairach-Raum dar, der mit Hilfe von markanten anatomischen Landmarken eine Normalisierung zulässt (Talairach u. Tournoux [1988]).

Bewegungskorrektur sowie räumliche Normalisierung führen zu dem gleichen mathematischen Problem. Es müssen Datensätze aufeinander registriert werden, d.h. Bildräume ineinander überführt werden. Der Unterschied liegt darin, dass bei der Bewegungskorrektur nur globale Rotationen und Translationen berücksichtigt werden müssen, wohingegen bei der räumlichen Normalisierung auch Skalierungen, Scherungen und nichtlineare Transformationen eine Rolle spielen (Ashburner u. Friston [1999]). Eine affine Transformation wie in der unten stehenden Matrix beschrieben, kann die linearen Bestandteile

beider Probleme lösen, wobei für die Bewegungskorrektur nur 6 Parameter des linearen Gleichungssystems berechnet werden müssen:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ 1 \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ 1 \end{pmatrix} \quad (2.12)$$

Um das Gleichungssystem zu lösen, braucht man entweder eindeutige Lösungen durch anatomische Landmarken oder ein Maß, welches die Güte der Registrierung abschätzt. Meist wird hierfür die Summe der quadrierten Differenzen der Signalwerte der Bildpunkte benutzt. Dieses Verfahren ist natürlich nur sinnvoll, wenn die zu registrierenden Bilder derselben Modalität entstammen. Bei der Inter-Modalitäten-Registrierung kann man zum Beispiel die korrelierte Information (Mutual Information) in den Histogrammen der beiden Bilder als Maß benutzen. In beiden Fällen ist ein Optimierungsproblem zu lösen, um die quadrierten Differenzen zu minimieren bzw. die korrelierte Information zu maximieren. Für nichtlineare Transformationen, wie lokale Verzerrungen, ist solch ein lineares Gleichungssystem nicht ausreichend, in diesem Fall werden lokale Transformationen z.B. mit Hilfe von elastischen Modellen realisiert (Ashburner u. Friston [1999]; Dössel [2000]).

Detrending, Räumliche und zeitliche Filterung

Funktionelle MRT-Daten enthalten aufgrund technischer Gegebenheiten oftmals einen kontinuierlichen globalen Drift in der Signalstärke als Funktion der Zeit. Diesen linearen Drift kann man durch das Anpassen einer linearen Funktion korrigieren, indem man diese errechnete Funktion vom Signal abzieht. Dieses Vorgehen nennt man *lineares Detrending*.

Für die räumliche Filterung (Glättung) werden sogenannte Filteroperatoren verwendet, um das Signal-Rausch-Verhältnis zu verbessern (Clare [1997]). Unter Filteroperator versteht man dabei eine Abbildung Φ , die einem Voxel v_0 einen neuen Wert zuordnet, welcher aus n Voxeln der Umgebung errechnet wurde:

$$\Phi : v_0 \rightarrow \Phi(v_0, v_1 \dots v_n). \quad (2.13)$$

Für die räumliche Filterung kann z.B. ein Gauss-Filter benutzt werden, wobei man beachten muss, dass eine räumliche Glättung mit einer Verringerung der räumlichen Auflösung einhergeht (Friston [2004]). Die zeitliche Filterung kann im Fourierraum ausgeführt wer-

den. Dabei werden hohe oder tiefe Frequenzen aus dem Spektrum entfernt, wenn sie nicht im Frequenzbereich der Stimuluspräsentation liegen. Man spricht dabei von Tiefpass- bzw. Hochpassfiltern. Tiefpassfilter dienen dabei der Verringerung von hochfrequentem Rauschen und Hochpassfilter filtern tieffrequente Anteile des Signals, welche z.B. durch respiratorische Effekte verursacht werden. Näheres hierzu findet sich in weiterführender Literatur wie z.B. Hu et al. [1995].

Slice Time Correction

Bei EPI-Aufnahmeverfahren, welche oft für die funktionelle MR-Bildgebung verwendet werden, wird das Signal zur Darstellung verschiedener Schichten zu getrennten Zeitpunkten innerhalb der Repetitionszeit aufgenommen. Die Schichten werden dabei entweder nacheinander (aufsteigend oder absteigend) oder abwechselnd (interleaved) aufgenommen, so dass zwischen der Aufnahme verschiedener Schichten im Extremfall mehrere Sekunden liegen können. Dieser Effekt spielt insbesondere bei den erwähnten ereigniskorrelierten Designs und bei Verwendung von langen Repetitionszeiten eine Rolle. Bei der *slice time correction* wird eine Interpolation der Intensitätswerte aus den einzelnen Schichten auf einen definierten Zeitpunkt innerhalb der Repetitionszeit vorgenommen, um dem beschriebenen Effekt entgegenzuwirken.

2.2.5. Auswertung funktioneller MRT-Daten: Univariate Verfahren und Klassifikation

In diesem Abschnitt werden Verfahren zur Auswertung von fMRT Daten vorgestellt. Dabei werden einfache statistische Verfahren wie der t-test und die Korrelationsanalyse ebenso betrachtet wie komplexere Methoden wie das *General Linear Model* (GLM) oder multivariate Ansätze. Unter letzterem Punkt werden insbesondere Klassifikationsverfahren wie *Support-Vector-Machines* (SVM) und *Relevance-Vector-Machines* (RVM) dargestellt, da diese im praktischen Teil der Arbeit in Kapitel 7 Anwendung finden.

t-Test und Korrelationsanalyse

Für die fMRT wurden zahlreiche statistische Verfahren zur Auswertung entwickelt (Lange [1999]). Zu den einfachsten zählen *t-Tests* wie der Welch-Test. Dabei wird gegen die Null-Hypothese getestet, dass der Zeitverlauf eines Voxels nicht durch rein zufällige Signalschwankungen erzeugt ist. Bei den t-Tests wird grundsätzlich die Differenz zwischen gemittelten Signalwerten aufgenommen während zweier verschiedener Blocktypen, also zweier Verteilungen (z.B. aktiv, inaktiv), mit der Standardabweichung gewichtet. Man

erhält für jedes Voxel einen t-Wert, der sich beim Welch-Test wie folgt berechnet:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \quad (2.14)$$

mit dem Freiheitsgrad:

$$f_W = \frac{((S_1^2/n_1) + (S_2^2/n_2))^2}{\frac{(S_1^2/n_1)^2}{n_1-1} + \frac{(S_2^2/n_2)^2}{n_2-1}} \quad (2.15)$$

Hier sind mit den Indizes 1 bzw. 2 die Werte verschiedener Blocktypen bezeichnet, somit sind \bar{X}_1 und \bar{X}_2 die Mittelwerte von n_1 bzw. n_2 verschiedenen Werten in der Zeitreihe. Dementsprechend stellen S_1 bzw. S_2 die Varianz in beiden Zeitreihen dar. Das Quantil lässt sich dann mit Hilfe dieses t-Wertes ermitteln und liefert die Aussage, ob sich die Mittelwerte beider Reihen signifikant unterscheiden. Dieser Test ist eingeschränkt bezüglich der Modellierung weiterer experimenteller Einflussgrößen.

Mehr Informationen über hämodynamische Prozesse kann man mit einem weiteren weitverbreiteten Verfahren, der *Korrelationsanalyse*, integrieren. Hierbei wird der Zeitverlauf des Signals mit einer Referenzfunktion verglichen (Huettel et al. [2004]). Diese Referenzfunktion modelliert die erwartete hämodynamische Antwort und kann experimentell ermittelt oder durch eine definierte Funktion gegeben sein (vgl. Abschnitt 2.2.2). Abbildung

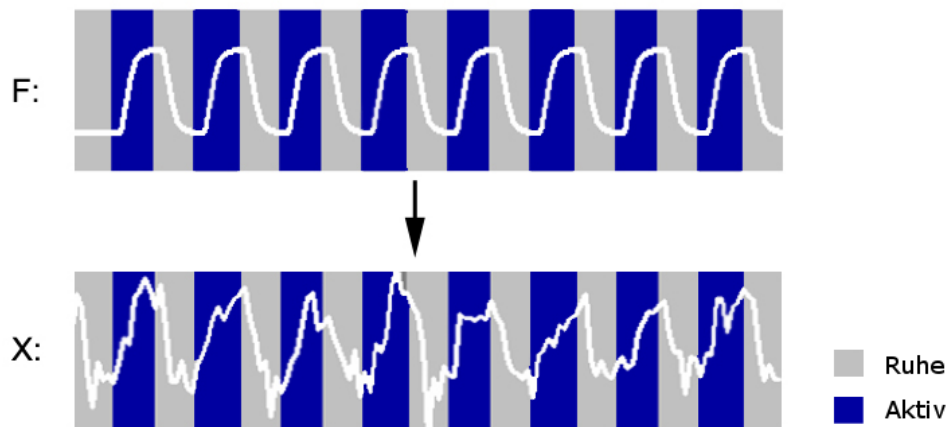


Abbildung 2.14.: Darstellung einer Funktion (F), welche für die Korrelationsanalyse einer Zeitreihe (X) benutzt werden kann. Graue Bereiche stellen zeitliche Abschnitte der Ruhebedingung und blaue Bereiche Abschnitte der Aktivbedingung dar.

2.14 zeigt eine modellierte Funktion (F), welche auf die Korrelation mit einer Zeitreihe (X) getestet wird. Der Korrelationskoeffizient r ergibt sich für die Zeitreihe X und der

Referenzfunktion F wie folgt:

$$r = \frac{\sum (X_i - \bar{X}) \cdot (F_i - \bar{F})}{\sqrt{\sum (X_i - \bar{X})^2 \cdot \sum (F_i - \bar{F})^2}}. \quad (2.16)$$

Für r ergibt sich ein Wert von +1 bei exakter Korrelation und -1 für exakte Antikorrelation. Wie man leicht erkennt, ist für dieses Verfahren die Wahl der richtigen Referenzfunktion essenziell.

General Linear Model

Das aufgrund seiner Zuverlässigkeit und Flexibilität am weitesten verbreitete Verfahren in der fMRT ist das *General Linear Model*. Mit dieser univariaten Methode wird versucht, Variationen in einer Zeitreihe Y als Linearkombination des mit β gewichteten Versuchsdesigns X und eines Fehlerterms ϵ darzustellen (Friston et al. [1995]):

$$Y = X\beta + \epsilon. \quad (2.17)$$

Zu einem Zeitpunkt i ergibt sich also das zu modellierende Signal wie folgt:

$$Y_i = \sum_j X_{ij} \cdot \beta_j + \epsilon_i. \quad (2.18)$$

Hierbei stellt Y den Vektor des Zeitverlaufs eines Voxels und X die Designmatrix mit den Regressoren dar. Ein Regressor steht dabei für einen Teilaspekt des Gesamtparadigmas, also z.B. für einen hypothetischen Signalverlauf für einen Stimulustyp. Der Parametervektor β kann nun mit Hilfe eines Schätzers $\hat{\beta}$ ermittelt werden, wenn der Fehlervektor ϵ als Gauss-verteilte Störgröße angenommen wird:

$$\hat{\beta} = (X^T X)^{-1} X^T Y. \quad (2.19)$$

Wenn man die Kombination der Parameter β mit der Designmatrix nun auf signifikante Unterschiede zwischen den Zeitpunkten untersucht, kann man Aktivierungen bzw. auch Differenzkontraste verschiedener Regressoren ableiten (Lange [1999]). Es ist hierbei zu beachten, dass man beim GLM die Schätzer $\hat{\beta}$ für jedes Voxel einzeln berechnet. Somit wird bei der Signifikanzprüfung eine Nullhypothese bei sehr vielen Voxeln (Multiple Tests) getestet und die Wahrscheinlichkeit falsch positiver Testergebnisse steigt. Dieses Problem kann man unter der Annahme der Unabhängigkeit der zu testenden Stichproben mit einer Bonferroni-Korrektur (also die Festlegung eines neuen Signifikanzlevels unter

Einbeziehung der Anzahl der getesteten Voxel) beheben. In fMRT-Daten findet man jedoch räumliche Abhängigkeiten womit sich diese Korrektur als zu konservativ herausgestellt hat. Man benutzt deshalb weitergehende Korrekturverfahren wie FDR (False Discovery Rate) oder FWE (Family Wise Error) um das Signifikanzlevel anzupassen (Friston [2004]; Lange [1999]).

Mit Hilfe des GLMs ist es möglich, neben verschiedenen Untersuchungs- und Kontrollbedingungen auch Respirationseffekte oder Bewegungsparameter zu modellieren. Allerdings ist die Lösung des linearen Gleichungssystems mit einigem Rechenaufwand verbunden und die Wahl der Regressoren des linearen Systems hat großen Einfluss auf das Ergebnis (Frackowiak et al. [2004]). Das General-Linear-Model wurde für die fMRT mit dem Softwarepaket *SPM* eingeführt, welches für die Auswertung von fMRT-Studien häufig verwendet wird (SPM [2008]). Weitere Applikation zur Auswertung von fMRT-Daten sind z.B. *Brain-Voyager* (BrainVoyager [2009]), *AFNI* (Cox [1996]) oder *FSL* (Smith et al. [2004]). Neben diesen Softwarepaketen ist gerade auch das kommerzielle Produkt zur Echtzeit-Auswertung von fMRT-Daten *Turbo-Brain-Voyager* für diese Arbeit interessant (BrainVoyager [2009]). Außer den hier skizzierten Strategien existieren noch weitere statistische Methoden zur Auswertung von fMRT-Daten, wie z.B. die Fourieranalyse für periodische Paradigmen, die Principal Component Analysis (PCA) und die Independent Component Analysis (ICA). Für nähere Informationen hierzu wird auf Huettel et al. [2004] verwiesen.

Multivariate Verfahren: Musterbasierte Klassifikationsverfahren

Standardisierte multivariate Verfahren wie die PCA und die ICA sind in der funktionellen Neurobildgebung weit verbreitet. Im Gegensatz zu univariaten Verfahren wie dem GLM, welches ein hypothesengetriebenes Verfahren ist, sind diese datengetrieben. Es ergeben sich bei beiden genannten Gruppen von Verfahren Limitationen bezüglich der Aussagekraft von erzielten Ergebnissen. Zum Beispiel werden beim GLM einzelne Voxel als unabhängige Stichproben betrachtet (vgl. Abschnitt 2.2.5), somit ist die Analyse von Netzwerken von Hirnregionen erschwert. Günstigerweise ist bei diesen Modellen eine enge Verbindung von Ergebnissen und experimentellen Variablen über die Design-Matrix gegeben. Bei den genannten multivariaten Ansätzen hingegen existiert nur eine schwache Verbindung zu experimentellen Variablen, da sie ohne Hypothesen die Varianz in Signalen analysieren (Friston [1998]).

In der funktionellen Neurobildgebung zeichnet sich eine Entwicklung ab, welche sich der Frage der 'Art der Repräsentation' von Hirnfunktionen widmet, anstatt wie bisher allein die Lokalisation zu betrachten (Haynes u. Rees [2006]; O'Toole et al. [2007]). Unter

der 'Art der Repräsentation' fällt z.B. die Frage, in welchen räumlich-zeitlichen Mustern das Hirn Informationen speichert. Eine Methodik, die in anderen Wissenschaftszweigen schon lange für die Analyse von Mustern benutzt wird, sind die sogenannten musterbasierte Klassifikationsverfahren (Duda et al. [2001]). Diese Verfahren sind auch in der Lage, Aussagen über räumlich-zeitliche Muster in funktionellen Hirnbilddaten zu liefern, wobei gleichzeitig eine enge Verbindung zu experimentellen Variablen hergestellt werden kann.

Bei musterbasierten Klassifikationsverfahren in der funktionellen Neurobildgebung lässt sich im Allgemeinen folgendes Vorgehen abstrahieren: Ein Algorithmus 'lernt' anhand von Trainingsdaten X und den gegebenen experimentellen Variablen Y räumliche und / oder zeitliche Muster in diesen Trainingsdaten bezüglich der Experimentkonditionen zu unterscheiden. Dies wird im Folgenden als Training bezeichnet. Anschließend erfolgt ein Test (Klassifikation), wobei unabhängigen Testdaten T , welche nicht Bestandteil der Trainingsmenge sind, jeweils Variablen (Klassen) aus Y zugeordnet werden. Neben der Güte der Klassifikation (prozentuale Anzahl an richtig klassifizierten Stichproben) kann man je nach verwendetem Algorithmus Informationen über die Muster gewinnen, welche der Klassifikation zugrunde lagen. Um Muster in den Daten zu finden, werden Trainingsdaten sowie Testdaten in einen Vektorraum transformiert, den sogenannten Merkmalsraum, wobei ein einzelnes Element der Trainings- bzw. Testdaten dann als 'feature' bezeichnet wird.

Um die Generalisierungsstärke eines Klassifikators zu testen, eignet sich ein Verfahren namens *Leave one out Cross Validation* (LOOCV). Das Vorgehen bei einer LOOCV besteht darin, aus einer Grundgesamtheit von Merkmalen eines zu entfernen, den Klassifikator mit den verbliebenen Merkmalen zu trainieren und anschließend den Test auf dem entfernten Merkmal anzuwenden. Dieses Vorgehen wird für jedes einzelne Merkmal wiederholt und die erzielte Klassifikationsgüte gibt Auskunft über die Leistungsfähigkeit des Klassifikators.

Oftmals werden in einem Vorverarbeitungsschritt Merkmale entfernt, welche keine oder nur wenig Varianz in Bezug auf die experimentellen Variablen erklären. Diesen Schritt nennt man *feature-selection*. Im Folgenden soll exemplarisch die Klassifikation mit Support Vector Machines (SVM) näher vorgestellt werden. Neben einer SVM wurde in den praktischen Experimenten dieser Arbeit auch eine Relevance Vector Machine (RVM) eingesetzt. Die RVM stellt eine probabilistische Erweiterung der SVM dar und wird nur kurz vorgestellt, da die Grundzüge von RVM und SVM sich stark ähneln.

Support Vector Machines Die Funktionsweise einer SVM beruht darauf, verschiedene Merkmale (features) im Merkmalsraum voneinander zu trennen, wobei features hierbei durch Vektoren repräsentiert sind. Für ein Klassifizierungsproblem reeller Zahlen mit genau 2 Klassen ergibt sich folgende Trainingsmenge X :

$$X = \{(x_i, y_i) | x_i \in \mathbf{R}^P, y_i \in \{1, -1\}\}. \quad (2.20)$$

Hierbei ist x_i ein p -dimensionaler Vektor und y_i gibt an, zu welcher Klasse (1 oder -1) x_i gehört. Beim Training der SVM wird nun eine $p-1$ dimensionale Hyperebene gesucht, welche die features der beiden Klassen mit größtmöglichem Abstand (Margin) zu dieser Ebene trennt. Abbildung 2.15 verdeutlicht beispielhaft ein zweidimensionales Klassifikationsproblem. Die Ebene H ist dabei über den Normalenvektor w und die Verschiebung

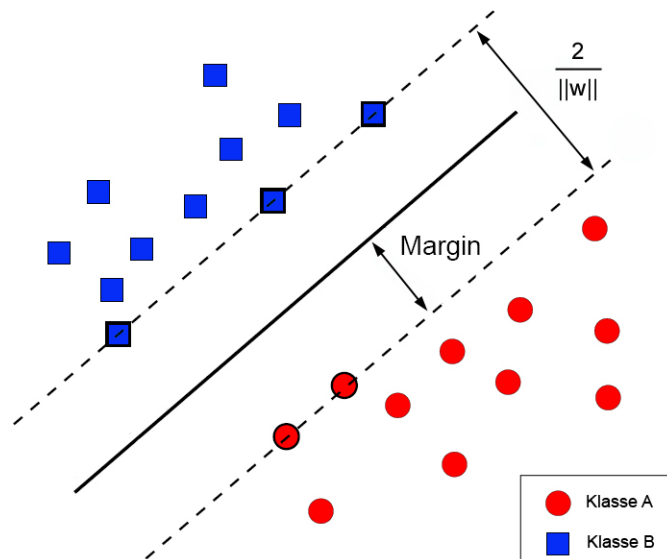


Abbildung 2.15.: Dargestellt ist ein Klassifikationsproblem mit zwei Klassen (A und B). Die Merkmale sind in diesem Fall Vektoren mit zwei Elementen, spannen also einen zweidimensionalen Merkmalsraum auf. Die trennende Hyperebene ist als durchgezogene Linie dargestellt. Die hervorgehobenen Objekte auf der Grenze der Margin bezeichnet man als 'Supportvektoren'. Der Abstand der Supportvektoren zur Hyperebene ergibt sich aus Gleichung 2.24.

(Bias) b definiert:

$$H = \{x | \langle w, x_i \rangle + b = 0\}. \quad (2.21)$$

Die Parameter w und b werden dabei skaliert, so dass für Merkmale auf den Grenzen der Margin (Supportvektoren) gilt:

$$\langle w, x_i \rangle + b = +1 \quad \text{bzw.} \quad \langle w, x_j \rangle + b = -1. \quad (2.22)$$

Die Entscheidungsfunktion ordnet dann jedem Element aus X eine Klasse aus Y zu:

$$y_i = \text{sgn}(\langle w, x_i \rangle + b). \quad (2.23)$$

Für zwei Supportvektoren x_l und x_r links und rechts von H ergibt sich der Abstand (und damit die Breite der Margin) über:

$$\left\langle \frac{w}{\|w\|}, (x_l - x_r) \right\rangle = \frac{2}{\|w\|}. \quad (2.24)$$

Der Ausdruck $\|w\|$ stellt dabei die Norm des Vektors w dar. Für die Maximierung der Margin folgt somit das Optimierungsproblem:

Minimiere für alle $1 < i < n$ bezüglich w, b :

$$\frac{1}{2} \|w\|^2 \quad (2.25)$$

unter der Nebenbedingung:

$$y_i \cdot (\langle w, x_i \rangle + b) \geq 1. \quad (2.26)$$

Diese Nebenbedingung stellt sicher, dass sich kein Merkmal innerhalb der Margin befindet. Für nichtlinear separierbare Daten erhöht man die Generalisierungsstärke des Klassifikators mit sogenannten 'Soft-Margins'. Es werden also Verletzungen der Nebenbedingung zugelassen, welche über die Schlupfvariable ξ_i und einen konstanten positiven Wichtungsfaktor C ausgedrückt wird. Das Optimierungsproblem stellt sich nun wie folgt dar:

Minimiere für alle $1 < i < n$ bezüglich w, b :

$$\frac{1}{2} \|w\|^2 + C \sum_i \xi_i \quad (2.27)$$

unter der Nebenbedingung:

$$y_i \cdot (\langle w, x_i \rangle + b) \geq 1 - \xi_i. \quad (2.28)$$

Dieses Optimierungsproblem lässt sich mit Hilfe einer Lagrange-Funktion $L(w, b, \alpha)$ lösen, wobei α einen Lagrange-Multiplikator darstellt. Weitere Informationen hierzu finden sich in Schölkopf u. Smola [2002]. Als Support-Vektoren bezeichnet man nun alle Vektoren, welche genau auf der Margin oder auf der 'falschen Seite' der Margin liegen und damit deren Lage bestimmen. Der Rest der Merkmale kann somit ignoriert werden.

Bisher wurde nur der lineare Ansatz im Rahmen der SVM beschrieben. Es ist jedoch möglich, dieses Verfahren auf nicht-linear trennbare Probleme zu erweitern, bei denen die Soft-Margin Methode nicht genügt, wie in Abbildung 2.16 dargestellt. Dazu kann man den Merkmalsraum in einen Raum höherer Dimension transformieren, in welchem sich wiederum eine lineare Trennung der Klassen vornehmen lässt:

$$\Phi : \mathbf{R}^{d_1} \rightarrow \mathbf{R}^{d_2}. \quad (2.29)$$

Hierbei gilt $d_1 < d_2$. Damit ist es möglich ein Skalarprodukt $\langle x_i, x_j \rangle$ in \mathbf{R}^{d_1} als Skalarprodukt in \mathbf{R}^{d_2} als $\langle \phi(x_i), \phi(x_j) \rangle$ zu berechnen. Zur effektiven Berechnung dieser Skalarprodukte verwendet man sogenannte Kernelfunktionen (Cristianini u. Shawe-Taylor [2004]):

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle. \quad (2.30)$$

Nachfolgend sind beispielhaft zwei symmetrische nicht-lineare Kernelfunktionen angegeben:

- Polynomiale Kernelfunktion: $k(x_i, x_j) = (\gamma \langle x_i, x_j \rangle + c_0)^d$, mit $c_0, \gamma > 0$
- Kernelfunktion mit radialer Basis: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, mit $\gamma > 0$

Wie man sich vorstellen kann, ist eine zu starke Anpassung der trennenden Hyperebene an gegebene Muster ungünstig, da somit die Generalisierungsfähigkeit des Klassifikators eingeschränkt wird. Man spricht dabei von 'Overfitting'. Um die optimalen Parameter einer SVM zu bestimmen, kann man die oben angesprochene Leave One Out Cross Validation benutzen, was jedoch je nach Komplexität der Daten mit einigem Rechenaufwand verbunden sein kann. SVM's gelten als sehr robuste Klassifizierer, wobei auch bei hochdimensionalen Merkmalsräumen eine sehr gute Generalisierungsfähigkeit nachgewiesen wurde (Schölkopf u. Smola [2002]). Zwei wichtige Einschränkungen sind jedoch anzumerken:

1. Die Anzahl von Support-Vektoren steigt typischerweise linear mit der Anzahl der Trainingsbeispiele, was die Komplexität des Modells und damit den Aufwand bei der Berechnung erhöht.

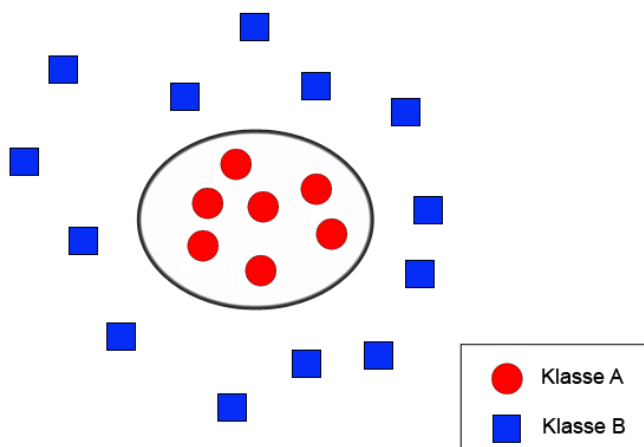


Abbildung 2.16.: Darstellung eines nicht-linear trennbaren Klassifikationsproblems

2. Die Vorhersagen sind nicht probabilistisch, sondern binäre Entscheidungen. Für Klassifikationsprobleme, welche auf Messdaten beruhen, ist es jedoch wünschenswert Wahrscheinlichkeiten der Klassenzugehörigkeit abzuleiten.

Um einige Einschränkungen der SVM's zu überwinden, wurde die Relevance Vector Machine eingeführt. Die RVM stellt eine Erweiterung der SVM unter Einbeziehung des Faktes dar, dass die Stichproben in Trainings- und Testmengen mit Messfehlern und Rauschen behaftet sind und somit ihre Beziehung nicht-deterministisch ist. Unter diesen Voraussetzungen kann man sowohl die Menge der Trainings- und Testdaten als auch die Parameter des Klassifikators als Zufallsvariablen betrachten, welche einer bestimmten Verteilung unterliegen. Es wird ein Wahrscheinlichkeitsmodell der Klassenzugehörigkeit Y der Merkmale unter der Voraussetzung der Werte dieser Merkmale X aufgestellt. Zusätzlich werden in diesem Modell sogenannte Hyperparameter eingeführt (sogenannte priors), welche benutzt werden, um die Parameter w des Modells zu wichten. Mit Hilfe des Bayestheorems wird für die Parameter in w eine Verteilung geschätzt, indem die Hyperparameter gesucht werden, welche eine optimale Trennung der Klassen bewirken. Wichtig ist dabei, dass nur Merkmale verwendet werden, welchen maximalen Einfluss auf die Trennung der Klassen haben, diese werden hier 'Relevance-Vektoren' genannt. Weitere Informationen zu Relevance Vector Machines finden sich in Tipping [2001].

3. Hirn-Computer Schnittstellen auf der Basis von Echtzeit-fMRT

Dieses Kapitel wird sich mit den Besonderheiten funktioneller MRT-Messungen befassen, wenn diese in Echtzeit ausgewertet werden sollen und somit verwendet werden können um Hirn-Computer Schnittstellen (Brain Computer Interface - BCI) zu implementieren. Die Technik der Echtzeit-fMRT wurde erstmals im Jahr 1995 von Cox et al. [1995] vorgestellt. Unter dem Begriff 'Echtzeit' ist in diesem Zusammenhang eine Auswertung eines Datensatzes innerhalb der Repetitionszeit der Messung zu verstehen. Wie Abschnitt 3.1 verdeutlichen wird, sind verschiedene Soft- und Hardwaresysteme an der Implementierung eines Gesamtkonzeptes für die Echtzeit-fMRT beteiligt.

Neben der Verlaufskontrolle bei herkömmlichen fMRT-Experimenten bilden zwei Schwerpunkte die Anwendung der Echtzeit-fMRT. Zum einen werden BCI's auf Basis der Echtzeit-fMRT verwendet, um Kommunikations- und Steuerungsanwendungen zwischen Mensch und Computer zu implementieren. Zum anderen werden mit Hilfe solcher Systeme Neurofeedback-Experimente durchgeführt (Weiskopf et al. [2004b]). In Abschnitt 3.2 werden die Anwendungsgebiete verdeutlicht und Echtzeit-fMRT Studien der letzten Jahre vorgestellt. Feedbackstrategien und Besonderheiten aufgrund der Verzögerung der hämodynamischen Antwort werden in Abschnitt 3.3 besprochen.

3.1. Technische Lösungen und Auswerteverfahren der Echtzeit-fMRT

Grundsätzlich können bei der Echtzeit-fMRT die gleichen Verfahren zur Signalverbesserung und Auswertung angewandt werden wie bei der herkömmlichen fMRT. Das Besondere ist die zeitliche Limitierung des Gesamtprozesses. Während eine vollständige Auswertung einer Zeitreihe mit einer Applikation wie SPM mehrere Stunden dauern kann, muss bei der Echtzeit-Auswertung das laufende Ergebnis innerhalb weniger Sekunden vorliegen (Weiskopf et al. [2007]). Bei der Verarbeitung innerhalb einer solchen Auswertung kann man verschiedene Schritte unterscheiden. Nach der Akquise eines ein-

zelen Volumendatensatzes wird dieser meist zur Verarbeitung zu einem externen System exportiert. In diesem System werden die Daten vorverarbeitet und eine statistische Analyse wird durchgeführt. Das Besondere an einem Echtzeit-fMRT-System ist nun, dass das Ergebnis der Analyse wiederum benutzt werden kann, um das Verhalten des Systems (z.B. die Stimuluspräsentation) zu beeinflussen. Abbildung 3.1 verdeutlicht den möglichen Aufbau eines Echtzeit-fMRT-Systems und die Datenströme in einer solchen Implementierung. Damit die einzelnen Prozesse in einem solchen System synchron ablaufen, werden oftmals Triggersignale verwendet. Im Allgemeinen wird das Triggersignal vom Tomographen für jeden Scan erzeugt und die externen Systeme (z.B. Datenauswertung und Stimuluspräsentation) verwenden dieses für ihre zeitliche Steuerung. Eine

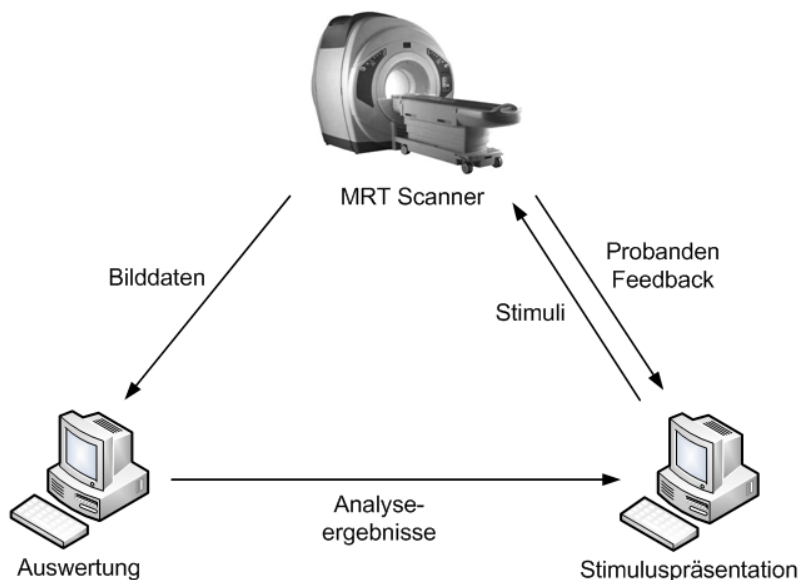


Abbildung 3.1.: Allgemeiner Aufbau eines Echtzeit-fMRT-Systems: Unter „Probanden Feedback“ sind in diesem Fall Rückmeldungen des Probanden durch Tastendrücke etc. zu verstehen. Die Pfeile bezeichnen Datenflüsse, welche über verschiedene technische Lösungen wie z.B. Netzwerkverbindungen, serielle oder optische Verbindungen realisiert werden.

typische Neurofeedback-Implementierung besteht beispielsweise darin, einem Probanden das BOLD-Signal aus einer vordefinierten funktionellen Region (Region of Interest - ROI) „rückzumelden“. Somit ist es dem Probanden möglich seine eigene Hirnaktivität in der Zielregion zu beeinflussen.

3.1.1. Vorverarbeitung

In der Echtzeit-fMRT kommen zur Vorverarbeitung und Auswertung ähnliche Verfahren wie in der fMRT (vgl. Abschnitt 2.2.4 und 2.2.5) zur Anwendung. Wie bereits angesprochen, wird jedoch oftmals das Signal in bestimmten vordefinierten Regionen (ROI's) ausgewertet. Somit kommt der Bewegungskorrektur eine besondere Rolle zu, da eine Verschiebung des analysierten Bereiches Fehler bei der Auswertung zur Folge hat. Eine Analyse von Bewegungsartefakten und Lösungsansätze für die Echtzeit-fMRT findet sich in Mathiak u. Posse [2001]. In diesem Zusammenhang muss auch auf Verschiebungen des Messvolumens geachtet werden, die z.B. durch die Erwärmung von Scannerbauteilen hervorgerufen werden können. Frequenzverschiebungen zwischen Messungen, welche sich auf die gleiche ROI beziehen, bewirken eine Translation des Messvolumens und müssen durch die sogenannte Frequenzjustage ausgeglichen werden. Ebenso sollten Suszeptibilitätsartefakte bei der Auswahl von ROI's beachtet werden (vgl. Weiskopf et al. [2004b]). Um physiologische Artefakte (z.B. durch Atmung verursacht) und lineare Signaldrifts zu korrigieren, werden in der Echtzeit-fMRT meist die in Abschnitt 2.2.4 beschriebenen Verfahren angewandt. Dabei ist zu beachten, dass zu Beginn einer Messung nicht genug Daten vorliegen, um globale Effekte zuverlässig zu extrahieren. Als Korrekturmöglichkeit wurde von Weiskopf et al. vorgeschlagen, das Ergebnis aus der Differenz des Signals zweier Regionen zu bestimmen (Weiskopf et al. [2004a, b]). Die Hypothese besteht dann darin, dass globale Effekte sich aufheben, Signalunterschiede aufgrund experimenteller Einflüsse in den Regionen jedoch bestehen bleiben. Ebenso gibt es Ansätze, welche die Korrektur globaler Signalstörungen in die statistische Auswertung von Echtzeit-fMRT-Daten integrieren (vgl. Gembris et al. [2000]).

3.1.2. Auswertung

Bei der statistischen Verarbeitung unterscheidet man zwei grundlegende Strategien. Bei der sogenannten *Growing-Window* Analyse wird das Ergebnis für den n-ten Datensatz (x_n) aus allen vorangegangenen Datensätzen ermittelt: $y_n = f([x_1; x_2; \dots; x_n])$. Bei der *Sliding-Window* Analyse hingegen verarbeitet man die Zeitreihen in einem festen Fenster: $y_n = f([x_{n-c+1}; x_{n-c+2}; \dots; x_n])$ mit der Fenstergröße $1 \leq c < n$. Somit ist man bei der Sliding-Window Analyse in der Lage dynamische Experimente zu realisieren, in denen das Ergebnis von der aktuellen Reaktion des Probanden abhängt und damit unabhängig ist von Werten experimenteller Variablen, die zeitlich vor dem Sliding-Window liegen (Gembris et al. [2000]). Als Sliding-Window in einem solchen Versuch eignet sich z.B. die zusammengefasste Zeitdauer für eine Ruhe- und eine Stimulationsperiode in einem

Blockdesign (vgl. hierzu Abschnitt 2.2.3). In Neurofeedback-Experimenten kommt oft ein kontinuierliches Feedback zur Anwendung. Dabei wird in jedem Scan einer Aktiv-Periode das Feedback generiert (Weiskopf et al. [2004a]).

Sowohl bei Growing- als auch bei Sliding-Window Auswertungen kann man die schon vorgestellten univariaten Analyseverfahren verwenden, wobei es Modifikationen für die schnelle Verarbeitung in Echtzeit-Anwendungen gibt (BrainVoyager [2009]; Gembris et al. [2000]; Nakai et al. [2006]). Hinzu gekommen ist in jüngerer Zeit die Echtzeit-Klassifikation funktioneller MRT-Daten (Hollmann et al. [2009]; LaConte et al. [2007]).

3.2. Anwendungsgebiete der Echtzeit-fMRT

Um mögliche Vorgehensweisen und konkrete Anwendungen der Echtzeit-fMRT zu verdeutlichen, werden hier einige veröffentlichte Studien zu diesem Thema angeführt. Eine Übersicht einiger Studien ist in Tabelle 3.1 dargestellt. Die Anwendungsgebiete der vorgestellten Technik beschränken sich im Wesentlichen auf drei Bereiche. Zum einen nutzt man Echtzeit-fMRT zur Abschätzung der Hirnaktivität von Probanden während einer fMRT-Messung, wobei die eigentliche Auswertung offline erfolgt (vgl. Voyvodic 1999 in Tabelle 3.1). Somit kann man schon während der Messung prüfen, ob das experimentelle Setup Fehler aufweist oder starke Bewegungsartefakte eine spätere Auswertung erschweren.

Das weitaus größere Anwendungsgebiet ist jedoch die Implementierung von Mensch-Maschinen-Interaktionen mit Hilfe von Hirn-Computer Schnittstellen (DeCharms [2008]; Weiskopf et al. [2007]). Dies beinhaltet die Wirkung der Hirnaktivität des Probanden auf Variablen in der Echtzeit-Umgebung insbesondere auf die Stimuluspräsentation. Diese Methodik kann z.B. als Neurofeedback zur Selbstregulierung von Hirnaktivität (Self-Regulation) verwendet werden, indem ein Proband lernt, mit Hilfe der Anzeige der Aktivierungsstärke einer bestimmten Region, die neuronale Aktivität in dieser zu beeinflussen (vgl. z.B. deCharms et al. 2004 und Weiskopf et al. 2003 in Tabelle 3.1). Es wurde gezeigt, dass Probanden mit Hilfe operanter Lernmethoden über Neurofeedback signifikant besser in der Lage waren die Aktivität in ihren Hirnregionen zu kontrollieren als ohne Feedbackstrategien (Weiskopf et al. [2007]). Ziel solcher Studien ist zu zeigen, dass über Feedbackmethoden die eigentliche Zielgröße, wie z.B. das subjektive Schmerzempfinden, beeinflusst werden kann (vgl. deCharms et al. 2005 in Tabelle 3.1).

Weiterhin gibt es Bestrebungen BCI's auf Basis der Echtzeit-fMRT zur Steuerung von Computern (vgl. Yoo et al. in Tabelle 3.1) oder zur Kommunikation mit Patienten zu verwenden, die auf herkömmliche Weise (verbal oder über Gebärden) nicht mit ihrer

Umwelt interagieren können (DeCharms [2008]; Owen et al. [2006]). In letztere Kategorie fällt z.B. auch eine Arbeit von Sorger et al. [2007], bei der Probanden lernten die Buchstaben des Alphabetes über ihre Hirnaktivierung zu kodieren.

3.3. Zeitverlauf der HRF und Feedbackstrategien

Einige Besonderheiten im Zusammenhang mit BCI's, welche mit Hilfe von Echtzeit-fMRT implementiert sind, sollen im Folgenden betrachtet werden. Zum einen ist zu beachten, dass die hämodynamische Antwort auf Probandenverhalten eine Verzögerung von ca. 4 bis 6 Sekunden aufweist. Damit erfolgt auch die Antwort eines BCI's verzögert. Über diesen Umstand müssen Probanden aufgeklärt werden und sie müssen z.B. über Trainingsstrategien oder das sogenannte „Shaping“ (z.B. langsame Steigerung der Anforderungen) an den Umgang mit BCI's gewöhnt werden (Weiskopf et al. [2004b]). Oft kommt bei BCI's die Analyse einzelner anatomischer oder funktioneller Gebiete mit Hilfe von ROI's zur Anwendung. Diese müssen dann vor einem Messdurchgang mit Feedback definiert sein, was oft über Vorexperimente (sogenannte *functional localizer*) oder die Auswahl von Regionen anhand anatomischer Merkmale erfolgt. Das abgeleitete und bearbeitete Signal in solch einer ROI dient dann z.B. als Einflussgröße für die Stimuluspräsentation. Soll der Proband über die eigene Aktivierungsstärke einer Region informiert werden, muss dies einfach und übersichtlich geschehen (Weiskopf et al. [2004b]). Neben wenigen Studien wie z.B. Posse et al. [2003], welche den Probanden verbal über seine Aktivierung informierten, hat sich die visuelle Darstellung durchgesetzt. Oft wird hier vereinfacht durch Balken oder Kurvendarstellungen die Stärke der Aktivierung angezeigt (vgl. Abbildung 3.2 a) (Weiskopf et al. [2004b]). In (DeCharms et al. [2005]) wurde die Stärke der Aktivierung im rostralen anterioren Cingulum (rACC) über die Größe von virtuellen Flammen visualisiert (siehe Abbildung 3.2 b), wobei Probanden die Kontrolle ihres Schmerzempfindens mit Hilfe dieses Feedbacks erlernten.

Für motorische, sprach- und stimmungsbezogene Aufgaben wurden von LaConte et al. Klassifikationsmethoden zur Analyse verteilter Aktivitätsmuster eingesetzt (siehe Tabelle 3.1). Als Feedback wurden hier die Entscheidungen des Klassifikators visualisiert. Eine Studie, welche die Echtzeit-Klassifikation von Probandenentscheidungen in sozialen Interaktionsszenarien umsetzt (Hollmann et al. in Tabelle 3.1) wird in Abschnitt 7 näher betrachtet werden.

Echtzeit-fMRT verspricht weitreichende Möglichkeiten in Bezug auf Selbstregulierung und Steuerung von Feedback-Systemen durch Hirnaktivität. Ein großer Vorteil ist die

Studie	Untersuchte Hirnareale	Hintergrund	Technische Verfahren
Voyvodic [1999]	Visueller Kortex	Verlaufskontrolle bei fMRT-Studien	t-Test
Posse et al. [2003]	Amygdala	Kontrolle der Hirnaktivierung durch selbst induzierte Traurigkeit	Korrelationsanalyse, Paralleles Rechnen
Weiskopf et al. [2003]	Anteriores Cingulum (ACC)	Kontrolle der Aktivierung durch den Probanden	Echtzeit-GLM
DeCharms et al. [2004]	Motorische und sensorische Areale (M1 und S1)	Untersuchung von Trainingseffekten bei Selbstregulierung	Analyse von ROI-Signaldifferenzen
Yoo et al. [2004]	M1, Medialer Frontaler Gyrus, Broca Areal, Auditorischer Kortex	Navigation durch ein Labyrinth mit Hilfe von Hirnaktivierung	t-Test
DeCharms et al. [2005]	rostraler ACC	Erlernen von Schmerzkontrolle durch Patienten	Analyse von ROI-Signaldifferenzen
Caria et al. [2007]	Anteriore Insula	Regulierung durch Erinnerung an affektauslösende Ereignisse	Echtzeit-GLM, Analyse von ROI-Signaldifferenzen
LaConte et al. [2007]	Verteilte Analyse	Klassifikation von kognitiven Zuständen	Echtzeit-Klassifikation (SVM)
Hollmann et al. [2009]	Anteriore Insula, Lateraler Präfrontaler Kortex (LPFC), Occipitaler Kortex	Vorhersage von Entscheidungen bei sozialer Interaktion	t-Test, Echtzeit-Klassifikation (RVM)

Tabelle 3.1.: Auswahl wichtiger Echtzeit-fMRT-Studien (chronologische Ordnung)

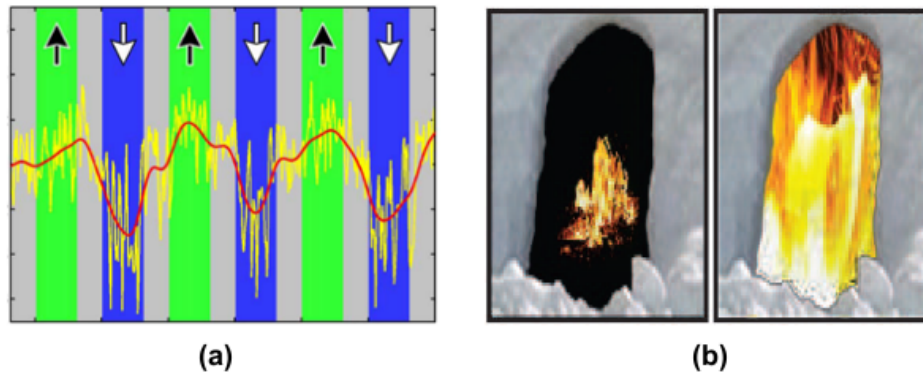


Abbildung 3.2.: Visualisierung der Aktivierungsstärke: **(a)** Visualisierung der Aktivierungsstärke über den geglätteten Signalverlauf (rote Linie) (Weiskopf et al. [2004a]) **(b)** Darstellung der Aktivierungsstärke im rostralen anterioren Cingulum über virtuelle Flammen (DeCharms et al. [2005])

hohe räumliche Auflösung der fMRT. Somit ist man in der Lage räumlich getrennte aufgabenspezifische Aktivierungen mit Hilfe von ROI's zu verarbeiten (DeCharms et al. [2004]; Weiskopf et al. [2004b]). Klassifizierungsverfahren erweitern die Möglichkeiten um spezifische räumliche oder zeitliche Muster in der Hirnaktivität von Probanden auch in der Echtzeit-fMRT.

4. Problembeschreibung

4.1. Informationstheoretische Anforderungen an die Echtzeit-fMRT

Analysiert man Echtzeit-fMRT-Systeme, kann man folgende Subsysteme unterscheiden (Hollmann et al. [2008b]):

- *Datenakquisition*: Hierzu wird in diesem Zusammenhang der Magnetresonanztomograph mit zugehörigen Hard- und Softwaresystemen gerechnet. Es kommen die bereits beschriebenen komplexen Steuersequenzen zum Einsatz, welche z.B. die zeitgenaue Ansteuerung der Gradientensysteme übernehmen (vgl. Abschnitt 2.1.5). Dem Bereich Datenakquisition wird hier auch die Bildrekonstruktion zugeordnet.
- *Vorverarbeitung und Statistische Auswertung*: Dieses System übernimmt rekonstruierte Daten und implementiert die Vorverarbeitung sowie die Extraktion von statistischen Größen, welche den Aktivierungsstatus beschreiben.
- *Aktivierungsverarbeitung*: Die Weiterverarbeitung bzw. Interpretation der statistischen Ergebnisse wurde bewusst von dem vorhergehenden Punkt getrennt, da die Analyse mitunter eine eigenständige und komplexe Verarbeitung beinhaltet. Hierzu zählen z.B. Klassifikationsroutinen oder die Integration statistischer Ergebnisse in Routinen zur abstrakten Visualisierung eines Sachverhaltes wie z.B. die Bewegung in einem virtuellen Labyrinth (vgl. Yoo et al. [2004] und Moench et al. [2008]).
- *Stimuluspräsentation*: Ein System zur Stimuluspräsentation muss in der Lage sein unter festen zeitlichen Bedingungen verschiedene Reize (visuell, auditiv, taktil etc.) zu präsentieren. Dies beinhaltet auch die Kommunikation mit dem Tomographen und den Systemen zur Datenauswertung.

Abbildung 4.1 stellt die oben genannten Subsysteme und mögliche Datenströme zwischen diesen Systemen dar. Jedes Subsystem ist dabei standardmäßig durch autarke Soft- und

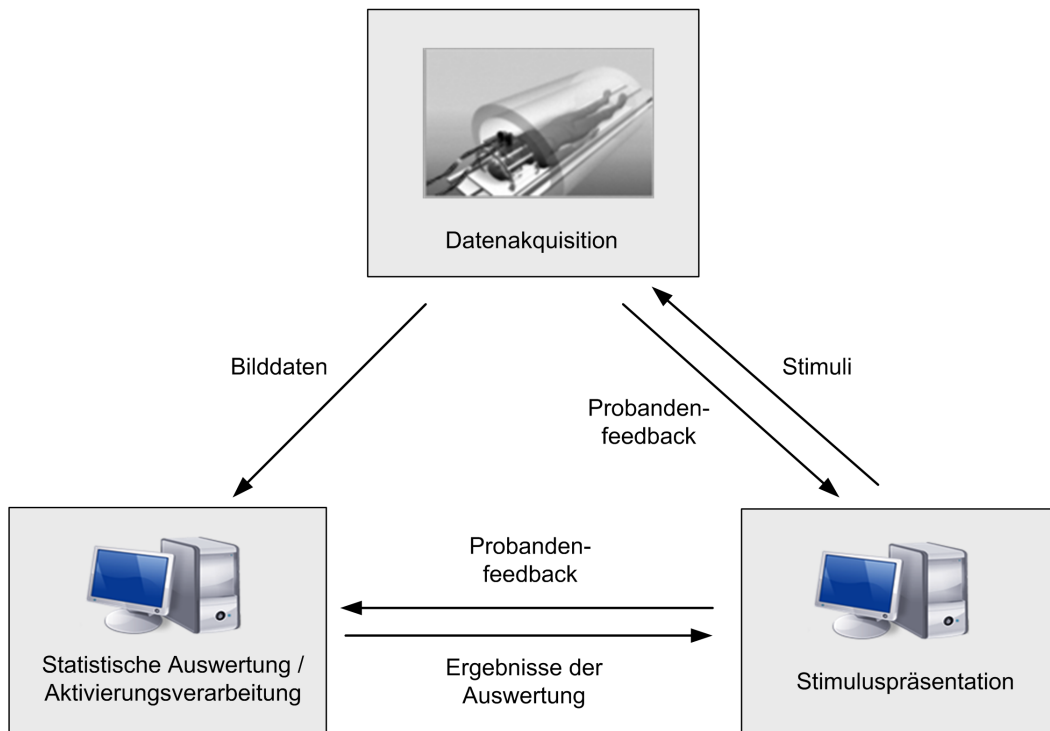


Abbildung 4.1.: Subsysteme und Datenströme einer möglichen Ausprägung eines Echtzeit-fMRT-Systems: Jedes System verfügt über einen eigenen Parameterraum. Die Konsistenz übergreifender Parameter, die zeitliche Synchronität sowie die Kommunikation zwischen den Systemen müssen sichergestellt werden.

Hardwarelösungen repräsentiert. Die Systeme arbeiten dabei mit einer Vielzahl von Parametern deren Konsistenz im Allgemeinen von den Systemen selbst sichergestellt wird. Die Trennung der einzelnen Komponenten, die an einem Echtzeit-fMRT-Experiment beteiligt sind ist problematisch, da Abhängigkeiten auch zwischen Parametern der Subsysteme bestehen. Die Integration dieser Abhängigkeiten ist üblicherweise dem Nutzer überlassen. Aus diesem Umstand ergibt sich eine hohe Fehleranfälligkeit beim Aufbau eines Echtzeit-fMRT-Experimentes (Smyser et al. [2001]).

Die Simulation eines Experimentes wäre aufgrund der Komplexität dieses Aufbaus und des Anspruchs der Echtzeitfähigkeit wünschenswert, ist jedoch ebenfalls aufgrund des genannten Aufbaus erschwert. Für dynamische Paradigmen, also Experimente in denen z.B. die Stimuluspräsentation durch die Analyse der Hirnaktivierung beeinflusst wird, ist es nötig eine Kommunikation zwischen den Systemen zu etablieren. Auch dies ist bei getrennten Komponenten nicht ohne Weiteres auf verifizierbare Weise zu implementieren.

Um systemübergreifende Parameterabhängigkeiten testen zu können und Simulationen

eines kompletten Experimentes automatisieren zu können, ist also eine zentrale Verwaltung von Parametern des Gesamtsystems nötig. Ein weiteres Problem ist, dass herkömmliche Scannersysteme meist nicht für den Echtzeit-Export der Bilddaten ausgelegt sind. Um solch ein System zu etablieren, muss also oftmals die Software des MRT modifiziert werden. Der letztgenannte Umstand wird nicht Gegenstand weiterer Betrachtungen sein, vielmehr wird in dieser Arbeit auf die zentralisierte Parameterverwaltung eingegangen.

4.2. Einführung eines Beispiexperimentes

Beispielhaft soll an dieser Stelle die vorgestellte Problemstellung der systemübergreifenden Parameterabhängigkeiten verdeutlicht werden. Das im Folgenden eingeführte einfache Beispiel wird im weiteren Verlauf der Arbeit dazu dienen die Funktionsweise des in dieser Arbeit entwickelten Ansatzes zu demonstrieren und wird in den Kapiteln 5 und 6 fortgeführt. Das folgende Szenario soll dabei als Grundlage dienen:

Ein Nutzer möchte ein einfaches Echtzeit-fMRT-Experiment durchführen. Im Rahmen der Datenakquisition wird ein TR von 1,5 Sekunden definiert und die Systeme (Auswertung und Stimuluspräsentation) sollen mit dem Triggersignal des ersten Scans starten. Bei der Definition des Designs für die statistische Auswertung erfolgen die Angaben in Scans. Die ersten sechs Scans sollen verworfen werden. Insgesamt sollen vier Blöcke mit jeweils einer Länge von acht Scans gemessen werden, wobei die Pause zwischen den Blöcken zwischen null und zwei Scans variiert. Die Auswertung des Echtzeit-Experimentes soll mit Hilfe eines Sliding-Windows für jeden Block vorgenommen werden. Die Gesamtlänge des Experimentes ergibt sich als 41 Scans, also 61,5 Sekunden:

$$6 + 8 + 0 + 8 + 2 + 8 + 1 + 8 = 41. \quad (4.1)$$

Als Stimulus dient die visuelle Präsentation eines Checkerboard-Stimulus¹ für jeweils 1,5 Sekunden, wobei in der Ruhephase ein schwarzer Bildschirm mit einem Fixationspunkt gezeigt werden soll. Das Stimulusdesign ist in Abbildung 4.2 dargestellt, die Definition erfolgt hierbei in Sekunden. Wie man in diesem Beispiel erkennen kann, ist dem Nutzer ein Fehler unterlaufen. In der Angabe des Stimulusdesigns wurden die sechs Scans, welche zu Beginn des Experimentes ignoriert werden sollen, nicht einbezogen. Da jedoch Stimuluspräsentation und Auswertung gleichzeitig starten, sind beide Systeme nicht synchronisiert und die Stimuluspräsentation endet vor dem im Paradigma vorgesehenen Zeitpunkt.

¹ein Schachbrettmuster in welchem schwarze und weiße Flächen mit festgelegter Frequenz wechseln



Abbildung 4.2.: Darstellung des Stimulusdesigns des Beispielerperimentes in Sekunden

Ein Mitarbeiter soll nun das oben angeführte Experiment durchführen und bezieht bei der Definition der Parameter zur Akquise die ignorierten Scans ebenfalls nicht ein, weil er sich an der Länge der Stimuluspräsentation orientiert. Somit werden 35 statt 41 Scans in der Benutzeroberfläche des MR-Tomographen als Scananzahl definiert.

In getrennten Systemen können diese strukturellen Fehler durchaus verborgen bleiben, wenn nicht weitere Tests durchgeführt werden. Mit Hilfe von explizitem Wissen über den Aufbau eines Experimentes kann jedoch ein automatisches Testverfahren diesen Fehler finden. Wie das in dieser Arbeit vorgestellte integrierte System das nötige Wissen repräsentiert um die beispielbezogene Problemstellung zu lösen, wird im Abschnitt 5.6 beschrieben. Die softwaretechnische Umsetzung der automatisierten Tests hierzu werden in Abschnitt 6.5 verdeutlicht.

4.3. Verwandte Arbeiten

Es gibt bereits Arbeiten, welche sich mit den beschriebenen Herausforderungen beschäftigt haben. Insbesondere werden hier die Arbeiten von Voyvodic (Voyvodic [1999]) und Smyser (Smyser et al. [2001]) besprochen. Des Weiteren wird ein kommerzielles Produkt vorgestellt, welches eine Lösung für angesprochene Probleme beinhaltet. In der Arbeit von Voyvodic [1999] wird ein Ansatz vorgestellt, in welchem Stimuluspräsentation, Echtzeit-fMRT-Statistik und die Aufnahme physiologischer Parameter mit Hilfe eines zentralen Kommunikationsmodells gesteuert wurden. Hier lagen physisch getrennte Rechnersysteme vor, welche über Netzwerkverbindungen gekoppelt waren. Die Statistische Auswertung wurde durch automatisch erzeugte Befehle im ASCII Format gesteuert, welche während der Messung vom Stimulussystem erzeugt und übertragen wurden. Somit wurden dynamische Paradigmen möglich, ohne im Vorhinein das Timing der statistischen Auswertung festzulegen. Ebenso konnten physiologische Parameter (Atmung und Herzschlag) in die statistische Auswertung integriert werden. Es wurden parallele

Prozesse und Echtzeit-Hardware verwendet um eine genügend schnelle Kommunikation zwischen den Systemen sicherzustellen. Implizit wird damit eine Verifizierung von Parametern der Stimuluspräsentation und der statistischen Analyse vorgenommen, da die Menge der nutzbaren Steuerbefehle bekannt ist und die zeitliche Kohärenz sichergestellt ist.

Im Ansatz von Smyser et al. [2001] wurden zeitabhängige Parameter wie Signale der Stimulusumgebung, physiologische Parameter und Probandenrückmeldungen (wie z.B. das Drücken eines Knopfes) mit exakten Zeitstempeln versehen abgespeichert. Diese LOG-Files wurden in einem speziell entwickelten Format (CVIO - Cardiovascular In/Out) entweder im Hauptspeicher (shared Memory) oder auf Festplatten abgelegt und konnten für die statistische Echtzeit-Auswertung mit linearen Regressionsmethoden verwendet werden.

Eine kommerzielle Anwendung, welche ein Gesamtsystem zur fMRT und Echtzeit-fMRT darstellt ist Eloquence/IFIS von der Firma Invivo (Invivo [2009]). Zu dem Konzept von Eloquence/IFIS gehört zum einen die Hardware wie spezielle Empfangspulen mit integrierten Stimulussystemen, Rechnerhardware zur Verarbeitung von fMRT-Daten oder auch Hardware zur Kommunikation mit dem Probanden (Taster etc.). Zum anderen werden aber auch die Softwarelösungen, welche die angegebenen Hardwaresysteme steuern als integriertes System betrieben. Der Nutzer kann in diesem System sowohl Stimuli definieren, als auch eine Auswertung der Daten in Echtzeit mit Hilfe von dem bereits angesprochenen Softwaretool Turbo-Brain Voyager (BrainVoyager [2009]) steuern. Innerhalb des Systems kann somit die Konsistenz der Parameter über implementierte Routinen sichergestellt werden (wobei Parameter des MRT-Scanners nicht berücksichtigt sind). Es werden jedoch keine Schnittstellen angeboten um eigene Funktionalität oder eine Erweiterung der Parameterräume zu implementieren.

In den vorgestellten Arbeiten wurden Lösungen für die schnelle Kommunikation bzw. zeitlich kohärente Aufnahme von experimentellen Parametern beschrieben. Implizit wurde die Gültigkeit von verschiedenen Parameterkombinationen wie z.B. das Stimulustiming in Kombination mit dem Timing der statistischen Auswertung sichergestellt. Es fällt jedoch auf, dass in allen dargestellten Ansätzen die Parameterrepräsentation nicht von der Implementation der Systeme entkoppelt wurde. Dies bringt mehrere Nachteile mit sich. Zum einen wird nur der Teil der Parameter auf Konsistenz geprüft, welcher zu den integrierten Systemen gehört. Im Ansatz von Voyvodic et al. beispielsweise sind dies nur die Parameter der Stimuluspräsentation und der statistischen Auswertung. Hierbei entziehen sich z.B. Parameter der Datenakquisition und systemübergreifende Abhängigkeiten der Prüfung. Zum anderen kann das Framework nicht leicht erweitert werden,

weil kein Zugang zu den Systemparametern vorgesehen ist. Als Beispiel kann man die automatische Erstellung von Testdaten anführen. Mit einer zentralen Repräsentation hätte eine Applikation für solche Aufgaben alle nötigen Parameter zur Verfügung, was mit den abgeschlossenen Systemen, die zuvor vorgestellt wurden, nicht möglich ist. Eine zentrale Abbildung der Parameter und ihrer Abhängigkeiten ermöglicht die Formulierung von explizitem Wissen über ein Experiment und kann deshalb zur Automatisierung von Konsistenzprüfungen und Simulationen verwendet werden. Des Weiteren kann eine solche externe Repräsentation die Grundlage für die Erweiterung der Funktionalität eines Frameworks durch grundsätzlich unabhängige Applikationen sein und somit z.B. automatisierte Dokumentationen von Experimenten ermöglichen. Solch eine Repräsentation eignet sich auch für den einfachen Austausch von Informationen zu Experimenten, wodurch Arbeitsabläufe bei neurologischen Experimenten vereinfacht würden. Das nachfolgende Kapitel wird eine integrierte Lösung zu der vorliegenden Problemstellung behandeln.

5. Wissensrepräsentation zur Beschreibung von Experimenten in der Echtzeit-fMRT

In diesem Kapitel wird eine Wissensbasis zur Beschreibung von Experimenten im Kontext der Echtzeit fMRT vorgestellt. Wichtige Teile des hier entwickelten Ansatzes sind in Hollmann et al. [2007b] und Hollmann et al. [2008b] beschrieben.

Die Definition des Begriffes „Wissen“ beschäftigt die Menschheit schon lange. In zahlreichen philosophischen Abhandlungen wurde versucht eine Abgrenzung von Glauben, Wahrheit und Wissen zu finden (Armstrong [1973]; Zins [2007]). Diese Diskussion ist noch nicht zu einem allgemeingültigen Ergebnis gekommen und soll hier auch nicht weiter thematisiert werden. In dieser Arbeit wird vielmehr ein Wissensbegriff aus der Psychologie nach Solso Grundlage sein: „Wissen ist organisierte Information, es ist Teil eines Systems oder Netzes aus strukturierten Informationen.“ (Solso [2005]). Im Folgenden sind die informationstheoretischen Hintergründe zur Wissensrepräsentation beschrieben, um eine geeignete Repräsentationsform von Wissen im Kontext von Experimenten in der Echtzeit-fMRT zu finden. Eine mögliche Darstellung von Parameterzusammenhängen in Experimenten findet sich in der strukturierten Modellierung von Information wie sie in Abschnitt 5.1 eingeführt wird. Der Verbindung dieser Information in regelbasierten Systemen werden im Abschnitt 5.2 näher beschrieben. Welche Anforderungen an die hier entwickelte Repräsentation gestellt werden ist im Abschnitt 5.3 dargestellt, wobei der aus diesen Anforderungen resultierende Entwurf im Abschnitt 5.4 vorgestellt wird.

Die Grundzüge des hier entwickelten Ansatzes werden in Abschnitt 5.5 erläutert, dieser teilt sich in zwei Unterabschnitte. Die Beschreibung der praktischen Umsetzung in einer XML-basierten Beschreibungssprache namens *Experiment Description Language* (EDL) findet sich in Abschnitt 5.5.1 und die Darstellung der Umsetzung einer Regelbasis zu EDL erfolgt in Abschnitt 5.5.2. Es folgt der Bezug auf das in der Problembeschreibung eingeführte Beispiexperiment indem die Mechanismen zur Darstellung der nötigen Informationen vorgestellt werden (Abschnitt 5.6) und die Diskussion in Abschnitt 5.7.

5.1. Strukturierte Informationsrepräsentation mit XML

XML (Extensible Markup Language) ist eine Meta-Sprache, die zur strukturierten Modellierung von Informationen verwendet werden kann. XML wurde vom World Wide Web Consortium (W3C) im Jahr 1998 in der Version 1.0 eingeführt (W3C [2009a]). Mittlerweile gilt XML als Standard zur Erstellung von maschinenlesbaren Dokumenten. Information wird in XML allein auf Grundlage ihrer internen Abhängigkeiten und unabhängig von verarbeitenden Soft- und Hardwaresystemen repräsentiert (Lobin [2000]). Der XML-Standard wurde aus einem umfangreicheren Standard namens SGML (Standard Generalized Markup Language) entwickelt. XML stellt dabei eine Untermenge von SGML dar und soll die Verwendung von Beschreibungssprachen vereinfachen. Eine weit verbreitete Anwendung von SGML ist HTML (Hypertext Markup Language), welche mit ihrer Bedeutung für das Internet das Potenzial von Beschreibungssprachen verdeutlicht. Bei der Verwendung von Meta-Sprachen wird zwischen dem Standard selbst und der Anwendung des Standards unterschieden. Als Anwendung wird hier die konkrete Beschreibung einer formalen Sprache (abstrakter Elemente und einer Grammatik auf diesen Elementen) mit Hilfe des Standards verstanden. In XML werden verschiedene abstrakte Einheiten unterschieden:

- Container-Elemente: Diese Elemente enthalten wiederum Elemente und ermöglichen damit die Relation *'ist Teil von'* in der Informationsbeschreibung.
- Daten-Elemente: Terminale Elemente, welche konkrete Daten enthalten.
- Leere Elemente: Elemente die mit keinem Inhalt versehen sind.

Die Elemente dienen dabei als Informationseinheiten, wobei die strukturierte Information sich aus der regelgeleiteten Anordnung der Elemente ergibt. Um die Flexibilität der zu entwickelnden Grammatik zu erhöhen, kann in XML jedes Element zusätzlich mit Parametern (Attributen) versehen werden. Die Definition einer Grammatik über Elementtypen und deren strukturellen Beziehungen erfolgt in sogenannten Typdefinitionen. Eine Instanz eines durch eine Typdefinition vorgegebenen Dokumentes stellt hingegen die konkrete Ausprägung mit Wertzuweisungen dar. Die Syntax von Instanzen (XML-Dokumenten) ist wie bei vielen Auszeichnungssprachen *Tag*-orientiert:

```
<Element1>
  <Element1.1 Attribut1.1="Attribut-Inhalt">
    Daten-Inhalt
  <\Element1.1>
  <Element1.2 Attribut1.2="Attribut-Inhalt">
```

```

    Daten-Inhalt
  <\Element1.2>
<\Element1>

```

Es ergibt sich in einem solchen Dokument eine Baumstruktur, die für die algorithmische Verarbeitung von Vorteil ist. Sowohl terminalen Elementen mit Daten-Inhalt und auch Attributen kann ein vom Standard definierter oder intern beschriebener Datentyp zugeordnet werden. Ausserdem können gültige Wertebereiche für Elemente und Attribute definiert werden. In dieser Arbeit wird die Technik der Typdefinition per XML-Schema verwendet (W3C [2009b]). Die Typdefinition wird dabei in einem Standard namens XSD (XML Schema Definition) beschrieben. Der wichtigste Vorteil der Beschreibung über ein XML-Schema liegt dabei in den bereits definierten Datentypen (built-in datatypes), welche neben numerischen und textuellen auch zeitbezogene Typen wie Zeitdauer und Datumsangaben beinhalten.

Man kann ein XML-Dokument hinsichtlich zweier Eigenschaften charakterisieren. Zum einen hinsichtlich der *Wohlgeformtheit* und zum anderen hinsichtlich der *Validierbarkeit*. Dabei stellt die Wohlgeformtheit eine schwächere Eigenschaft dar, da hier lediglich Aussagen über eine korrekte Annotation des Dokumentes getroffen werden. Ein Dokument muss nicht zwingend über eine Typdefinition bzw. ein XML-Schema verfügen oder kann hinsichtlich dieser falsch sein und trotzdem 'wohlgeformt' sein (Lobin [2000]). Die Validierbarkeit hingegen trifft eine Aussage über die syntaktische und semantische Korrektheit eines Dokumentes bezüglich seiner Typdefinition. Dabei spielen z.B. auch in der Typdefinition angegebene Wertebereiche und Datentypen eine Rolle. Sowohl die Wohlgeformtheit als auch die Validierbarkeit von XML-Dokumenten werden von sogenannten XML-Parsern getestet, welche weit verbreitete Anwendungen sind und in effizienten Implementierungen vorliegen (Lobin [2000]).

5.2. Wissensrepräsentation in regelbasierten Systemen

Wesentlich an der Definition des Wissensbegriffes nach Solso (siehe Beginn dieses Kapitels) ist, dass sich Wissen ergibt, wenn Information vernetzt ist. Dieser Wissenbegriff stellt weder Anforderungen an die Wahrheit der Information noch an ein Bewusstsein des informationsverarbeitenden Systems, womit also auch ein Computersystem über Wissen verfügen kann (Schmitz u. Zucker [2003]; Solso [2005]). Die Hauptkomponenten eines wissensbasierten Systems sind eine Wissensbasis, also eine irgendwie geartete Darstellung von Wissen und eine Inferenzkomponente. Unter Inferenzkomponente versteht man dabei ein System, welches in der Lage ist auf formalen Repräsentationen von Wissen

Schlussfolgerungen zu ziehen (Beierle u. Kern-Isberner [2008]). Bereits repräsentiertes Wissen W und neues Wissen B sind dann über die Inferenzrelation R miteinander verbunden:

$$(W, B) \in R. \quad (5.1)$$

Das hier betrachtete regelbasierte System basiert auf der Aussagenlogik, einer Untermenge der Prädikatenlogik. Ein solches System besteht aus Propositionen (Aussagen), die in einer Faktenmenge F zusammengefasst sind und die Verknüpfung dieser Propositionen¹ über Regeln (Brachman u. Levesque [2004]). Regeln sind in diesem Fall formalisierte Konditionalsätze der Form: *Wenn A dann B*. Im Weiteren wird die Notation $A \Rightarrow B$ für Konditionalsätze verwendet werden. Hierbei wird A als Prämisse und B als Konklusion bezeichnet (Beierle u. Kern-Isberner [2008]). In dieser Arbeit wird davon ausgegangen, dass A und B Aussagen sind, welche wiederum über die logischen Verknüpfungen \wedge (Konjunktion) oder \vee (Disjunktion) von Aussagen zusammengesetzt sein können. Terminale Aussagen, welche nicht aus anderen Aussagen zusammengesetzt sind nennt man Literale. Für die hier betrachteten regelbasierten Systeme gelten weiterhin zwei Aussagen:

1. Es werden nur deterministische Regeln verwandt (alle definierten Regeln gelten ohne Ausnahme).
2. Es ist kein unsicheres Wissen repräsentiert.

Das Schließen von Aussagen in einem solchen System nennt man Inferenz. Die grundlegende Inferenzregel in einem regelbasierten System ist der *modus ponens*: aus der Regel $(A \Rightarrow B)$ und dem Faktum A , folgt B . Die Umkehrung dieser Inferenz stellt der *modus tollens* dar: aus der Regel $(A \Rightarrow B)$ und dem Faktum $\neg B$, folgt $\neg A$. Ein System, welches nur den *modus ponens* implementiert muss bei Bedarf beide Regeln $(A \Rightarrow B)$ und $(\neg B \Rightarrow \neg A)$ definieren (Beierle u. Kern-Isberner [2008]; Brachman u. Levesque [2004]). Ein Standardverfahren um in einem regelbasierten System auf Wissen zu schließen ist die *Vorwärtsverkettung* oder auch *datengetriebene Inferenz*. Dabei wird jede Regel auf Gültigkeit der Prämisse getestet und bei deren Gültigkeit die Konklusion in die Faktenmenge aufgenommen (Altenkrüger u. Büttner [1992]). Das Schließen auf ausgewählte Fakten ist bei diesem Verfahren ineffizient, da alle Regeln durchlaufen werden. Ein effizienteres Verfahren stellt die *zielorientierte Inferenz* dar. In Listing 5.1 ist ein Algorithmus zur zielorientierten Inferenz skizziert. Dieser testet, ob eine Reihe atomarer Anfragen (Literale) $[b_1, \dots, b_n]$ aus einer gegebenen evidenziellen Faktenmenge $F = a_1, \dots, a_n$ und einer

¹Hinweis: Produktionen (Aktionen) werden hier nicht als Bestandteil von Regeln betrachtet, da diese eine im Verlauf der Arbeit nicht verwendete Sonderform darstellen.

Regelbasis RB ableitbar sind. Die Regeln in RB sollen dabei nur in der Normalform $a_1 \wedge \dots \wedge a_m \Rightarrow b$ vorliegen (modifiziert nach Beierle u. Kern-Isberner [2008]).

```

ZIEL-INFERENZ([ $b_1, \dots, b_n$ ])
  if n=0 then return(true);
  if  $b_1 \in F$ 
    then ZIEL-INFERENZ([ $b_2, \dots, b_n$ ])
  else
    foreach Rule  $a_1 \wedge \dots \wedge a_m \Rightarrow b \in RB$ , mit  $b_1 = b$  do
      if ZIEL-INFERENZ([ $a_1, \dots, a_m, b_2, \dots, b_n$ ])
        then return(true)
    endfor
  endif
return(false)

```

Listing 5.1: Ein Algorithmus zur zielorientierten Inferenz

Zu beachten ist, dass dieser Algorithmus nicht terminiert, wenn rekursive Regelmengen wie z.B. $a \Rightarrow b, b \Rightarrow a$ gegeben sind. Oft werden logische Programmiersprachen wie LISP und PROLOG für die Erstellung von wissensbasierten Systemen verwendet. Es wird aber zunehmend auch auf Standardsprachen wie C++ zurückgegriffen (Beierle u. Kern-Isberner [2008]; Brachman u. Levesque [2004]).

5.3. Anforderungen

Aus der in Abschnitt 4 eingeführten Problemstellung ergeben sich verschiedene Anforderungen an die Repräsentation von Wissen im Kontext von Echtzeit-fMRT-Messungen:

1. Die Wissensrepräsentation soll in der Lage sein die verschiedenen Teilsysteme einer Echtzeit-fMRT-Implementierung zu verbinden.
2. Die Beschreibung soll unabhängig von der tatsächlichen Implementierung der Funktionalität in einer Applikation sein. Es sollte also keine Rolle spielen, mit welcher Applikation und auf welchem Zielsystem die hier beschriebenen Parameter und Prozesse umgesetzt sind.
3. Die Repräsentation muss mit vertretbarem Aufwand auch hinsichtlich systemübergreifender Parameterabhängigkeiten validierbar sein. Sowohl die syntaktische als auch die semantische Korrektheit der Beschreibung müssen innerhalb von wenigen Sekunden getestet werden können.

4. Wie in der Problembeschreibung deutlich wird, ist es wichtig, leicht neue Funktionalität in ein Echtzeit-fMRT-System integrieren zu können. Deshalb sollte eine Experimentbeschreibung so konzipiert sein, dass Zusatzfunktionalität, welche in der Typdefinition einer Beschreibung nicht näher spezifiziert ist an festgelegten Punkten trotzdem eingebunden werden kann.
5. Es ist wünschenswert, dass eine Wissensrepräsentation einen großen Teil des Expertenwissens im Zielbereich abdeckt. Diese Forderung lässt sich mit einer leicht erweiterbaren Repräsentation erfüllen. Es ist wichtig, dass ein Nutzer in der Lage ist die Konzepte und Regeln in einem solchen System nachzuvollziehen um eigene Konzepte und Regeln integrieren zu können. Die Übersichtlichkeit und damit eine überschaubare Komplexität ist also eine zentrale Forderung.

Besonders die Forderung nach der Repräsentation von umfangreichem Wissen wird bei einem so komplexen Thema wie der Beschreibung von neurologischen Experimenten kaum vollständig erfüllt werden können. Die Möglichkeit der Erweiterung der Wissensbasis wirkt diesem Problem entgegen. Der Ansatz der Validierbarkeit übergreifender Parameterabhängigkeiten birgt jedoch ein hohes Potenzial, auch wenn Teilbereiche des Wissens über Echtzeit-fMRT-Experimente nicht abgebildet werden.

5.4. Entwurf

Um die zuvor definierten Anforderungen zu erfüllen, eignet sich eine zentrale Beschreibung der Parameter der in der Problembeschreibung vorgestellten Teilsysteme eines Echtzeit-fMRT-Systems. Abbildung 5.1 verdeutlicht diese Informationsstruktur. Diese Beschreibung repräsentiert Wissen und kann mit Hilfe eines regelbasierten Systems implementiert werden. Dabei erfolgt eine strukturelle Trennung von Aussagen (Propositionen) und einem Regelsystem aus Konditionalsätzen, welche die Propositionen verbinden. Die hier beschriebene Wissensbasis soll allein auf der Aussagenlogik basieren. Es gibt keine Unifikationsmechanismen², wie sie in der Prädikatenlogik verwendet werden. Für die Repräsentation von Propositionen wird die Darstellung der Fakten in Form einer XML-Beschreibung (EDL) der Parameter eines Echtzeit-fMRT-Experimentes gewählt. Dies hat den Vorteil, dass syntaktische und strukturelle Korrektheit sowie Beschränkungen der Parameterbereiche getestet werden können. Für diese Tests stehen zahlreiche hoch entwickelte Applikationen (XML-Parser) zur Verfügung. Mit der Beschreibung von Fakten über EDL ist die Forderung nach syntaktischer und semantischer Validierbarkeit der

²Unifikation = Vereinheitlichung prädikatenlogischer Ausdrücke

Experimentbeschreibung erfüllt.

Die systemübergreifenden Parameterabhängigkeiten sind in Regeln festgelegt, welche mit Algorithmen auf ihre Gültigkeit geprüft werden können. Für die Repräsentation der Konditionalsätze der Regelbasis wird ebenfalls eine XML-Beschreibung (*edlRules*) gewählt. Hiermit wird die syntaktische und strukturelle Form dieser Sätze vorgegeben. Folgende Bedingungen gelten für die Wissensrepräsentation:

1. Es wird kein unsicheres Wissen repräsentiert.
2. Die Prämissen der Konditionalsätze der Regelbasis bestehen nur aus Konjunktionen.
3. Die Konklusionen der Konditionalsätze der Regelbasis bestehen nur aus Literalen.
4. Alle Regeln der Regelbasis sind deterministisch, gelten also in jedem Falle. Es gibt keine Zensoren (Regeln, welche andere Regeln blockieren).
5. Bei der Inferenz kommt ausschließlich der *modus ponens* zur Anwendung (vgl. Abschnitt 5.2).
6. Rekursionen und Widersprüche in Regeln sind nicht zugelassen.

Dieser Ansatz ist systemunabhängig und insofern erweiterbar, als dass Benutzer eigene Parameter in *EDL* als auch eigene Regeln in die Regelbasis einfügen können. Der Aufbau der Beschreibungssprache EDL wird in Abschnitt 5.5.1 vorgestellt. Die syntaktische Formulierung der Regeln (*edlRules*) ist in Abschnitt 5.5.2 beschrieben.

5.5. EDL und EDL-Regeln als Wissensrepräsentation

Die Definition von EDL (in Form von XSD) und die definierten Regeln in der Regelbasis *edlRules* repräsentieren in ihrer Einheit abstraktes Wissen über Experimente. Die Ausprägung eines Experimentes in einem EDL-File stellt somit das evidenzielle, fallbasierte Wissen (Faktenmenge) dar. Im Folgenden werden die Spezifikationen von EDL und *edlRules* beschrieben.

5.5.1. EDL-Spezifikation

Für die Struktur der Parameterbeschreibung in EDL bietet sich die Strukturierung hinsichtlich der Parameterräume der in Abschnitt 4 vorgeschlagenen Trennung der Echtzeit-fMRT-Subsysteme an. Die Strukturierung von EDL ist in Abbildung 5.2 dargestellt. Die

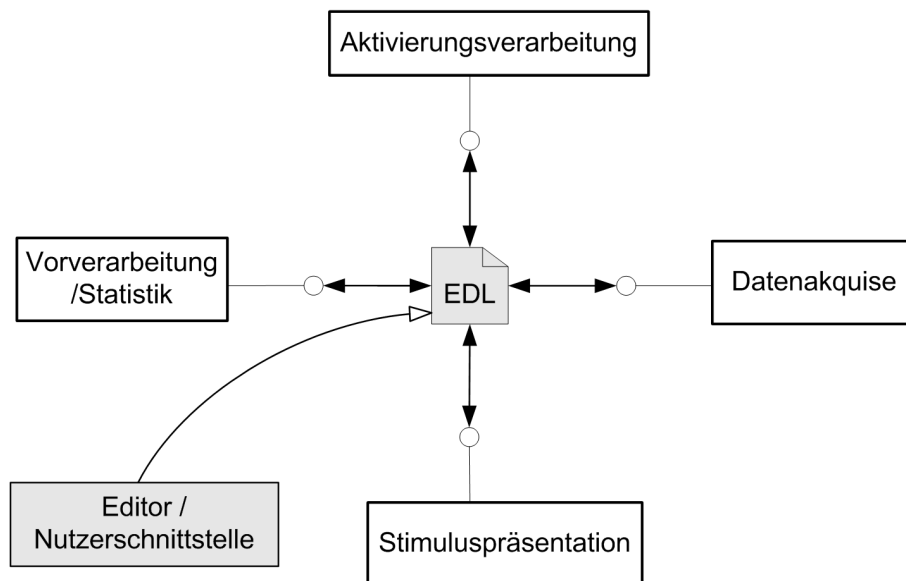


Abbildung 5.1.: EDL verbindet als zentrale Parameterrepräsentation die Teilsysteme einer Echtzeit-fMRT-Implementierung. Jedes Teilsystem ist dabei über Schnittstellen (als Kreise dargestellt) mit der zentralen Beschreibung verbunden.

Spezifikation der Beschreibungssprache erfolgt in den folgenden Absätzen beispielhaft nach den einzelnen Elementen der ersten Hierarchiestufe. Die Definition von EDL erfolgt über eine Schemadefinition (XSD) und basiert auf der XML-Version 1.0 in einer UTF-8 Kodierung. Es werden verschiedene primitive und abgeleitete XSD-Datentypen³, sogenannte „built-in datatypes“, wie z.B. *int*, *double*, *string*, *anyURI*, *id*, *idref* verwendet. Eine Implementierung eigener Datentypen erfolgte nicht.

Informationen zur Systemumgebung: *environment*

Diese Sektion enthält Informationen, die für die Funktionen der funktionellen Auswertung wichtig sind. Hier werden z.B. Angaben zum Empfang von Triggersignalen vom MR-Tomographen gemacht und Informationen zu Loggingfunktionen (z.B. Angabe eines Logfiles) gespeichert. Des Weiteren finden sich hier alle Angaben zur Lokalisation von Daten:

1. Lokalisation der zu verarbeitenden funktionellen Daten
2. Ziellokalisation eventueller Ergebnisdaten

³Eine Übersicht über XSD-Datentypen findet sich in der Spezifikation des Standards unter W3C [2009b].

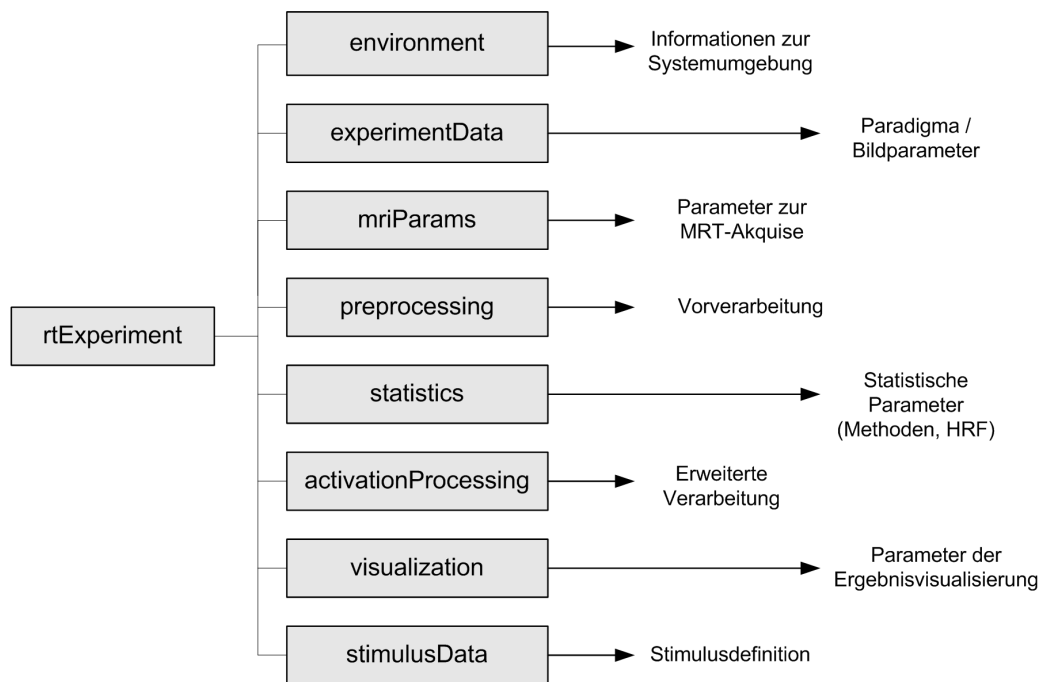


Abbildung 5.2.: Grundstruktur der Beschreibungssprache EDL: Dargestellt sind die Elemente der ersten Hierarchiestufe.

3. Angabe einer Targetlokalisierung in welche die funktionellen Daten automatisch gesichert werden können (optional)

Alle Lokalisationsangaben können als globale oder relative Pfade bzw. Netzwerkadressen angegeben werden, wobei relative Angaben sich auf den Ort beziehen an dem das EDL-File gespeichert ist. Mit Hilfe der in dieser Sektion beschriebenen Informationen ist eine Applikation in der Lage beim Laden eines EDL-Files zu testen, ob alle Lokalisationen existieren und ein Logfile an den definierten Ort geschrieben werden kann. Zu beachten ist, dass es sich bei jeder Lokalisationsangabe in der zugrunde liegenden Schemadefinition um ein Element des XSD-Typs *anyURI*⁴ handelt und diese Angaben somit bestimmten syntaktischen Anforderungen unterliegen.

Informationen zum Paradigma: *experimentData*

Die Unterstruktur *experimentData* beschreibt Informationen, die die Paradigmenstruktur betreffen. Hierzu gehören der Aufbau des Paradigmas und Angaben zu Bilddaten, welche vom System verarbeitet werden. Der folgende Auszug beschreibt ein Sliding-

⁴URI = Uniform Resource Identifier

Window Design mit einem Regressor, der sich aus der fünffachen Wiederholung eines Fensters der konstanten Größe zehn ergibt. Die Anzahl der zu ignorierenden Scans am Beginn des Experimentes wird über das Attribut *ignoreScansAtStart* angegeben:

```
<paradigm ignoreScansAtStart="2">
  <swDesignStruct>
    <scanBasedRegressor regressorID="sbr1" name="l1" size="54" useRefFct="g1">
      <sbrDesign size="10" repetitions="5">
        <scan>0</scan>
        .
        .
        <scan>1</scan>
      </sbrDesign>
      <sbrIgnoreStruct overallIgnores="4">
        <insertIgnores repetitionNr="1" nmbIgnores="0"/>
        .
        .
      </sbrIgnoreStruct>
    </scanBasedRegressor>
  </swDesignStruct>
</paradigm >
```

Mit der Angabe von ignorierten Aufnahmen in den Repetitionen (*sbrIgnoreStruct*) und dem Verweis auf eine Referenzfunktion über das Attribut *useRefFct* wird der erwartete Verlauf der hämodynamischen Antwortfunktion vorgegeben. Innerhalb der Struktur *sbrDesign* steht jeder Scaneintrag 1 für einen aktiven Anteil und 0 dementsprechend für einen inaktiven Anteil im Regressor. Analog zu dieser Definition kann auch ein *Growing-Window Design* angegeben werden. Ebenso ist es möglich die Angabe der Regressoren basierend auf Zeitpunkten zu realisieren. Hierbei werden die Angaben in Millisekunden getätigt. Dies ermöglicht die Implementierung von ereigniskorrelierten Designs. Eine vollständige Instanz des Elementes *paradigm* findet sich im Anhang A.1.1.

Der zweite wichtige Bestandteil in der Sektion *experimentData* besteht aus der Definition der Eigenschaften der zu verarbeitenden Bilddaten. Die nachfolgenden Angaben initiieren die Verarbeitung von Bilddaten im IMA-Bildformat, wie sie von den in Abschnitt 6.3 vorgestellten Methoden während einer Messung exportiert werden.

```
<imageModalities imgDataType="image" imgType="IMA" imgDataExtension=".ima">
  <imgBase>IBMI_fmri_</imgBase>
  <byteOrder>littleEndian</byteOrder>
  <TR>2000</TR>
```

```

<imgMosaic>
  <MimgWidth>384</MimgWidth>
  <MimgHeight>384</MimgHeight>
  <Mslices>31</Mslices>
  <MtileWidth>64</MtileWidth>
  <MtileHeight>64</MtileHeight>
  <Mbpp>uint16</Mbpp>
</imgMosaic>
</imageModalities>

```

Die Echtzeit-Verarbeitung erwartet aufgrund dieser Angaben einen Datensatz in der Aufteilung als MOSAIC⁵ mit 31 Schichten und einer Matrix von 64 x 64 Bildpunkten. Neben dem Bildformat IMA werden von EDL die Formate ANALYZE und DICOM unterstützt.

Parameter der MRT Akquise: *mriParams*

Die Parameter zur Akquise der MRT-Daten der Messung sind in der EDL Sektion *mriParams* abgelegt. Nachfolgend ist beispielhaft ein Auszug dieser Sektion angegeben, wobei alle derzeit integrierten MR-Parameter angegeben sind.

```

<mriParams fieldStrength="3T" vendorName="Siemens" systemVersion="VA25A">
  <MR_TAG_BANDWIDTH vendorTag="bw">1800</MR_TAG_BANDWIDTH>
  <MR_TAG_BASE_RESOLUTION vendorTag="matrix">64</MR_TAG_BASE_RESOLUTION>
  <MR_TAG_PHASE_RESOLUTION vendorTag="ph_res">100</MR_TAG_PHASE_RESOLUTION>
  <MR_TAG_PHASE_FOV vendorTag="phase_fov">100</MR_TAG_PHASE_FOV>
  <MR_TAG_TE vendorTag="te">27</MR_TAG_TE>
  <MR_TAG_TR vendorTag="tr">2000</MR_TAG_TR>
  <MR_TAG_SG_SIZE vendorTag="sg.0.size">31</MR_TAG_SG_SIZE>
  <MR_TAG_SLICE_THICKNESS vendorTag="sl_thick">4.0</MR_TAG_SLICE_THICKNESS>
  <MR_TAG_SG_DISTANCE_FACTOR vendorTag="sg.0.df">0</MR_TAG_SG_DISTANCE_FACTOR>
  <MR_TAG_READOUT_FOV vendorTag="readout_fov">210</MR_TAG_READOUT_FOV>
  <MR_TAG_FLIP_ANGLE vendorTag="angle_array.0">90</MR_TAG_FLIP_ANGLE>
  <MR_TAG_MEASUREMENTS vendorTag="meas">180</MR_TAG_MEASUREMENTS>
</mriParams>

```

Die angegebenen Parameter werden durch die in Abschnitt 6.3 vorgestellte Schnittstelle zum Scanner gelesen und können von dieser auch in der EDL-Beschreibung verändert werden. In den Attributen der Sektion sind die Feldstärke sowie die Bezeichnung des

⁵MOSAIC bedeutet die Anordnung der Schichten eines 3D-Datensatzes in einer 2D-Matrix

Herstellers und die Systemversion als Pflichtfelder definiert. Somit kann ein Verarbeitungsprozess auf diese Informationen zugreifen und eventuell die Verarbeitung anpassen. Für jeden MR-Parameter ist ein sogenanntes *vendorTag* angegeben. Dieses bestimmt unter welchem Namen der entsprechende Parameter im Zielsystem repräsentiert ist. Ist dieses Attribut leer, muss die entsprechende Schnittstelle eine eigene Strategie zur Namensauflösung implementieren. Eine Besonderheit an dieser Sektion ist, dass die Reihenfolge der Elemente eine wichtige Rolle spielt. Die Schnittstelle fügt die Parameter entsprechend ihrer Reihenfolge in das Messprotokoll ein und es ist zum Beispiel sinnvoll erst die Bandbreite festzulegen und dann das TE zu definieren, da die Echozeit sich unter anderem aufgrund der Bandbreite bestimmt und nicht umgekehrt. Alle Elemente in diesem Abschnitt von EDL sind laut Definition auch ohne Inhalt gültig.

Vorverarbeitung: *preprocessing*

In der Sektion *preprocessing* sind Angaben zur Vorverarbeitung der funktionellen Daten abgelegt. In der aktuellen Version können drei standardisierte Vorverarbeitungsverfahren angegeben werden: die räumliche Filterung, eine zeitliche Filterung und ein lineares Detrending (vgl. Abschnitt 2.2.4). Nachfolgend ist eine beispielhafte Definition dieser EDL-Sektion angegeben:

```
<preprocessing>
  <spatialFilter>
    <kernelType>GAUSS</kernelType>
    <kernelSize>6</kernelSize>
  </spatialFilter>
  <linearDetrending></linearDetrending>
</preprocessing>
```

In diesem Beispiel ist ein Gaussfilter mit einer Größe von 6 mm definiert. Die Korrektur linearer Drifts ist allein durch die Anwesenheit des Elements *linearDetrending* gefordert. Dieses Element beinhaltet per Definition keine weiteren Angaben. Eine zeitliche Filterung ist im oben angegebenen Beispiel nicht integriert und wird demnach vom Nutzer nicht in den Vorverarbeitungsprozess eingebunden. Auch in dieser Sektion ist die Reihenfolge der Elemente von Bedeutung, da diese auch der Reihenfolge der Ausführung entspricht.

Statistische Methoden: *statistics*

Welche Methodik mit welchen Parametern für die statistische Verarbeitung der funktionellen Daten verwendet werden soll ist in der EDL-Sektion *statistics* definiert. Hier stehen zwei Verfahren zur Auswahl: die Korrelationsanalyse und der t-Test (Welch-Test).

Im unten stehenden Beispiel ist eine Korrelationsanalyse gewählt worden, wobei die Referenzfunktion über eine doppelte Gamma-Variate Funktion definiert ist. Die angegebene Funktion dient als Modell der HRF. Für beide Anteile der Referenzfunktion (Overshoot und Undershoot) sind jeweils drei Parameter gegeben: die Zeit bis zum Maximum (*tPeak* in Sekunden), die Breite der Funktion auf halber Höhe (*mWidth* in Sekunden) und ein Skalierungsfaktor (*scale*), welcher das Verhältnis der beiden Funktionen zueinander bestimmt:

```
<statistics>
  <corrAnalysis></corrAnalysis>
  <referenceFunctions>
    <dGamma refFctID="g1">
      <tPeak1>5</tPeak1>
      <mWidth1>5.1</mWidth1>
      <scale1>1</scale1>
      <tPeak2>14</tPeak2>
      <mWidth2>7</mWidth2>
      <scale2>0.09</scale2>
      <offset>0</offset>
    </dGamma>
  </referenceFunctions>
</statistics>
```

Es können im Element *referenceFunctions* auch mehrere Referenzfunktionen angegeben werden, um verschiedene Regressoren spezifisch für einen Stimulus modellieren zu können. Für die Korrelationsanalyse ist weiterhin die Angabe einer *HRF-Map* möglich. Dies ist ein Datensatz in dem für jeden einzelnen Voxel eine solche Referenzfunktion angegeben werden kann (vgl. hierzu Abschnitt 6.1.2 und Hollmann et al. [2008a]). Für die Verarbeitung der funktionellen Daten im Experiment werden Parameter dieser Sektion im Zusammenhang mit der Definition des Paradigmas verwendet. Zum Beispiel erfolgt die Auswertung der Daten in der Korrelationsanalyse mit einer Funktion, die sich aus der Faltung der hier angegebenen Referenzfunktion und dem in der Sektion *experimentData* definierten Paradigma ergibt.

Erweiterte Verarbeitung: *activationProcessing*

Für den in dieser Arbeit beschriebenen Ansatz stellt die Definition der erweiterten Verarbeitung in der Sektion *activationProcessing* einen besonders wichtigen Punkt dar. Es wird davon ausgegangen, dass der Nutzer eines Echtzeit-fMRT-Systems neben standardisierten statistischen Auswertungen meist weiterführende komplexe Funktionalität einbinden

möchte. Dieses Vorgehen ist auch bei den in Kapitel 7 implementierten Experimenten gewählt worden. In der Umsetzung wird folgendes Design gewählt: Der Prozess der weiterführenden Verarbeitung wird in verschiedene Abschnitte unterteilt. Jeder Abschnitt ist durch ein Laufzeitmodul repräsentiert. Es wird davon ausgegangen, dass jedes in EDL definierte Modul in einer von der Applikation bestimmten Taktung (z.B. bei jedem neu eingegangenen Datensatz) gerufen wird. Sind mehrere Module angegeben erfolgt der Aufruf in der Reihenfolge der Definition. In der derzeitigen Umsetzung existieren zwei Modultypen:

- *analysisRuntimeModuleBOLD* - extrahiert das BOLD-Signal in einer angegebenen ROI, wobei diese ROI auch über den Verweis auf ein Vorexperiment (wiederum ein EDL-File) definiert werden kann
- *analysisRuntimeModuleUserDefined* - führt vom Nutzer implementierte Funktionalität aus

Nachfolgend ist beispielhaft die Definition eines Modules vom Typ *analysisRuntimeModuleBOLD* dargestellt:

```
<analysisRuntimeModuleBOLD moduleID="aMRT1" useRegressor="sbr1">
  <ROI>
    <preExperiment>
      <targetEDL>..\1_Init_Links\Init_Links.edl</targetEDL>
      <initialScaledTreshold>0.4</initialScaledTreshold>
    </preExperiment>
    <label>links</label>
  </ROI>
  <baselineStart>1</baselineStart>
  <baselineEnd>3</baselineEnd>
  <activeStart>4</activeStart>
  <activeEnd>6</activeEnd>
  <globalTargetVariables>
    <targetVar tVarID="tvBoldW1" tVarName="aMRT1_BoldLevelWeighted">
    <targetVar tVarID="tvBoldN1" tVarName="aMRT1_BoldLevelNative">
  </globalTargetVariables>
  <doOutput useOutputModule="om1">
    <outputAtSWSscan>6</outputAtSWSscan>
    <outPutVariables useVarIDS="tvBoldW1"/>
  </doOutput>
</analysisRuntimeModuleBOLD>
```

Unter dem Abschnitt *ROI* ist in dieser Ausprägung des Modules der Verweis auf ein EDL-File und somit auf ein Vorexperiment gemacht worden. Dieses kann zum Beispiel ein so genannter „functional localizer“ sein, wie er bereits im Abschnitt 3.2 angesprochen wurde. Der Bezug zum Paradigma wird über die Referenz zum Regressor über das Attribut *useRegressor* hergestellt. In jedem Modul können Variablen definiert werden, die zur Laufzeit beschrieben und global zur Verfügung gestellt werden. Somit sind diese auch für andere Module oder für Kommunikationsprozesse erreichbar. In diesem Beispiel wird eine der beiden erstellten Variablen (*tvBoldW1*) über ein definiertes Ausgabemodul im sechsten Scan des Sliding-Window Designs versendet.

Der Modultyp *analysisRuntimeModuleUserDefined* kann genau wie ein BOLD-Modul die Elemente *globalTargetVariables* und *doOutput* implementieren. In einem nutzerdefinierten Modul werden neben Initialisierungsvariablen ein Verweis (Handle) auf eine Funktion zur Initialisierung des Moduls und ein Verweis auf eine Laufzeitfunktion angegeben:

```
<analysisRuntimeModuleUserDefined moduleID="aM_UG1" useRegressor="sbr1">
  <functionHandlePrepare>initUDAnalysisModule_UG</functionHandlePrepare>
  <functionHandleRun>runUDAnalysisModule_UG</functionHandleRun>
  <prepareParamsUD>
    <param>
      <paramName>UG_nmbTrials</paramName>
      <paramValue>60</paramValue>
      <paramType>INT</paramType>
    </param>
    .
    .
  </prepareParamsUD>
</analysisRuntimeModuleUserDefined>
```

In Abschnitt 6.1.2 wird die Verwendung einer solchen Definition näher beleuchtet.

Zu der hier beschriebenen Sektion gehören auch Kommunikationsmodule, die den Austausch von Informationen beschreiben. Laufzeitmodule können auf diese Funktionalität über den Verweis auf ein Element vom Typ *output*- bzw. *receiveModule* zugreifen. Ein Element zur Ausgabe von Informationen ist beispielhaft mit der folgenden EDL-Definition gegeben:

```
<outputModuleTCPIP oModuleID="om1">
  <targetIP>192.68.2.4</targetIP>
  <targetPort>16666</targetPort>
  <waitForAcknowledge>1</waitForAcknowledge>
</outputModuleTCPIP>
```

Neben der Kommunikation über TCP/IP können Informationen auch in ein ASCII-File geschrieben bzw. aus einem ASCII-File gelesen werden.

Informationen zur Ergebnisvisualisierung: *viewProperties*

Diese Sektion wurde in die Beschreibungssprache aufgenommen, um die Visualisierung der statistischen Ergebnisse zu steuern. Hier kann ein anatomischer Datensatz angegeben werden, auf welchen Ergebnisse der Auswertung projiziert werden. In der im Rahmen dieser Arbeit implementierten Applikation ist diese Projektion nicht umgesetzt und die Visualisierung erfolgt mit Hilfe der funktionellen Daten.

Informationen zur Stimuluspräsentation: *stimulusData*

Die Stimuluspräsentation in einem Experiment wird mit Hilfe der EDL-Sektion *stimulusData* definiert. Diese Sektion teilt sich wiederum in drei Abschnitte:

1. *stimEnvironment* - Parameter zu Triggersignalen, Logging und Präsentationsmedien (z.B. Auflösung des Präsentationsbildschirms etc.)
2. *mediaObjectList* - Eine Liste aller Medienobjekte, welche dargestellt werden sollen (Bilder, Text, Audioobjekte etc.)
3. *timeTable* - Die millisekundengenaue Angabe des Ablaufes der Präsentation mit Referenzen auf die Medienobjekte

Die verfügbaren Medienobjekte können in der derzeitigen Fassung fünf Typen ausprägen: *TEXT*, *IMAGE*, *SOUND*, *COM* und *COMPLEX*. Nachfolgend ist ein Beispiel für die Definition eines Textobjektes angegeben:

```
<mediaObject moID="1" name="text1" type="TEXT">
  <contentText>
    <text>+</text>
    <tColor>
      <tcRed>250</tcRed>
      <tcGreen>250</tcGreen>
      <tcBlue>250</tcBlue>
    </tColor>
    <tSize>40</tSize>
    <posX>150</posX>
    <posY>150</posY>
  </contentText>
</mediaObject>
```


Dieses Objekt stellt ein Pluszeichen auf dem Bildschirm an der angegebenen Position und in der definierten Größe und Farbe dar. Jedes Objekt ist über seine Attribute mit Informationen zur Identifikation und seinem nativen Typ versehen. Der Objekttyp *COM* steht für ein Kommunikationsobjekt, also den Empfang oder das Versenden von Variablen zur Laufzeit, während über den Typ *COMPLEX* umfangreichere Funktionalität wie z.B. dynamische Prozesse eingebunden werden können. Dabei wird für *COMPLEX*-Objekte eine Schnittstelle angegeben, welche zur Initialisierung bzw. zur Laufzeit vom System gerufen werden soll. Der folgende Auszug aus der Definition eines komplexen Medienobjektes gehört zu der für die praktischen Experimente (siehe Kapitel 7) verwendeten Stimuluspräsentation. Eine vollständige Angabe des EDL-Codes findet sich in Anhang A.1.2.

```
<mediaObject moID="2" name="ug1" type="COMPLEX">
  <contentComplex type="ULTIMATUMGAME" overallPresLength="22000">
    <functionHandlePrepare>predef_prepareObjects_UG</functionHandlePrepare>
    <functionHandleRun>predef_runtObjects_UG</functionHandleRun>
    <stimPrepareParamsCOMPLEX>
      .
      .
    </stimPrepareParamsCOMPLEX>
    <stimRunParamsCOMPLEX>
      .
      .
    </stimRunParamsCOMPLEX>
  </contentComplex>
</mediaObject>
```

Die Verwendung dieser Objekte dient für Benutzer zur Definition selbst implementierter Funktionalitäten im Stimulussystem. Dies ist auch der Grund, warum die Parameter zur Initialisierung und Laufzeitparameter (*stimPrepareParamsCOMPLEX* und *stimRunParamsCOMPLEX*) mit Namen und Typ angegeben werden müssen. So erhält der Nutzer die Möglichkeit eine unbegrenzte Anzahl an eigenen Parametern zu definieren, welche nicht in der zugrunde liegenden XSD-Definition für EDL enthalten sind. In der Unterstruktur *timeTable* (vgl. Anhang A.1.3) wird der genaue Ablauf der Präsentation festgelegt. Dies beinhaltet den Zeitpunkt und die Dauer der Präsentation eines Medienobjektes. Des Weiteren kann eine Zeittoleranz in Millisekunden angegeben werden. Dieser Toleranzwert dient der Stimulusapplikation zur Bestimmung des Abbruchkriteriums, falls der Ablauf der *timeTable* nicht eingehalten wird. Als Teil der *timeTable* ist außerdem ein Element namens *outro* vorgesehen, das ebenfalls ein Medienobjekt beinhal-

tet. Es wird verwendet, um den Probanden auf das Ende der Präsentation vorzubereiten (z.B. über die Anweisung die Augen zu schließen).

5.5.2. Beschreibung der EDL-Regelbasis

Die im vorhergehenden Abschnitt beschriebene Parameterrepräsentation über EDL ist die Basis für die automatisierte Auswertung der Parameterzusammenhänge über Algorithmen. Mit Hilfe von wiederum extern definierten Regeln muss ein solcher Algorithmus im Quellcode nicht einmal über explizite Informationen zu den Parameterzusammenhängen verfügen. Die unabhängige Repräsentation der Regelbasis erfüllt die Forderung der leichten Erweiterbarkeit, da ein Nutzer Regeln (und damit Wissen) hinzufügen kann, ohne den Quellcode der Software ändern zu müssen, welche die Tests der Abhängigkeiten durchführt.

Die Regeln zu den in EDL beschriebenen Fakten sind in einem XML-File namens *edl-Rules.xml* beschrieben, welches wie ein EDL-File selbst eine Ausprägung einer XSD-Definition darstellt. Jede Regel folgt dem folgendem strukturellen Aufbau:

```
<rule ruleID="...">
  <param pID="..."> ... </param>
  .
  .
  <premise>
    <literal> ... </literal>
    .
    .
  </premise>
  <conclusion>
    <literal> ... </literal>
  </conclusion>
  <message type="..."> ... </message>
</rule>
```

Alle Regeln sind mit einem Identifier versehen (*ruleID*). Der Verweis auf die einzelnen Elemente von EDL erfolgt mit Hilfe der Inhalte der Elemente *param*, wobei eine Regel eine unbegrenzte Anzahl von Parametern beinhalten kann. Der eindeutige Verweis auf ein EDL-Element erfolgt über die Struktur dieser Beschreibung. Der folgende Inhalt eines *param*-Elementes verweist demnach auf das in der EDL-Beschreibung der MR-Akquiseparameter definierte TR:

```
<param pID="r20A">mriParams{1}.MR_TAG_TR{1}.CONTENT</param>
```

Die Phrase *CONTENT* zeigt dabei auf den Dateninhalt eines EDL-Elementes, wobei Attribute über die Phrase *ATTRIBUTE* angesprochen werden. Der so erzeugte Parameter ist dann über den Identifier *pID* eindeutig identifiziert. Mit der Struktur *premise* wird die Prämisse einer Regel festgelegt. Gemäß dem Entwurf werden in den Prämissen der hier definierten Regeln nur Konjunktionen (UND-Verknüpfungen) von Literalen zugelassen. Somit ist es ausreichend in einer Prämisse die Literale über Elemente vom Typ *literal* anzugeben. Literale sind über Textinhalt repräsentiert, der Verweise auf die Parameter über deren *pID* enthält:

```
<literal>(r20A >= r20B)</literal>
```

Eine Regel mit dem oben definierten Literal in der Prämisse kommt nur zur Anwendung wenn der Parameter mit dem Verweis *r20A* größer oder gleich dem Parameter *r20B* ist. In Literalen können neben den Werten *true*, *false* und der Negierung NOT() auch die Vergleichsoperatoren ==, >= und <= und die mathematischen Operatoren +, -, * und / definiert werden. Des Weiteren sind zwei Funktionen zugelassen: *edlValidation_exists()* und *edlValidation_strcmp()*. Erstere testet die Existenz eines Elementes und letztere vergleicht Elemente vom Typ String (beide Funktionen liefern Werte vom Typ Boolean). Um den Prinzipien der klassischen Aussagenlogik zu folgen, müssen Literale in EDL-Regeln zwingend als Boolean-Werte (*true* oder *false*) aufgelöst werden können. Die Konklusion einer Regel erfolgt wiederum in einem Element vom Typ *literal*, welches vom Element *conclusion* gekapselt ist. Dem Entwurf der Regelbasis folgend fungiert nur ein einziges Literal als Konklusion einer Regel. Dies ist durch die XSD-Definition der Regelbasis sichergestellt. Alle Regeln müssen, wenn sie zur Anwendung kommen auch erfüllt sein. Das bedeutet, das Literal der Konklusion muss den Wert *true* liefern. Über das Element *message* wird eine Warnung oder ein Fehlertext bestimmt, welcher von einer Applikation verwendet werden kann, um dem Nutzer die passenden Hinweise zu geben wenn eine Konklusion nicht als wahr terminiert. Der Typ des Fehlers wird dabei über das Attribut *type* der Nachricht bestimmt. Hier sind die Werte *WARNING* und *ERROR* zugelassen. Derzeit sind in der Regelbasis zu EDL 87 Regeln definiert. Vollständige Definitionen von Regeln finden sich beispielhaft im folgenden Abschnitt. Dort sind die Regeln aufgeführt, welche direkt die Problemstellung des Beispiexperimentes betreffen.

5.6. Anwendung auf das Beispiexperiment

In diesem Abschnitt wird auf das in Abschnitt 4.2 eingeführte Beispiexperiment eingegangen und beschrieben, wie die für dieses Experiment nötigen Parameter in EDL definiert sind. Des Weiteren werden die Regeln vorgestellt, welche im Zusammenhang

mit der EDL-Instanz dieses Experimentes die Inkonsistenzen auflösen. Das in Abschnitt 4.2 vorgestellte Paradigma kann in EDL mit folgender Paradigmenstruktur repräsentiert werden:

```
<paradigm ignoreScansAtStart="6">
<swDesignStruct>
  <scanBasedRegressor regressorID="sbr1" name="t11" size="35" useRefFct="g1">
    <sbrDesign size="8" repetitions="4">
      <scan>1</scan>
      <scan>0</scan>
      <scan>0</scan>
      <scan>0</scan>
      <scan>0</scan>
      <scan>0</scan>
      <scan>0</scan>
      <scan>0</scan>
      <scan>0</scan>
    </sbrDesign>
    <sbrIgnoreStruct overallIgnores="3">
      <insertIgnores repetitionNr="1" nmbIgnores="0"/>
      <insertIgnores repetitionNr="2" nmbIgnores="2"/>
      <insertIgnores repetitionNr="3" nmbIgnores="1"/>
      <insertIgnores repetitionNr="4" nmbIgnores="0"/>
    </sbrIgnoreStruct>
  </scanBasedRegressor>
</swDesignStruct>
</paradigm>
```

Gemäß der EDL-Definition der zu verarbeitenden Bilddaten, wird für das Beispiexperiment die folgende Struktur definiert:

```
<imageModalities imgDataType="image" imgType="IMA" imgDataExtension=".ima">
  <imgBase>IBMI_FMRI_</imgBase>
  <byteOrder>littleEndian</byteOrder>
  <TR>1500</TR>
  .
  .
</imageModalities>
```

Die Definition des Elementes *TR* mit 1500 Millisekunden ist essenziell für die Konsistenzprüfung. Die Stimuluspräsentation wird über das im folgenden Auszug dargestellte *timeTable*-Element gesteuert. Die Medienobjekte mit den Identifikatoren 1, 2 und 3

stellen jeweils ein Fixationskreuz, den Checkerboard-Stimulus und einen Text mit der Anweisung „Bitte die Augen schließen!“ dar.

```
<timeTable timeTolerance="20">
  <freeStimulusDesign overallPresLength="52500">
    <stimEvent time="0" duration="1500">
      <mObjectID>2</mObjectID>
    </stimEvent>
    <stimEvent time="1500" duration="10500">
      <mObjectID>1</mObjectID>
    </stimEvent>
    <stimEvent time="12000" duration="1500">
      <mObjectID>2</mObjectID>
    </stimEvent>
    .
    .
    <stimEvent time="42000" duration="10500">
      <mObjectID>1</mObjectID>
    </stimEvent>
  </freeStimulusDesign>
  <outro>
    <stimEvent time="0" duration="4000">
      <mObjectID>3</mObjectID>
    </stimEvent>
  </outro>
</timeTable>
```

Gemäß der Angaben aus der Problemstellung sind in diesem Stimulusdesign die ersten sechs ignorierten Scans nicht enthalten. Ebenso erfolgt die Angabe der Anzahl der Scans über die Sektion *mriParams* nur mit 35 statt der nötigen 41 Scans:

```
<mriParams fieldStrength="3T" vendorName="Siemens" systemVersion="VA25A">
  .
  .
  <MR_TAG_MEASUREMENTS vendorTag="meas">35</MR_TAG_MEASUREMENTS>
</mriParams>
```

Der angeführte EDL-Code führt zu einer wohlgeformten und validen Beschreibung laut der XSD-Definition von EDL, obwohl für das verwendete Paradigma eine zu kurze Stimuluspräsentation und die falsche Anzahl an Scans definiert wird.

Über die im Folgenden aufgeführten Regeln der EDL-Regelbasis wird nun ein Zusammenhang zwischen dem Paradigmen-Design, der Stimuluspräsentation und dem Parameter der MRT-Akquise hergestellt. Aus Platzgründen sind in beiden hier angegebenen Regeln gegenüber dem Originalquelltext Textstellen ausgelassen. Diese sind mit „...“ gekennzeichnet.

Die folgende Regel bestimmt das Verhältnis zwischen der im Paradigma definierten Scananzahl (Parameter $r12A$ und $r12B$), der Repetitionszeit (Parameter $r12D$) und der Länge der Stimuluspräsentation (Parameter $r12C$):

```
<rule ruleID="r112">
  <param pID="r12A">
    experimentData{1} ... scanBasedRegressor{1}.ATTRIBUTE.size
  </param>
  <param pID="r12B">
    experimentData{1}.paradigm{1}.ATTRIBUTE.ignoreScansAtStart
  </param>
  <param pID="r12C">
    stimulusData{1} ... ATTRIBUTE.overallPresLength
  </param>
  <param pID="r12D">
    experimentData{1}.imageModalities{1}.TR{1}.CONTENT
  </param>
  <premise>
    <literal>edlValidation_exists(r12A)</literal>
    <literal>edlValidation_exists(r12C)</literal>
  </premise>
  <conclusion>
    <literal>
      (r12C >= ((r12A + r12B) * r12D))
    </literal>
  </conclusion>
  <message type="WARNING">
    EDL Warning: Stimuluspresentation length is lower ...
  </message>
</rule>
```

Diese Regel kommt zur Anwendung, wenn die Parameter $r12A$ und $r12C$ existieren und bestimmt in der Konklusion, dass die Stimuluspräsentation mindestens so lange andauert wie die Akquise einschließlich ignorierte Anfangsscans. Die Regel erzwingt allerdings

nicht den Abbruch, sondern führt zu der Warnmeldung: *EDL Warning: Stimuluspresentation length is lower than overall scantime in paradigm. Check design?*

Die nachfolgende Regel ist restriktiver und führt bei Verletzung zu einem Fehler. Diese Regel bestimmt, dass die Anzahl der Scans in der Sektion *mriParameter* (Parameter *r15A*), mit der Gesamtanzahl der im Paradigma angegebenen Scans (Parameter *r15B* und *r15C*) übereinstimmen muss:

```
<rule ruleID="r115">
  <param pID="r15A">
    mriParams{1}.MR_TAG_MEASUREMENTS{1}.CONTENT
  </param>
  <param pID="r15B">
    experimentData{1} ... scanBasedRegressor{1}.ATTRIBUTE.size
  </param>
  <param pID="r15C">
    experimentData{1}.paradigm{1}.ATTRIBUTE.ignoreScansAtStart
  </param>
  <premise>
    <literal>edlValidation_exists(r15A)</literal>
    <literal>edlValidation_exists(r15B)</literal>
  </premise>
  <conclusion>
    <literal>
      (r15A == (r15B + r15C))
    </literal>
  </conclusion>
  <message type="ERROR">
    EDL Error: Number of scans ...
  </message>
</rule>
```

Diese Regel empfiehlt die Fehlermeldung: *EDL Error: Number of scans in paradigm is NOT equal to number of measurements in mriParams (MR_TAG_MEASUREMENTS). Experiment is not valid!* Mit den vorgestellten Repräsentationen der Informationen zum experimentellen Paradigma, der Stimuluspräsentation und den Akquiseparametern ist es möglich algorithmisch die korrekte Definition der Parameter zu testen. Mit Hilfe der zwei vorgestellten Regeln ist genug Wissen repräsentiert um die Diskrepanz zwischen Paradigma, Stimulusdesign und MR-Parametern zu finden. Beide Regeln werden nach der hier angegebenen Definition verletzt. In welchen Softwarekomponenten die nötigen

Tests zur Identifikation der Fehler praktisch umgesetzt sind, wird in Abschnitt 6.5 näher beleuchtet.

5.7. Diskussion

Die Idee, Informationen strukturiert und maschinenlesbar darzustellen, ist in der Informationsverarbeitung etabliert. Gerade die Metasprache XML eignet sich, um wiederum Standards zur systemunabhängigen Beschreibung von Informationen zu definieren. Es gibt eine Vielzahl von Beschreibungssprachen aus den verschiedensten Wissenschaftsgebieten. Im Bereich der Compilerentwicklung wurde z.B. bereits 1979 die Sprache MDL (Machine Description Language) zur Beschreibung von Rechnerarchitekturen eingeführt (Boulton u. Goguen [1979]). Um geografische Daten zu beschreiben, wurde die Sprache GML (Geography Markup Language) (OGC [2009]) entwickelt. Viele weitere XML-basierte Ansätze finden sich unter den Empfehlungen des World Wide Web Consortiums (W3C [2009a]).

Wissensbasen sind typischerweise die Grundlage von Expertensystemen. Zu bekannten regelbasierten Expertensystemen zählt z.B. MYCIN (Beierle u. Kern-Isberner [2008]; Buchanan u. Shortliffe [1984]), welches an der Stanford University in der Programmiersprache LISP entwickelt wurde. Es wird zur Prüfung von Verfahren zur Diagnose und Therapie von Infektionskrankheiten durch Antibiotika eingesetzt. Für die Versuchsplanung molekulargenetischer Experimente wurde ein System namens MOLGEN (Grüner et al. [2009]) entwickelt und viele weitere Beispiele finden sich in den verschiedensten Anwendungsbereichen wie Militär, Finanzplanung und Geologie (Beierle u. Kern-Isberner [2008]; Brachman u. Levesque [2004]). Bei den meisten Expertensystemen sind die Wissensbasen nicht extern repräsentiert, sondern bestehen als interne Konstrukte des Expertensystems. Mit XML-basierten Beschreibungen hat man die Möglichkeit die Informationen in einer solchen Wissensbasis gleichzeitig für den unkomplizierten Austausch zwischen Systemen bereitzustellen.

Die hier vorgestellte Wissensbasis kann nach Altenkrüger u. Büttner [1992] als logikbasierte Wissensbasis charakterisiert werden. Zunächst muss die vorliegende Wissensbasis zu herkömmlichen Systemen der logischen Wissensrepräsentation abgegrenzt werden. Während die Grundidee ist, dass neue Fakten ohne Einschränkung aufgenommen werden können, wenn sie anderen Fakten nicht widersprechen, gilt diese Annahme bei diesem Ansatz nicht. Ein Nutzer kann z.B. nicht ohne Weiteres die Aussage „ein Rad ist rund“ integrieren, da der Faktenbereich durch die XSD-Beschreibung von EDL als Struktur vorgegeben ist. Um diese Aussage aufzunehmen, müsste in EDL das Konzept *Rad* mit

der Eigenschaft *rund* aufgenommen werden. Die Integration eines Konzeptes ist nicht trivial, da in der Strukturierung der Elemente in EDL bereits Wissen über den Aufbau von Experimenten enthalten ist. Dieser Punkt spielt bei den weiteren Betrachtungen eine wichtige Rolle.

Als Eigenschaften von Wissensrepräsentationen ergeben sich nach Altenkrüger u. Büttner [1992] neben der *Ausdrucksstärke* die *akquisitorische Effizienz* (also die Frage nach Integrationsmöglichkeiten für neues Wissen) und die *inferentielle Effizienz*. Letzterer Punkt beinhaltet die Betrachtung des rechnerischen Aufwandes für Inferenzprozesse in einem solchen System.

Ausdrucksstärke

Zunächst soll eine Diskussion der Ausdrucksstärke der Wissensbasis erfolgen. Die zentrale Frage ist dabei, welches Wissen man ausdrücken kann und welches nicht. In den vorherigen Abschnitten wurde gezeigt, dass man strukturelle Zusammenhänge und Abhängigkeiten mit dem Ansatz gut darstellen kann. Diese Darstellung ist jedoch eingeschränkt hinsichtlich der Komplexität der Zusammenhänge. Wenn man z.B. komplexe physikalische Abhängigkeiten (wie sie auch zwischen Parametern der MR-Akquise herrschen) abbilden will, müsste man umfangreiche mathematische und algorithmische Methoden in die Literale der Regelbasis aufnehmen. Dies würde zu einer Komplexität führen, die der Umsetzung in Quellcode einer Programmiersprache gleicht, was der Forderung nach der leichten Erweiterbarkeit der Wissensbasis widersprechen würde. Ebenso sind applikationsspezifische Zusammenhänge in EDL nicht abgebildet, da die Beschreibung unabhängig vom Zielsystem sein soll. In der praktischen Umsetzung von Systemen, welche EDL und EDL-Rules verwenden, kann die herkömmliche algorithmische Prüfung spezieller Zusammenhänge demnach nicht vermieden werden.

Akquisitorische Effizienz

Bei der Betrachtung der akquisitorische Effizienz ist zu bestimmen, wie leicht ein Nutzer eigene Informationen integrieren kann. Die Wissensdarstellung mit EDL und `edlRules` ist rein deklarativ-propositionsorientiert, es werden keine Produktionen sondern nur unabhängige Aussagen verarbeitet. Die Struktur der Wissensbasis ist modular. Dies vereinfacht die Aufnahme neuer Parameter und Regeln in die vorhandene Struktur. Wie im Abschnitt 5.5.1 dargestellt wurde, gibt es bereits Elemente in EDL mit denen ein Nutzer eigene Informationen in EDL aufnehmen kann (z.B. in der Sektion *activationProcessing*). In diesen Elementen erfordert die Erweiterung keine Veränderung der XSD-Definition von EDL und es ist trotzdem möglich in selbst definierten Regeln auf eingeführte Parameter

zu verweisen. Eine zweite Möglichkeit der Aufnahme von neuem Wissen bietet die Erweiterung von EDL selbst. Dazu muss die zugrunde liegende XSD-Definition verändert bzw. erweitert werden. Aufgrund der bereits erwähnten strukturellen Zusammenhänge in EDL kann diese Aufgabe eine hohe Komplexität erreichen. Für das Hinzufügen von Regeln besteht nicht die Notwendigkeit die XSD-Definition von `edlRules` zu bearbeiten. Es ist somit vereinfacht neue Regeln hinzuzufügen. Wichtig ist, dass neue Regeln nicht bereits vorhandenen Regeln widersprechen und keine Rekursionen definieren. Der Test auf die Erfüllung dieser Forderungen kann algorithmisch gelöst werden.

Inferentielle Effizienz

Um die inferentielle Effizienz des Systems zu beschreiben, sind zunächst zwei Szenarien zu unterscheiden. Zum einen erfolgt der Test der Regeln wenn ein vollständig definiertes EDL-File vorliegt. In diesem Fall wird mit Hilfe der Vorwärtsverkettung (vgl. Abschnitt 5.2) geprüft, ob die Anforderungen der Regeln erfüllt sind. Sollte eine Parameterbelegung noch nicht vollständig sein, können mit Hilfe dieses Inferenzverfahrens fehlende Belegungen ergänzt werden. In Abschnitt 6.1.2 wird der Algorithmus skizziert, welcher für die datengetriebene Inferenz bei der Anwendung der Regeln auf die EDL-Faktenmenge verwendet wird. Das zweite Szenario ist die gezielte Anfrage auf ein bestimmtes Faktum, wofür die zielorientierte Inferenz zur Anwendung kommt. Diese kann z.B. bei der Eingabe von Parametern für den laufenden Test auf Gültigkeit der Parameterkonstellation verwendet werden. Nähere Betrachtungen hierzu sind in der Beschreibung eines Editors für EDL in Abschnitt 6.4 zu finden.

Für beide Testszenarien gilt, dass der Aufwand für Inferenzverfahren mit steigender Anzahl von Regeln und Fakten ansteigt. Das Kriterium der Anzahl an benötigten Regeln ist hierbei ein Merkmal für die inferentielle Effizienz. Bei dem vorliegenden System werden aufgrund der Übersichtlichkeit und Nutzerfreundlichkeit nur Konjunktionen in Prämissen zugelassen. Mit dieser Forderung steigt die Anzahl der Regeln, da in diesem Fall jede ODER-Verknüpfung von n Literalen in n Regeln mündet (Altenkrüger u. Büttner [1992]). Gemäß der Anforderungsanalyse ist der *modus tollens* nicht als Inferenzverfahren vorgesehen. Dies vereinfacht die Verarbeitung und erhöht die Übersichtlichkeit, bedingt aber auch die Definition zusätzlicher Regeln (vgl. Abschnitt 5.2). Über die Einführung von Zensoren, also sich gegenseitig blockierende Regeln, kann die inferentielle Effizienz gesteigert werden, da unnötige Tests wegfallen. Zensoren beeinflussen wiederum die Übersichtlichkeit negativ und wurden deshalb nicht umgesetzt, hier wird jedoch noch Potenzial für zukünftige Entwicklungen gesehen.

Einige allgemeine Bemerkungen schließen diesen Abschnitt ab. Es wurde bereits im Abschnitt 5.5.1 bemerkt, dass Informationen in EDL redundant abgelegt sein können.

Folgendes Beispiel verdeutlicht noch einmal den Sachverhalt:

```
<paradigm ignoreScansAtStart="0">
  <swDesignStruct>
    <scanBasedRegressor regressorID="sbr1" name="l1" size="50" useRefFct="g1">
      <sbrDesign size="10" repetitions="5">
        .
      </sbrDesign>
    </scanBasedRegressor>
  </swDesignStruct>
</paradigm >
```

Man kann in diesem Beispiel die Gesamtanzahl der Scans in einem Sliding-Window Design (Designgröße: *scanBasedRegressor.size*) auch aus der Größe des Sliding-Window (*sbrDesign.size*) und der Anzahl der Wiederholungen (*sbrDesign.repetitions*) errechnen. In diesem Fall muss jede Regel, welche auf die Information der Designgröße zugreift diese Berechnung in Literalen ausdrücken, was den Inferenzaufwand erhöht. Wenn eine Variable oft benötigt wird, lohnt es sich daher redundante Informationen einzuführen um Rechenzeit zu sparen und die Komplexität der Regeln zu reduzieren.

In der derzeitigen Implementierung der Regelbasis erfolgt die Angabe der Literale in Form von Dateninhalt vom Typ Text. Jedes Literal muss also gelesen werden und hinsichtlich des Inhaltes interpretiert werden. Hinsichtlich der Operatoren wurde eine Syntax gewählt, die bei herkömmlichen Programmiersprachen direkt verwendet werden kann. Für komplexere Ausdrücke bleibt dieses Problem jedoch bestehen. Es bietet sich an die Bestandteile der Literale aufzuspalten (in dem man z.B. Operatoren trennt und in eigene Elemente kapselt) und somit die Verarbeitung des Ausdrucks zu vereinfachen.

EDL kann keine allumfassende Repräsentation für alle denkbaren Experimentszenarien darstellen. Es wird immer Aspekte geben, welche mit einem solchen Ansatz nicht beschrieben werden können. Einige Parameterbereiche eignen sich jedoch besonders gut für die Beschreibung mit EDL. Zum Beispiel könnte man die Sektion der MR-Akquise am DICOM-Standard ausrichten und die dort verwendeten Parameter integrieren. Ebenso sind Informationen zur Vorverarbeitung und probandenspezifische Informationen noch unterrepräsentiert und sollten bei der künftigen Entwicklung näher betrachtet werden. Durch die flexible Architektur von EDL und *edlRules* sind zukünftige Erweiterungen jedoch mit geringem Aufwand ausführbar. Besonders an letztgenannten Punkten zeigt sich die Eignung von EDL zum Austausch von Experimentinformationen und die Nutzbarkeit für automatisierte Dokumentationsmechanismen.

Zusammenfassend ist festzustellen, dass EDL eine zentrale Wissensrepräsentation darstellt, welche in der Lage ist die Parameterräume der Teilsysteme einer Echtzeit-fMRT-

Implementierung zu verbinden. Die Repräsentation ist anhand von Regeln und Strukturbeschreibungen verifizierbar, es werden somit die in Abschnitt 5.3 definierten Anforderungen erfüllt.

6. Eine Softwareumgebung für die Echtzeit-fMRT auf Basis von EDL

Der Ansatz einer Wissensrepräsentation zur Experimentbeschreibung ist nur sinnvoll, wenn es auch Applikationen gibt, die diese Beschreibung verarbeiten können. In dem vorliegenden Kapitel soll die im Rahmen dieser Arbeit entwickelte Softwareumgebung vorgestellt werden. Die dargestellten Softwareanwendungen folgen in ihrer Funktionalität den in der Problemstellung beschriebenen Aufgaben des Gesamtsystems: Datenakquisition, Vorverarbeitung/Statistische Auswertung, Aktivierungsverarbeitung und Stimuluspräsentation. Dies bedeutet, dass für diese Teilaufgaben einzelne Applikationen entwickelt wurden.

Den überwiegenden Anteil dieses Kapitels wird die Beschreibung der Applikation zur Verarbeitung der Echtzeit-fMRT-Daten in Abschnitt 6.1 einnehmen, da diese Applikation die umfangreichste Funktionalität bietet. Diese Funktionalität umfasst die Punkte Vorverarbeitung/Statistik und Aktivierungsverarbeitung. Die Softwarelösungen zur Stimuluspräsentation und eine Schnittstelle zur Datenaquise auf Basis von EDL werden in Abschnitt 6.2 bzw. 6.3 beschrieben. Allen vorgestellten Anwendungen ist gemein, dass sie EDL-Beschreibungen verarbeiten können und somit zum Bestandteil des EDL-Frameworks werden, obwohl sie für sich betrachtet unabhängig sind. In Abschnitt 6.4 wird ein Editor für EDL-Beschreibungen vorgestellt und in Abschnitt 6.5 der Bezug zum laufenden Beispiexperiment hergestellt. Eine Studie zur Nutzbarkeit der Softwareumgebung in Abschnitt 6.6 und die anschließende Diskussion runden dieses Kapitel ab.

6.1. Eine Echtzeit-fMRT-Applikation auf Basis einer Experimentbeschreibung mit EDL: rtExplorer

Im Folgenden werden die Anforderungsanalyse, der Entwurf und die Implementierung der Applikation zur Auswertung von Echtzeit-fMRT-Daten beschrieben. Eine Untersuchung zu Leistungsmerkmalen der Anwendung wird in Abschnitt 6.1.3 vorgestellt.

6.1.1. Anforderungen und Entwurf

Für den Anwender stehen bei einer Applikation für die Echtzeit-fMRT folgende allgemeine Aufgaben im Vordergrund:

- Die Vorverarbeitung der Daten
- Die statistische Auswertung
- Die Visualisierung und Speicherung der Ergebnisse

Aufgrund der vorliegenden Experimentbeschreibung auf Basis von EDL und der damit geschaffenen Möglichkeiten die Konsistenz des Gesamtsystems zu prüfen, ergeben sich weitere Aufgaben der Applikation:

- Die Verarbeitung von EDL-Beschreibungen sowie die Steuerung interner Routinen auf Basis dieser Beschreibungen (hierzu gehören auch Routinen zur Prüfung der Gültigkeit einer Beschreibung)
- Die Kommunikation mit anderen Systemen
- Die Möglichkeit der Simulation der Datenverarbeitung aufgrund einer EDL-Beschreibung

Die Echtzeit-fMRT erlaubt vielfältige Anwendungen und die fortschreitende methodische Entwicklung der Systeme ist beträchtlich. Zu dieser Entwicklung zählt z.B. auch die Einbindung von musterbasierten Klassifikationsmethoden in Echtzeit-fMRT-Anwendungen (vgl. Kapitel 7). Die Echtzeit-Verarbeitung kann Ereignisse auslösen, welche z.B. auf die Stimuluspräsentation wirken und umgekehrt können Ereignisse von außen auf die Verarbeitung Einfluss nehmen. Somit greift das System in experimentelle Variablen ein, die ggf. von einer weiteren Analyse nach dem Echtzeit-Experiment berücksichtigt werden müssen. Unter anderem ergeben sich aus diesen Überlegungen folgende Anforderungen an den Entwurf und die Implementierung der Applikation:

- Die Gesamtfunktionalität sollte leicht erweiterbar sein. Schnittstellen, mit denen eigene Funktionalität in den Auswerteprozess eingebunden werden können, sind dafür die Grundvoraussetzung. Dies bedeutet auch, dass Teilfunktionalitäten gekapselt vorliegen sollten, um von anderen Komponenten verwendet werden zu können.
- Die Unabhängigkeit der Anwendung von anderen Applikationen der Umgebung muss gewährleistet sein. Sie sollte z.B. mit Daten von anderen Scannersystemen

ohne Veränderungen am Quellcode arbeiten können, wenn diese in einem verarbeitbaren Format vorliegen. Ebenso sollte es keine Rolle spielen mit welcher Applikation die Stimuli in einem Experiment erzeugt wurden oder mit welcher Applikation kommuniziert wird.

- In der Applikation muss es möglich sein, die zeitlichen Abläufe während eines Echtzeit-Experimentes nachzuvollziehen, um den Einfluss auf experimentelle Variablen verfolgen zu können. Es müssen also Logging-Mechanismen zur Verfügung gestellt werden.
- Wie in Abschnitt 3.1 deutlich wurde, ist es in der Echtzeit-fMRT wichtig die beteiligten Systeme zeitlich zu synchronisieren. Hier wird der Umstand ausgenutzt, dass Scannersysteme über Triggermechanismen verfügen, welche von externen Applikationen verwendet werden können, um die synchrone Verarbeitung sicherzustellen. Das Triggersignal des Scanners soll auch im *rtExplorer* zur Synchronisierung dienen.
- Mit Hinblick auf die Echtzeit-Auswertung ist besonders auf die Effizienz von verwendeten Datenstrukturen und Algorithmen zu achten.
- Innerhalb der Anwendung ist eine effiziente und einheitliche Fehlerbehandlung umzusetzen, damit Fehler für Entwickler und Anwender nachvollziehbar bleiben.

Die vorliegenden Anforderungen führen zu einem Entwurf, welcher im Folgenden beschrieben ist. Die einzelnen Komponenten der erstellten Anwendung sind in Abbildung 6.1 dargestellt. Die zentrale Funktion der *Steuerungskomponente* wird von einer Laufzeitmethode dargestellt, welche je nach Auslastung des Prozessors mit maximaler Frequenz prüft, welche äußeren Bedingungen vorliegen und danach die entsprechenden Aufrufe tätigt. Liegt beispielsweise in einem Echtzeit-Experiment ein neuer Datensatz in dem beobachteten Zielverzeichnis vor, wird die Auswertung dieses Datensatzes initiiert. Dabei hängt die Verarbeitung von den in der EDL-Beschreibung formulierten Informationen ab. Die Steuerungskomponente liefert als Ergebnis der Auswerteroutinen eine Visualisierung der Hirnaktivierung des Probanden, welche in der *Benutzeroberfläche* angezeigt wird. Die Routinen der Anwendungsschicht sind dabei nur über Schnittstellen zu erreichen, um ihre Funktionalität zu kapseln.

Über die Komponente *Eingabe/Ausgabe* stehen im System Methoden zum Lesen und Schreiben von Bild- und Textdaten zur Verfügung. Routinen zur Behandlung von EDL-Beschreibungen werden von der Komponente *EDL-Verarbeitung* gekapselt. Hierbei steht

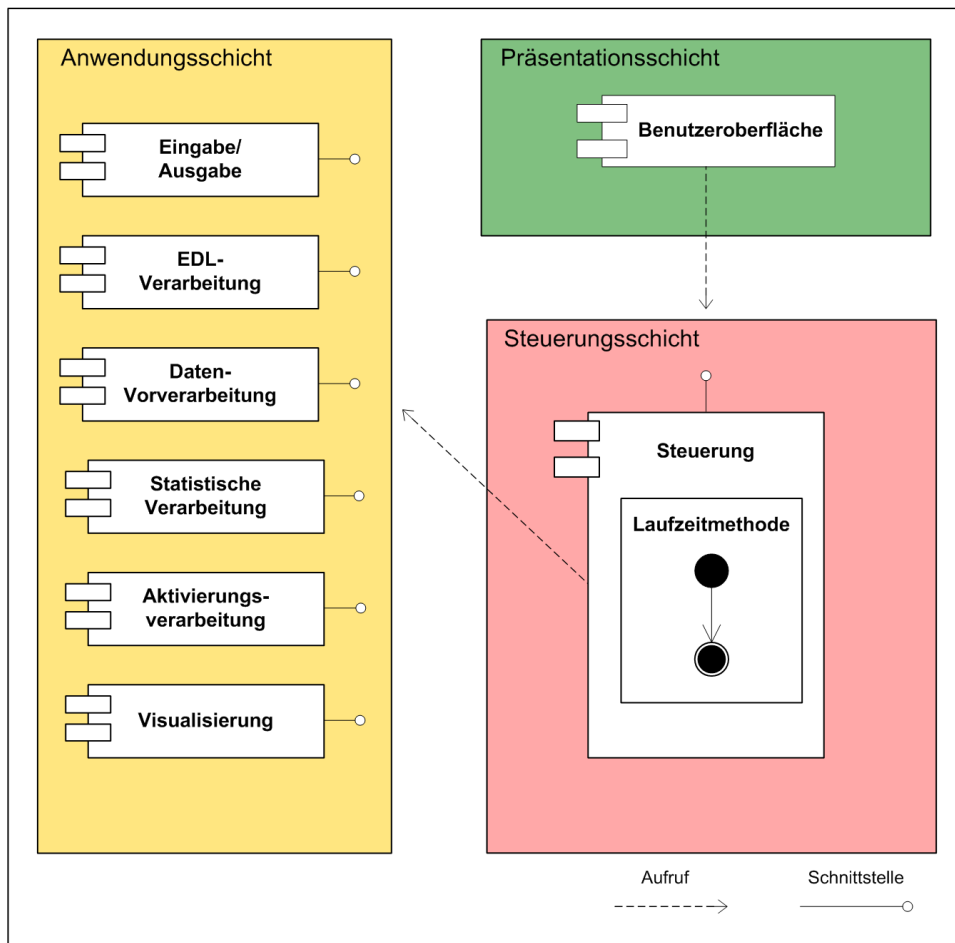


Abbildung 6.1.: Komponenten der Applikation unterteilt nach der Architekturschicht der sie angehören

die Umwandlung der XML-basierten Beschreibung in interne Strukturen sowie die Validierung der Beschreibung und das Testen der Gültigkeit der definierten Regeln im Vordergrund (vgl. Abschnitt 5.5). In den Komponenten *Daten-Vorverarbeitung* und *Statistische Verarbeitung* sind Funktionen gekapselt, welche die Vorverarbeitung von fMRT-Daten und die statistische Auswertung betreffen (vgl. Abschnitt 2.2.4 und 2.2.5). Welche Algorithmen und Parameter dafür verwendet werden, ist wiederum in der EDL-Beschreibung formuliert. Eine besondere Stellung nimmt die Komponente *Aktivierungsverarbeitung* ein. Hier sind Funktionalitäten zur Weiterverarbeitung der Ergebnisse vorangegangener Prozesse zusammengefasst. Dazu gehört z.B. die Kommunikation von Ergebnissen. Ebenso wird eine Schnittstelle zur Verfügung gestellt, welche vom Nutzer für eigene Implementierungen verwendet werden kann. Die Visualisierung von Ergebnissen erfolgt in der Komponente *Visualisierung*. Hierbei ist allein die voxelbasierte Darstellung von

Hirnaktivierung auf anatomischen oder funktionellen Daten gemeint.

6.1.2. Implementierung

Die Anwendung *rtExplorer* wurde in Matlab unter der Version 2006b implementiert. Matlab [2009]. Da es sich bei Matlab nicht um eine objektorientierte Sprache handelt, ist die Umsetzung des Entwurfes nicht in Klassen sondern in Funktionen realisiert. Funktionen, die zu einer Komponente gehören, sind dabei in *Packages* gekapselt, wobei jedes Package über eine Schnittstelle verfügt. Somit findet sich die Struktur aus Abbildung 6.1 in der Implementierung wieder. Innerhalb eines Packages wird die Funktionalität in Modulen implementiert. Diese Module werden wiederum über Schnittstellen angesprochen. Eine Kapselung der Variablen in der Anwendung wird dadurch erreicht, dass es getrennte Strukturen für Variablen der Packages und Module gibt, welche über *get-* und *set-*Methoden angesprochen werden.

Die Benutzeroberfläche (Graphical User Interface - GUI) ist in Abbildung 6.2 dargestellt. Über die GUI interagiert der Benutzer mit der Applikation. Aus der GUI heraus kann der Nutzer zum Beispiel die Laufzeitmethode starten und zur Laufzeit Parameter der Visualisierung wie Schwellwerte oder Visualisierungsmethoden beeinflussen. Die Laufzeitmethode kann zu Beginn eines Experimentes ein Triggersignal vom MRT-Scanner empfangen, somit wird die Synchronität mit der Datenakquisition erreicht. Über welche Modalität Trigger empfangen werden können ist in der EDL-Beschreibung festgelegt (EDL-Sektion *Environment*). Das Package *IO* (Eingabe/Ausgabe) erlaubt das Lesen und Schreiben von Bilddaten in ANALYZE-, DICOM- und IMA-Format, wobei unter IMA-Format ein speziell von Siemens verwendetes Bilddatenformat zu verstehen ist. Ebenso können textuelle Daten im ASCII-Format gelesen und geschrieben werden, was unter anderem für Loggingprozesse verwendet wird.

Im *rtExplorer* müssen Funktionen zum Lesen und Schreiben von XML-Files (also auch EDL-Files) sowie zur Validierung dieser Beschreibungen verfügbar sein. Diese sind in den Packages *XMLParsing* und *EDLValidation* zusammengefasst. Zum Parsen von XML-Files wird eine frei verfügbare XML-Toolbox für Matlab (XMLToolbox [2009]) verwendet. Über diese Toolbox wird lediglich die syntaktische Korrektheit eines EDL-Files geprüft und eine interne Datenstruktur erstellt, welche die EDL-Parameter im System abbildet. Ob formale und strukturelle Bedingungen laut der XSD-Definition von EDL eingehalten werden (Validierung), wird mit Hilfe der freien Java-Bibliothek *javax.xml* Sun Microsystems [2009] getestet. Diese Tests werden für die Beschreibung des Experimentes selbst aber auch für die Beschreibung der EDL-Regeln durchgeführt. Anschließend wird über Routinen aus dem Package *EDLValidation* geprüft, ob die EDL-Beschreibung den

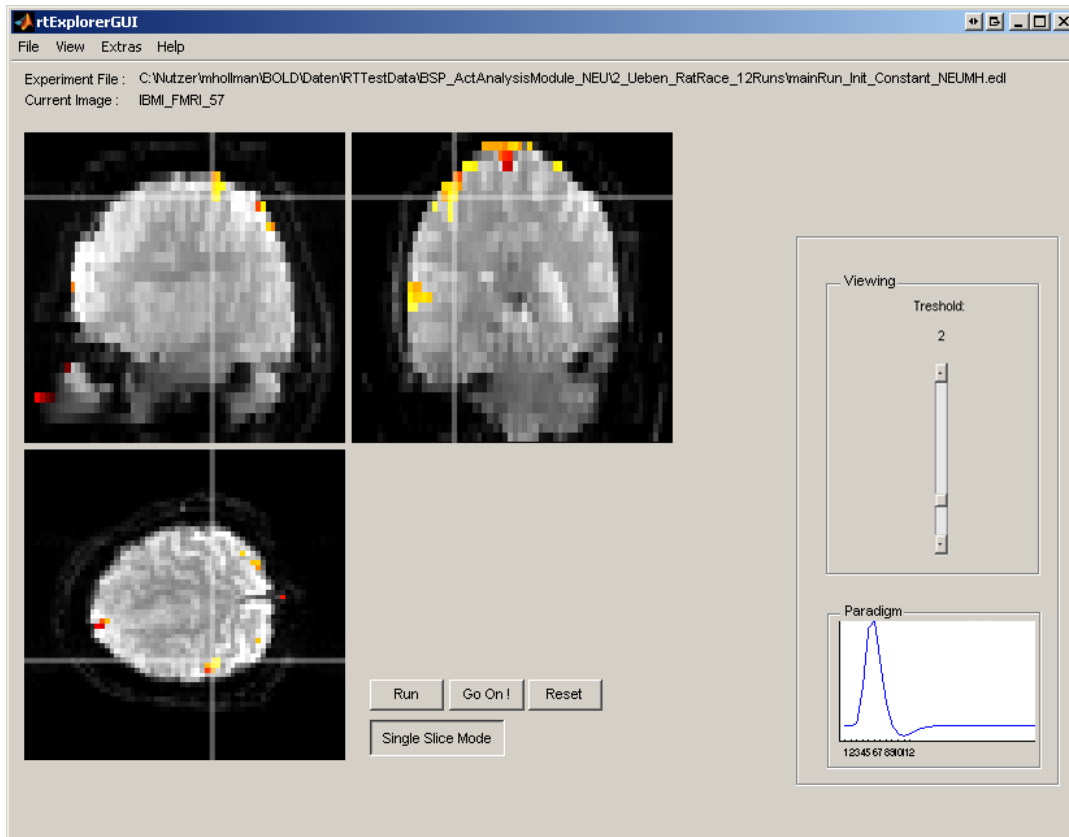


Abbildung 6.2.: Dargestellt ist die grafische Benutzeroberfläche des *rtExplorers*. Die Abbildung zeigt die Visualisierung der lokalen Aktivierung eines Probanden zur Laufzeit als Projektion auf eine einzelne Schicht in sagittaler, coronarer und axialer Ansicht (nur die Aktivierungswerte der aktuell gewählten Schichten in den drei Ansichten werden angezeigt). Auf der rechten Seite sind der Regler zur Steuerung des Anzeigeschwellwertes und die in der Echtzeit-Korrelation verwendete Referenzfunktion dargestellt.

Regeln aus der Regelbasis entspricht. Der Prüfprozess besteht aus einer datengetriebenen Inferenz über der Wissensbasis, wobei die Erkenntnisse aus dem Inferenzprozess mit den Informationen der Faktenmenge im EDL-File verglichen werden (vgl. Abschnitt 5.5.2). Bei Regelverletzungen wird die jeweilige Warn- oder Fehlermeldung, die in der Regelbasis vorgegeben ist, in den sogenannten *messagestack* gespeichert und dem Nutzer angezeigt. Im Rahmen der Vorverarbeitung (Package *Preprocessing*) sind folgende Verfahren implementiert (vgl. Abschnitt 2.2.4):

- Lineares Detrending - Globale lineare Trends werden entfernt¹.
- Räumliche Filterung - Mit Hilfe eines Gauss-Filters wird das SNR verbessert.

¹Das Verfahren beruht auf Least-Square-Fits linearer Funktionen.

- Zeitliche Filterung - Es werden mit Hilfe der Fourieranalyse bestimmte Frequenzen aus dem Signal entfernt.

Die Parameter für diese Verfahren sind in der EDL-Sektion *preprocessing* beschrieben. Die implementierten statistischen Verfahren beschränken sich auf den t-Test (Welch-Test) und die Korrelationsanalyse mit einer Korrelationsfunktion, die sich aus der Faltung einer doppelten Gamma-Funktion mit dem Paradigma ergibt (vgl. Abschnitt 2.2.5). Mit Hilfe einer HRF-Map kann eine separate Korrelationsfunktion für jedes Voxel angegeben werden (vgl. Hollmann et al. [2008a]). Beide Verfahren sind im Package *Statistics* implementiert. Die für die Definition der nötigen Parameter verwendete Sektion in EDL ist: *statistics*.

Die im Entwurf beschriebene Komponente Aktivierungsverarbeitung ist in einem Package mit der Bezeichnung *Activationprocessing* verwirklicht. Die Funktionalität dieses Packages ist in Module unterteilt. Zu bereits implementierten Funktionalitäten gehört das Modul *analysisRuntimeModuleBOLD* (vgl. EDL-Spezifikation in Abschnitt 5.5.1). Damit kann das Signal aktueller Scans aus einer definierten ROI mit dem Signal früherer Scans (Baseline-Signal) in dieser ROI verglichen werden. Das Ergebnis kann z.B. über ein Output-Modul (*outputModuleTCPIP*) versandt werden. Ist die verwendete ROI Ergebnis eines Vorexperimentes, wird hier auf eine EDL-Beschreibung dieses Vorexperimentes verwiesen. Um eigene Routinen einzubinden, kann der Nutzer in der EDL-Beschreibung eines Modules vom Typ *analysisRuntimeModuleUserDefined* auf den Aufruf einer eigenen Methode verweisen. Analog zu den Ausführungen in Abschnitt 5.5.1 wird dafür via EDL das Handle einer Funktion angegeben. Die Definition der Parameter aller Module dieses Packages finden sich in der EDL-Sektion *activationProcessing*.

Als zusätzliche Funktionalität wird die automatische Erstellung einer SPM-Design-Matrix aufgrund der EDL-Beschreibung zur Verfügung gestellt (siehe Abschnitt 2.2.5). Hierbei basieren die Routinen auf der SPM-Version 2. Des Weiteren ist es möglich künstliche fMRT-Daten zu erzeugen. Dabei werden vom System Bilddaten mit einem künstlichen Signal aufgrund der EDL-Beschreibung generiert, welche wiederum im *rtExplorer* verarbeitet werden können. Somit wird die Simulation von Experimenten ermöglicht.

Eine weitere Anforderung wird erfüllt, indem eine einheitliche Fehlerbehandlung implementiert wurde. Entstandene Fehler und Warnungen werden an die jeweils aufrufende Funktion zurückgegeben und zentral nach ihrem Schweregrad behandelt. Eventuelle Fehlermeldungen und Warnungen werden dem Nutzer angezeigt. Um die Speicher- und Rechenperformance zu erhöhen, wurden von MathWorks empfohlene Optimierungsstrategien für Matlab-Anwendungen verwendet. In nachfolgendem Abschnitt sind die Ergebnisse

	$64 \times 64 \times 31$ Voxel	$128 \times 128 \times 31$ Voxel
Growing-Window	0,63 \pm 0.04	1,78 \pm 0.11
Sliding-Window	0,51 \pm 0.03	1,52 \pm 0.07

Tabelle 6.1.: Durchschnittliche Verarbeitungszeit (\pm Standardabweichung) für künstliche 3D-Datensätze in den Testprozeduren (Angaben in Sekunden)

der Tests zu Rechenzeit und Speicherauslastung dokumentiert.

6.1.3. Untersuchungen zu Leistungsmerkmalen

Um nachzuweisen, dass die Applikation mit den integrierten Auswertemethoden in der Lage ist Echtzeit-fMRT-Daten in der nötigen Geschwindigkeit und vertretbarem Speicheraufwand zu verarbeiten, wurden leistungsbezogene Tests durchgeführt. Hierzu wurden künstliche funktionelle Daten mit den Dimensionen $64 \times 64 \times 31$ und $128 \times 128 \times 31$ Voxel herangezogen. Die Daten enthielten jeweils zwei Aktivierungscluster mit jeweils 216 Voxeln. Die Testprozeduren wurden auf einem PC mit folgenden Leistungsmerkmalen ausgeführt:

- Prozessor: Pentium IV
- Taktung: 3.0 GHz
- Random Access Memory (RAM): 2048 MB
- Arbeitsspeicher-Lesedurchsatz: 2563 MB/s
- Betriebssystem: Windows XP

Die Testprozedur bestand aus dem Einlesen der Daten, der Filterung mit einem Gauss-Filter der Größe $6 \times 6 \times 6$ mm, der anschließenden Korrelationsanalyse und dem Speichern der Ergebnisse in einem Datensatz im ANALYZE-Format. Tabelle 6.1 zeigt die durchschnittliche Verarbeitungszeit bei 500 getesteten Volumendatensätzen. Weitere Informationen finden sich in Hollmann et al. [2008b]. Bei realen Daten von Probanden wurden keine signifikanten Abweichungen von diesen Werten festgestellt. Die Speicherauslastung erreichte im Maximum 298 MB Hauptspeicher, was hauptsächlich dem Umstand

geschuldet war, dass für jedes Voxel eine separate Korrelationsfunktion angegeben wurde. Die Gesamtverarbeitungszeit in einem Echtzeit-fMRT-System hängt natürlich auch von anderen Faktoren wie der Rekonstruktionsgeschwindigkeit am Tomographen oder Übertragungsgeschwindigkeiten zwischen den Systemen ab. Diese Merkmale müssen bei der Einrichtung eines Echtzeit-fMRT-Systems separat betrachtet werden.

6.2. Stimuluspräsentation auf Basis einer Experimentbeschreibung mit EDL

Mit Hilfe der Matlab-Bibliothek COGENT wurde ein System zur Präsentation von visuellen (Text, Bilder, Film) und auditiven Stimuli erstellt (COGENT [2000]). Die Applikation wurde *rtPresent* genannt.

Die Beschreibung in EDL (EDL-Sektion *stimulusData*) definiert dabei millisekunden-genau den Ablauf der Präsentation über die Struktur *timingTable* (vgl. Spezifikation der Stimulusdefinition in Abschnitt 5.5.1). Wird ein EDL-File in *rtPresent* geladen, werden alle definierten Medienobjekte in internen Strukturen gespeichert und in einen von COGENT zur Verfügung gestellten Zwischenspeicher geladen. Wenn die Präsentation gestartet wird, werden die Medienobjekte nach den in der *timingTable* vorgegebenen zeitlichen Angaben ausgeführt. Dabei testet die Applikation zur Laufzeit ob der definierte zeitliche Ablauf unter Berücksichtigung der im EDL-File angegebenen Toleranz eingehalten wird. Sollte dieses Kriterium verletzt werden, bricht der Präsentationsprozess ab. Die zeitliche Synchronität mit der Akquise wird mit Hilfe von Triggersignalen erreicht, welche vom MRT-Scanner gesendet werden. Ebenso ist in diesem System Funktionalität implementiert, die externe Signale (z.B. Tastendrucke von Probanden) über eine USB-Schnittstelle oder via TCP/IP-Protokoll empfängt oder eigene Informationen via TCP/IP versendet. Somit besteht auch die Möglichkeit, die Stimuluspräsentation im Experiment zu beeinflussen bzw. Informationen zu kommunizieren. Alle Aktivitäten der Anwendung werden dabei aufgezeichnet und können als Matlab-Struktur oder ASCII-File gespeichert werden. Wie in Abschnitt 5.5.1 beschrieben wurde, können in EDL fünf Typen von Medienobjekten definiert werden: *TEXT*, *IMAGE*, *SOUND*, *COM* und *COMPLEX*. Diese finden sich in der Implementation von *rtPresent* wieder. Von besonderem Interesse sind dabei Medienobjekte vom Typ *COMPLEX*. Diese dienen dazu komplexe dynamische Prozesse darzustellen und können gleichzeitig als Schnittstelle für nutzereigene Funktionalität verwendet werden. Dabei wird innerhalb der EDL-Beschreibung ein Methodenaufruf definiert, welcher vom Stimulussystem zum in der *timingTable* angegebenen Zeitpunkt ausgeführt wird. Über diese Methodik wurden bereits verschiedene komplexe Stimuli

implementiert, hierzu zählt z.B. ein visueller Schachbrett-Stimulus (Flickering Checkerboard). Nähere Informationen finden sich in Hollmann et al. [2008b]. Zusätzlich wurde eine Schnittstelle zu einer virtuellen 3D-Umgebung implementiert, wobei die 3D-Engine Trinigy (Trinigy [2006]) als Grundlage diente (vgl. Hollmann et al. [2008b] und Moench et al. [2008]).

Diese Applikation implementiert keine weiteren Tests für die Validierung der EDL-Beschreibung und Gültigkeit der EDL-Regeln. Die Testprozeduren sollten also über den *rtExplorer* oder den EDL-Editor erfolgen, bevor ein EDL-File in die Anwendung geladen wird.

6.3. Die EDL-Schnittstelle zur Datenakquisition

Die Möglichkeit der Übertragung der Parameter zwischen EDL und der internen Repräsentation dieser Parameter in der Steuersoftware des Magnetresonanztomographen stellt einen wichtigen Teil des Gesamtkonzeptes dar. Mit Hilfe einer Schnittstelle zum MRT ist es möglich auch die Parameter der Datenakquisition in die Beschreibung aufzunehmen und sie der pre-experimentellen Validierung zugänglich zu machen.

Im Allgemeinen besteht ein Sequenzprotokoll, also die Parameterrepräsentation in der Messesequenz am MRT aus einer Vielzahl von Parametern. Es finden sich Parameter, die unter anderem von der Messesequenz selbst, aber auch von der verwendeten Hardware abhängig sind. Um die Kohärenz innerhalb der MR-Parameter zu prüfen, werden in den Messesequenzen komplexe Algorithmen implementiert. Es ist nicht ohne Weiteres möglich, diese Funktionalität in eigenen Quellcode des hier beschriebenen Frameworks aufzunehmen. Deshalb sollte eine Lösung gefunden werden, die Kommunikation mit dem Sequenzprotokoll zu ermöglichen und trotzdem die sequenzinterne Validierung der MR-Parameter beizubehalten.

Die hier beschriebene Schnittstelle wurde für die Siemens Scannersoftware Systemversion Numaris 4 VA25A (Siemens [2004a, d]) implementiert. Eine Messesequenz wird bei diesem System durch eine dynamisch verlinkte C++ Bibliothek (dll) repräsentiert, welche den gesamten Scanprozess steuert. Innerhalb einer Sequenz ist das Messprotokoll über ein Objekt namens *mrProt* gespeichert und die Parameter können am Scanner über eine Benutzeroberfläche eingegeben werden.

Die Schnittstelle zum MRT stellt nun eine Erweiterung der Funktionalität der Siemens EPI-Sequenz *ep2d_bold* dar, welche für fMRT-Messungen verwendet werden kann. Die zusätzliche Funktionalität besteht darin, dass der Nutzer über die GUI der Sequenz ein EDL-File auswählen kann und die Parameter in das aktuelle Messprotokoll übertragen

werden. Dabei wird von der Sequenz selbst die Konsistenz der übertragenen Parameter geprüft. Diese „kontrollierte“ Übertragung geschieht über einen von den Bibliotheken der Scannersoftware zur Verfügung gestellten Mechanismus, den *Direct Protocol Access* (DPA) (Siemens [2004b]). Sollten bei der Übertragung Inkonsistenzen festgestellt werden, wird der Nutzer aufgefordert diese zu beheben. Ebenso können über dieses Interface MR-Parameter in ein EDL-File geschrieben werden. Abbildung 6.3 verdeutlicht die Einbindung der Schnittstelle in die Scannerumgebung.

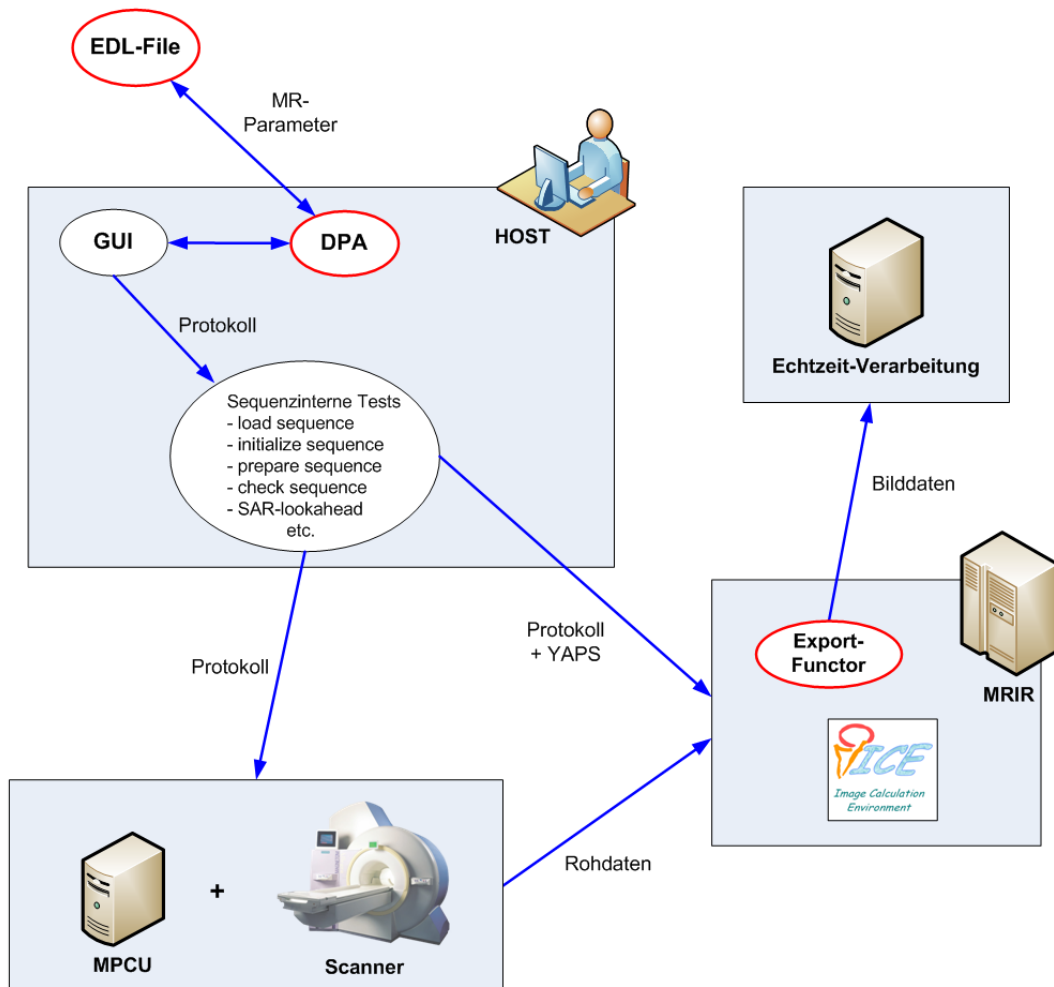


Abbildung 6.3.: Dargestellt ist die Einbindung der implementierten Schnittstelle in die Scannerumgebung. Funktionalität, welche zur Schnittstelle gehört, ist rot markiert. MR-Parameter werden durch die Applikation *Direct Protocol Access* (DPA) in die grafische Nutzerschnittstelle der Sequenz übertragen. Somit können sequenzinterne Tests die Gültigkeit dieser Parameter prüfen. Über den sogenannten YAPS-Buffer wird der Export-Functor angewiesen, die rekonstruierten Bilddaten zu exportieren.

Um zu verhindern, dass Nutzer erst am Scanner feststellen wenn in MR-Parametern Inkonsistenzen auftreten, kann man die von Siemens zur Sequenzentwicklung bereitgestellte Softwareumgebung IDEA (Integrated Development Environment for Applications) verwenden (Siemens [2004b]). IDEA kann auf einem lokalen PC installiert werden und emuliert die Scannerumgebung. Innerhalb dieser Applikation finden sich Werkzeuge um Sequenzen zu simulieren, wobei auch die Benutzeroberfläche zur Parametereingabe verwendet wird. In diesem Simulationsmodus kann man analog zur Bedienung der Schnittstelle am Tomographen EDL-Files einlesen und die Konsistenz der MR-Parameter von der Sequenz prüfen lassen. Die MR-Parameter sind in EDL in der Sektion *MRIParams* definiert. Es muss über die EDL-Variable *vendorTag* jeweils die systeminterne Bezeichnung der Parameter angegeben werden, da die Schnittstelle keine eigenen Routinen zur Auflösung der Parameternamen implementiert. In Abschnitt 5.5.1 wurden bereits die von EDL unterstützten MR-Parameter beschrieben.

Eine weitere Aufgabe der hier beschriebenen Schnittstelle ist der Echtzeit-Export der rekonstruierten Bilddaten während eines Experimentes. Hierzu wurde zusätzliche Funktionalität in das von der EPI-Sequenz verwendete Rekonstruktionsprogramm integriert. Über ein sogenanntes EVA-Protokoll² wird dabei ein Teil des Rekonstruktionsprozesses im sogenannten ICE (Image Calculation Environment) konfiguriert. Die Implementierung der Funktionalität erfolgt in diesem System in Klassen, welche Funktionalität innerhalb des Rekonstruktionsprozesses kapseln. Diese Klassen nennt man EVA-Funktoren. Um den Export zu realisieren, wurde ein neuer EVA-Funktor in den Prozess eingefügt, welcher in der Lage ist zur Laufzeit die Bilddaten im siemenseigenen IMA-Format auf eine im Dateisystem vorhandene Adresse zu schreiben. Auf diese Weise ist es möglich, die durch Standard ICE-Prozeduren zur Verfügung gestellte Bewegungskorrektur der funktionellen Daten zu nutzen und den Export der Daten erst nach dieser Bewegungskorrektur auszuführen. Ob in einer Messung der Echtzeit-Export der Bilder erfolgen soll, wird vom Nutzer über die grafische Benutzeroberfläche eingestellt. Der entsprechende Parameter vom Typ *Boolean* wird dabei über den YAPS-Buffer³ an die ICE-Funktionalität übertragen (vgl. Abbildung 6.3). Abbildung 6.4 zeigt die Elemente, die zusätzlich in die herkömmliche Nutzeroberfläche der EPI-Sequenz eingebunden wurden. Für solche nutzeroigene Funktionalität stellt die Nutzeroberfläche eine sogenannte „Special-Card“ zur Verfügung. Näheres zu dieser Schnittstelle findet sich in Hollmann et al. [2008b]; Mueller [2007] und Mueller u. Bernarding [2008].

²EVA = Kurzform für Evaluation

³YAPS = Abkürzung für yet another parameter set

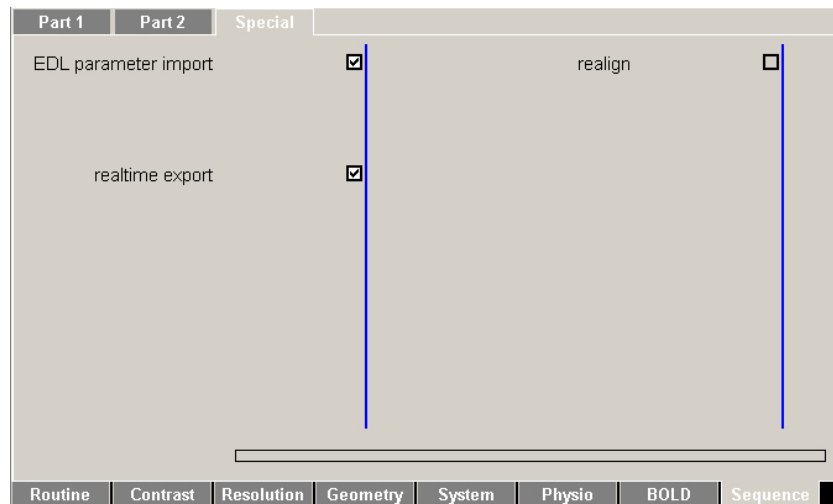


Abbildung 6.4.: Die Erweiterung der grafischen Oberfläche der Sequenz besteht aus zwei Checkboxes für den Parameterimport und den Echtzeitexport der Bilddaten (linke Seite der Abbildung). Da die Standardfunktionalität der Nutzeroberfläche keine Buttons zulässt, wurde eine Checkbox eingebunden, bei deren Aktivierung ein Dialog zum Öffnen eines EDL-Files angezeigt wird.

6.4. Der EDL-Editor

Für das Erstellen von EDL-Files wurde eine C++ Applikation implementiert, welche möglichst flexibel auf Änderungen in der EDL-Spezifikation reagieren kann. Das Grundprinzip ist dabei, dass die Struktur der Eingabelemente (Textfelder, Auswahlfelder etc.) sich allein aus der XSD-Definition von EDL bestimmt. Aus einem EDL-File, welches man zusätzlich laden kann, ergeben sich dann die Inhalte der Eingabelemente. Sollten in der XSD-Definition von EDL Änderungen vorgenommen werden, muss somit die Implementierung des Editors nicht verändert werden. Die Benutzerschnittstelle des EDL-Editors ist in Abbildung 6.5 dargestellt. In diesem Beispiel wurde ein EDL-File geladen aus welchem sich die Inhalte der Eingabefelder ergeben.

Zur Implementierung der GUI wurde die open-source Bibliothek *wxWidgets* (wxWidgets [2009]) in der Version 2.8.9 verwendet, die für verschiedene Plattformen erhältlich ist. Die Applikation nutzt weiterhin XML Bibliotheken von Microsoft, wobei DOM (Document Object Model) und SOM (Schema Object Model) Funktionalitäten verwendet werden. Zu den Funktionalitäten des Editors gehören das Laden und Speichern von EDL-Files (über den Menüpunkt *File* erreichbar) sowie Funktionen zum Validieren und Testen der Gültigkeit der *edlRules* (über den Menüpunkt *Tools* erreichbar). Für die Testroutinen der *edlRules* ist ein Algorithmus zur datengetriebenen Inferenz implementiert. Die Implementierung ist konform mit den Inferenzmechanismen im *rtExplorer* (vgl. Abschnitt

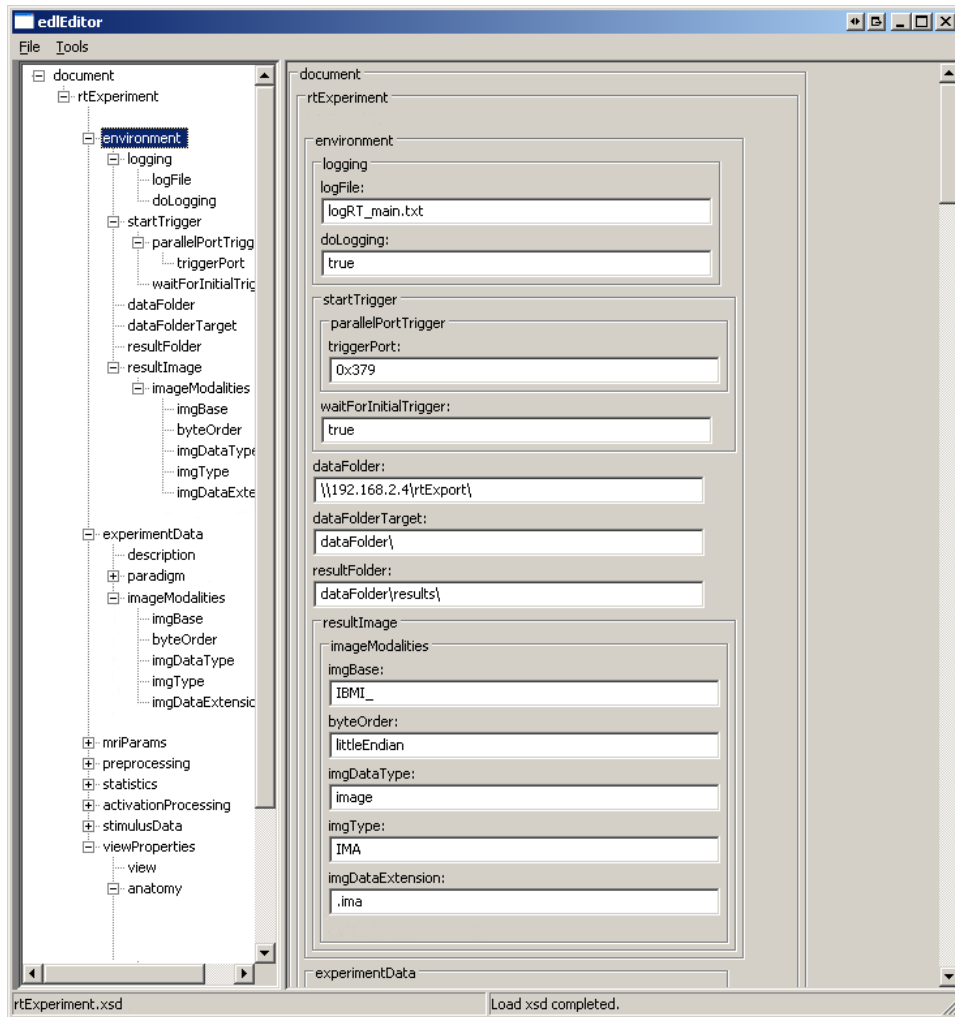


Abbildung 6.5.: Die grafische Benutzerschnittstelle des EDL-Editors: Dargestellt ist der Inhalt eines zuvor geladenen EDL-Files. Die Baumstruktur im linken Teil dient der Übersicht und kann zur Navigation verwendet werden.

6.1.2). Der Testmechanismus kann vom Nutzer zu jedem Zeitpunkt gestartet werden. Dem Test auf die Gültigkeit der Regeln geht dabei immer die Validierung der erzeugten EDL-Repräsentation über die angesprochene DOM-Funktionalität voraus. Sollten Fehler bei der Validierung oder dem Regeltest auftauchen, werden dem Nutzer diese als Meldungen angezeigt. Für den Anwender besteht die Möglichkeit verschiedene Templates zu laden und somit vordefinierte Werte in die Eingabeelemente einzufügen. Diese Templates liegen im System als EDL-Files vor und erleichtern den Eingabeprozess, da Standardwerte vorgegeben werden.

6.5. Anwendung auf das Beispiexperiment

In Abschnitt 5.6 wurde geschildert, wie die für die Lösung der Beispielpromblemstellung essenziellen Elemente in EDL und in `edlRules` formuliert werden. Hier soll dargestellt werden zu welchen Zeitpunkten der Experimenterstellung und mit welchen Methoden die Experimentinformationen auf ihre Gültigkeit getestet werden. Die Testprozedur für eine Experimentbeschreibung mit einem EDL-File besteht zum einen aus der Validierung des Files anhand der XSD-Beschreibung von EDL. Zum anderen werden die Regeln der Regelbasis über Inferenzmechanismen getestet. In diesem Beispiel würden also zwei Dateien die Wissensrepräsentation bilden (der Inhalt bestimmt sich nach den in Abschnitt 5.6 definierten Angaben):

1. Ein EDL-File: *Beispiexperiment.edl*
2. Ein EDL-Rules-File: *edlRules.xml*

Es gibt zwei Möglichkeiten im Erstellungsprozess die Gültigkeit des Experimentes zu prüfen. Die erste Möglichkeit ist, dass der Nutzer das Experiment mit Hilfe des EDL-Editors erstellt. Nach der Definition der Parameter würde der Benutzer über die im vorherigen Abschnitt angesprochene Funktionalität die datengetriebene Inferenz auslösen und die folgende Ausgabe erhalten:

WARNINGS:

```
EDL Warning: Stimuluspresentation length is  
lower than overall scantime in paradigm. Check design?
```

ERRORS:

```
EDL Error: Number of scans in paradigm is NOT equal to  
number of measurements in mriParams (MR_TAG_MEASUREMENTS).
```

Die zweite Möglichkeit ist, dass ein komplettes EDL-File vorliegt, welches über den *rtExplorer* ausgeführt werden soll. In diesem Fall würden beim Laden des EDL-Files erst der Validierungsprozess über die *javax.xml* Bibliotheken erfolgen und bei positiver Rückmeldung der in Abschnitt 6.1.2 beschriebene datengetriebene Inferenzalgorithmus zur Anwendung kommen. Im Beispiel würde die Validierung der EDL-Beschreibung keine Fehler ergeben und die Prüfung der Regeln die bekannte Warn- bzw. Fehlermeldung zum *messagestack* hinzufügen. Der *rtExplorer* würde in diesem Fall die Ausführung des Experimentes verweigern, da eine Meldung vom Typ *ERROR* im *messagestack* gespeichert ist. Die Inkonsistenz in der Definition des Beispiexperimentes wird also vom System erkannt und der Nutzer kann entsprechend reagieren.

6.6. Untersuchungen zur Nutzerfreundlichkeit der Softwareumgebung

Im Rahmen einer Nutzerstudie wurde die Nutzerfreundlichkeit des Gesamtsystems untersucht. Hierfür wurden fünf Freiwillige mit Erfahrung im Bereich der Neurowissenschaft in die Studie einbezogen. Zu Beginn wurden die Probanden gebeten, ihre Erfahrungen im Bereich der funktionellen MRT auf einer Skala von 1 bis 10 (keine Erfahrung bis professionelle Erfahrung) einzuschätzen. Dabei schätzten drei Testteilnehmer ihre Fähigkeiten mit einem Wert von neun ein, einer mit einem Wert von acht und einer mit einem Wert von sechs. Alle Teilnehmer bejahten Erfahrung mit SPM 2 (SPM [2008]) und einer Applikation zur Erstellung von Stimuli für neurowissenschaftliche Experimente namens *Presentation* in der Version 10.0 (Presentation [2009]). Die Studie bestand aus insgesamt drei Teilaufgaben:

1. Die erste Aufgabe der Testteilnehmer bestand darin, ein (Echtzeit-)fMRT Experiment zu erstellen. Dazu sollte ein Stimulationsprogramm implementiert werden, welches ein flickerndes Schachbrettmuster (Frequenz 8 Hz) als visuellen Stimulus darstellt. Das Paradigma sollte aus einem Blockdesign mit 20 Blöcken und einer wechselnden Blocklänge zwischen 10 bis 13 Scans bei einem TR von 2 Sekunden bestehen. Es sollte nur die Länge der Ruhebedingung eines Blockes variiert werden.
2. Als zweiter Schritt sollte für das Paradigma eine SPM-Designmatrix für die SPM Version 2 erstellt werden.
3. Die dritte und letzte Aufgabe war die Vorbereitung der Messung in Form der Eingabe der MR-Parameter in der Benutzeroberfläche, welche über IDEA auf dem für die Tests verwendeten PC zur Verfügung gestellt wurde (vgl. Abschnitt 6.3).

Die Teilaufgaben sollten dabei jeweils unter der Verwendung von *Presentation* und SPM sowie unter Verwendung des beschriebenen EDL-Softwareframeworks ausgeführt werden. Die Nutzer wurden jeweils in die Bedienung des EDL-Editors, des *rtExplorers*, des *rtPresent*-Tools und der Schnittstelle zur Datenakquisition eingewiesen und die Tests verliefen jeweils unter Aufsicht eines Testleiters.

	Presentation / SPM 2	EDL-Framework
Erstellung des Stimulusprogramms	22,3 ± 14,1	16,4 ± 3,2
Erstellung der SPM-Designmatrix	2,2 ± 0,5	automatisiert
Eingabe der MR-Parameter	3,4 ± 0,7	1,2 ± 0,1
Summe	27,9	18,1

Tabelle 6.2.: Durchschnittliche Bearbeitungszeit der Teilaufgaben der Nutzbarkeitsanalyse und die Standardabweichung (die Werte sind in Minuten angegeben)

Am Ende der Testprozedur sollte der gesamte Prozess vergleichend für beide Ansätze unter den folgenden Gesichtspunkten bewertet werden:

- Effizienz
- Verständlichkeit und Nutzerfreundlichkeit
- Fehleranfälligkeit

Dabei wurde eine Skala von 1-10 (schlecht bis sehr gut) zur Bewertung angesetzt. Zur Bewertung der Ergebnisse wurden drei Merkmale verwendet.

1. Es wurde die Zeit gemessen, die jeder Teilnehmer für die Durchführung der Aufgaben benötigte.
2. Es wurden die erstellten Ergebnisse begutachtet und auf eventuelle Fehler geprüft.
3. Die subjektive Einschätzung der Nutzer nach den oben genannten Gesichtspunkten diente als Qualitätsmerkmal für das System.

Tabelle 6.2 stellt die Bearbeitungszeit dar, die von den Teilnehmern für die einzelnen Arbeitsschritte des Tests benötigt wurde. Die Bewertungen der Probanden sind in Tabelle 6.3 dargestellt. In den Experimenten wurde lediglich ein Fehler im Quellcode der Implementierung des visuellen Checkerboard-Stimulus mit *Presentation* gefunden. Die Probanden wurden gebeten, ihre Bewertungen zu kommentieren und eigene Anregungen zum System zu geben. Eine Auswertung dieser Kommentare und die Diskussion der hier vorgestellten Ergebnisse finden sich im folgenden Abschnitt.

	Presentation / SPM 2	EDL-Framework
Effizienz	(7,9)	(7,9)
Bearbeitungsgeschwindigkeit	6,6 ± 0,7	8,4 ± 0,5
Ausdrucksmöglichkeiten	9,2 ± 0,6	7,4 ± 0,7
Verständlichkeit/Nutzerfreundlichkeit	(4,0)	(6,4)
Parametereingabe	3,6 ± 1,1	4,4 ± 0,9
Testmechanismen	2,8 ± 1,4	7,8 ± 0,7
Überblick	5,6 ± 0,9	7,0 ± 1,2
Fehleranfälligkeit	(3,5)	(8,2)
Flüchtigkeitsfehler	2,8 ± 0,6	9,0 ± 0,4
Strukturelle Fehler	4,2 ± 1,4	7,4 ± 0,7
Durchschnitt	(5,1)	(7,5)

Tabelle 6.3.: Durchschnittliche Bewertungen der fünf Testteilnehmer für die beiden Systeme. Die Benotung erfolgte in Werten von 1-10 (schlecht bis sehr gut). In Klammern sind die Durchschnittswerte für die jeweiligen Kategorien angegeben.

6.7. Ergebnisse und Diskussion

In Abschnitt 4.3 wurden bereits Ansätze besprochen, welche sich mit Echtzeit-fMRT und der Sicherstellung der Parameterkonsistenz beschäftigen (BrainVoyager [2009]; In-vivo [2009]; Smyser et al. [2001]; Voyvodic [1999]). Es wurde in diesem Abschnitt auch festgestellt, dass die vorliegenden Arbeiten keine übergreifende Repräsentation von Parametern aller Teilsysteme einer Echtzeit-fMRT-Implementierung bereitstellen. Somit ist in den vorgestellten Lösungen keine vollständige Konsistenzprüfung und Experiment simulation möglich.

Der in den vorangegangenen Abschnitten entwickelte Ansatz einer zentralen Wissensrepräsentation im Zusammenhang mit der implementierten Softwareumgebung lösen die Problemstellung durch eine integrierte Darstellung der getrennten Parameterräume und der Nutzung dieser Darstellung in den entwickelten Applikationen. Wie gezeigt wurde, sind in diesem Ansatz sowohl Methodiken zur Konsistenzprüfung von Experimenten, als auch zur Erzeugung von Simulationsdaten entwickelt worden (vgl. Abschnitte 6.1.2 und 6.4).

Im Folgenden sollen wichtige Eigenschaften der Echtzeit-fMRT-Applikation *rtExplorer* hervorgehoben und diskutiert werden. Wie man erkennen kann, wurde nicht auf die

standardmäßige Implementierung verfeinerter statistischer Verfahren wie dem GLM etc. Wert gelegt, sondern vielmehr auf die Schaffung einer flexiblen Lösung, in welche weitere Funktionalität leicht eingebunden werden kann. Die angesprochene Realisierung über Schnittstellen und unabhängige Module erfüllt diese Forderung. Die Ergebnisse der Leistungsanalyse haben gezeigt, dass die Applikation auf handelsüblichen Rechnern ab 1024 MB RAM lauffähig ist. Es wurde jedoch auch deutlich, dass für höhere Auflösungen von MR-Datensätzen kritische Zeitgrenzen erreicht werden können. Zum Beispiel ist eine Verarbeitungszeit von 1,78 Sekunden für einen Datensatz der Auflösung 128 x 128 x 31 im Grenzbereich. Gerade bei höheren Feldstärken der MRT-Scanner kommt dieser Umstand zum Tragen, weil dort durch ein erhöhtes SNR höhere Matrixauflösungen benutzt werden können. Die Verwendung von Mehrkernprozessoren und die Implementierung der Verarbeitung in parallelen Prozessen birgt dabei noch erhebliches Potenzial zur Optimierung der Performance.

Ein zentraler Designaspekt der Applikation ist, dass die Verarbeitung über den scanbasierten Aufruf aller Funktionalitäten erfolgt. Wie im Entwurf beschrieben wurde, erfolgt der Aufruf der Verarbeitungsprozesse durch die Laufzeitroutine immer dann, wenn ein neuer Datensatz im Zielverzeichnis vorliegt. Die Zeitauflösung mit der die Applikation Aktionen auslösen kann ist damit auf das TR beschränkt. Zum einen vereinfacht dieser Umstand die Implementierung von Auswertefunktionalität, zum anderen folgen daraus jedoch Einschränkungen hinsichtlich der Flexibilität des Systems. In der Echtzeit-fMRT spielen dynamische Prozesse eine große Rolle und es sind Szenarien vorstellbar, in denen eine Auswertesoftware mit einer höheren Frequenz als $1/TR$ agieren muss, auch wenn in diesem Fall noch keine neuen funktionellen Daten vorliegen. Zur Lösung des Problems könnten multithreading Prozesse über Matlab Schnittstellen zu C++ Code in die Implementierung des *rtExplorers* eingeführt werden.

Wie ausgeführt wurde, ist es über EDL und EDL-Regeln nicht möglich, alle Zusammenhänge zwischen den Parametern der MRT-Akquise abzubilden. Als Lösung wurde in diesem Ansatz empfohlen, die Parameterkonstellation vor dem eigentlichen Experiment über die Siemens Entwicklungsumgebung IDEA und die hier vorgestellte Schnittstelle zu prüfen. Dazu muss IDEA lokal auf dem Rechner installiert sein, auf welchem auch dieser Test durchgeführt werden soll. Man muss als Nutzer also Zugang zu dieser Entwicklungsumgebung haben und den relativ aufwendigen Installationsprozess durchführen, was für weniger versierte Anwender zu einer abschreckenden Hürde werden kann. Eine umfassende Lösung hierfür müsste die sequenzspezifischen Algorithmen, welche in der Entwicklungsumgebung verwendet werden, in eigenen Testverfahren implementieren, was unter anderem urheberrechtliche Konsequenzen hätte. Grundlegend ist weiterhin, dass

Bilddatenexport und die Schnittstelle zur Parameterübertragung in das MR-System nur für die beschriebene Siemens-Software entwickelt wurde. Für andere Systeme müssen also eigene Lösungen implementiert werden.

Wie die Auswertung der Untersuchungen zur Nutzerfreundlichkeit noch zeigen wird, gibt es bei der Anwendung zur Erstellung von EDL-Files (*EDL-Editor*) noch Verbesserungsbedarf. Ein Aspekt, der von den Testteilnehmern nicht angesprochen wurde, aber aus Anwendersicht sehr wichtig ist besteht darin, dass der EDL-Editor in der derzeitigen Implementierung die datengetriebene Inferenz nur als Testmethode der vollständigen Faktenmenge anbietet. Daraus folgt, dass erst alle Angaben vollständig eingegeben sein müssen, ehe ein Test der Abhängigkeiten über die Regelbasis erfolgt. Für den Anwender wäre es jedoch wünschenswert, direkt bei der Zuweisung eines Wertes zu einer Variablen zu erfahren, ob dies eine gültige Zuweisung im Sinne der Regelbasis ist. Dies könnte effizient über einen Algorithmus zur zielorientierten Inferenz gelöst werden, wie er im Abschnitt 5.2 skizziert wurde. Eine solche Implementierung muss schnell genug sein um während des Eingabeprozesses laufend die Gültigkeit der Regeln prüfen zu können, ohne spürbare Verzögerungen hervorzurufen. Insbesondere wenn die Faktenmenge und die Regelbasis in Zukunft noch erweitert werden, gewinnt die laufende zielorientierte Inferenz an Gewicht, da Mehrfacheingaben bzw. Korrekturen vermieden werden.

Um diesen Abschnitt abzuschließen, sollen nun die Ergebnisse der Untersuchung der Nutzerfreundlichkeit (Abschnitt 6.6) diskutiert werden. Die hier vorgestellte Nutzerstudie erlaubt die Bewertung des Systems nur in eingeschränktem Umfang. Zum einen sind die erzielten Bearbeitungszeiten und Benotungen von den jeweiligen Fertigkeiten des Nutzers abhängig. Zum anderen ist der Stichprobenumfang mit fünf Teilnehmern relativ gering. Trotzdem gibt die Studie interessante Hinweise auf Stärken und Schwächen des vorgestellten Systems. Es wurde gezeigt, dass die Bearbeitungszeiten für die Teilaufgaben unter Benutzung des EDL-Frameworks geringer ausfallen. Besonders die Erstellung der Stimuluspräsentation mit Presentation stellte sich als kompliziert dar, da die von Presentation verwendete Skriptsprache umständlich im Umgang ist. Es wird jedoch angemerkt, dass für nicht vordefinierte komplexe Stimulusprogramme in rtPresent (vgl. Abschnitt 6.2) ebenso Programmierkenntnisse in Matlab erforderlich sind. Als positiv wurde die geringe Anfälligkeit des EDL-Frameworks auf Flüchtigkeitsfehler bewertet, wie auch aus den Ergebnissen in Tabelle 6.3 zu erkennen ist. Ebenso gab es positive Kommentare zur automatischen Erstellung der SPM-Designmatrix.

Ein wichtiger Kritikpunkt betraf die Eingabe der Parameter im EDL-Framework. Die Nutzerführung und Übersichtlichkeit im EDL-Editor wurden als mangelhaft empfunden, was sich in der schlechten Benotung (4,4) für diesen Punkt widerspiegelt. Ebenso wurde

in Kommentaren angemerkt, dass die Hilfsfunktionen und Informationen zu den Elementen von EDL nicht ausreichend repräsentiert sind. Es ist verständlich, dass es für den naiven Benutzer initiale Schwierigkeiten gibt mit EDL umzugehen. Wie in Abschnitt bereits diskutiert wurde, ist ein Teil des Wissens über den Aufbau und die Funktionsweise eines Experimentes schon durch die Struktur und die verwendeten Konzepte in EDL repräsentiert. Diese Konzepte müssen durch den Anwender erst verinnerlicht werden, bevor er effektiv ein Experiment mit dem EDL-Editor erstellen kann. In diesem Punkt unterscheidet sich dieser Ansatz nicht von anderen Systemen. Aus Sicht der Nutzerführung könnte der EDL-Editor jedoch noch stark verbessert werden, z.B. könnten Übersichtsgrafiken und ein intelligentes Hilfe- und Dokumentationssystem die Erstellung von Experimenten vereinfachen. Im Bereich der Softwareergonomie gibt es bereits weitreichende Forschungen Dahm [2005]; Preim [1999]. In der derzeitigen Implementierung ergeben sich für didaktische Konzepte die spezielle Parameter jedoch noch Probleme. Wie in Abschnitt 6.4 dargelegt wurde, wird die Struktur und Form der Eingabeelemente automatisch aus der XSD-Definition von EDL erzeugt. Dieser Ansatz ist sehr flexibel, weil es im Quellcode des Editors keine spezifischen Routinen für Elemente in EDL gibt und somit Änderungen an der Beschreibungssprache keine Änderungen in der Implementierung des Editors nach sich ziehen. Gleichzeitig ist dadurch jedoch die individuelle Behandlung einzelner Eingabeelemente und die Bearbeitung spezieller Sachverhalte erschwert, da dann im Quellcode individuelle Routinen für EDL-Elemente vorliegen müssten. Über flexible Programmierkonzepte können zukünftige Versionen des Editors mit diesen Herausforderungen umgehen. Wie die Benotung der Fehleranfälligkeit in Tabelle 6.3 und das Beispiexperiment verdeutlichen, ist eine Hauptintention des Ansatzes erfüllt worden. Übergreifende Zusammenhänge können vom System analysiert werden, was dazu beiträgt, dass Nutzer leicht konsistente Experimente erstellen können. Flüchtigkeitsfehler und strukturelle Fehler werden erkannt, wenn die nötigen Informationen der Regelbasis abgebildet sind.

Eine zentrale Konsequenz dieser Arbeit betrifft die Ausdrucksmöglichkeiten der Wissensbasis, also die Flexibilität hinsichtlich der Abbildung verschiedener Experimentszenarien. Diese Frage tauchte auch in den Kommentaren der Testanwender auf. Dieser zentrale Punkt wurde in Abschnitt 5.7 schon angesprochen und wird in den abschließenden Betrachtungen in Abschnitt 8 noch einmal Eingang finden. Zwei Nutzer regten in ihren Kommentaren die Aufnahme von Probandendaten in EDL an, um die Verwaltung von Experimenten automatisieren zu können. Diese Erweiterung von EDL erscheint durchaus sinnvoll (vgl. Abschnitt 5.7) und wird in zukünftigen Versionen von EDL eine Rolle spielen.

7. Vorhersage von Entscheidungen in sozialen Interaktionsszenarien mit Hilfe der Echtzeit-fMRT

Das vorliegende Kapitel behandelt die Anwendung der in den vorherigen Abschnitten entwickelten Methodik zur Echtzeit-fMRT auf eine Studie aus dem Bereich der Neuroökonomie. Den Kern dieser Studie bildet ein soziales Interaktionsszenario. Soziale Interaktion stellt einen wichtigen Teil des menschlichen Zusammenlebens in sozialen Gemeinschaften dar. Unter anderem ist sie Bestandteil von Verhandlungen und wirtschaftlichen Beziehungen. Besondere Brisanz erhält das Thema, wenn man davon ausgeht, dass es z.B. möglich wäre einem Verhandlungspartner Informationen über die Entscheidungstendenz seines Gegenübers zukommen zu lassen, bevor dieser seine Entscheidung aktiv kund tut. In einer diesem Kapitel zugrunde liegenden Studie wurde versucht, durch die Kombination von Echtzeit-fMRT und musterbasierten Klassifikationsverfahren herauszufinden, ob Entscheidungen von Probanden in sozialen Interaktionsszenarien vorhersagbar sind. Diese Problemstellung wird im Abschnitt 7.1 näher beschrieben. Um sich der Thematik zu nähern, werden im Abschnitt 7.2 die Grundlagen der sozialen Kognition betrachtet. Als Interaktionsszenario wurde ein bekanntes Paradigma aus der ökonomischen Spieltheorie, das Ultimatum Game, verwendet. In Abschnitt 7.3 werden die Merkmale dieses Paradigmas sowie ökonomische und neuroökonomische Implikationen mit Hilfe verwandter Arbeiten beleuchtet. Die daraufhin entwickelte Methodik wird in Abschnitt 7.4 beschrieben. Es folgt die Betrachtung der Ergebnisse der angewandten Klassifikationsverfahren und die Diskussion der Methodik und der Ergebnisse in Abschnitt 7.5. Die diesem Kapitel zugrunde liegende Studie wurde in Hollmann et al. [2009] veröffentlicht.

7.1. Problembeschreibung

Das Augenmerk der vorgestellten Studie liegt auf der *Voraussage* einer Entscheidung. Das System soll also über die Information der Entscheidung verfügen, *bevor* der Pro-

band diese kommuniziert. Hierzu muss eine Methodik gefunden werden, die in der Lage ist in Echtzeit, also während eines Experimentes, die nötigen Informationen zu extrahieren und zu bewerten. Es wurde bereits gezeigt, dass man mit Hilfe der fMRT in der Lage ist, mentale Zustände aufgrund der Hirnaktivität von Probanden zu unterscheiden (Haynes u. Rees [2005, 2006]; Mourao-Miranda et al. [2005]; Soon et al. [2008]). Echtzeit-fMRT bietet sich also an, diese Extraktion mentaler Zustände in Echtzeit-Anwendungen zu übertragen.

Die in o.g. Arbeiten verwendete und bereits in Abschnitt 2.2.5 eingeführte informationstechnische Methode der Klassifikation bietet sich auch für die Anwendung in der Echtzeit-fMRT an. Da es Limitierungen bezüglich der Verarbeitungszeit gibt, müssen schnelle Klassifikationsverfahren verwendet werden. Um diesem Umstand Rechnung zu tragen, ist es z.B. möglich den Merkmalsraum einzuschränken und nur BOLD-Signale vordefinierter Regionen als Merkmale zu verwenden. Dieses Vorgehen erfordert eine Analyse neuronaler Korrelate sozialer Interaktion und Entscheidungsfindung. Diese Analyse liefert die benötigten a-priori Informationen, um schnelle Klassifikationsverfahren zu implementieren, welche eine Echtzeit-Klassifikation in dem oben beschriebenen Problemfeld erst ermöglichen.

7.2. Soziale Kognition

In Form eines kurzen Überblickes wird dieser Abschnitt die Grundlagen der humanen sozialen Kognition behandeln. Dieses Forschungsgebiet ist sehr weitläufig und berührt verschiedene Bereiche aus Psychologie, Neurowissenschaft, Verhaltensbiologie und Sozialwissenschaft. Diese Einführung soll lediglich einen Einblick vermitteln, welche zum Verständnis der Arbeit nötig ist und erhebt keinen Anspruch auf Vollständigkeit.

Eng verknüpft mit der sozialen Kognition ist der Begriff der Emotion. Emotionen kann man als Reaktionsmuster auf positiv verstärkende oder aversive körperexterne oder -interne Reize verstehen (Birbaumer u. Schmidt [2006]; Plutchik [1980]). Diese laufen auf verschiedenen Ebenen ab:

1. Der motorischen Ebene
2. Der physiologischen Ebene (einschließlich hormoneller Reaktionen)
3. Der subjektiv-psychologischen Ebene

Hirnregionen, die Gefühlszustände erzeugen oder verarbeiten, liegen hauptsächlich zwischen phylogenetisch alten Strukturen im Stammhirn und neokortikalen Regionen. Neben der Amygdala werden insbesondere der cinguläre Cortex, orbitofrontale Areale, der

präfrontale Kortex sowie der insuläre Kortex als wichtige Strukturen bei der Emotionsverarbeitung betrachtet (Schneider u. Fink [2007]). Im Verlauf dieses Abschnittes wird sich zeigen, dass weitere Strukturen bei den komplexen Prozessen der sozialen Kognition beteiligt sind. Entscheidend ist, dass Emotionen psychologische Variablen wie Gedächtnis, Verhalten, Lernen, Wahrnehmung etc. beeinflussen und somit einen entscheidenden Anteil an sozialen Interaktionsprozessen haben. Dieser Umstand wird sich auch in den Betrachtungen über das hier implementierte Interaktionsszenario widerspiegeln. Im Folgenden sollen neuronale Korrelate im Kontext der verschiedenen Prozesse bei sozialer Interaktion betrachtet werden. Die betrachteten Regionen sind schematisch in Abbildung 7.1 dargestellt.

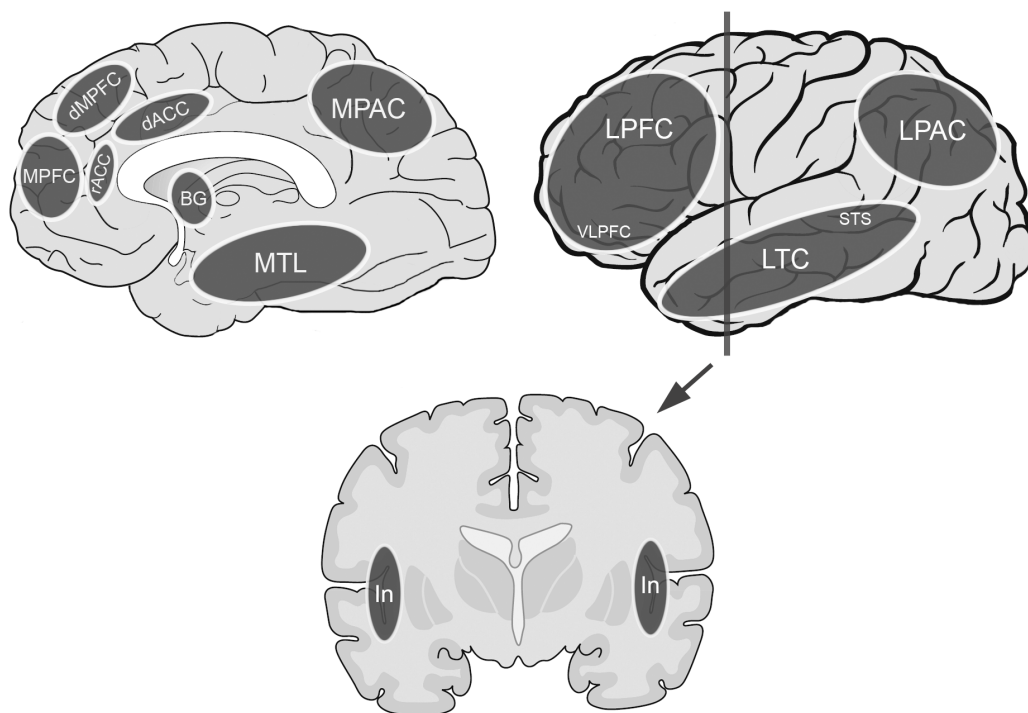


Abbildung 7.1.: Verschiedene Hirnregionen welche im Zusammenhang mit sozialer Kognition stehen (Critchley [2005]; Gallagher u. Frith [2003]; Lieberman [2007]; Rilling et al. [2004]; Sanfey [2007]; Schneider u. Fink [2007]): **Mediale Ansicht:** Medialer Präfrontaler Kortex (MPFC), Dorsaler Medialer Präfrontaler Kortex (dMPFC), Rostrales Anteriores Cingulum (rACC), Dorsales Anteriores Cingulum (dACC), Basalganglien (BG), Medialer Temporallappen (MTL), Medialer Parietaler Kortex (MPAC) **Laterale Ansicht:** Lateraler Präfrontaler Kortex (LPFC) mit ventrolateralem Anteil (VLPCF), Lateraler Temporaler Kortex (LTC) mit Superior Temporal Sulcus (STS) , Lateraler Parietaler Kortex (LPAC) **Coronare Ansicht:** Insula (In)

Automatisierte vs. kontrollierte Prozesse

Man kann zwei Arten von Kognitionsprozessen unterscheiden. Zum einen gibt es kontrollierte Prozesse (wie z.B. das bewusste Erinnern an eine Telefonnummer), die mit den Attributen Bewusstheit, Intention und der Möglichkeit der Unterbrechung verbunden sind (Wegner u. Bargh [1998]). Zum anderen gibt es automatisierte Prozesse, denen die genannten Attribute nicht zugeordnet werden können. Automatisierte Prozesse (wie die Ausschüttung von Stresshormonen in einer Gefahrensituation) verlaufen in der Regel sehr schnell und man findet neuronale Korrelate eher in phylogenetisch älteren Gebieten als Korrelate kontrollierter Prozesse (Lieberman [2007]; Wegner u. Bargh [1998]). Das bewusste Regulieren einer emotionalen Reaktion auf einen Reiz gehört zu diesen kontrollierten Prozessen, wobei ein unbewusster affektiver Zustand und eine autonome physische Reaktion (vegetativ oder muskulär) auf den gleichen Reiz nicht beeinflusst werden kann (Schneider u. Fink [2007]).

Selbstwahrnehmung

Die menschliche Selbstwahrnehmung basiert zum einen auf der physischen Rückkopplung des Körpers, steht aber auch in engem Zusammenhang mit Konzepten (Schemata), die das „Selbst“ in unserer Vorstellung repräsentieren (Lieberman [2007]; Wegner u. Bargh [1998]). MacDonald und Paus zeigten in Experimenten aus dem Jahr 2003, dass bei Unstimmigkeiten zwischen der visuellen Präsentation des menschlichen Körpers und der inneren physischen Wahrnehmung der Probanden Aktivierung des LPAC nachgewiesen werden kann (MacDonald u. Paus [2003]). Bei aktiver Selbstreflektion im Sinne vom Nachdenken über den eigenen Zustand (z.B. „Wie fühle ich mich gerade?“) wurden wie auch bei autobiografischen Erinnerungen Aktivierungen im MPFC und MTL gefunden (Lieberman [2007]; Ochsner et al. [2005]). Aktivierung des MPFC und MPAC wird mit dem Nachdenken über Selbstkonzept, also der eigenen Rolle in der Umwelt (z.B. Wie nehme ich mich wahr?) in Verbindung gebracht (Lieberman [2007]; Ochsner et al. [2005]).

Wahrnehmung sozialer Partner

Die Wahrnehmung sozialer Partner in der Umwelt und die neuronalen Korrelate von Handlungen wie z.B. das Schlussfolgern von emotionalen Zuständen und Absichten anderer Menschen werden in der sogenannten „Theory of Mind“ (ToM) beschrieben (Lieberman [2007]; Rilling et al. [2004]). In verschiedenen Arbeiten hierzu wurden z.B. STS und DMPFC mit dem Schlussfolgern über Intentionen eines sozialen Partners in Verbindung gebracht (Gallagher u. Frith [2003]; Rilling et al. [2004]; Sanfey [2007]). Die

Unterdrückung eigener Erfahrungen und Ansichten bei diesem Schlussfolgerungsprozess könnte dabei mit Aktivierungen im LPFC einhergehen (Gallagher u. Frith [2003]; Samson et al. [2005]).

Selbstregulation

Um eigene Impulse zu kontrollieren und das Verhalten an soziale Normen anzupassen, bedarf es Mechanismen der Selbstregulation. Hierbei rücken Strukturen wie die Insula und der dorsale ACC in den Mittelpunkt. Die Insula wird als wichtige Struktur bei der Verarbeitung affektiver Reize und der folgenden autonomen Reaktion betrachtet, während der dACC mit dem Finden von Konflikten zwischen Impulsen und eigenen Intentionen (wie z.B. sozial angepasstes Handeln) in Zusammenhang gebracht wird (Critchley [2005]; MacDonald et al. [2000]; Rilling et al. [2004]). Es wird vermutet, dass Prozesse der Ziel-Fokussierung (Goal-Maintenance) in solchen Situationen mit einer Aktivierung des LPFC einhergehen (Lieberman [2007]; MacDonald et al. [2000]; Sanfey [2007]), wobei insbesondere die Strategie der Neubewertung von impulsauslösenden Stimuli mit der Aktivierung des rechten VLPFC verbunden sein könnte (Ochsner u. Gross [2005]).

Soziale Interaktion

Soziale Interaktion kann je nach Situation von Kooperation und Vertrauen geprägt sein, aber auch von Rückweisung, Misstrauen und als unfair empfundenem Verhalten. In Szenarien menschlicher Interaktion, die von Vertrauen und Kooperation geprägt sind, werden insbesondere Aktivierungen des MPFC, des MPAC und der Basalganglien berichtet (Aron et al. [2005]; Gallagher u. Frith [2003]; Lieberman [2007]; Rilling et al. [2004]). Eisenberger et al. wiesen dACC-Aktivierung bei sozialer Rückweisung nach (Eisenberger et al. [2003]). Ebenso sind ACC und anteriore Insula bei nicht-kooperativer und unfairer Interaktion involviert (Rilling et al. [2004]; Sanfey [2007]). Dies ist von besonderer Bedeutung, da besonders die anteriore Insula auch mit der Verarbeitung von autonomen Reaktionen auf Ekel und Schmerz in Verbindung gebracht wird (Lieberman [2007]; Wicker et al. [2003]).

7.3. Interaktionsszenarien in Neuroökonomischen Experimenten: Das Ultimatum Game

Menschliches Verhalten in ökonomische Modelle aufzunehmen, ist ein wesentlicher Bestandteil der Bestrebungen eines besonderen Zweiges der Wirtschaftswissenschaften,

der Verhaltensökonomie (Camerer et al. [2003]). Nach welchen Mustern verhalten sich Verhandlungspartner in Interaktionsszenarien? Welchen Einfluss haben soziale Normen, Emotionen und Framing-Effekte¹? Empirische Ergebnisse und Modelle aus der ökonomischen Spieltheorie fließen in diese Fragestellungen genauso ein wie sozialpsychologische und neurowissenschaftliche Erkenntnisse (Aumann [2002]; Camerer et al. [2003]; DeMartino et al. [2006]). Die Frage neuronaler Korrelate in Verbindung mit ökonomischen Fragestellungen wird im Bereich der Neuroökonomie untersucht (Glimcher et al. [2008]; Kenning et al. [2007]), wobei starke Überschneidungen mit Arbeiten im Bereich der „Theory of Mind“ und „Social Decision-Making“ vorliegen (Rilling et al. [2008]; Sanfey et al. [2006]). In Untersuchungen im Rahmen der Neuroökonomie werden oft Interaktionsszenarien aus der Spieltheorie verwendet. Paradigmen aus der ökonomischen Spieltheorie sind deswegen besonders interessant, weil sie meist über empirisch belegte Modelle des menschlichen Verhaltens in diesen Szenarien verfügen und somit einen guten Ausgangspunkt für neurowissenschaftliche Experimente bieten. Ein solches Interaktionsszenario aus der Spieltheorie ist das sogenannte *Ultimatum Game* (UG).

Im Ultimatum Game mit zwei Parteien wird einem Spieler die Rolle des Anbieters und einem anderen Spieler die Rolle des Angebotsnehmers zugewiesen. Im Verlauf des Spiels bietet der Anbieter dem Angebotsnehmer von einem fest vorgegebenen Geldbetrag einen Teil an. Bei Annahme durch den Angebotsnehmer bekommt der Anbieter den verbleibenden Teil ausgezahlt und der Angebotsnehmer erhält den ihm angebotenen Teil des Startbetrages. Lehnt der Angebotsnehmer jedoch ab, erhält keiner der beiden Mitspieler eine Auszahlung (Aumann [2002]; Henrich et al. [2004]). Wenn man strikt logisches profitmaximierendes Handeln unterstellt, wie es bei der Modellannahme des „Homo oeconomicus“ der Fall ist, sollte der Anbieter immer den kleinstmöglichen Betrag bieten und der Angebotsnehmer diesen in jedem Fall annehmen. Dies liegt darin begründet, dass für den Angebotsnehmer selbst der kleinste Gewinn gegenüber keinem Gewinn im Ablehnungsfall vorzuziehen ist (Aumann [2002]). Robuste experimentelle Ergebnisse belegen, dass dieses spieltheoretische Gleichgewicht nicht erreicht wird. Vielmehr werden unvorteilhafte (unfaire) Angebote von unter 20 Prozent des Startbetrages in mehr als 50 Prozent der Fälle abgelehnt (Fehr u. Schmidt [1999]). Die hohe Ablehnungsrate unfairer Angebote im UG deutet auf nicht rein rationales Verhalten und unterstützt die Vorstellung des Einflusses von anderen Faktoren, wie Emotionen oder sozialen Normen auf Entscheidungen humaner Spielpartner (Camerer et al. [2003]).

Im Rahmen einer Studie zum ökonomischen „Decision-Making“ implementierten Sanfey

¹Unter Framing-Effekt versteht man den Einfluss der Art der Präsentation eines Entscheidungsproblems (Entscheidungsrahmen) auf die Entscheidung selbst.

et. al eine Version des Ultimatum Games in Kombination mit herkömmlichen funktionellen MRT-Messungen (Sanfey et al. [2003]). Hier wurden den Probanden zehn Mitspieler vor einem Experiment persönlich als Anbieter vorgestellt und während des Spieles wurde der jeweilige Mitspieler eines Durchganges per Foto identifiziert. Hiermit versuchte man eine soziale Verbindung zu etablieren und diese durch visuelle Präsentation des Spielpartners während des Experimentes aufrecht zu halten. Es wurden jedoch nicht nur Durchgänge mit menschlichen Partnern durchgeführt, sondern auch ein Computer als Mitspieler ausgewiesen. Dieses Vorgehen diente dazu Differenzen in Verhaltens- und Kognitionsprozessen zwischen Spieldurchgängen gegen humane Partner und gegen nicht-humane Partner zu finden.

Die Angebotshöhe in dieser Implementierung setzte sich bei einem Startbetrag von 10\$ wie folgt zusammen (Anteil Anbieter : Anteil Angebotsnehmer): 5\$: 5\$, 7\$: 3\$, 8\$: 2\$, 9\$: 1\$. Um die experimentelle Variable der Angebotshöhe kontrollieren zu können und die volle Breite an Angebotsmöglichkeiten auszuschöpfen, wurden während der Experimente vordefinierte Angebote an die Probanden gegeben und nicht wie in der ökonomischen Spieltheorie üblich freie Entscheidungen der Anbieter zugelassen. Insgesamt spielte jeder Proband 30 Durchgänge mit jeweils zehn Durchgängen mit einem menschlichen Partner, zehn Durchgängen mit einem Computer als Partner und zehn Kontrolldurchgängen ohne Partner. In diesen Experimenten kam die sogenannte „Single-Shot“-Variante des UG zur Anwendung, d.h. jeder Proband erhielt von einem bestimmten Partner jeweils nur ein Angebot. Somit gibt es keine vordergründige strategische Komponente, da Entscheidungen der Angebotsnehmer keinen Einfluss auf zukünftige Angebote haben.

In dieser Studie wurden nur die Ablehnungsraten der Angebote 8\$: 2\$ und 9\$: 1\$ mit 47% bzw. 61% angegeben. Die Ergebnisse dieser Studie zeigten signifikant erhöhte Aktivität des ACC, DLPFC und der anterioren Insula bei unfairen Angeboten menschlicher Spielpartner im Gegensatz zu fairen Angeboten menschlicher Spielpartner (siehe Abbildung 7.2). Ebenso wurde festgestellt, dass unfaire Angebote von als „Computer“ gekennzeichneten Spielpartnern in signifikant weniger Fällen abgelehnt wurden als unfaire Angebote humaner Partner.

7.4. Methodik und Materialien

Im Folgenden werden die Methoden dargestellt, welche verwendet wurden um die in Abschnitt 7.1 vorgestellte Problemstellung zu bearbeiten. Dabei spielen das Versuchsdesign, sowie hard- und softwaretechnische Methoden eine Rolle. Hierzu gehört insbesondere die technische Umsetzung der Experimente mit Hilfe des EDL-Frameworks. Die Beschrei-

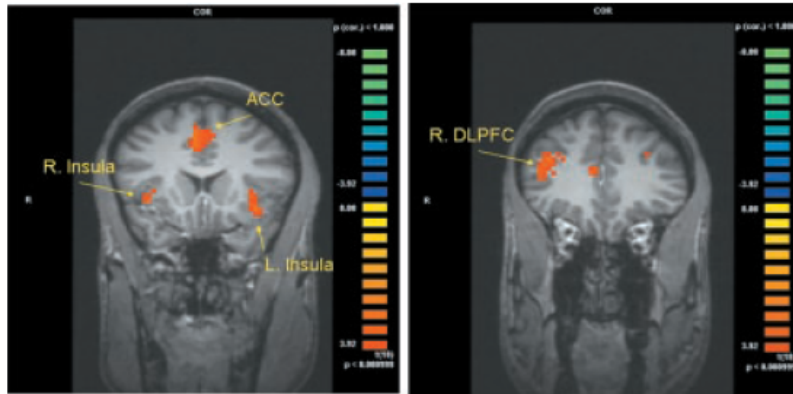


Abbildung 7.2.: Darstellung der Ergebnisse der t-Statistik von Sanfey et al.: Kontrast unfaire vs. faire Angebote humaner Partner ($P < 0.001$) (Sanfey et al. [2003])

bung der Analysemethoden zur Ermittlung der Klassifikationsgüte und die Darstellung der Offline-Klassifikation schließen diesen Abschnitt ab.

7.4.1. Probanden und Paradigma

Es wurden zehn gesunde männliche Probanden in die Studie einbezogen. Das Alter der Teilnehmer reichte von 23 bis 28 Jahren (Mittelwert: $24,7 \pm$ Standardabweichung: 1,64). Alle Probanden gaben schriftlich ihre Einwilligung in die Experimente, welche in Übereinstimmung mit den Bestimmungen der Ethikkommission der Medizinischen Fakultät der Universität Magdeburg durchgeführt wurden. Um geschlechterbezogene Effekte, wie z.B. von Eckel u. Grossman [1996] berichtet, auszuschließen, wurden nur männliche Teilnehmer zugelassen. Ein Teilnehmer wurde aus der Studie ausgeschlossen, nachdem er nach dem Experiment Zweifel äußerte gegen einen humanen Partner gespielt zu haben. Die hier vorgestellten Daten und Ergebnisse beziehen sich also auf neun Probanden. Im Folgenden soll der Ablauf eines Experimentes geschildert werden.

Vor einem Experiment wurden dem jeweiligen Probanden zwei männliche Mitspieler persönlich vorgestellt. Es wurde erläutert, dass diese beiden Mitspieler die Rolle der Anbieter im folgenden Versuch übernehmen und für jeden einzelnen Durchgang (Trial) der aktuelle Anbieter zufällig aus diesen beiden ausgewählt würde. Tatsächlich wurden die Angebote jedoch in einer vordefinierten Reihenfolge abgegeben um diese experimentelle Variable genau kontrollieren zu können (vgl. hierzu auch Sanfey et al. [2003] und Rilling et al. [2004]). Anschließend wurde allen drei Spielern am Computer der Ablauf des Experimentes erklärt und dem Probanden die Rolle des Angebotsnehmers zugewiesen. Es

wurde erläutert, dass die gesamte Entlohnung für das Experiment sich allein aus dem Gewinn im Spiel bestimmt. Nach den Experimenten füllten die Probanden Fragebögen aus, um z.B. zu klären, ob Ihnen Zweifel gekommen waren gegen menschliche Partner gespielt zu haben. Die Probanden spielten, während sie sich im Scanner befanden, insgesamt 60 Trials des Ultimatum Games. Sie bekamen also 60 Angebote und mussten 60 mal entscheiden ob, sie das aktuelle Angebot annehmen oder ablehnen. Abbildung

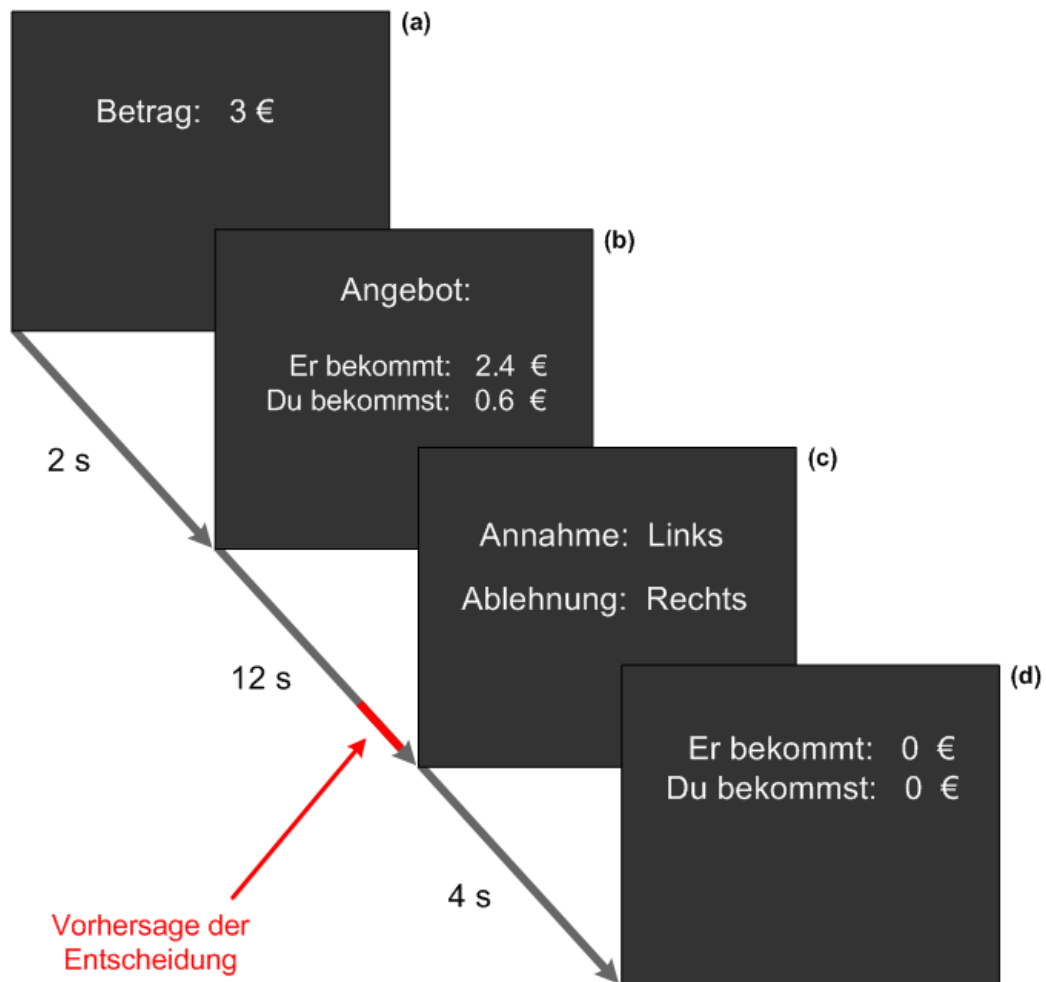


Abbildung 7.3.: Ablauf eines einzelnen Trials im Experiment: Die Visualisierung zeigt die Bildschirmanzeige, welche dem Probanden im Scanner präsentiert wurde. Die Vorhersage der Entscheidung erfolgte ein bis zwei Sekunden vor dem Einblenden des Antwort-Bildschirms (c). Im dargestellten Fall wurde das Angebot abgelehnt.

7.3 zeigt den Ablauf eines einzelnen Trials. In jedem Durchgang wurde dem Probanden zuerst für zwei Sekunden der Startbetrag gezeigt (Abbildung 7.3 a). Dieser Betrag war festgesetzt auf drei Euro für alle Trials. Anschließend (Abbildung 7.3 b) wurde das Ange-

bot des aktuellen Mitspielers für 12 Sekunden gezeigt. In dieser Zeit konnte der Proband sich entscheiden das Angebot abzulehnen oder anzunehmen. Zeitgleich leitete das System aus der Hirnaktivierung des Probanden die Vorhersage seiner Entscheidung ab und zeigte diese dem Experimentator ein bis zwei Sekunden vor dem Einblenden des Antwort-Bildschirms (Abbildung 7.3 c) an. Der Proband hatte während der Antwortphase von vier Sekunden die Möglichkeit mit Tastendrücker des Zeigefingers der rechten bzw. linken Hand seine Entscheidung kundzutun. Wichtig ist dabei, dass auf dem Bildschirm dargestellt wurde, welcher Taster (links oder rechts) für die Ablehnung bzw. Annahme benutzt werden sollte. Diese Funktion der Tasten wurde zufällig vertauscht, um auszuschließen, dass folgende Klassifikationsverfahren Informationen aus der Vorbereitung auf motorische Aktivität ableiten (vgl. hierzu Haggard u. Eimer [1999]). Nachdem der Proband seine Entscheidung kommuniziert hatte, wurde für vier Sekunden die endgültige Verteilung des Startbetrages angezeigt (Abbildung 7.3 d). In dieser Abbildung ist der Ausgang eines Trials mit einem abgelehnten Angebot zu sehen: Keiner der beiden Spieler erhält eine Auszahlung.

Die Anzahl der verschiedenen Angebote war über 60 Trials wie folgt verteilt (Angaben in Prozent von 3 Euro - Anbieter/Angebotsnehmer):

- 6 mal 50/50 (1,5 Euro / 1,5 Euro)
- 8 mal 65/35 (1,95 Euro / 1,05 Euro)
- 12 mal 70/30 (2,1 Euro / 0,9 Euro)
- 21 mal 80/20 (2,4 Euro / 0,6 Euro)
- 13 mal 90/10 (2,7 Euro / 0,3 Euro)

Die Angebote in einer Höhe von 80/20 und 90/10 werden im Folgenden als unfaire Angebote bezeichnet.

7.4.2. Technischer Aufbau und Datenakquisition

Der technische Aufbau des Echtzeit-fMRI Systems sowie die vorliegenden Datenströme im UG-Experiment sind in Abbildung 7.4 dargestellt. Die bereits beschriebene Anwendung *rtExplorer* (vgl. Abschnitt 6.1) wird dabei auf dem externen PC (External-PC) unter der Matlab-Version 2006b ausgeführt. Die visuelle Stimulation erfolgte über einen getrennten PC (Stimulus-PC), der die Bildinformationen über einen Beamer und ein Spiegelsystem für den Probanden sichtbar macht. Die Antworten der Probanden in Form von Tastendrücker wurden vom Stimulus-PC über eine USB-Schnittstelle empfangen und an

den externen PC weitergeleitet. Es wurde ein Siemens Ganzkörper-MRT Scannersystem

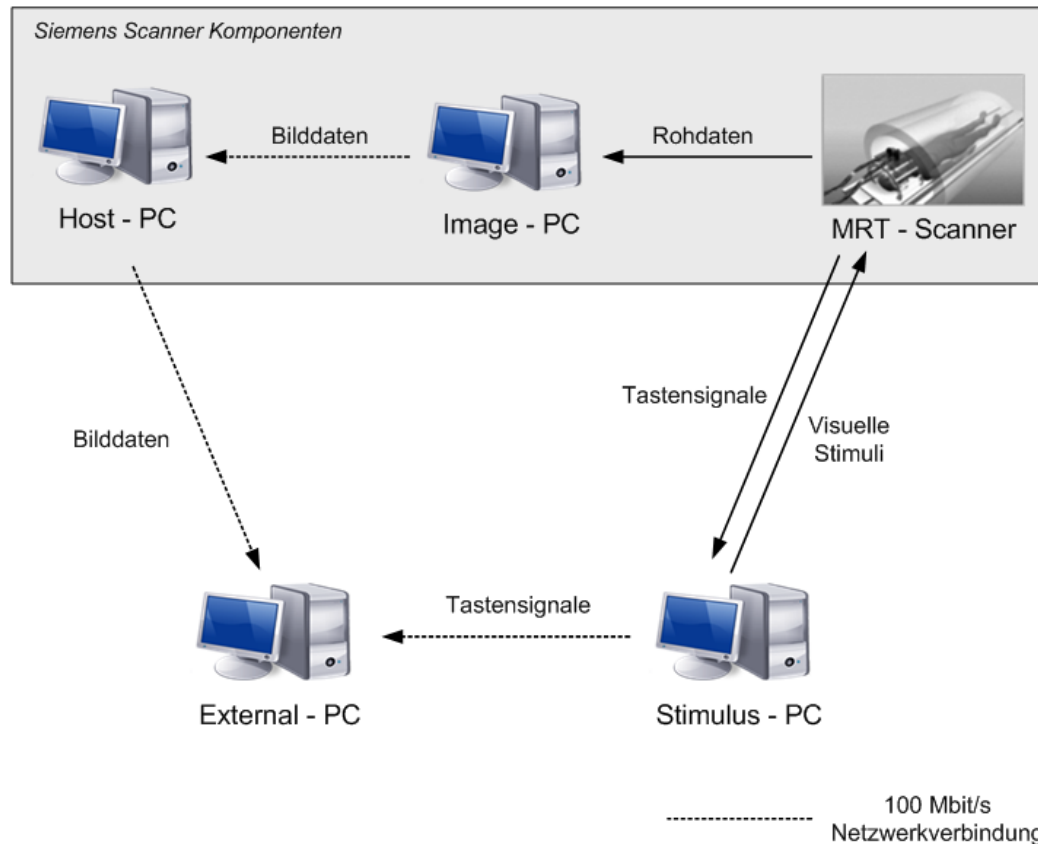


Abbildung 7.4.: Aufbau des Echtzeit-fMRT-Systems und interne Datenströme

mit einer Feldstärke von drei Tesla und einem Trio Gradientensystem (Siemens Medical Systems, Erlangen) verwendet. Die installierte Systemversion ist VA25A, wobei der in Abschnitt 6.3 beschriebene Mechanismus zum Export der rekonstruierten Bilddaten zur Anwendung kam. Zur Bildgebung diente eine 8-Kanal phased-array Kopfspule. Es wurde eine Gradientenecho-EPI Sequenz mit den folgenden Parametern verwendet: TR: 2000 ms, TE: 29 ms, Flipwinkel: 90° , Schichtanzahl (axiale Orientierung): 31, Matrix: 64×64 , Räumliche Auflösung: $3,4 \times 3,4 \times 4$ mm. Mit einem so erzeugten 3D-Datensatz wurde das gesamte Gehirn der Probanden akquiriert, wobei schon vor dem Export die Bewegungskorrektur durch in den Standard-Rekonstruktionsprogrammen implementierte Algorithmen erfolgt (vgl. Abschnitt 6.3).

Die Leistungsdaten des externen PC's wurden bereits im Abschnitt 6.1.3 beschrieben. Im nachfolgenden Abschnitt ist die Verbindung dieses Aufbaus mit dem EDL-Framework (Kapitel 5 und 6) näher beschrieben.

7.4.3. Integration in das EDL-Framework

Ein wichtiger technischer Aspekt ist, dass sowohl die Messequenz am Scanner wie auch die Datenauswertung am externen PC und die Stimuluspräsentation über den Stimulus-PC (vgl. Abbildung 7.4) auf ein und dasselbe EDL-File zugreifen und somit ihre Parameter aus der in Kapitel 5 vorgestellten Experimentrepräsentation beziehen. Die Parameter der MRT-Akquise finden sich als Auszug aus diesem zentralen EDL-File in Anhang A.1.4. In den UG-Experimenten wurden diese Parameter durch die in Abschnitt 6.3 beschriebene Schnittstelle zum Messprotokoll des MR-Scanners übertragen.

Die Implementierung der Datenauswertung erfolgte unter Nutzung der von EDL und dem *rtExplorer* angebotenen Schnittstellen. Über ein Modul zur erweiterten Auswertung (vgl. Abschnitt 5.5.1) wurden Normalisierungsroutinen, statistische Tests und Klassifikationsroutinen in den laufenden Verarbeitungsprozess des *rtExplorers* eingebunden. Die zugehörige Definition des EDL-Laufzeitmoduls findet sich in Anhang A.1.5. Die Stimuluspräsentation erfolgte über die in Abschnitt 6.2 vorgestellte Applikation *rtPresent*. Es wurden die von *rtPresent* zur Verfügung gestellten Schnittstellen verwendet um die Präsentation zu steuern und die Tastendrucke der Probanden zu kommunizieren. Im Anhang A.1.2 ist das entsprechende EDL-Medienobjekt dokumentiert.

7.4.4. Echtzeit-Verarbeitung und Klassifikation

Um anhand der Hirnaktivität eines Probanden zu unterscheiden, ob er ein ihm unterbreitetes Angebot annehmen oder ablehnen wird, wurden die gemittelten Signale aus den folgenden drei Regionen analysiert: *Occipitaler Kortex*, *Anteriore Insula* und *LPFC*. Diese drei Regionen sind in Abbildung 7.8 schematisch dargestellt und die entsprechenden MNI-Koordinaten² sind in Tabelle 7.1 angegeben. Die linken und rechten Anteile von LPFC und Anteriorer Insula wurden jeweils zu einem Mittelwert zusammengefasst. Die Definition der Regionen ergab sich aus der Analyse der Daten von zwei Testprobanden und anhand anatomischer Merkmale, und erfolgte als feste Regions of Interest (ROI) in einem normalisierten 3 x 3 x 3 mm MNI-Raum. Die Abbildungen 7.5 und 7.6 zeigen die signifikant aktivierten Regionen bei einem der beiden Testprobanden unter den Angebotsbedingungen: *fair vs. unfair* und *unfair vs. fair* (ermittelt mit SPM 5). Während eines laufenden Experimentes wurden die funktionellen Datensätze direkt nach der Rekonstruktion exportiert und in den angesprochenen MNI-Raum transformiert (vgl. Ablaufschema der Echtzeit-Verarbeitung in Abbildung 7.7). Anschließend wurde das gemittelte Signal in den vordefinierten ROI's ermittelt. Globale lineare Trends der

²MNI = Standardraum des *Montreal Neurological Institute* (Evans et al. [1993])

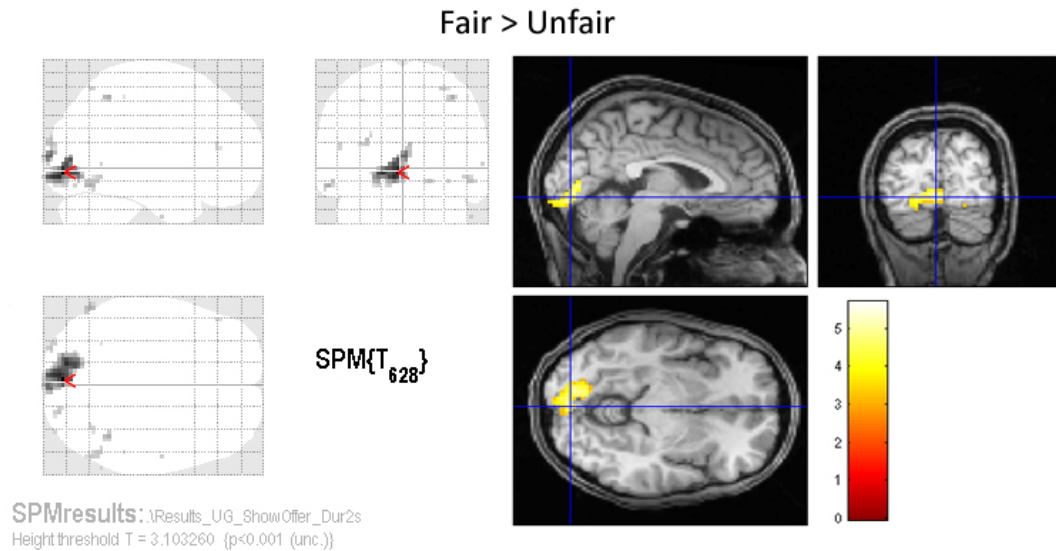


Abbildung 7.5.: Aktivierte Bereiche für den Kontrast: faire Angebote vs. unfaire Angebote ($p < 0.001$, unkorrigiert) bei einem der Testprobanden. Im linken Teil ist die Maximum-Intensity-Projektion der aktivierten Voxel dargestellt. Die rechte Seite zeigt die schichtweise Überlagerung auf einen anatomischen Datensatz des Probanden. Man kann die Aktivierung des visuellen Kortex erkennen (siehe Markierung).

Signalverläufe wurden mit einem Fitting-Verfahren³ korrigiert. Anschließend wurden die t-Werte für ein Sliding-Window von zehn Sekunden ab der Anzeige eines Angebotes berechnet. Die Berechnung der t-Werte erfolgte mit einem Welch-Test, wie in Gleichung 2.14 in Abschnitt 2.2.5 dargestellt. Als zu vergleichende Mittelwerte dienten dabei die Signalwerte der ersten beiden Scans im Sliding-Window bzw. der folgenden drei Scans in diesem Fenster.

Zehn Sekunden nach Anzeige des Angebotes lagen also drei t-Werte vor. Diese drei Werte wurden an einen nicht-linearen RVM-Klassifikator übergeben, welcher als Matlab-Bibliothek (Implementierung von Tipping [2001]) im *rtExplorer* eingebunden war. Das Klassifikationsproblem liegt also in einem dreidimensionalen Merkmalsraum \mathbf{R}^3 und besteht aus zwei Klassen (Annahme/Ablehnung). Das Ergebnis des Klassifikators wurde angezeigt und dokumentiert, *bevor* der Antwort-Bildschirm (Abbildung 7.3 c) angezeigt wurde und demnach bevor der Proband seine Entscheidung per Knopfdruck kommunizierte. Der Klassifikator wurde mit einem radialen Gauss-Kernel und maximal 500 Iterationen trainiert (vgl. Tipping [2001] und Abschnitt 2.2.5). Das initiale Training des Klassifikators erfolgte auf Grundlage der t-Werte und Entscheidungen von zwei Probanden. Somit lagen 120 Stichproben vor, aufgrund derer der Klassifikator die Trennung der

³Dieses Verfahren beruht auf einem in Matlab implementierten least-square Fit einer Geraden.

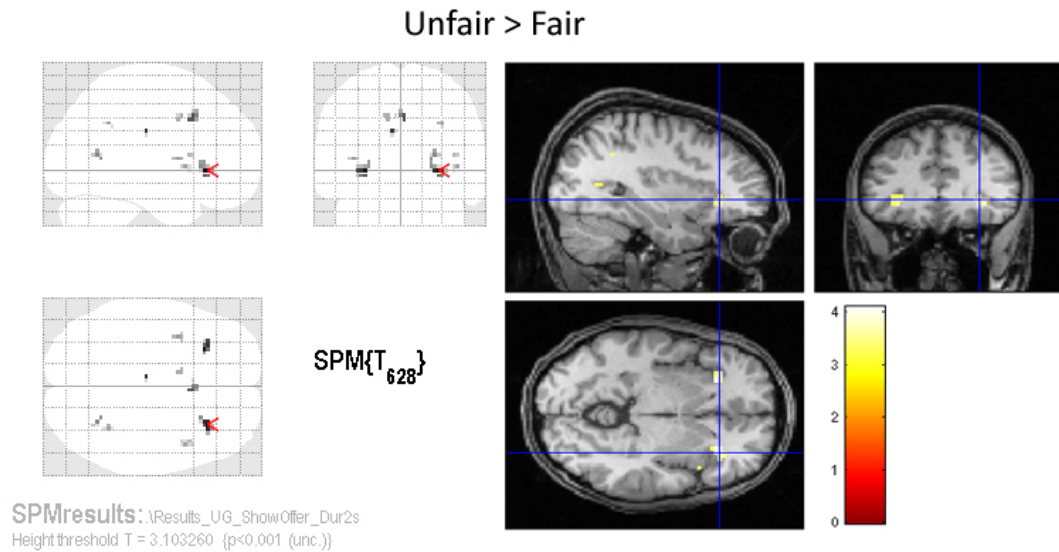


Abbildung 7.6.: Aktivierte Bereiche für den Kontrast: unfaire Angebote vs. faire Angebote ($p < 0.001$, unkorrigiert) bei einem der Testprobanden (vgl. Abbildung 7.5). Die Markierung zeigt aktivierte Voxel in der anterioren Insula.

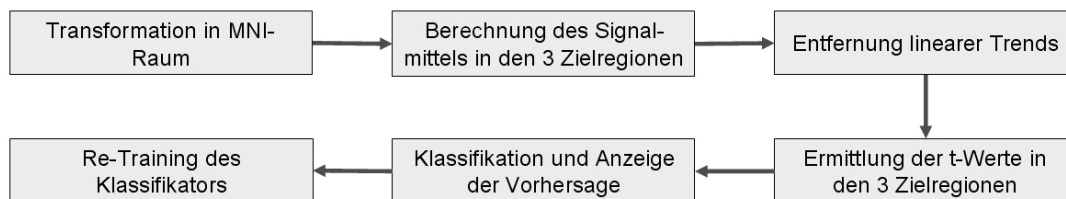


Abbildung 7.7.: Ablaufschema der Echtzeit-Verarbeitung für jeden Trial

Klassen *abgelehnte* und *angenommene* Angebote im Merkmalsraum bestimmte. Der Entscheidungsvektor bestand hierbei aus den Elementen -1 und $+1$, wobei -1 für Ablehnung und $+1$ für die Annahme eines Angebotes steht. Während der Experimente wurde diese initiale Trainingsmenge $X_{initial}$ zur Klassifikation des Trials n jeweils um die t -Werte und Entscheidungen des Probanden aus den Trials 1 bis $n - 1$ erweitert und der Klassifikator neu trainiert um die Güte zukünftiger Voraussagen zu erhöhen. Die Trainingsmenge X_n für den Trial n ergibt sich also als:

$$X_n = X_{initial} \cup (x_{1..n-1}, y_{1..n-1}), \text{ mit } x_i \in \mathbf{R}^3 \text{ und } y_i \in \{1, -1\}. \quad (7.1)$$

Die Ausführung des Klassifikationsalgorithmus (Klassifikation und Re-Training) wurde auf der in Abschnitt 6.1.3 beschriebenen Hardware in durchschnittlich $0,9$ Sekunden ausgeführt.

	X (mm)	Y (mm)	Z (mm)
Occipitaler Kortex	0	-88	3
Anteriore Insula			
Links	38.5	20	-1
Rechts	-38.5	20	-1
LPFC			
Links	50	28	11
Rechts	-50	28	11

Tabelle 7.1.: Mittelpunkte der verwendeten ROI's wie sie im MNI-Raum definiert wurden

7.4.5. Ermittlung des Rateniveaus

Um die Klassifikationsgüte eines Klassifikators beurteilen zu können, empfiehlt sich ein Vergleich der erzielten Ergebnisse mit dem Rateniveau des Klassifikators (Good [2005]; Polyn et al. [2005]; Rieger et al. [2008]). Dieses Vorgehen soll sicherstellen, dass erzielte Aussagen nicht auf purem „Raten“ des Klassifikators beruhen. Das theoretische Rateniveau in diesem zwei-Klassen Experiment beträgt 50 Prozent, jedoch wird das Rateniveau auch von anderen Faktoren wie z.B. der Anzahl der Elemente in einer Klasse bestimmt. Wenn in einem Merkmalsraum beispielsweise 70 von 100 Elementen zur Klasse 1 von zwei Klassen gehören und der Klassifikator sich bei jedem Element für Klasse 1 entscheidet liegt die Klassifikationsgüte bei 70 %, ohne dass der Klassifikator gelernt hätte die Klassen aufgrund struktureller Informationen zu trennen. Um das Rateniveau des Echtzeit-Klassifikators zu bestimmen, wurden die Klassifikationsroutinen in einer offline-Analyse wiederholt, jedoch mit dem Unterschied, dass die Entscheidungsvektoren, welche die Grundlage des Trainings des Klassifikators sind, permutiert wurden. Somit wurde die Korrelation zwischen beobachteten t-Werten und Probandenentscheidungen zerstört und trotzdem die relativen Klassengrößen beibehalten. Insgesamt wurden für jeden Probanden 500 Durchläufe mit jeweils permutierten Entscheidungsvektoren durchgeführt um das mittlere Rateniveau und das 95 % Konfidenzintervall mit einer Abweichung von 0.9 % zu berechnen. Dabei kam für jeden der 500 Durchläufe exakt der Algorithmus aus der Online-Klassifikation zur Anwendung.

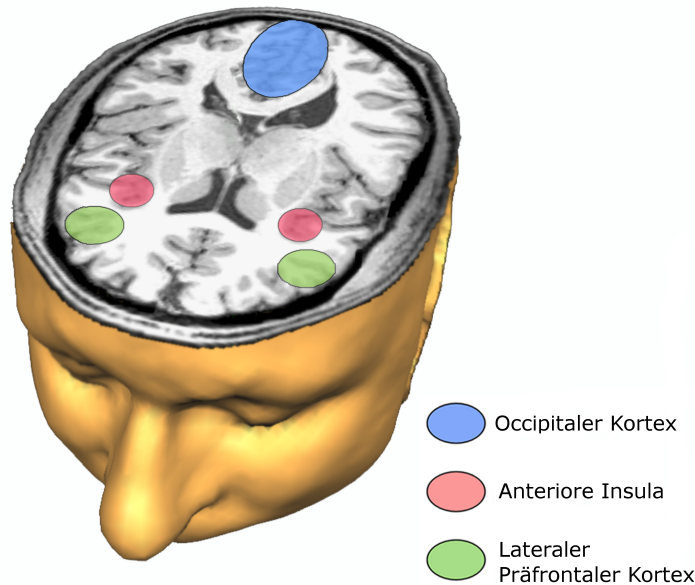


Abbildung 7.8.: Schematische Darstellung der Lokalisation der vordefinierten ROI's

7.4.6. Offline-Klassifikation

In einer postexperimentellen Analyse der Daten wurde getestet, wie zuverlässig man die Probandenentscheidungen aus der Hirnaktivierung bestimmen kann, wenn die t-Werte aller Voxel als Merkmale in die Verarbeitung eingehen. Im Gegensatz zur Echtzeit-Klassifikation wurden hier umfangreiche Vorverarbeitungsschritte ausgeführt. Abbildung 7.9 verdeutlicht den Ablauf der Datenverarbeitung in der Offline-Analyse. Analog zur

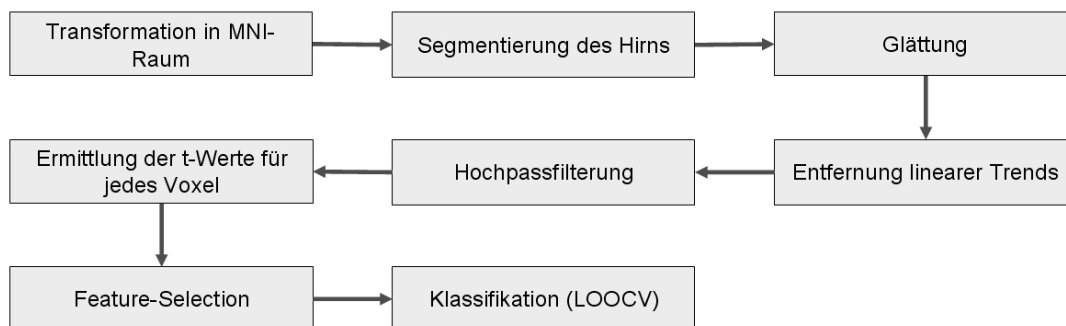


Abbildung 7.9.: Ablaufschema für die Offline-Verarbeitung

Online-Verarbeitung wurde jeder Datensatz in einen 3 x 3 x 3 mm MNI-Raum transformiert. Bei jedem Datensatz wurde der Hintergrund mit Hilfe einer Maske (MNI-Template) segmentiert und von der weiteren Verarbeitung ausgeschlossen. Die Daten

wurden mit einem räumlichen Gauss-Filter der Größe 6 mm geglättet und analog zur Online-Verarbeitung lineare Trends des Signalverlaufes für alle Voxel entfernt. Im Anschluss wurden die Daten zeitlich gefiltert, wobei ein Hochpassfilter mit der cut-off Frequenz von 0.01 Hz verwendet wurde (vgl. Abschnitt 2.2.4). Anschließend wurden entsprechend der Online-Analyse die t-Werte für ein Sliding-Window von 10 Sekunden nach der Anzeige des Angebotes berechnet. Im Unterschied zur Online-Methodik wurden die t-Werte für jedes einzelne Voxel berechnet. Zur Selektion von relevanten Voxeln wurde eine feature-selection durchgeführt (vgl. Abschnitt 2.2.5). Diese bestand darin den Korrelationskoeffizienten c zwischen den berechneten t-Werten und dem Entscheidungsvektor des Probanden zu berechnen und alle Voxel mit einem errechneten Koeffizienten von $|c| < 0.2$ aus der weiteren Analyse auszuschließen. Bei allen Probanden lag die Anzahl der verbleibenden Voxel zwischen 8000 und 10000.

Im Gegensatz zur Echtzeit-Klassifikation wurde für diese Analyse kein RVM-Klassifikator verwendet, da die vorliegende Implementierung sich bei einer hohen Anzahl an Merkmalen als instabil erwies. Stattdessen wurde eine Implementierung einer Support-Vector-Machine von Chang und Lin (Chang u. Lin [2001]) verwendet. Es wurde ein Klassifikator mit einer polynomialen Kernel-Funktion (Parameter $\gamma = 3.0$) verwendet (vgl. Abschnitt 2.2.5). Mit einer Leave One Out Cross Validation, welche in Abschnitt 2.2.5 bereits beschrieben wurde, wurde die Klassifikationsgüte ermittelt. Dies bedeutet, dass für jeden Probanden zur Ermittlung der korrekt klassifizierten Trials 60 Durchläufe mit der jeweils maximalen Trainingsmenge von 59 Trials durchgeführt wurden. Analog zum Vorgehen bei der Echtzeit-Klassifikation wurde auch hier das Rateniveau des Klassifikators ermittelt. Um die Klassifikation eines Trials durchzuführen, was das Trainieren von 59 Trials und den finalen Klassifikationsschritt beinhaltet, benötigte der Algorithmus auf einem Dual-Core Pentium Prozessor mit 1800 MHz Taktfrequenz und 3072 MB RAM durchschnittlich 35 Sekunden.

7.5. Ergebnisse und Diskussion

7.5.1. Klassifikationsergebnisse

Wie in Abschnitt 7.4.1 dargestellt wurde, wurde ein Proband aus der Studie ausgeschlossen. Die Daten von zwei Probanden dienten zur Voranalyse und der Erstellung der initialen Trainingsmenge. Demnach beziehen sich die hier dargestellten Ergebnisse auf sieben Probanden.

Die korrekte Voraussage der Entscheidungen während der Experimente wurde in 69.73

% (Standardabweichung 2.42 %) aller Trials getroffen. Das Rateniveau des Klassifikators beträgt 52.2 % (Standardabweichung 2.83 %). Setzt man eine Binomialverteilung voraus, zeigt die Vorhersage einen signifikanten Unterschied zum Rateniveau ($p = 0.00384$). In einer postexperimentellen Analyse wurde der Online-Algorithmus ohne die Funktionalität des laufenden Trainings des Klassifikators für alle Probanden angewandt. Somit ist es möglich, den Effekt des kontinuierlichen Trainings abzuschätzen. Die probanden-spezifischen Ergebnisse sind in Abbildung 7.10 dargestellt. Erwartungsgemäß führte das kontinuierliche Training des Klassifikators bei allen Probanden zu einer höheren Klassifikationsrate. Die durchschnittliche Steigerung der Klassifikationsrate beträgt 10,35 %. In

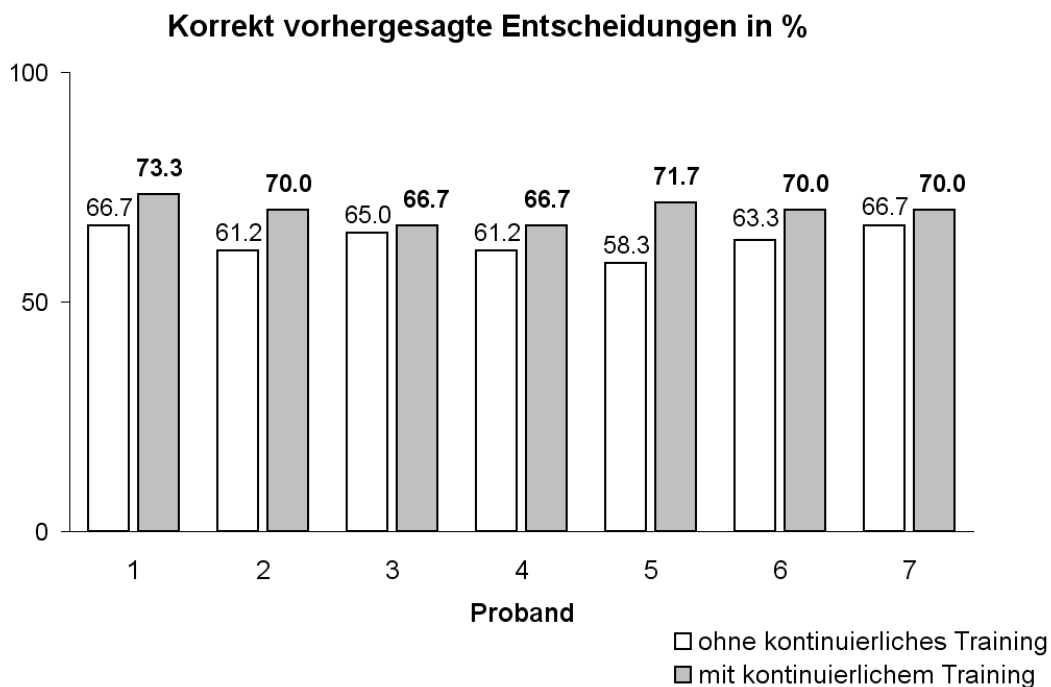


Abbildung 7.10.: Anzahl korrekt vorhergesagter Entscheidungen während der Experimente (Echtzeit-Klassifikation mit kontinuierlichem Training) im Vergleich zur korrekten postexperimentellen Klassifikation ohne kontinuierliches Training

der Offline-Klassifikation wurden mit 87.84 % (Standardabweichung 5.32 %) mehr Entscheidungen als im Online-Experiment korrekt erkannt. Das Rateniveau beträgt 54.31 % (Standardabweichung 3.28 %). Abbildung 7.11 zeigt die Klassifikationsgüte für die sieben Probanden. Die höhere Klassifikationsgüte kann auf den erweiterten Informationsraum zurückgeführt werden, der hier verwendet wurde. Zum einen wurden statt drei Merkmalen in der Online-Methodik zwischen 8000 und 10000 Merkmale verwendet. Zum anderen diente in der LOOCV die maximale Trialanzahl von 59 Trials als Trainingsmenge für die Klassifikation eines einzelnen Trials. Bei der initialen Trainings-

menge im Online-Fall handelt es sich nicht um probandenspezifische Daten, sondern um Daten von anderen Individuen. Im Gegensatz dazu waren die Trainingsdaten in der Offline-Klassifikation probandenspezifisch und tragen somit mehr Informationen zu individuellen Aktivierungsmustern in sich.

Bei Klassifikationsverfahren ist es möglich die Merkmale zu extrahieren, welche der Klassifikator vorrangig verwendet hat, um die verschiedenen Klassen im Merkmalsraum zu trennen (vgl. Abschnitt 2.2.5). In dieser Analyse wurde geprüft, ob motorische oder supplementärmotorische Areale als prädiktive Merkmale in der Offline-Klassifikation verwendet wurden. Dies war nicht der Fall. Es wird geschlussfolgert, dass das Randomisieren der Tasten für Annahme bzw. Ablehnung (vgl. Abschnitt 7.4.1) den gewünschten Effekt hatte und motorische Prozesse aufgrund der Vorbereitung der Probanden auf das Drücken der Tasten nicht in die Klassifikation eingingen.

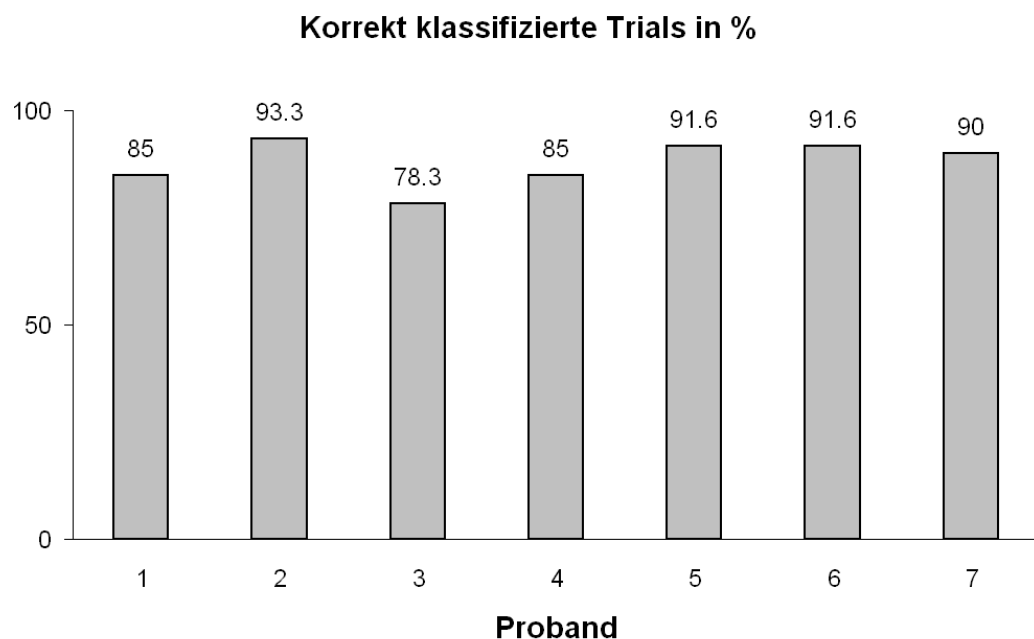


Abbildung 7.11.: Korrekt klassifizierte Trials in der Offline-Analyse

7.5.2. Verhaltensbezogene Ergebnisse

Die vorliegende Implementierung des Ultimatum Games ist iterativ gestaltet. Dies bedeutet, dass der Angebotsnehmer auf mehrere Angebote ein und desselben Anbieters reagiert. Es wurde gezeigt, dass in dieser Variante des UG eine höhere Ablehnungsrate vorgefunden wird, als in der „Single-Shot“-Version, in welcher jeder Anbieter nur ein

Angebot an einen Angebotsnehmer richten kann (Darcet u. Sornette [2008]; Gueth et al. [1982]) (vgl. auch Abschnitt 7.3). Dieser Umstand spiegelt sich in den verhaltensbezogenen Ergebnissen wider, welche in Abbildung 7.12 dargestellt sind. Unfaire Angebote (unter 30 % des Startbetrages) wurden in den Experimenten in 86,97 % der Fälle abgelehnt. In der iterativen Variante des UG findet sich also eine strategische Komponente,

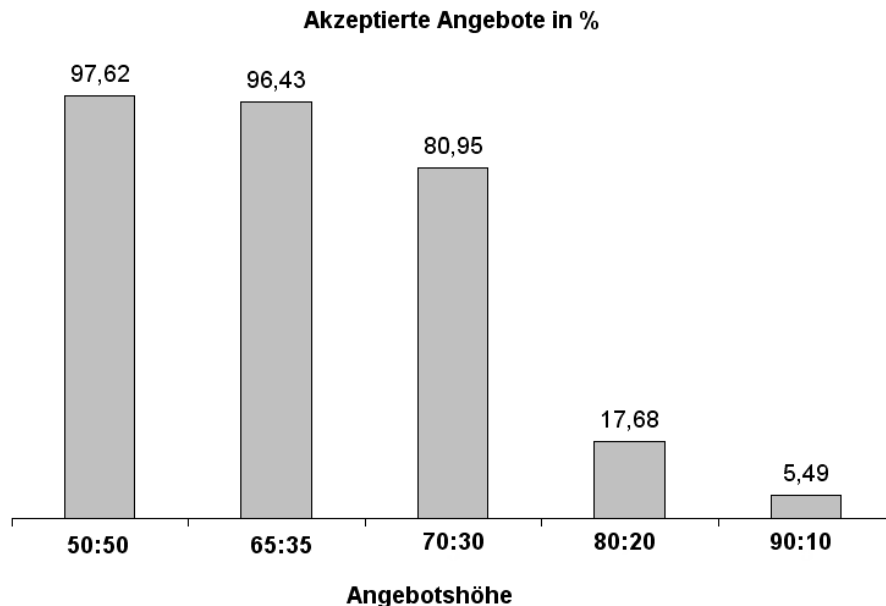


Abbildung 7.12.: Prozentuale Akzeptanzrate der Angebote bezogen auf die Höhe der Angebote. Die Angaben beziehen sich auf sieben Probanden und demnach auf insgesamt 420 Angebote.

weil Probanden versuchen die zukünftigen Angebote ihres Gegenüber durch ihr Annahmeverhalten zu beeinflussen. Dieser Umstand muss bei der Interpretation der neuronalen Korrelate, welche den hier verwendeten Methoden zugrunde liegen beachtet werden. Eine weitere Ursache der höheren Ablehnungsrate in diesem Experiment könnte sich in dem Umstand finden, dass nur männliche Probanden interagierten und diese sich vor dem Experiment persönlich vorgestellt wurden. Zum Beispiel stellten Eckel u. Grossman [1996] Unterschiede in der Bewertung von relativer Fairness zwischen Männern und Frauen fest. Männer neigten dabei stärker zu kategorischen Beurteilungen, während Frauen stärker rational geprägt waren. Die kategorische Einstufung von Angeboten als „zu niedrig“ kann dabei zu erhöhten Ablehnungsraten führen. In den meisten spieltheoretischen Umsetzungen des UG's sind die Geschlechter gleich verteilt und die Spielpartner sind sich nicht bekannt (Fehr u. Schmidt [1999]). Das Geschlecht der Spieler spielt in diesen Versuchen somit keine Rolle.

7.5.3. Diskussion und Ausblick

Ziel dieser Studie war es zu zeigen, dass Voraussagen von Probandenentscheidungen in sozialen Interaktionsszenarien möglich sind. Die Bereitstellung der Vorhersage sollte dabei *vor* der Kundgabe der Entscheidung liegen. In den meisten Implementationen im Bereich des „Mind-Reading“ wurden die Vorhersagen nur in Offline-Analysen getroffen (vgl. Davatzikos et al. [2005]; Haynes u. Rees [2005, 2006]; Kamitani u. Tong [2005]; Mitchell et al. [2004]; Mourao-Miranda et al. [2005]; Soon et al. [2008]). In einer Studie von LaConte et al. [2007] wurden zum ersten Mal Klassifikationsmethoden und Echtzeit-fMRT verbunden, um motorische und stimmungsbezogene Zustände (traurig vs. glücklich) von Probanden zu unterscheiden. Diese Studie war jedoch nicht auf die Vorhersage zukünftiger Probandenentscheidungen ausgelegt, vielmehr wurden die Klassifikationsroutinen für die Implementierung von Neurofeedback verwendet. Im Gegensatz zu bisherigen Studien wurde in dieser Arbeit zum ersten mal die *Vorhersage* von Verhalten mit Hilfe der Echtzeit-fMRT implementiert. Der Umstand des „zeitlichen Vorsprungs“ bei der Klassifikation von zukünftigen Entscheidungen soll hier besonders hervorgehoben werden, weil nur dann diese Vorhersage von Relevanz für die jeweilige Situation ist. In einer Verhandlungssituation zum Beispiel kann ein Verhandlungspartner eine Informationen über die Entscheidungstendenz seines Gegenübers nicht mehr verwenden, wenn die betreffende Situation schon in der Vergangenheit liegt.

Von besonderem Interesse im Zusammenhang mit Online-Klassifikation ist die Frage der Trainingsdaten. Wie auch in den vorliegenden Ergebnissen gezeigt wurde, sind probandenspezifische Trainingsdaten (wie hier in der Offline-Klassifikation verwendet) von Vorteil gegenüber nicht probandenspezifischen Daten in der Echtzeit-Klassifikation. Um probandenspezifische Trainingsdaten zu erhalten, muss man jedoch schon vor der Klassifikation die jeweiligen experimentellen Zustände erzeugen, die auch klassifiziert werden sollen (vgl. LaConte et al. [2007]). In der Implementation des Ultimatum Games hätte dies bedeutet, nur einen geringen Anteil der 60 Trials klassifizieren zu können, da erst die entsprechenden Trainingsdaten hätten akquiriert werden müssen. Neben dem Vorteil der sehr schnellen Verarbeitung durch die Reduktion des Merkmalsraumes auf drei Dimensionen, konnte in der umgesetzten Methodik vom ersten Trial an klassifiziert werden. Des Weiteren wurde somit eine emotionale Prekonditionierung der Probanden durch die Trainingsphase vermieden. Um die Prekonditionierung zu vermeiden, könnte man die Trainingsphase zeitlich stärker von der Klassifikationsphase trennen. Man müsste also mit einer Pause von mehreren Tagen das Experiment wiederholen. Dies birgt im wesentlichen zwei Risiken. Zum einen muss dann die korrekte Registrierung der zu klassifizierenden Messdaten auf die individuellen Trainingsdaten sichergestellt werden, da

einzelne Voxel als getrennte Merkmale verwendet werden. Aus dem gleichen Grund muss die a-priori Annahme gelten, dass genau die gleichen neuronalen Muster durch das Klassifikationsexperiment erzeugt werden, welche auch zum mehrere Tage zurückliegenden Training des Klassifikators benutzt wurden. Es ist vorstellbar, niedrig dimensionierte, nicht probandenspezifische Merkmalsräume für die Klassifikation von Probandenzuständen zu Beginn eines Experimentes zu verwenden, und die Leistungsfähigkeit der Methode zu verstärken, in dem man im weiteren Verlauf des Experimentes auch probandenspezifische hochdimensionierte Merkmalsräume in die endgültige Voraussage einfließen lässt. In solch einem hybriden System ist es möglich, die Entscheidung eines Klassifikators der den hochdimensionalen Raum verarbeitet für spätere Trials stärker zu wichten als die Ergebnisse des besonders für den Beginn wichtigen Klassifikators im niedrigdimensionalen, nicht probandenspezifischen Raum.

Die Analyse der Ergebnisse der Online-Klassifikation zeigte, dass ein Signalanstieg in anteriorer Insula und LPFC vom Klassifikator als Merkmale verwendet wurden, welche auf die Ablehnung eines Angebotes hindeuten. Umgekehrt deutete ein Signalanstieg im occipitalen Kortex auf die Annahme eines Angebotes durch den Probanden. Es wird geschlossen, dass unfaire Angebote mit autonomen Reaktionen auf als negativ eingeordnete Stimuli verbunden sein können. Solch negative Stimuli können Ekel, physischer Schmerz oder auch Angst sein, welche bereits mit Aktivierung der anterioren Insula in Verbindung gebracht wurden (Botvinick et al. [2005]; Caria et al. [2007]; Lieberman [2007]; Wicker et al. [2003]). Untersuchungen im Bereich der Theory of Mind lassen vermuten, dass auch soziale Rückweisung und als unfair empfundenen Verhalten eines sozialen Partners als negative Stimuli in diesem Sinne betrachtet werden können und ebenso Aktivierung der anterioren Insula nach sich ziehen (Rilling et al. [2004]; Sanfey [2007]; Singer et al. [2005]). In diesem Zusammenhang bekommt die persönliche Vorstellung der Probanden vor einem Experiment ihre besondere Bedeutung, da davon ausgegangen wird, dass solch eine persönliche Beziehung die beschriebenen autonomen Reaktionen verstärken kann (Sanfey [2007]; Sanfey et al. [2003]). Neuronale Aktivität im Bereich des LPFC wurde bereits mehrfach mit Funktionen wie Selbstregulation von autonomen Reaktionen und der Fokussierung von Probanden auf eine bestimmte Aufgabe in Verbindung gebracht (Aron et al. [2005]; MacDonald et al. [2000]; Rilling et al. [2004]). Ein Signalanstieg im LPFC bei unfairen Angeboten könnte genau diese höhere Funktion der Selbstregulierung widerspiegeln und Teil der Wirkung des strategischen Denkens der Probanden sein (vgl. auch Gallagher u. Frith [2003] und Samson et al. [2005]).

In einer beträchtlichen Anzahl von Studien wurde der enge Zusammenhang von Aufmerksamkeit und kortikaler Aktivität aufgrund visueller Stimuli belegt (Maunsell u. Cook

[2002]; Maunsell u. McAdams [2001]; Pashler [1998]). Dabei wurde herausgefunden, dass erhöhte visuelle Aufmerksamkeit auch erhöhte neuronale Antworten des visuellen Kortex auslöst. Das Signalanstiege im occipitalen Kortex vom hier verwendeten Klassifikator als Merkmal positiver Angebote eingestuft wurden, könnte genau diesen Zusammenhang widerspiegeln. Positive Angebote bedeuten für den Probanden eine sichere Auszahlung und können somit seine Aufmerksamkeit auf die visuelle Präsentation dieser Angebote erhöhen. Eine Folge wäre erhöhte Aktivität in frühen visuellen Arealen im occipitalen Kortex. Die weitere Analyse der vorliegenden Daten wird zeigen, welche funktionalen Gebiete für den Offline-Klassifikator besonderen Informationsgehalt bezüglich der Probandenentscheidung hatten und wird weitere Schlüsse auf neuronale Korrelate in diesem Paradigma ermöglichen.

Diese Betrachtungen ziehen die Frage nach sich, wie spezifisch die vorgestellten Ergebnisse für das verwendete Paradigma sind, inwiefern man also Aussagen über die allgemeingültige Anwendung ableiten kann. Neuronale Aktivierung im visuellen Kortex kann als spezifisch für den vorliegenden Experimentaufbau angesehen werden, da die Angebote visuell präsentiert wurden. Wenn also als positiv oder salient⁴ empfundene Reize die visuelle Aufmerksamkeit erhöhen, können Klassifikatoren Informationen aus einem Signalanstieg im primären visuellen Kortex ziehen. Unter Berücksichtigung der vorliegenden experimentellen Bedingungen kann dieser Zusammenhang von Nutzen sein, ebenso aber auch störend wirken. Autonome Reaktionen sowie höhere Kontrollfunktionen und die Detektion von Konflikten finden sich als Komponenten in vielen Paradigmen der sozialen Interaktion (Lieberman [2007]). Für die Detektion von internen Konflikten und autonomer Erregung bei sozialer Interaktion scheint Aktivierung der anterioren Insula und dem dorsalen ACC besonders prädiktiv zu sein, was sich in Ergebnissen von Experimenten mit Interaktionsszenarien widerspiegelt (MacDonald et al. [2000]; Rilling et al. [2004]; Singer et al. [2005]). Für diese Experimente ist die Möglichkeit des Online-Zugangs zu Informationen über mentale Zustände von Probanden besonders interessant. Es scheint z.B. möglich, mit der vorliegenden Technik Informationen über den Erregungszustand eines Partners an Kommunikationspartner weiterzugeben und zu untersuchen wie diese „augmented communication“ sich auf das Interaktionsverhalten selbst auswirkt. Ebenso könnte man mit Hilfe dieser Technik untersuchen, inwieweit Probanden in der Lage sind eigene Intentionen bzw. emotionale Zustände vor solch einem System zu verbergen. Dies würde die Anwendung von Hirn-Computer Schnittstellen, welche zum Lernen der Regulation von Hirnaktivierung verwendet werden, erweitern (DeCharms [2008]; Posse et al. [2003]; Weiskopf et al. [2007]).

⁴Als *salient* bezeichnet man ein gegenüber seinem Kontext besonders hervorgehobenes Ereignis.

In diesem Zusammenhang ist die Frage interessant, ob sich die vorgestellte Technik auch für Neurofeedback-Anwendungen eignet. LaConte et al. haben bereits gezeigt, dass sich mit Echtzeit-Klassifikation Neurofeedback implementieren lässt (LaConte et al. [2007]). Eine Diskussion der spezifischen Wirkung der Selbstregulation auf bestimmte Hirnareale ist in dieser Arbeit jedoch nicht enthalten. Gerade für Neurofeedback bei neurologischen Störungen ist die Wirkung der Selbstregulation auf definierte Regionen jedoch wichtig, da Hypothesen über die Funktion dieser Areale bestehen (Posse et al. [2003]; Weiskopf et al. [2004b, 2007]). Die Klassifikation unter Einbeziehung aller Voxel des Hirns hingegen berücksichtigt diesen Zusammenhang nicht. Somit ist nicht sichergestellt, welche Merkmale (also Signalveränderungen in Voxeln) den Klassifikator zu einer Entscheidung führen. Der hier vorgestellte Ansatz der Klassifikation mit vordefinierten Regionen ist in diesem Fall von Vorteil, da der Zusammenhang von Signalveränderungen in festen Arealen und Probandenverhalten berücksichtigt wird. Inwiefern regionenbasierte Klassifikation für Neurofeedback nutzbar ist, bedarf der genaueren Untersuchung.

Die zentrale Steuerung der verschiedenen an diesem Experiment beteiligten Systeme (Stimuluspräsentation, Datenanalyse und Datenakquisition) durch ein einziges EDL-File vereinfachte die Durchführung der Experimente und half dabei Fehler bei der Definition der experimentellen Parameter zu vermeiden. Alle hier beschriebenen Online-Verarbeitungsschritte wurden mit Hilfe der in Abschnitt 6.1.2 vorgestellten Schnittstellen in das Echtzeit-System eingebunden, was sich als komfortabler Weg zur Erweiterung der Funktionalität herausgestellt hat. Ein weiterer Vorteil dieses Frameworks besteht darin, dass die Kommunikation von Laufzeitvariablen schon integriert ist. Es wäre also möglich, die Klassifikationsergebnisse an ein System zu übergeben, dass diese visualisiert und für Neurofeedback nutzbar macht. Mit schnellen Algorithmen zur Vorverarbeitung und Klassifikation von hochdimensionalen Merkmalsräumen wird es möglich sein, auch diese für eine Echtzeit-Klassifikation bzw. ein Training während der Messung zu verwenden. Besonders die vorgestellten Methodiken zur Filterung und Merkmalsselektion eignen sich für die Implementierung in parallelen Prozessen und können somit auf Prozessoren mit mehreren Kernen oder Clustersystemen noch stark beschleunigt werden. Eine weitere Offline-Analyse der funktionellen Daten aus diesem Experiment wird zeigen, wie man ggf. den Merkmalsraum weiter einschränken kann, ohne dass man an Klassifikationsgüte einbüßt.

8. Zusammenfassung und Ausblick

Dieses Kapitel dient der Zusammenfassung der Ergebnisse dieser Arbeit. Aufgeworfene Fragen werden diskutiert und zukünftige Entwicklungsmöglichkeiten skizziert.

8.1. Zusammenfassung

Wie gezeigt wurde, kann man mit Hilfe der Echtzeit-fMRT wichtige neurowissenschaftliche Erkenntnisse gewinnen. Diese Arbeit trägt dazu bei, Experimente in der Echtzeit-fMRT zu vereinfachen. Komplexe Zusammenhänge der Parameter eines Experimentes werden abgebildet und für die automatisierte Verarbeitung durch Software zur Verfügung gestellt. Abstrakte Beschreibungen wie EDL und edlRules führen zu Dokumenten, welche maschinell verarbeitet werden können und somit helfen, Kontroll- und Verarbeitungsprozesse zu automatisieren und in der Konsequenz Fehler zu vermeiden.

Die hier vorgestellte Experimentbeschreibung kann nicht nur zur Beschreibung von Abhängigkeiten zwischen Parametern verwendet werden. Mit Hilfe von EDL können Experimentbeschreibungen auch zwischen Nutzern ausgetauscht und für Dokumentationszwecke verarbeitet werden. Eine weitere Anwendungsmöglichkeit ist die Implementierung von Standardmessungen in EDL. Zu solchen Standardmessungen gehören z.B. funktionelle Localizer für bestimmte Hirnareale, wobei Stimulusdesign und Auswertung oft einem festen Muster folgen. Da in einem EDL-File alle nötigen Parameter¹ abgebildet sind, kann im Zusammenspiel mit den hier vorgestellten Applikationen solch eine Lokalisierungsmessung inklusive der Auswertung stark automatisiert werden.

EDL dient als Bindeglied zwischen Systemen, welche an einer fMRT- bzw. Echtzeit-fMRT-Messung beteiligt sind. Da auch Kommunikationsprozesse zwischen den Systemen abgebildet werden können, werden dynamische Experimente vereinfacht. Diese Technik kommt verschiedenen, mit dem vorgestellten Softwareframework umgesetzten Arbeiten zugute (vgl. z.B. Kapitel 7, Hollmann et al. [2008a, 2009, 2007a]; Moench et al. [2008]; Mueller u. Bernarding [2008]; Rieger et al. [2009]).

Um Wissen für Maschinen zugänglich zu machen, sind Formalisierungen notwendig (Bis-

¹Stimulusdesign, Paradigma der Auswertung, Parameter der MR-Akquise etc.

kup [1995]; Zins [2007]). Diese Formalisierungen münden in Konzepten, mit denen Informationen strukturiert und verknüpft werden. Wie gezeigt wurde, nutzen auch EDL und edlRules solche Konzepte um Information zu beschreiben und zu verbinden. Eine natürliche Folge ist, dass die Ausdrucksmöglichkeiten solcher Beschreibungen auf die verwendeten Konzepte beschränkt sind. Um so komplexer und umfangreicher eine Wissensbasis ist, um so mehr Wissen kann beschrieben werden. Genauso steigt aber auch der Aufwand für den Benutzer, wenn er die angesprochenen Konzepte verstehen und erweitern will. In dem hier vorgestellten System macht die Einführung eigener Parameter und das Formulieren von eigenen Regeln durch den Anwender nur Sinn, wenn der benötigte Aufwand geringer ist als der Aufwand für das Implementieren einer individuellen Softwarelösung. Diese Überlegung spricht gegen die Abbildung komplexer Spezialfunktionalität in EDL und unterstützt die hier umgesetzte Technik der Schnittstellen, welche über EDL angeboten werden.

Wichtig für den vorgestellten Ansatz ist, dass die Experimentbeschreibung von den verarbeitenden Applikationen unabhängig ist. Jede Anwendung in der fMRT und Echtzeit-fMRT verwendet intern eigene Parameter, um die gestellten Aufgaben zu erledigen. Es bietet sich also an, über Schnittstellen die EDL-Parameter in den internen Parameterraum einer Anwendung zu überführen. Die in dieser Arbeit beschriebenen Applikationen des EDL-Frameworks nutzen diese Technik. So können auch bereits bestehende Anwendungen EDL verwenden.

Im Rahmen der in Kapitel 7 vorgestellten Studie wurde gezeigt, dass es mit Hilfe der Echtzeit-fMRT möglich ist, Probandenentscheidungen in sozialen Interaktionsszenarien vorherzusagen. Gerade die Technik der Echtzeit-Klassifikation von fMRT-Daten birgt durch ihre hohe Aussagekraft entscheidendes Zukunftspotenzial. Die Nutzbarkeit für Neurofeedback-Anwendungen wurde in Abschnitt 7.5.3 bereits diskutiert und wird als ein wichtiges Anwendungsgebiet angesehen, gerade weil Hirnaktivierung in verteilten Netzwerken für Feedback verwendet werden kann. Die angesprochenen Limitierungen hinsichtlich der Spezifität des Feedbacks sind dabei zu berücksichtigen.

Bei der vorgestellten Studie konnten die Vorteile des EDL-Frameworks genutzt werden. Es war über die in Abschnitt 5.5.1 und 6.1.2 beschriebenen Mechanismen sehr einfach möglich, zusätzliche Funktionalität in Form der Klassifikationsroutinen in den Auswerteprozess des *rtExplorers* einzubinden. Gleichzeitig hat diese Anwendung jedoch einen wichtigen Punkt verdeutlicht: Alle Funktionalität, welche über den Aufruf unabhängiger Routinen hinzugefügt wird, entzieht sich der Überprüfung der verwendeten Parameter durch automatische Mechanismen über EDL. Dies gilt solange bis der Nutzer die entsprechenden Regeln in die Regelbasis einpflegt. Im Falle der Klassifikationsroutinen waren

die Parameter der verwendeten Klassifikatoren nicht in EDL abgelegt, sondern im Quellcode der Routinen selbst definiert. Es konnten somit keine Regeln zum Test eventueller Abhängigkeiten dieser Parameter in die Regelbasis aufgenommen werden. In solch einem Fall ist die Prüfung der Konsistenz unabhängiger Parameter wieder dem Nutzer überlassen. Für die Prüfung unabhängiger Parameter ist noch keine überzeugende Lösung gefunden worden.

8.2. Ausblick

Der vorgestellte Ansatz einer Wissensbasis ist noch in der Entwicklung. Er zeigt jedoch bereits, dass wissensbasierte Methoden auch in Software für Experimente in der Neurowissenschaft eingehen können, wie sie das in vielen anderen Bereichen bereits getan haben (vgl. Ansätze wie MDL, GML, MOLGEN und MYCIN). Wie gezeigt wurde, birgt die Beschreibung neurowissenschaftlicher Experimente über eine Wissensbasis viele Vorteile. Ob die Ausdrucksmöglichkeiten einer solchen Beschreibung ausreichen, um einen allgemeingültigen Standard für Experimente darzustellen, bleibt jedoch offen. Nur eine konsequente Untersuchung und Weiterentwicklung kann die Antwort auf diese Frage geben. In diese Überlegungen fließt die Idee ein, die Beschreibung nicht auf Experimente der fMRT oder Echtzeit-fMRT zu beschränken. Auch andere Modalitäten wie PET, MEG oder EEG kann man in EDL einbeziehen, wenn die entsprechenden Konzepte eingebunden werden.

Der vorgestellte Ansatz eignet sich auch für die automatisierte Verwaltung von Experimenten. Jeder Nutzer, der neurowissenschaftliche Experimente implementiert hat, kennt die umfangreichen Aufgaben, welche zur Dokumentation und Datensicherung nötig sind. Viele der für die automatisierte Bearbeitung solcher Aufgaben benötigten Parameter sind in EDL bereits enthalten. Die Integration von Probandendaten könnte ein weiterer Schritt sein diese Funktionalität auszubauen. Hierbei sind jedoch datenschutzrechtliche Einschränkungen zu berücksichtigen und es müssen ggf. Konzepte zur Verschlüsselung von EDL-Inhalten etabliert werden.

Die leichte Erweiterbarkeit der Wissensbasis stellt eine zentrale Eigenschaft des EDL-Ansatzes dar (vgl. Abschnitt 5.7). In der derzeitigen Implementierung ist die Regelbasis in einem `edlRules`-File beschrieben. Der Nutzer fügt hier eigene Regeln hinzu, um weitere Funktionalität zu etablieren. Dies bedeutet, dass systeminherente Regeln nicht von nutzereigenen Regeln getrennt vorliegen. Um Probleme wie das versehentliche Löschen oder Bearbeiten systeminherenter Regeln zu vermeiden, wird in zukünftigen Versionen im EDL-File ein Verweis auf ein nutzereigenes `edlRules`-File möglich sein. Dieses `edlRules`-

File stellt dann natürlich auch eine Ausprägung der XSD-Definition von `edlRules` dar. Entwicklungspotenzial findet sich auch bei den Applikationen des EDL-Frameworks. Wie in der Nutzerstudie deutlich wurde, ist die Funktionalität zum Erstellen von Experimenten noch nicht zufriedenstellend. Dies wird in unten stehendem Abschnitt 8.2.1 noch einmal separat betrachtet. Wichtig für das Grundanliegen der Arbeit ist der Ausbau der Funktionalitäten zur Simulation von Echtzeit-fMRT-Experimenten. Zur Zeit können lediglich künstliche Simulationsdaten erzeugt werden. Es ist jedoch vorstellbar, noch mehr Informationen der EDL-Beschreibung zu nutzen und z.B. komplette Kommunikationsvorgänge zu simulieren. Dazu wäre es denkbar, eine Simulationsumgebung als eigenständige Applikation zu etablieren, welche Simulationsdaten zur Verfügung stellt und den Gesamtprozess koordiniert.

8.2.1. Erstellungsmöglichkeiten für Experimente

In der hier verwendeten Applikation, dem *EDL-Editor*, bleiben einige Möglichkeiten hinsichtlich der Nutzerführung ungenutzt. Neben der Anwendung der zielorientierten Inferenz zum Prüfen der Parameterzusammenhänge während der Eingabe, gibt es effektive grafische Methoden um Zusammenhänge und Parametergrenzen zu visualisieren (vgl. Dahm [2005]; Preim [1999]; Strothotte [1998]). Weiterhin ist es vorstellbar, auch die Erstellung der Parameter der Stimuluspräsentation über interaktive Funktionalität zu unterstützen. Diese Methoden vermeiden die textuelle Repräsentation von Informationen, wenn diese intuitiver über grafische Objekte dargestellt und manipuliert werden können. Professionelle Tools wie *DirectRT* (Empirisoft [2008]) oder *E-Prime* (Psychology-Software-Tools [2009]) verwenden diese Mechanismen und vereinfachen den Erstellungsprozess der Stimuluspräsentation (Stahl [2006]). Ein intuitives System zum Editieren von EDL- und `edlRules`-Files würde einen leichteren Zugang zu den in EDL verwendeten Konzepten schaffen und somit die Verwendbarkeit und Nutzerfreundlichkeit der Beschreibungssprache erhöhen. In eine Editorumgebung könnten zusätzlich auch oben angesprochene Simulationsmöglichkeiten eingebunden werden.

8.2.2. Neuroökonomische Experimente und Echtzeit-fMRT

Experimente in der Neuroökonomie bieten die Möglichkeit Erkenntnisse in verschiedenen Bereichen wie Ökonomie, Neurowissenschaft, Psychologie und Soziologie zu gewinnen (vgl. Abschnitt 7.3). Es bietet sich daher an, in Verbundprojekten die Expertise der verschiedenen Bereiche zu verknüpfen. In verhaltensökonomischen Experimenten werden oftmals Softwarelösungen in speziellen Laboren eingesetzt, um Experimente mit meh-

renen Teilnehmern zu synchronisieren (Aumann [2002]; Camerer et al. [2003]). Es ist vorstellbar, ein Echtzeit-fMRT-System über Serverlösungen mit solch einer Software zu verbinden. Der hier vorgestellte Ansatz beinhaltet bereits Kommunikationsmöglichkeiten zwischen den Systemen einer Echtzeit-fMRT-Umgebung. Es ist möglich, diese Kommunikation auch zwischen einem Echtzeit-fMRT-System und einer verhaltensökonomischen Anwendung zu etablieren und Probanden an solchen Experimenten teilnehmen zu lassen, während sie sich im MRT-Scanner befinden. Solch ein Ansatz bedarf noch näherer Untersuchung. Die weitreichenden Möglichkeiten einer flexiblen Lösung, wie sie in dieser Arbeit vorgestellt wurde, sind dabei von Nutzen.

Interaktionsszenarien in der Neuroökonomie bilden ein interessantes Anwendungsgebiet für die Echtzeit-fMRT. Der erwartete Nutzen eines Spielers beschränkt sich in typischen Implementierungen solcher Szenarien nicht allein auf seine eigenen Handlungen, sondern auch auf das strategische Verhalten seines Gegenübers. Spezialisierte Nutzenfunktionen in Modellen der Verhaltensökonomie beinhalten demnach auch Strategien und sogenannte „Beliefs“, also Einschätzungen des Verhaltens von Spielpartnern (Battigalli u. Dufwenberg [2009]; Battigalli u. Siniscalchi [2002]; Fehr u. Schmidt [1999]). Wenn ein Spieler nun Informationen über den emotionalen Zustand seines Gegenübers erhält (z.B. durch ein BCI wie es in Kapitel 7 beschrieben wurde), hat dies Einfluss auf seine Einschätzung der Situation. Dieser Umstand birgt vielfältige Ansätze zur Untersuchung psychologischer Szenarien in der Verhaltensökonomie und bietet damit ein zukünftiges Anwendungsgebiet der Echtzeit-fMRT.

8.2.3. Echtzeit-Klassifikation und Neurofeedback

In Abschnitt 7.5.3 wurde angedeutet, dass die Nutzbarkeit der Echtzeit-Klassifikation für Neurofeedback-Anwendungen der näheren Untersuchung bedarf. In der Studie von LaConte et al. [2007] wurde diese Technik für Neurofeedback verwendet, es bleibt jedoch offen, inwiefern sich die lokal unspezifische Antwort des Systems und die Rückmeldung von lediglich zwei Zuständen (Klasse 1 bzw. Klasse 2) auf Lernerfolge auswirken. Gerade der letzte Punkt scheint kritisch für konditioniertes Lernen zu sein, da eine qualitativ genauere Bewertung (z.B. eine Skala 1 bis 10) dem Probanden mehr Raum für die Einschätzung seiner Leistung und damit seiner Strategieentwicklung lässt. Nahezu alle betrachteten Neurofeedback-Studien (vgl. Abschnitt 3.2) verwenden erweiterte Skalen anstatt die Ausgabe einer Klasse, wie sie originär von Klassifikatoren geliefert werden. Eine Lösung liegt in der Extraktion von qualitativen Aussagen über Merkmale, wie z.B. den Abstand von der trennenden Hyperebene bei SVM-Klassifikation oder die Wahrscheinlichkeit der Klassenzugehörigkeit von Objekten im Merkmalsraum (vgl. Abschnitt

2.2.5 und Pessoa u. Padmala [2005]; Rieger et al. [2008]; Schölkopf et al. [1999]). Die Untersuchung von Feedbackmöglichkeiten mit Echtzeit-Klassifikatoren eröffnet ein weiteres interessantes Gebiet zukünftiger Forschung.

Literaturverzeichnis

- Aguirre et al. 1998** AGUIRRE, G.K. ; ZARAHN, E. ; D'ESPOSITO, M.: The variability of human, bold hemodynamic responses. In: *NeuroImage* 8 (1998), S. 360–369
- Aguirre u. D'Esposito 1999** *Kapitel 30*. In: AGUIRRE, G.K. ; D'ESPOSITO, M.: *Experimental Design for Brain fMRI*. Springer-Verlag, 1999
- Altenkrüger u. Büttner 1992** ALTENKRÜGER, D. ; BÜTTNER, W.: *Wissensbasierte Systeme*. Vieweg, 1992
- Armstrong 1973** ARMSTRONG, D.M.: *Belief, Truth, and Knowledge*. Cambridge University Press, 1973
- Aron et al. 2005** ARON, A. ; FISHER, H. ; MASHEK, D.J. ; STRONG, G. ; LI, H. ; BROWN, L.L.: Reward, motivation, and emotion systems associated with early-stage intense romantic love. In: *Journal of Neurophysiology* 94 (2005), S. 327–337
- Ashburner u. Friston 1999** *Kapitel 26*. In: ASHBURNER, J. ; FRISTON, K.J.: *Image Registration*. Springer-Verlag, 1999
- Aumann 2002** AUMANN, R.J.: *Handbook of Game Theory with Economic Applications*. Bd. 3. North-Holland, 2002
- Baert et al. 1999** BAERT, A.L. ; HEUCK, F.H.W. ; YOUKER, J.E.: *Functional MRI*. Springer-Verlag, 1999
- Battigalli u. Siniscalchi 2002** BATTIGALLI, P. ; SINISCHALCHI, M.: Strong Belief and Forward Induction Reasoning. In: *Journal of Economic Theory* 106 (2002), S. 356–391
- Battigalli u. Dufwenberg 2009** BATTIGALLI, P. ; DUFWENBERG, M.: Dynamic psychological games. In: *Journal of Economic Theory* 144 (2009), Nr. 1, S. 1–35
- Beierle u. Kern-Isberner 2008** BEIERLE, C. ; KERN-ISBERNER, G.: *Methoden wissenschaftlicher Systeme*. Vieweg und Teubner, 2008

- Bernarding 1997** BERNARDING, J.: *Implementierung kernspintomographischer Messsequenzen zur rechnergestützten Diffusions-und Relaxationsbildgebung*. 1997. – Dissertation am Institut für Medizinische Statistik, Epidemiologie und Informatik der Freien Universität Berlin
- Birbaumer u. Schmidt 2006** BIRBAUMER, N. ; SCHMIDT, R.F.: *Biologische Psychologie*. Springer Medizin Verlag Heidelberg, 2006
- Biskup 1995** BISKUP, J.: *Grundlagen von Informationssystemen*. Vieweg, 1995
- Bloch et al. 1946** BLOCH, F. ; HANSEN, W.W. ; PACKARD, M.E.: Nuclear induction. In: *Physiological Review* 69 (1946), S. 127–128
- Botvinick et al. 2005** BOTVINICK, M. ; JHA, A.P ; BYLSMA, L.M. ; FABIAN, S.A. ; SOLOMON, P.E. ; PRKACHIN, K.M.: Viewing facial expressions of pain engages cortical areas involved in the direct experience of pain. In: *Neuroimage* 25 (2005), S. 312–319
- Boulton u. Goguen 1979** BOULTON, P.I.P. ; GOGUEN, J.R.: A machine description language. In: *The Computer Journal* 22 (1979), Nr. 2, S. 132–135
- Boynton et al. 1996** BOYNTON, G.M. ; ENGEL, S.A. ; GLOVER, G.H. ; HEEGER, D.J.: Linear systems analysis of functional magnetic resonance imaging in human V1. In: *Journal of Neuroscience* 16 (1996), Nr. 13, S. 4207–4221
- Brachman u. Levesque 2004** BRACHMAN, R.J. ; LEVESQUE, H.J.: *Knowledge Representation and Reasoning*. Elsevier, 2004
- BrainVoyager 2009** BRAINVOYAGER: *BrainVoyager - A product from Brain Innovation*. <http://www.brainvoyager.com/>, 2009. – Abgerufen am 10.11.2008
- Brown u. Semelka 2003** BROWN, M.A. ; SEMELKA, R.C.: *MRI Basic Principles and Applications*. Wiley-Liss, 2003
- Buchanan u. Shortliffe 1984** BUCHANAN, B.G. ; SHORTLIFFE, E.H.: *Rule Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, 1984
- Buxton et al. 1998** BUXTON, R.B. ; WONG, E.C. ; FRANK, L.R.: Dynamics of blood flow and oxygenation changes during brain activation: The balloon model. In: *Magnetic Resonance in Medicine* 39 (1998), Nr. 6, S. 855–864

- Camerer et al. 2003** CAMERER, C.F. ; LOEWENSTEIN, G. ; M.RABIN: *Advances in Behavioral Economics*. Princeton University Press, 2003
- Canet 1994** CANET, D.: *NMR-Konzepte und Methoden*. Springer Verlag, 1994
- Caria et al. 2007** CARIA, A. ; VEIT, R. ; SITARAM, R. ; LOTZE, M. ; WEISKOPF, N. ; GRODD, W. ; BIRBAUMER, N.: Regulation of anterior insular cortex activity using real-time fMRI. In: *NeuroImage* 35 (2007), Nr. 3, S. 1238–1246
- Chang u. Lin 2001** CHANG, C.-C. ; LIN, C.-J.: *LIBSVM: A Library for Support Vector Machines*. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001. – Abgerufen am 12.03.2009
- Clare 1997** CLARE, S.: *Phd Thesis, Stuart Clare, FMRI*. <http://www.fmrib.ox.ac.uk/~stuart/thesis/index.html>, 1997. – Dissertationschrift an der Universität von Nottingham. Abgerufen am 10.03.2008
- COGENT 2000** COGENT: *COGENT, 2000. Laboratory of neurobiology. University College London*. <http://www.vislab.ucl.ac.uk/Cogent2000/index.html>, 2000. – Abgerufen am 20.03.2007
- Cox et al. 1995** COX, R.W. ; JESMANOWICZ, A. ; HYDE, J.S.: Real-time functional magnetic resonance imaging. In: *Magnetic Resonance in Medicine* 33 (1995), S. 230–236
- Cox 1996** COX, R. W.: AFNI: Software for analysis and visualization of functional magnetic resonance neuroimages. In: *Computers and Biomedical Research* 29 (1996), Nr. 3, S. 162–173
- Cristianini u. Shawe-Taylor 2004** CRISTIANINI, N. ; SHAWE-TAYLOR, J.: *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004
- Critchley 2005** CRITCHLEY, H.D.: Neural mechanisms of autonomic, affective, and cognitive integration. In: *Journal of Comparative Neurology* 493 (2005), Nr. 1, S. 154–166
- Dahm 2005** DAHM, M.: *Grundlagen der Mensch-Computer-Interaktion*. Verlag Pearson Studium, 2005

- Darcet u. Sornette 2008** DARCET, D. ; SORNETTE, D.: Quantitative determination of the level of cooperation in the presence of punishment in three public good experiments. In: *Journal of Economic Interaction and Coordination* 3 (2008), Nr. 2, S. 137–163
- Davatzikos et al. 2005** DAVATZIKOS, C. ; RUPAREL, K. ; FAN, Y. ; SHEN, D.G. ; ACHARYYA, M. ; LOUGHEAD, J.W. ; GUR, R.C. ; LANGLEBEN, D.D.: Classifying spatial patterns of brain activity with machine learning methods: Application to lie detection. In: *NeuroImage* 28 (2005), Nr. 3, S. 663–668
- DeCharms et al. 2004** DECHARMS, R.C. ; CHRISTOFF, K. ; GLOVER, G.H. ; PAULY, J.M. ; WHITFIELD, S. ; GABRIELI, J.D.E.: Learned regulation of spatially localized brain activation using real-time fMRI. In: *NeuroImage* 21 (2004), S. 436–443
- DeCharms et al. 2005** DECHARMS, R.C. ; MAEDA, F. ; GLOVER, G.H. ; PAULY, J.M. ; LUDLOV, D. ; GABRIELI, J.D.E. ; SONEJI, D. ; MACKEY, S.C.: Control over brain activation and pain learned by using real-time functional MRI. In: *PNAS* 102 (2005), S. 18626–18631
- DeCharms 2008** DECHARMS, C.: Applications of real-time fMRI. In: *Nature Reviews Neuroscience* 9 (2008), S. 720–729
- DeMartino et al. 2006** DEMARTINO, B. ; KUMARAN, D. ; SEYMOUR, B. ; DOLAN, R.J.: Frames, Biases, and Rational Decision-Making in the Human Brain. In: *Science* 313 (2006), S. 684–687
- Dössel 2000** DÖSSEL, O.: *Bildgebende Verfahren in der Medizin - Von der Technik zur medizinischen Anwendung*. Springer-Verlag, 2000
- Duda et al. 2001** DUDA, R.O. ; HART, P.E. ; STORK, D.G.: *Pattern Classification*. Wiley, 2001
- Eckel u. Grossman 1996** ECKEL, C.C. ; GROSSMAN, P.J.: The Relative Price of Fairness: Gender Differences in a Punishment Game. In: *Journal of Economic Behavior & Organization* 30 (1996), S. 143–158
- Eisenberger et al. 2003** EISENBERGER, N.I. ; LIEBERMAN, M.D. ; WILLIAMS, K.D.: Does rejection hurt? An fMRI study of social exclusion. In: *Science* 302 (2003), S. 290–292

Empirisoft 2008 EMPIRISOFT: *Direct RT*.

<http://www.empirisoft.com>, 2008. – Abgerufen am 05.07.2008

Evans et al. 1993 EVANS, A.C. ; COLLINS, D.L. ; MILLS, S.R. ; BROWN, E.D. ; KELLY, R.L. ; PETERS, T.M.: 3D statistical neuroanatomical models from 305 MRI volumes. In: *Proceedings of the IEEE -Nuclear Science Symposium and Medical Imaging Conference*, 1993

Fehr u. Schmidt 1999 FEHR, E. ; SCHMIDT, K.M.: A theory of fairness, competition, and cooperation. In: *Quarterly Journal of Economics* 114 (1999), Nr. 3, S. 817–868

Frackowiak et al. 2004 FRACKOWIAK, R.S.J. ; FRISTON, K.J. ; FRITH, C.D. ; DOLAN, R.J. ; PRICE, C.J. ; ZEKI, S. ; ASHBURNER, J. ; PENNY, W.: *Human Brain Function*. Elsevier Academic Press, 2004 (Second Edition)

Friston et al. 1994 FRISTON, K. ; JEZZARD, P. ; TURNER, R. und: Analysis of functional MRI time-series. In: *Human Brain Mapping* 1 (1994), S. 153–171

Friston et al. 1995 FRISTON, K.J. ; HOLMS, A.P. ; WORSLEY, K.J. ; POLINE, J.B. ; FRITH, C.D. ; FRACKOWIAK, R.S.J.: Statistical Parametric Maps in Functional Imaging - A General Linear Approach. In: *Human Brain Mapping* 2 (1995), S. 189–210

Friston 1998 FRISTON, K.J.: Modes or models: A critique of independent component analysis for fMRI. In: *Trends in Cognitive Sciences* 2 (1998), S. 568–573

Friston et al. 2000 FRISTON, K.J. ; MECHELLI, A. ; TURNER, R. ; PRICE, C.J.: Nonlinear responses in fMRI: The balloon model, Volterra kernels, and other hemodynamics. In: *NeuroImage* 12 (2000), Nr. 4, S. 466–477

Friston 2004 FRISTON, K.J.: Experimental Design and Statistical Parametric Mapping. In: *Human Brain Function*. Elsevier Academic Press, 2004 (Second Edition), Kapitel 31, S. 604–605

Gallagher u. Frith 2003 GALLAGHER, H.L. ; FRITH, C.D.: Functional imaging of 'Theory of Mind'. In: *Trends in Cognitive Sciences* 7 (2003), Nr. 2, S. 77–83

Gembris et al. 2000 GEMBRIS, D. ; TAYLOR, J.G. ; SCHOR, S. ; FRINGS, W. ; SUTER, D. ; POSSE, S.: Functional Magnetic Resonance Imaging in Real Time (FIRE): Sliding-Window Correlation Analysis and Reference-Vector Optimization. In: *Magnetic Resonance in Medicine* 43 (2000), S. 259–268

- Glimcher et al. 2008** GLIMCHER, P.W. ; CAMERER, C.F. ; FEHR, E. ; POLDRACK, R.A.: *Neuroeconomics: Decision Making and the Brain*. Elsevier Academic Press, 2008
- Good 2005** GOOD, P.: *Permutation, Parametric and Bootstrap Tests of Hypothesis*. Springer-Verlag, New York, 2005
- Grüner et al. 2009** GRÜNER, T. ; KERBER, A. ; LAUE, Re. ; RUCKDESCHEL, A.: *Molecular Structure Generation*. <http://www.molgen.de>, 2009. – Abgerufen am 04.03.09
- Gueth et al. 1982** GUETH, W. ; SCHMITTBERGER, R. ; SCHWARZE, B.: An experimental analysis of ultimatum bargaining. In: *Journal of Economic Behavior and Organization* 3 (1982), Nr. 4, S. 367–388
- Haake et al. 1999** HAAKE, E.M. ; BROWN, R.W. ; THOMPSON, M.R. ; VENKATESAN, R.: *Magnetic Resonance Imaging - Principles and Sequence Design*. Wiley-Liss, 1999
- Haggard u. Eimer 1999** HAGGARD, P. ; EIMER, M.: On the relation between brain potentials and the awareness of voluntary movements. In: *Exp Brain Res* 126 (1999), S. 128–133
- Handwerker et al. 2004** HANDWERKER, D.A. ; OLLINGER, J.M. ; D’ESPOSITO, M.: Variation of BOLD hemodynamic response across subjects and brain regions and their effect on statistical analysis. In: *NeuroImage* 21 (2004), S. 1639–1651
- Haynes u. Rees 2005** HAYNES, J.-D. ; REES, G.: Predicting the orientation of invisible stimuli from activity in human primary visual cortex. In: *Nature Neuroscience* 8 (2005), Nr. 5, S. 686–691
- Haynes u. Rees 2006** HAYNES, J.-D. ; REES, G.: Decoding mental states from brain activity in humans. In: *Nature Reviews Neuroscience* 7 (2006), Nr. 7, S. 523–534
- Henrich et al. 2004** HENRICH, J. ; BOYD, R. ; BOWLES, S. ; CAMERER, C. ; FEHR, E. ; GINTIS, H.: *Foundations of Human Sociality: Economic Experiments and Ethnographic Evidence from Fifteen Small-Scale Societies*. Oxford University Press, 2004
- Hollmann et al. 2007a** HOLLMANN, M. ; MÖNCH, T. ; STADLER, J. ; BERNARDING, J.: A unified approach for data acquisition and classification in real-time fMRI (rfMRI): analysis of single-event experiments at 7 Tesla. In: *Proceedings of the ISMRM*, 2007

- Hollmann et al. 2007b** HOLLMANN, M. ; MÖNCH, T. ; TEMPELMANN, C. ; BERNARDING, J.: A Unified Approach for fMRI-Measurements Used by a New Real-Time fMRI Analysis System. In: *Proceedings of the BVM*, 2007, S. 56–60
- Hollmann et al. 2008a** HOLLMANN, M. ; MÖNCH, T. ; BAECKE, S. ; LUCHTMANN, M. ; BERNARDING, J.: New Experimental Setup using Individual Hemodynamic Response Functions used to Increase the Statistical Power in Event-Related Real-Time fMRI (erfMRI). In: *Proceedings of the GMDS*, 2008
- Hollmann et al. 2008b** HOLLMANN, M. ; MÖNCH, T. ; MULLA-OSMAN, S. ; TEMPELMANN, C. ; STADLER, J. ; BERNARDING, J.: A new concept of a unified parameter management, experiment control, and data analysis in fMRI: Application to real-time fMRI at 3 T and 7 T. In: *Journal of Neuroscience Methods* 175 (2008), Nr. 1, S. 154–162
- Hollmann et al. 2009** HOLLMANN, M. ; MÖNCH, T. ; MÜLLER, C. ; BERNARDING, J.: Predicting Human Decisions in socioeconomic Interaction using Real-Time Functional Magnetic Resonance Imaging (rtfMRI). In: *Proceedings of the SPIE - Medical Imaging*, 2009
- Horowitz 1995** HOROWITZ, A.L.: *MRI Physics for Radiologists - A Visual Approach*. Springer Verlag, 1995 (Second Edition)
- Hu et al. 1995** HU, X. ; LE, T.H. ; PARRISH, T. ; ERHARD, P.: Retrospective Estimation and Correction of Physiological Fluctuation in Functional MRI. In: *Magnetic Resonance in Medicine* 35 (1995), S. 201–212
- Huettel et al. 2004** HUETTEL, S.A. ; SONG, A.W. ; MCCARTHY, G.: *Functional Magnetic Resonance Imaging*. Sinauer, 2004
- Iadecola et al. 1997** IADECOLA, C. ; YANG, G. ; EBNER, T.J. ; CHEN, G. und: Local and propagated vascular responses evoked by focal synaptic activity in cerebellar cortex. In: *Neurophysiology* 78 (1997), Nr. 2, S. 651–659
- Invivo 2009** INVIVO: *Eloquence/IFIS fMRI-System*.
<http://www.invivocorp.com/fmri/eloquence.php>, 2009. – Abgerufen am 14.01.09
- Jezzard u. Balaban 1995** JEZZARD, P. ; BALABAN, R. S.: Correction for geometric distortion in echo planar images from B(o) field variations. In: *Magnetic Resonance in Medicine* 34 (1995), Nr. 1, S. 65–73

- Kamitani u. Tong 2005** KAMITANI, Y. ; TONG, F.: Decoding the visual and subjective contents of the human brain. In: *Nature Neuroscience* 8 (2005), Nr. 5, S. 679–685
- Kennan 1999** *Kapitel 12*. In: KENNAN, R.P.: *Gradient Echo and Spin Echo Methods for Functional MRI*. Springer-Verlag, 1999
- Kenning et al. 2007** KENNING, P. ; PLASSMANN, H. ; AHLERT, D.: Applications of functional magnetic resonance imaging for market research. In: *Qualitative Market Research* 10 (2007), Nr. 2, S. 135–152
- LaConte et al. 2007** LACONTE, S.M. ; PELTIER, S.J. ; HU, X.P.: Real-time fMRI using brain-state classification. In: *Human Brain Mapping* 28 (2007), Nr. 10, S. 1033–1044
- Lange 1999** *Kapitel 27*. In: LANGE, N.: *Statistical Procedures for Functional MRI*. Springer-Verlag, 1999
- Lieberman 2007** LIEBERMAN, M.D: Social cognitive neuroscience: A review of core processes. In: *Annual Review of Psychology* 58 (2007), S. 259–289
- Lobin 2000** LOBIN, H.: *Informationsmodellierung in XML und SGML*. Springer Verlag, 2000
- MacDonald et al. 2000** MACDONALD, A.W. ; COHEN, J.D. ; STENGER, V.A. ; CARTER, C.S.: Dissociating the role of the dorsolateral prefrontal and anterior cingulate cortex in cognitive control. In: *Science* 288 (2000), S. 1835–1838
- MacDonald u. Paus 2003** MACDONALD, P.A. ; PAUS, T.: The role of parietal cortex in awareness of self-generated movements: a transcranial magnetic stimulation study. In: *Cerebral Cortex* 13 (2003), S. 962–967
- Mathiak u. Posse 2001** MATHIAK, K. ; POSSE, S.: Evaluation of motion and realignment for functional magnetic resonance imaging in real time. In: *Magnetic Resonance in Medicine* 45 (2001), S. 167–171
- Matlab 2009** MATLAB: *Matlab - The Language of Technical Computing*. <http://www.mathworks.com/products/matlab>, 2009. – Abgerufen am 25.01.2009
- Maunsell u. McAdams 2001** MAUNSELL, J.H.R. ; MCADAMS, C.J.: Effects of attention on the responsiveness and selectivity of individual neurons in visual cerebral cortex. In: *Visual Attention and Cortical Circuits* (2001), S. 103–119

- Maunsell u. Cook 2002** MAUNSELL, J.H.R. ; COOK, E.P.: The role of attention in visual processing. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 357 (2002), Nr. 1424, S. 1063–1072
- Mayo-Research 2008** MAYO-RESEARCH: *Analyze Data Format*.
<http://mayoresearch.mayo.edu/mayo/research/robblab/analyze.cfm>, 2008. – Abgerufen am 03.11.2008
- Mitchell et al. 2004** MITCHELL, T.M. ; HUTCHINSON, R. ; NICULESCU, R.S. ; PEREIRA, F. ; WANG, X. ; JUST, M. ; NEWMAN, S.: Learning to decode cognitive states from brain images. In: *Machine Learning* 57 (2004), Nr. 1-2 SPEC. ISS., S. 145–175
- Moench et al. 2008** MOENCH, T. ; HOLLMANN, M. ; GRZESCHIK, R. ; MUELLER, C. ; LUETZKENDORF, R. ; BAECKE, S. ; LUCHTMANN, M. ; WAGEGG, D. ; BERNARDING, J.: Real-time classification of activated brain areas for fMRI-based human-brain-interfaces. In: *Proceedings of the SPIE - Medical Imaging* Bd. 6916, 2008
- Mourao-Miranda et al. 2005** MOURAO-MIRANDA, J. ; BOKDE, A.L.W. ; BORN, C. ; HAMPEL, H. ; STETTER, M.: Classifying brain states and determining the discriminating activation patterns: Support Vector Machine on functional MRI data. In: *NeuroImage* 28 (2005), Nr. 4, S. 980–995
- Mueller 2007** MUELLER, C.: *Intelligent-Imaging: Entwurf und Implementierung eines Werkzeuges zur semi-automatischen Optimierung in der MR-Bildgebung*. Diplomarbeit am Institut für Biometrie und Medizinische Informatik, Medizinische Fakultät, Universität Magdeburg, 2007
- Mueller u. Bernarding 2008** MUELLER, C. ; BERNARDING, J.: Intelligent-Imaging: Entwurf und Implementierung eines Werkzeuges zur semi-automatischen Optimierung in der MR-Bildgebung. In: *Proceedings of the DSISMRM*, 2008
- Nakai et al. 2006** NAKAI, T. ; BAGARINAO, E. ; MATSUO, K. ; OHGAMIC, Y. ; KATO, C.: Dynamic monitoring of brain activation under visual stimulation using fMRI - The advantage of real-time fMRI with sliding window GLM analysis. In: *Journal of Neuroscience Methods* 157 (2006), S. 158–167
- NEMA 2008** NEMA: *DICOM - Digital Imaging and Communications in Medicine*.
<http://medical.nema.org>, 2008. – Abgerufen am 02.03.2009

- Ochsner et al. 2005** OCHSNER, K.N. ; BEER, J.S. ; ROBERTSON, E.R. ; COOPER, J.C. ; GABRIELI, J.D.: The neural correlates of direct and reflected self-knowledge. In: *NeuroImage* 28 (2005), S. 797–814
- Ochsner u. Gross 2005** OCHSNER, K.N. ; GROSS, J.J.: The cognitive control of emotion. In: *Trends in Cognitive Science* 9 (2005), S. 242–249
- Ogawa et al. 1990** OGAWA, S. ; LEE, T.M. ; KAY, A.R. ; TANK, D.W.: Brain magnetic resonance imaging with contrast dependent on blood oxygenation. In: *Proceedings of the National Academy of Sciences of the United States of America* Bd. 87, 1990, S. 9868–9872
- OGC 2009** OGC: *Geography Markup Language (Open Geospatial Consortium)*. <http://www.opengeospatial.org/standards/gml>, 2009. – Abgerufen am 04.03.2009
- O’Toole et al. 2007** O’TOOLE, A.J. ; JIANG, F. ; ABDI, H. ; PENARD, N. ; DUNLOP, J.P. ; PARENT, M.A.: Theoretical, Statistical, and Practical Perspectives on Pattern-based Classification Approaches to the Analysis of Functional Neuroimaging Data. In: *Journal of Cognitive Neuroscience* 19 (2007), S. 1735–1752
- Owen et al. 2006** OWEN, A.M. ; COLEMAN, M.R. ; BOLY, M. ; DAVIS, M.H. ; LAUREYS, S. ; PICKARD, J.D.: Detecting awareness in the vegetative state. In: *Science* 313 (2006), S. 1402
- Pashler 1998** PASHLER, H.E.: *The Psychology of Attention*. Cambridge MA, 1998
- Pessoa u. Padmala 2005** PESSOA, L. ; PADMALA, S.: Quantitative prediction of perceptual decisions during near-threshold fear detection. In: *Proceedings of the National Academy of Sciences of the USA* 102 (2005), Nr. 15, S. 5612–5617
- Plutchik 1980** *Kapitel A* A general psychoevolutionary theory of emotion. In: PLUTCHIK, R.: *Emotion: Theory, research, and experience*. New York: Academic, 1980, S. 3–33
- Polyn et al. 2005** POLYN, S.M. ; NATU, V.S. ; COHEN, J.D. ; NORMAN, K.A.: Neuroscience: Category-specific cortical activity precedes retrieval during memory search. In: *Science* 310 (2005), Nr. 5756, S. 1963–1966
- Posse et al. 2003** POSSE, S. ; FITZGERALD, D. ; GAO, K. ; HABEL, U. ; ROSENBERG, D. ; MOORE, G.J. ; SCHNEIDER, F.: Real-time fMRI of temporolimbic regions detects amygdala activation during single-trial self-induced sadness. In: *NeuroImage* 18 (2003), März, Nr. 3, S. 760–768

- Preim 1999** PREIM, B.: *Entwicklung interaktiver Systeme*. Springer Verlag, 1999
- Presentation 2009** PRESENTATION: *Presentation Version 10.0. Neurobehavioral Systems*. <http://www.neurobs.com/presentation>, 2009. – Abgerufen am 07.02.2009
- Psychology-Software-Tools 2009** PSYCHOLOGY-SOFTWARE-TOOLS: *E-Prime*. <http://www.pstnet.com>, 2009. – Abgerufen am 06.07.2008
- Reiser u. Semmler 1997** REISER, M. ; SEMMLER, W.: *Magnetresonanztomographie*. Springer-Verlag, 1997
- Rieger et al. 2008** RIEGER, J.W. ; REICHERT, C. ; GEGENFURTNER, K.R. ; NOESELT, T. ; BRAUN, C. ; HEINZE, H.-J. ; KRUSE, R. ; HINRICHS, H.: Predicting the recognition of natural scenes from single trial MEG recordings of brain activity. In: *NeuroImage* 42 (2008), Nr. 3, S. 1056–1068
- Rieger et al. 2009** RIEGER, J.W. ; REICHERT, C. ; HOLLMANN, M. ; BERNARDING, J. ; HÖFER, M. ; TEMPELMANN, C. ; HEINZE, H.-J.: Time Resolved Determination of the Content of Visual Perception from fMRI-Data. In: *Proceedings of the HBM*, 2009
- Rilling et al. 2004** RILLING, J. K. ; SANFEY, A.G. ; ARONSON, J.A. ; NYSTROM, L.E. ; COHEN, J.D.: The neural correlates of theory of mind within interpersonal interactions. In: *NeuroImage* 22 (2004), Nr. 4, S. 1694–1703
- Rilling et al. 2008** RILLING, J.K. ; KING-CASAS, B. ; SANFEY, A.G.: The neurobiology of social decision-making. In: *Current Opinion in Neurobiology* 18 (2008), Nr. 2, S. 159–165
- Rooney et al. 2007** ROONEY, W.D. ; JOHNSON, G. ; LI, X. ; COHEN, E.R. ; KIM, S.-G. ; UGURBIL, K. ; SPRINGER, C.S.: Magnetic field and tissue dependencies of human brain longitudinal 1H_2O relaxation in vivo. In: *Magnetic Resonance in Medicine* 57 (2007), Nr. 2, S. 308–318
- Samson et al. 2005** SAMSON, D. ; APPERLY, I.A. ; KATHIRGAMANATHAN, U. ; HUMPHREYS, G.W.: Seeing it my way: a case of selective deficit in inhibiting self-perspective. In: *Brain* 128 (2005), S. 1102–1111
- Sanfey et al. 2003** SANFEY, A.G. ; RILLING, J.K. ; ARONSON, J.A. ; NYSTROM, L.E. ; COHEN, J.D.: The neural basis of economic decision-making in the Ultimatum Game. In: *Science* 300 (2003), Nr. 5626, S. 1755–1758

- Sanfey et al. 2006** SANFEY, A.G. ; LOEWENSTEIN, G. ; MCCLURE, S.M. ; COHEN, J.D.: Neuroeconomics: Cross-currents in research on decision-making. In: *Trends in Cognitive Sciences* 10 (2006), Nr. 3, S. 108–116
- Sanfey 2007** SANFEY, A.G.: Social decision-making: Insights from game theory and neuroscience. In: *Science* 318 (2007), Nr. 5850, S. 598–602
- Schölkopf et al. 1999** SCHÖLKOPF, B. ; MIKA, S. ; BURGESS, C.J.C. ; KNIRSCH, P. ; MULLER, K.R. ; RATSCH, G. ; SMOLA, A.J.: Input space versus feature space in kernel-based methods. In: *IEEE Transactions on Neural Networks* 10 (1999), S. 1000–1017
- Schölkopf u. Smola 2002** SCHÖLKOPF, B. ; SMOLA, A.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. Cambridge: MIT Press, 2002
- Schmitz u. Zucker 2003** SCHMITZ, C. ; ZUCKER, B.: *Wissensmanagement*. Metropolitan, 2003
- Schneider u. Fink 2007** SCHNEIDER, F. ; FINK, G.R.: *Funktionelle MRT in Psychiatrie und Neurologie*. Springer Medizin Verlag, 2007
- Siemens 2003** SIEMENS: *Siemens AG - Medical Solutions: Magnete, Spins und Resonanzen - Eine Einführung in die Grundlagen der Magnetresonanztomographie*. 2003
- Siemens 2004a** SIEMENS: *Siemens AG - Medical Solutions: ICE User's Guide - Manual for Ice programmers*. 2004
- Siemens 2004b** SIEMENS: *Siemens AG - Medical Solutions: IDEA Manual*. 2004
- Siemens 2004c** SIEMENS: *Siemens AG - Medical Solutions: Magnete, Fluss und Artefakte*. 2004
- Siemens 2004d** SIEMENS: *Siemens AG - Medical Solutions: The Sequence in Numaris 4*. 2004
- Singer et al. 2005** SINGER, T. ; FEHR, E. ; LAIBSON, D. ; CAMERER, C.F. ; MCCABE, K.: The neuroeconomics of mind reading and empathy. In: *American Economic Review* 95 (2005), Nr. 2, S. 340–345
- Smith et al. 2004** SMITH, S. M. ; JENKINSON, M. ; WOOLRICH, M. W. ; BECKMANN, C. F. ; BEHRENS, T. E. J. ; JOHANSEN-BERG, H. ; BANNISTER, P. R. ; DE LUCA,

- M. ; DROBNJAK, I. ; FLITNEY, D. E. ; NIAZY, R. K. ; SAUNDERS, J. ; VICKERS, J. ; ZHANG, Y. ; DE STEFANO, N. ; BRADY, J. M. ; MATTHEWS, P. M.: Advances in functional and structural MR image analysis and implementation as FSL. In: *NeuroImage* 23 (2004), Nr. SUPPL. 1, S. S208–S219
- Smyser et al. 2001** SMYSER, C. ; GRABOWSKI, T.J. ; FRANK, R.J. ; HALLER, J.W. ; BOLINGER, L.: Real-time multiple linear regression for fMRI supported by time-aware acquisition and processing. In: *Magnetic Resonance in Medicine* 45 (2001), S. 289–98
- Solso 2005** SOLSO, R.: *Kognitive Psychologie*. Springer, 2005
- Soon et al. 2008** SOON, C.S. ; BRASS, M. ; HEINZE, H.-J. ; HAYNES, J.-D.: Unconscious determinants of free decisions in the human brain. In: *Nature Neuroscience* 11 (2008), Nr. 5, S. 543–545
- Sorger et al. 2007** SORGER, B. ; DAHMEN, B. ; REITHLER, J. ; GOEBEL, R.: BOLD communication: When the Brain speaks for itself. In: *Proceedings of the HBM, 2007*
- SPM 2008** SPM: *SPM - Statistical Parametric Mapping, UCL*.
<http://www.fil.ion.ucl.ac.uk/spm>, 2008. – Abgerufen am 14.11.2008
- Stahl 2006** STAHL, C.: Software for generating psychological experiments. In: *Journal of Experimental Psychology* 53 (2006), Nr. 3, S. 218–32
- Stephan et al. 2007** STEPHAN, K.E. ; WEISKOPF, N. ; DRYSDALE, P.M. ; ROBINSON, P.A. ; FRISTON, K.J.: Comparing hemodynamic models with DCM. In: *NeuroImage* 3 (2007), S. 387–401
- Strothotte 1998** STROTHOTTE, T.: *Computational Visualization: Graphics, Abstraction, and Interactivity*. Springer Verlag, 1998
- Sun-Microsystems 2009** SUN-MICROSYSTEMS: *javax.xml API*.
<http://java.sun.com/j2se/1.5.0/docs/api/javax/xml/package-summary.html>, 2009.
– Abgerufen am 06.03.2009
- Talairach u. Tournoux 1988** TALAIRACH, J. ; TOURNOUX, P.: *Co-Planar Stereotaxic Atlas of the Human Brain*. Thieme, 1988
- Tipping 2001** TIPPING, M.E.: Sparse Bayesian learning and the relevance vector machine. In: *Journal of Machine Learning Research* 1 (2001), S. 211–244

- Trinigy 2006** TRINIGY: *Trinigy 3D Engine 6.01*. Trinigy GmbH.
<http://www.trinigy.net/>, 2006. – Abgerufen am 21.04.2008
- Villringer 1999** *Kapitel 1*. In: VILLRINGER, A.: *Physiological Changes During Brain Activation*. Springer Verlag, 1999
- Voyvodic 1999** VOYVODIC, J.T.: Real-Time fMRI Paradigm Control, Physiology, and Behavior Combined with Near Real-Time Statistical Analysis. In: *NeuroImage* 10 (1999), S. 91–106
- W3C 2009a** W3C: *Extensible Markup Language - XML*. <http://www.w3.org/XML/>, 2009. – Abgerufen am 10.01.2009
- W3C 2009b** W3C: *XML Schema - XSD*. <http://www.w3.org/XML/Schema>, 2009. – Abgerufen am 12.01.2009
- Warmuth 2002** WARMUTH, C.: *Nichtinvasive Magnetresonanz-Perfusionsmessung des Gehirns mittels Magnetischer Blutbolusmarkierung (Spin-Labeling)*. 2002. – Dissertationsschrift an der Medizinischen Fakultät Charité der Humboldt-Universität Berlin, Abgerufen am 26.01.2006
- Wegner u. Bargh 1998** *Kapitel Control and automaticity in social life*. In: WEGNER, D.M. ; BARGH, J.A.: *The Handbook of Social Psychology*. New York: McGraw-Hill, 1998, S. 446–96
- Weiskopf et al. 2003** WEISKOPF, N. ; VEIT, R. ; ERB, M. ; MATHIAK, K. ; GRODD, W. ; GOEBEL, R. ; BIRBAUMER, N.: Physiological self-regulation of regional brain activity using real-time functional magnetic resonance imaging (fMRI): Methodology and exemplary data. In: *NeuroImage* 19 (2003), Nr. 3, S. 577–586
- Weiskopf et al. 2004a** WEISKOPF, N. ; MATHIAK, K. ; BOCK, S.W. ; SCHARNOWSKI, F. ; VEIT, R. ; GRODD, W. ; GOEBEL, R. ; BIRBAUMER, N.: Principles of a brain-computer interface (BCI) based on real-time functional magnetic resonance imaging (fMRI). In: *IEEE Transactions on Biomedical Engineering* 51 (2004), Nr. 6, S. 966–970
- Weiskopf et al. 2004b** WEISKOPF, N. ; SCHARNOWSKI, F. ; VEIT, R. ; GOEBEL, R. ; BIRBAUMER, N. ; MATHIAK, K.: Self-regulation of local brain activity using real-time functional magnetic resonance imaging (fMRI). In: *Journal of Physiology Paris* 98 (2004), Nr. 4-6 SPEC. ISS., S. 357–373

- Weiskopf et al. 2007** WEISKOPF, N. ; SITARAM, R. ; JOSEPHS, O. ; VEIT, R. ; SCHARNOWSKI, F. ; GOEBEL, R. ; BIRBAUMER, N. ; DEICHMANN, R. ; MATHIAK, K.: Real-time functional magnetic resonance imaging: Methods and applications. In: *Magnetic Resonance Imaging* 25 (2007), Nr. 6, S. 989–1003
- Wicker et al. 2003** WICKER, B. ; KEYSERS, C. ; PLAILLY, J. ; ROYET, J.-P. ; GALLESE, V. ; RIZZOLATTI, G.: Both of us disgusted in My insula: The common neural basis of seeing and feeling disgust. In: *Neuron* 40 (2003), Nr. 3, S. 655–664
- Wolpaw et al. 2002** WOLPAW, J.R. ; BIRBAUMER, N. ; MCFARLAND, D.J. ; PFURTSCHELLER, G. ; VAUGHAN, T.M.: Brain-computer interfaces for communication and control. In: *Clinical Neurophysiology* 113 (2002), Nr. 6, S. 767–791
- wxWidgets 2009** WXWIDGETS: *Open-source cross-platform GUI-library*. www.wxwidgets.org, 2009. – Abgerufen am 12.03.2009
- XMLToolbox 2009** XMLTOOLBOX: *XML-Toolbox for Matlab, Version 3.1.2 by GEODISE*. <http://www.geodise.org/>, 2009. – Abgerufen am 22.01.2009
- Yoo et al. 2004** YOO, S.-S. ; FAIRNENY, T. ; CHEN, N.-K. ; CHOO, S.-E. ; PANYCH, L. P. ; PARK, H. ; LEE, S.-Y. ; JOLESZ, F. A.: Brain-computer interface using fMRI: Spatial navigation by thoughts. In: *NeuroReport* 15 (2004), Nr. 10, S. 1591–1595
- Zins 2007** ZINS, C.: Conceptual Approaches for Defining Data, Information, and Knowledge. In: *Journal of the American Society for Information Science and Technology* 58 (2007), S. 479–493

A. Auszüge EDL-Spezifikation

A.1. EDL-Codeauszüge

A.1.1. Element *paradigm*

```
<paradigm ignoreScansAtStart="2">
  <swDesignStruct>
    <scanBasedRegressor regressorID="sbr1" name="l1" size="54" useRefFct="g1">
      <sbrDesign size="10" repetitions="5">
        <scan>0</scan>
        <scan>0</scan>
        <scan>0</scan>
        <scan>0</scan>
        <scan>0</scan>
        <scan>0</scan>
        <scan>0</scan>
        <scan>0</scan>
        <scan>1</scan>
        <scan>1</scan>
        <scan>1</scan>
      </sbrDesign>
      <sbrIgnoreStruct overallIgnores="4">
        <insertIgnores repetitionNr="1" nmbIgnores="0"/>
        <insertIgnores repetitionNr="2" nmbIgnores="1"/>
        <insertIgnores repetitionNr="3" nmbIgnores="0"/>
        <insertIgnores repetitionNr="4" nmbIgnores="3"/>
        <insertIgnores repetitionNr="5" nmbIgnores="0"/>
      </sbrIgnoreStruct>
    </scanBasedRegressor>
  </swDesignStruct>
</paradigm>
```

A.1.2. Element *mediaObject: contentComplex*

```
<mediaObject moID="2" name="ug1" type="COMPLEX">
  <contentComplex type="ULTIMATUMGAME" overallPresLength="22000">
    <functionHandlePrepare>predef_prepareObjects_UG</functionHandlePrepare>
    <functionHandleRun>predef_runtObjects_UG</functionHandleRun>
    <stimPrepareParamsCOMPLEX>
      <param>
        <paramName>splitMoney</paramName>
        <paramValue>3</paramValue>
        <paramType>INT</paramType>
      </param>
      <param>
        <paramName>overallPresLength</paramName>
        <paramValue>22000</paramValue>
        <paramType>INT</paramType>
      </param>
    <stimRunParamsCOMPLEX>
      <param>
        <paramName>showMoneyDur</paramName>
        <paramValue>2000</paramValue>
        <paramType>INT</paramType>
      </param>
      <param>
        <paramName>showOfferDur</paramName>
        <paramValue>1200</paramValue>
        <paramType>INT</paramType>
      </param>
      <param>
        <paramName>showDecisionDur</paramName>
        <paramValue>4000</paramValue>
        <paramType>INT</paramType>
      </param>
      <param>
        <paramName>showResultDur</paramName>
        <paramValue>4000</paramValue>
        <paramType>INT</paramType>
      </param>
    </stimRunParamsCOMPLEX>
  </contentComplex>
</mediaObject>
```

```
</mediaObject>
```

A.1.3. Element *timeTable*

```
<timeTable timeTolerance="30">
  <blockStimulusDesign repeats="3" overallPresLength="66000">
    <stimEvent time="0" duration="22000">
      <mObjectID>2</mObjectID>
    </stimEvent>
  </blockStimulusDesign>
  <outro>
    <stimEvent time="0" duration="4000">
      <mObjectID>3</mObjectID>
    </stimEvent>
  </outro>
</timeTable>
```

A.1.4. Ultimatum Game: *mriParams*

```
<mriParams fieldStrength="3T" vendorName="Siemens" systemVersion="VA25A">
  <MR_TAG_BANDWIDTH vendorTag="bw">1816</MR_TAG_BANDWIDTH>
  <MR_TAG_BASE_RESOLUTION vendorTag="matrix">64</MR_TAG_BASE_RESOLUTION>
  <MR_TAG_PHASE_RESOLUTION vendorTag="ph_res">100</MR_TAG_PHASE_RESOLUTION>
  <MR_TAG_PHASE_FOV vendorTag="phase_fov">100</MR_TAG_PHASE_FOV>
  <MR_TAG_TE vendorTag="te">29</MR_TAG_TE>
  <MR_TAG_TR vendorTag="tr">2000</MR_TAG_TR>
  <MR_TAG_SG_SIZE vendorTag="sg.0.size">31</MR_TAG_SG_SIZE>
  <MR_TAG_SLICE_THICKNESS vendorTag="sl_thick">4.5</MR_TAG_SLICE_THICKNESS>
  <MR_TAG_SG_DISTANCE_FACTOR vendorTag="sg.0.df">0</MR_TAG_SG_DISTANCE_FACTOR>
  <MR_TAG_READOUT_FOV vendorTag="readout_fov">210</MR_TAG_READOUT_FOV>
  <MR_TAG_FLIP_ANGLE vendorTag="angle_array.0">90</MR_TAG_FLIP_ANGLE>
  <MR_TAG_MEASUREMENTS vendorTag="meas">660</MR_TAG_MEASUREMENTS>
</mriParams>
```

A.1.5. Ultimatum Game: *analysisRuntimeModule*

```
<analysisRuntimeModuleUserDefined moduleID="aM_UG1" useRegressor="sbr1">
  <functionHandlePrepare>initUDAnalysisModule_UG</functionHandlePrepare>
  <functionHandleRun>runUDAnalysisModule_UG</functionHandleRun>
```

```
<prepareParamsUD>
  <param>
    <paramName>UG_nmbTrials</paramName>
    <paramValue>60</paramValue>
    <paramType>INT</paramType>
  </param>
  <param>
    <paramName>NormalizeImageFileName</paramName>
    <paramValue>../../IBMI_FMRI_10.ima</paramValue>
    <paramType>STRING</paramType>
  </param>
  <param>
    <paramName>NormalizeImageParams</paramName>
    <paramValue>FM_UG_NormaliseParams.mat</paramValue>
    <paramType>STRING</paramType>
  </param>
  <param>
    <paramName>mask_AI</paramName>
    <paramValue>AntIns_mask.hdr</paramValue>
    <paramType>STRING</paramType>
  </param>
  <param>
    <paramName>mask_LPFC</paramName>
    <paramValue>LPFC_mask.hdr</paramValue>
    <paramType>STRING</paramType>
  </param>
  <param>
    <paramName>mask_OC</paramName>
    <paramValue>OC_mask.hdr</paramValue>
    <paramType>STRING</paramType>
  </param>
</prepareParamsUD>
<runParamsUD></runParamsUD>
<globalTargetVariables></globalTargetVariables>
</analysisRuntimeModuleUserDefined>
```