



## **M A S T E R A R B E I T**

# **FuzzyDBSexpert – Webbasierte Entscheidungsunterstützung zur Auswahl von Parkinson-Patienten für die Tiefe Hirnstimulation**

Ausgeführt am Institut für  
Medizinische Experten- und Wissensbasierte Systeme  
Besondere Einrichtung für Medizinische Statistik und Informatik  
Medizinische Universität Wien  
Spitalgasse 23, A-1090 Wien

unter der Anleitung von Univ.-Prof. Dipl.-Ing. Dr. Klaus-Peter Adlassnig

durch

Ing. Marcus Vitek, Bakk. Techn.

Wehlistraße 71-81/7/2, A-1200 Wien

## **FuzzyDBSexpert – Webbasierte Entscheidungsunterstützung zur Auswahl von Parkinson-Patienten für die Tiefe Hirnstimulation**

### **Kurzfassung:**

Die Parkinson'sche Krankheit ist eine progressive, neurologische Krankheit, welche durch die Degeneration von Dopamin-produzierenden Neuronen entsteht. Die Behandlung dieser Krankheit beschränkt sich derzeit auf eine Dopamin-Ersatz-Therapie und eine Anzahl von Begleittherapien – Heilung ist derzeit nicht möglich. Damit werden die Kardinalsymptome, wie Ruhetremor, Rigidität und Bradykinesie gelindert. Mit der Zeit treten Langzeit-Nebenwirkungen der medikamentösen Therapie auf, wie zum Beispiel motorische Fluktuationen und Dyskinesien. Nur ein neurochirurgischer Eingriff kann die sonst inakzeptablen Nebenwirkungen mindern. Eine der möglichen Interventionen stellt die Tiefe Hirnstimulation dar. Die Indikation oder Kontraindikation der Operation für die Tiefe Hirnstimulation hängt von (a) dem bisherigen Erfolg der medikamentösen Therapie, (b) den Symptomen und Zeichen, welche sich in naher Vergangenheit gezeigt haben, (c) der subjektiven Lebensqualität des Patienten und (d) der Prognose des zu erwartenden Verlaufs der Krankheit ab. Aus diesem Grund wurde ein webbasiertes, entscheidungsunterstützendes System für Ärzte, welche nicht Experten auf diesem Gebiet sind, und Patienten und deren Verwandte geschaffen, welches auf einer Wissensbasis mit dem neuesten Stand des medizinischen Konsensus in dieser Domäne basiert. Das System, FuzzyDBSexpert genannt, besteht aus fünf Komponenten: (1) der Datenbank, welche die medizinischen Daten der Patienten enthält, welche mittels (2) dem Benutzerinterface – dargestellt via HTTP über das Internet –, welches die Unified Parkinson's Disease Rating Scale (UPDRS), die derzeit am häufigsten benutzte Skala zur Bewertung der Einschränkungen der Patienten durch die Parkinson'sche Krankheit, und einem Eingabeformular für andere Patientendaten, (3) einer Wissensbasis, welche Fuzzy-WENN-DANN-Regeln, eingegeben durch (4) eine Experten-Schnittstelle, gespeichert hat, und (5) einer Fuzzy-Inferenzmaschine. Die Entwicklung des Systems wurde abgeschlossen, weitere Studien über die Usability des Systems und zur Verfeinerung der Wissensbasis sind notwendig.

**Schlüsselwörter:** Expertensystem, Wissensbasiertes System, Entscheidungsunterstützendes System, Webbasiertes System, Fuzzy-Logik, Fuzzy-Regeln, Fuzzy-Inferenz, Parkinson, UPDRS

---

## **FuzzyDBSexpert – A web based decision support system for Deep Brain Stimulation in Parkinson's disease patients**

### **Abstract:**

Parkinson's disease is a progressive neurological disorder caused by the degeneration of dopamine-producing neurons. Treatment of this disease is limited to dopamine replacement therapy—at present, there is no etiological cure—and a number of concomitant measures. By doing this, major signs of the disease such as resting tremor, rigidity, bradykinesia are alleviated. With the time, long-term therapy side effects occur like motor fluctuations and dyskinesia. Only a neurosurgical intervention is able to detain the otherwise intolerable symptoms and signs. One of the possible interventions is deep brain stimulation (DBS). Indication or contraindication of DBS surgery heavily depends on (a) the past course of therapy, (b) the symptoms and signs presently seen in the patient, (c) the subjectively felt quality of life of the patient, and (d) the prognosis of the expected course of the disease. Therefore a web-based decision support system, relying on a state-of-the-art consensus expert knowledge base, for non-expert physicians and for patients and their confidants was established. The system, called FuzzyDBSexpert, consists of five main components: (1) the database that contains patient's medical data that are gathered via (2) a user interface—delivered over the web via HTTP—that implements the Unified Parkinson's Disease Rating Scale (UPDRS), the currently most commonly used scale for describing the impairments of patients suffering from Parkinson's disease and, in addition, an input form for other patient data, (3) a knowledge base that contains fuzzy IF-THEN-rules entered through (4) an expert user interface, and (5) a fuzzy inference engine. Currently, the system development has been finished, further studies regarding the usability of the system and for the refinement of the knowledge base will be necessary.

**Keywords:** expert system, knowledge-based system, decision support system, web-based system, fuzzy logic, fuzzy rules, fuzzy inference, Parkinson, UPDRS

**Danksagung**

An dieser Stelle möchte ich mich bei Univ.-Ass. Dipl.-Math. Dr. Rudolf Seising für die freundliche Unterstützung auf dem Gebiet der Fuzzy-Logik während meines Studiums, welches zu Beginn noch interuniversitär zwischen der Technischen Universität Wien und der Medizinischen Universität Wien durchgeführt wurde, bedanken. Weiters gilt mein Dank Frau Univ.-Doz. Dr. Michaela M. Pinter als beteiligte medizinische Expertin für die Parkinson-Krankheit und die Tiefe Hirnstimulation. Ich danke Frau Dipl.-Ing. Andrea Rappelsberger für die große und freundliche Unterstützung bei der Fertigstellung der Masterarbeit. Nicht zuletzt möchte ich mich für die Betreuung von Univ.-Prof. Dipl.-Ing. Dr. Klaus-Peter Adlassnig bedanken.

## **Konventionen**

In dieser Masterarbeit werden einige typografische Konventionen benutzt, um bestimmte Textteile von normalem Text abzuheben. Für Programmcode wird eine nicht-proportionale Schrift verwendet. *Paket*-, *Klassen*- und *Methodennamen* von Programmiersprachen (z. B. Java) werden in einer *serifenlosen Schrift kursiv* dargestellt.

In dieser Masterarbeit habe ich zugunsten einer besseren Lesbarkeit auf eine geschlechterneutrale Schreibweise verzichtet. Die entsprechenden Ausdrücke bitte ich, geschlechterneutral zu verstehen.

# Inhaltsverzeichnis

1	Einleitung .....	1
2	Medizinischer Hintergrund .....	3
2.1	Morbus Parkinson .....	3
2.1.1	Klinik.....	3
2.1.2	Symptome.....	5
2.1.2.1	Tremor.....	5
2.1.2.2	Rigor (Muskelsteifheit) .....	5
2.1.2.3	Bradykinesie.....	6
2.1.2.4	Posturale Instabilität .....	6
2.1.2.5	Weitere Symptome .....	6
2.1.3	Medikamentöse Therapie.....	7
2.2	Operationen.....	10
2.2.1	Allgemeines.....	10
2.2.2	Deep Brain Stimulation (DBS) .....	11
2.2.2.1	Einführung in die DBS .....	11
2.2.2.2	Patientenselektion für die DBS .....	13
2.2.2.3	Durchführung der DBS-Operation.....	17
2.2.2.4	Risiken der DBS .....	23
2.2.2.5	Nachsorge bei Patienten mit DBS .....	24
3	UPDRS.....	25
3.1	Allgemeine Bemerkungen zur UPDRS.....	25
3.2	Beschreibung der UPDRS .....	26
4	Systemanforderungen.....	29
4.1	Prinzipielle Anforderungen .....	29
4.2	Anforderungen an Expertensysteme im klinischen Einsatz .....	29
4.2.1	Elementare Bedingungen.....	30
4.2.2	Notwendige Bedingungen .....	30
4.2.2.1	Benutzeroberfläche/Datenerfassung .....	30
4.2.2.2	Erklärung/Rechtfertigung .....	31
4.2.2.3	Flexibilität .....	31
4.2.2.4	Arbeitserleichterung.....	32
4.2.3	Hinreichende Bedingungen .....	32
4.2.3.1	Organisatorische Einbettung in die Klinik- bzw. Praxisroutine .....	32
4.2.3.2	Validierung des Systems.....	32
4.2.3.3	Pflege der Wissensbasis (Erweiterung/Änderung) .....	33
5	Wissensbasierte Systeme.....	34
5.1	Aufbau eines wissensbasierten Systems .....	34
5.1.1	Wissensbasis.....	34
5.1.2	Inferenzkomponente .....	35
5.1.3	Wissensakquisitionskomponente.....	35
5.1.4	Benutzerschnittstelle .....	35
5.1.5	Erklärungskomponente.....	35
5.2	Arten von Wissen .....	36
5.2.1	Faktenwissen.....	36
5.2.2	Zusammenhänge (Beziehungen) .....	36
5.2.3	Methodisches Wissen.....	36

5.2.4	Meta-Wissen .....	36
5.3	Wissensrepräsentation .....	37
5.3.1	Prozedurale Methoden .....	37
5.3.2	Objektorientierte Methoden .....	38
5.3.2.1	Frames .....	38
5.3.2.2	Semantische Netze .....	38
5.3.3	Logikbasierte Methoden .....	38
5.3.3.1	Aussagenlogik .....	39
5.3.3.2	Syntax .....	41
5.3.3.3	Semantik .....	42
5.3.3.4	Elementare Resultate der Aussagenlogik .....	43
5.3.4	Prädikatenlogik .....	43
6	Regelbasierte Systeme .....	44
6.1	Was sind Regeln? .....	44
6.2	Die Wissensbasis eines regelbasierten Systems .....	45
6.3	Inferenz in einem regelbasierten System .....	46
6.4	Vorwärtsverkettung .....	47
6.5	Rückwärtsverkettung .....	47
6.6	Das Problem der Widersprüchlichkeit .....	47
6.7	Die Erklärungskomponente .....	48
6.8	Modularität und Effizienz regelbasierter Systeme .....	48
7	Quantitative Methoden .....	49
7.1	Probabilistische Netzwerke .....	49
7.2	Übergang zu anderen Methoden .....	50
7.2.1	Verallgemeinerte Wahrscheinlichkeitstheorie .....	50
7.2.2	Die Dempster-Shafer-Theorie .....	51
7.2.2.1	Grundlagen .....	51
7.2.2.2	Basismaße und Glaubensfunktion .....	51
8	Fuzzy-Logik .....	53
8.1	Einführung in die Fuzzy-Logik .....	53
8.2	Grundlegende Konzepte .....	54
8.2.1	Enge Deutung der Fuzzy-Logik .....	54
8.2.2	Weite Deutung der Fuzzy-Logik .....	54
8.2.3	Unscharfe Mengen .....	54
8.2.4	Linguistische Variable .....	55
8.2.5	Fuzzy-Inferenzsysteme .....	55
8.2.5.1	Fuzzifizierung der Eingangssignale .....	56
8.2.5.2	Anwendung der Fuzzy-Operatoren .....	56
8.2.5.3	Anwendung der Fuzzy-Implikation .....	57
8.2.5.4	Aggregierung der Ausgangsgrößen .....	58
8.2.5.5	Defuzzifizierung der Ausgangsgrößen .....	58
8.3	Spezielle Konzepte der Fuzzy-Logik .....	59
8.3.1	Zugehörigkeitsfunktionen .....	59
8.3.1.1	Grundlagen der Zugehörigkeitsfunktionen .....	59
8.3.1.2	Konstruktion einer Zugehörigkeitsfunktion .....	59
8.3.1.3	Verschiedene Zugehörigkeitsfunktionen .....	60
8.3.2	Logische Operatoren .....	61
8.3.2.1	Fuzzy-UND, -ODER und -Komplement .....	61
8.3.2.2	t-Norm und s-Norm .....	62
8.3.3	Defuzzifizierungsmethoden .....	64
8.3.3.1	Center of Area .....	64

---

8.3.3.2	Mean of Maxima.....	64
8.3.4	Fuzzy-Systeme.....	65
8.3.4.1	Mamdani-Fuzzy-Systeme.....	65
8.3.4.2	TSK-Fuzzy-Systeme.....	65
8.3.4.3	Singleton-Fuzzy-Systeme.....	66
9	Projektbeschreibung.....	67
9.1	Realisierungsmodell.....	67
9.2	Prinzipielle Struktur.....	68
9.2.1	Benutzerschnittstelle.....	69
9.2.1.1	Dateneingabe.....	69
9.2.1.2	Datenausgabe.....	70
9.2.2	Expertenschnittstelle.....	70
9.2.3	Wissensverarbeitung.....	71
9.2.3.1	Gewichtung der UPDRS-Antworten.....	71
9.2.3.2	Ausschlusskriterien.....	72
9.2.3.3	Fuzzy-Inferenz.....	72
9.2.3.4	Defuzzifizierung der Ausgangsgrößen.....	72
9.2.4	Wissensspeicherung.....	73
9.3	Programmmodule.....	73
9.4	Testverfahren.....	76
10	Methode.....	77
10.1	Grundlagen.....	77
10.1.1	Wissensbasis.....	77
10.1.1.1	Fallspezifisches Wissen (Patientendaten).....	77
10.1.1.2	Generisches Wissen (Regel-Set).....	82
10.1.2	Inferenz.....	87
10.1.2.1	Fuzzifizierung der Eingangssignale.....	87
10.1.2.2	Anwendung der Fuzzy-Operatoren und -Implikation.....	87
10.1.2.3	Aggregation der Ausgangsgrößen.....	89
10.1.2.4	Defuzzifizierung der Ausgangsgrößen.....	89
10.1.3	Ergebnisse.....	89
10.1.3.1	DBS-Eignung.....	89
10.1.3.2	Variablen.....	90
10.1.3.3	UPDRS-Ergebnisse.....	90
10.1.3.4	Regeln.....	90
10.2	Programmbeschreibung.....	90
10.2.1	Administrator-Teil.....	91
10.2.1.1	Benutzerverwaltung.....	91
10.2.1.2	UPDRS-Set-Verwaltung.....	92
10.2.1.3	Variablenverwaltung.....	98
10.2.2	Experten-Teil.....	100
10.2.2.1	Regel-Sets.....	100
10.2.2.2	Regel-Sets-Manager.....	101
10.2.2.3	Regel-Set-Regeln.....	104
10.2.2.4	Regel-Set-Gewichte.....	107
10.2.3	Anwender-Teil.....	109
10.2.3.1	Patienten-Manager.....	109
10.2.3.2	Evaluierungs-Teil.....	113
10.2.4	Bereichsübergreifende Features.....	119
10.2.4.1	Lokalisierung.....	119
10.2.4.2	Flexibilität.....	122
10.2.4.3	DatabaseCRUDPanel.....	125

---

10.2.4.4	Kaskadiertes Löschen .....	126
10.2.4.5	LinguisticExpressionConstructor.....	128
10.3	Technische Implementierung .....	129
10.3.1	Überblick über die eingesetzten Technologien .....	129
10.3.1.1	Datenbank.....	129
10.3.1.2	Softwareentwicklung .....	130
10.3.1.3	Abgrenzung zu anderen Technologien.....	132
10.3.1.4	Entwicklungsumgebung.....	133
10.3.1.5	Web-Container.....	134
10.3.2	Datenbank.....	134
10.3.2.1	Entwurf .....	134
10.3.2.2	Primärschlüssel .....	136
10.3.2.3	Konzeptuelle Modellierung mit UML.....	137
10.3.2.4	SQL-Script .....	144
10.3.3	Programm .....	144
10.3.3.1	Grundlegende Konzepte.....	144
10.3.3.2	Pakete.....	147
10.3.3.3	DatabaseCRUDPanel.....	153
10.3.3.4	LinguisticExpressionConstructor.....	160
10.3.3.5	Handhabung des Session-Timeouts .....	162
10.3.3.6	Verwendung von Reflection .....	163
10.3.3.7	Ablauf der Inferenz.....	166
10.3.4	Literaturhinweise.....	170
10.3.4.1	Java.....	170
10.3.4.2	Servlets & JSP .....	171
10.3.4.3	JSF.....	171
10.3.4.4	Creator .....	172
10.3.4.5	Datenbank .....	172
10.3.4.6	Web-Container.....	172
11	Schlussdiskussion .....	173
11.1	Zusammenfassung .....	173
11.2	Ausblick.....	174
	Literaturverzeichnis .....	176
	Abbildungsverzeichnis .....	181
	Tabellenverzeichnis .....	183
	Anhang.....	184
UPDRS-Skala	.....	184
SQL-Script	.....	194

# 1 Einleitung

Morbus Parkinson ist eine progressive, neurologische Erkrankung, welche durch eine Degenerierung von Dopamin-Neuronen verursacht wird. Die Behandlung dieser Erkrankung beschränkt sich auf die Linderung der Symptome, wie Tremor, Rigor und Bradykinesie. Die Haupttherapie besteht derzeit in der Gabe von L-Dopa, einem Vorläufer von Dopamin, welches das fehlende Dopamin nach Umwandlung ersetzen soll. In der Langzeittherapie kommt es jedoch zu Langzeiteffekten, wie motorischen Schwankungen und schmerzhaften Dyskinesien, da die Dosis des L-Dopa kontinuierlich erhöht werden muss. Nach dem derzeitigen Stand der Forschung kann nur eine neurochirurgische Intervention verlässlich helfen, diese Langzeiteffekte durch eine ermöglichte Dosisreduzierung drastisch zu mindern. Eine der möglichen Interventionen, welche zusätzlich reversibel und adaptierbar ist, ist die Tiefe Hirnstimulation (Deep Brain Stimulation, DBS). Die Entscheidung, dass ein Parkinson-Patient eine entsprechende Operation bekommen soll, basiert auf sehr vielen verschiedenen Parametern, welche nicht immer exakt messbar sind oder wo es keinen Sinn macht, scharfe Abgrenzungen zu benutzen.

“So, it is easily understood that the transition from healthy to ill and from normal to pathological, which is modelled by fuzzy sets in the above-mentioned knowledge-based systems, is a gradual transition and not a crisp one, and that the partial firing of rules with partially valid antecedents naturally diminishes the validity of the resulting consequence—a mechanism inherent to fuzzy logic” [Adlassnig, 2001, S. 141].

Nachdem in der Medizin ein Großteil der Begriffe und Entscheidungen unscharf ist, ist es vielleicht ein Schritt in die richtige Richtung, wenn zu diesem Zweck auch eine unscharfe Logik verwendet wird, die nicht nur zwei verschiedene Zustände kennt, wie dies zum Beispiel bei der Aristoteles'schen Logik der Fall ist. Eine Logik, mit der man unscharfe Daten fassen kann, wurde 1965 vom Elektrotechniker Lotfi Zadeh [Zadeh, 1965] – er nannte sie Fuzzy-Logik – formuliert. Der Gedanke dahinter ist, ein System mathematisch so zu beschreiben, dass es der linguistischen Beschreibung eines Menschen nahe kommt.

„It will thus be correct and advantageous to postulate that

Everything in medicine is fuzzy (2)

rendering the entire medicine an application domain of fuzzy theory. We in medicine are grateful to Professor Lotfi Zadeh for having enabled us to establish the basic paradigm (2) and to state accordingly that about 2370 years after medicine's constitution as a discipline and profession by Hippocrates, finally *habemus methodologiam*” [Sadegh-Zadeh, 2001, S. 19].

So wird ein Alter von über 75 Jahren als Kontraindikation für eine Tiefe Hirnstimulation genannt [vgl. Gerlach, 2003, S. 214–215]. Wenn nun ein sonst gesunder, lebensfroher Mensch um 75 Jahre alt ist, soll er dann keine Operation mehr erhalten? Es ist vielmehr so, dass dieser Parameter im Kontext mit anderen gesehen wird und auch keine scharfe Grenze darstellt. Die Fuzzy-Logik hilft uns bei der Frage, ob ein Mensch mit 77 Jahren vielleicht doch noch eine Tiefe Hirnstimulation bekommen kann.

Das Ziel dieser Arbeit ist es, ein webbasiertes, entscheidungsunterstützendes System für die Benutzung durch Mediziner, welche über kein Spezialwissen auf diesem Gebiet verfügen, und für Patienten beziehungsweise deren Angehörige, zu schaffen, welches dem heutigen Stand der Softwaretechnik und dem neuesten Stand des medizinischen Konsensus, auch in der Zukunft, entspricht. Das System, FuzzyDBSexpert genannt, soll zur Unterstützung bei der Entscheidung dienen, ob ein Patient mit idiopathischem Parkinson für eine Tiefe Hirnstimulation indiziert ist oder nicht. FuzzyDBSexpert besteht aus einer Datenbank, in der das generische und das fallspezifische Wissen gespeichert werden, und einer Webapplikation, welche die erforderlichen Benutzerschnittstellen und Programmlogiken realisiert, welche die Wissenserfassung, -verarbeitung und Präsentation der Ergebnisse ermöglichen. Hauptbestandteile der Webapplikation sind Benutzerschnittstellen zur Verwaltung der Patienten, Erfassung der Unified Parkinson Disease Rating Scale (UPDRS) und anderer Patientendaten und die Ergebnispräsentation für Benutzer zur Erfassung des generischen Wissens von Experten und teilweise zur Erfassung von generischem Wissen und zur Benutzerverwaltung durch Administratoren. Einen weiteren großen Teil der Anwendung macht die Fuzzy-Inferenzmaschine aus, welche das fallspezifische Wissen mit Hilfe des generischen Wissens verarbeitet und die Ergebnisse präsentiert, unter anderem, wie gut ein Patient für eine Tiefe Hirnstimulation geeignet ist.

## 2 Medizinischer Hintergrund

In diesem Kapitel werden die medizinischen Hintergründe über die Krankheit Morbus Parkinson, deren Symptome, medikamentöse Therapie, chirurgische Interventionen und schließlich die Tiefe Hirnstimulation beschrieben.

### 2.1 Morbus Parkinson

#### 2.1.1 Klinik

Morbus Parkinson, kurz Parkinson, ist eine progressive, neurologische Erkrankung. Sie wird durch eine massive Degeneration von dopaminergischen, nigrostriatalen Neuronen verursacht. Dopamin ist ein Neurotransmitter, ein Botenstoff, der von Nervenzellen im Gehirn, den Neuronen, produziert wird. Der Landarzt James Parkinson hat diese Krankheit erstmalig 1817 im „Essay on the Shaking Palsy“ [Parkinson, 1817] beschrieben.

Bewegungen des menschlichen Körpers werden im motorischen Kortex initiiert. Die wichtigsten motorischen Bahnen bestehen aus dem Pyramidenbahnensystem, welches sich vom motorischen Kortex bis hin zum Rückenmark ausbreitet. Die unteren Motor-Neuronen leiten die Signale vom Rückenmark zu den entsprechenden Muskeln, um Bewegungen zu erzeugen. Das Pyramidenbahnensystem wird vom extrapyramidalmotorischen System moduliert, welches die Substantia nigra (schwarze Substanz), das Striatum (Nucleus caudatus und Putamen), den Nucleus subthalamicus, die externen und internen Segmente des Globus pallidus (Bleichkörper, innerer Teil des Linsenkerns) und den Thalamus (Sehhügel) beinhaltet. Das Pyramidenbahnensystem kann, abhängig von der tonischen Dopamin-Innervierung des Striatum, Bewegungen entweder fördern oder hemmen. Wenn ungefähr 60–80 % der Dopamin produzierenden Neuronen der Substantia nigra degeneriert sind, ist das extrapyramidalmotorische System nicht mehr länger in der Lage, Bewegungen effektiv zu unterstützen und somit treten die Symptome der Parkinson'schen Krankheit auf.

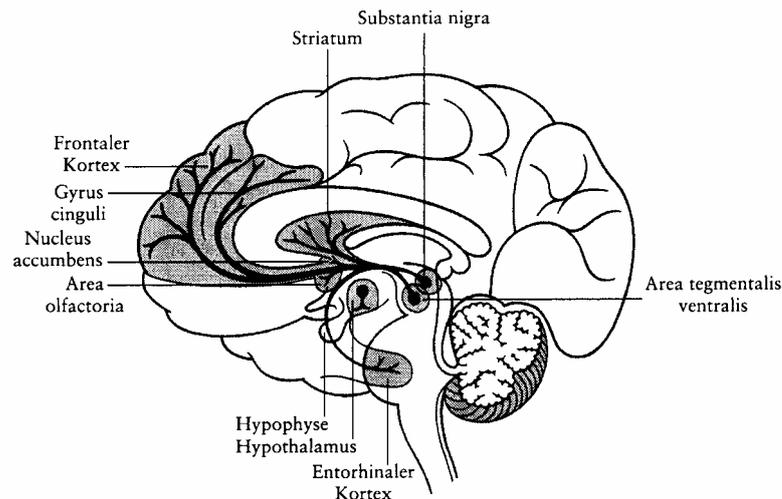


Abbildung 2.1: Die wichtigsten dopaminergen Neuronensysteme im menschlichen Gehirn [Gerlach, 2003, S. 47]

Die Ursache für das Absterben der Dopamin erzeugenden Nervenzellen (siehe Abbildung 2.1) ist in neun von zehn Fällen nicht bekannt, man spricht hier vom idiopathischen Parkinson. Bekannte Ursachen hingegen sind Gehirnhautentzündung (postenzephalitischer Parkinson), Schädigung durch Gifte (toxischer Parkinson), Stoffwechselerkrankungen, dessen Abfallprodukte hirnschädigend sind (metabolischer Parkinson), und Hirntumore, welche die Substantia nigra schädigen. Für die Entstehung des normalen idiopathischen Parkinson nimmt man an, dass genetische und Umwelteinflüsse (Umweltgifte) eine Rolle spielen, eine genaue Ursache ist bis heute jedoch noch nicht erforscht.

Die Prävalenz beträgt ca. 120–180/100.000 in der weißen Bevölkerung. Die Inzidenz beträgt 20/100.000, wobei der Durchschnitt des Beginns der Krankheit bei ca. 60 Jahren liegt, die Krankheit normalerweise bei Patienten über 50 Jahren auftritt und ein Auftreten der Krankheit in einem Alter von unter 25 Jahren ungewöhnlich ist [vgl. Hauser, 2003, S. 12].

„Rund 17.000 Menschen leiden in Österreich am Morbus Parkinson. Zwar tritt die Krankheit vorwiegend bei älteren Menschen auf, jeder zehnte Betroffene spürt die ersten Anzeichen jedoch bereits im Alter von 30 Jahren. Mit Fortschreiten der Krankheit lässt die Wirksamkeit der oralen L-Dopa-Medikation nach“ [ÄrzteWoche, 2006].

Es gibt derzeit keine Heilungsmöglichkeiten für Morbus Parkinson, es können nur die Symptome gelindert werden [vgl. Hauser, 2003, S. 73–74].

## 2.1.2 Symptome

Charakteristisch sind vier klinische Eigenschaften bei Parkinson, die vier Kardinalsymptome, nämlich Tremor, Rigor, Bradykinesie und posturale Instabilität, die wegen der Wichtigkeit zum Verständnis der Beeinträchtigung des Patienten nachfolgend etwas detaillierter beschrieben werden.

### 2.1.2.1 Tremor

Der Tremor, es handelt sich dabei in der Regel um einen Ruhetremor, beginnt einseitig und äußert sich durch ein rhythmisches Hin- und Herbewegen einer Extremität oder eines Extremitätenendes mit einer Frequenz von 4–6/s. Der Ruhetremor verschwindet bei Bewegung der Extremität. Häufig ist neben dem Ruhetremor auch ein Haltetremor zu finden, hierbei ist die Differentialdiagnose zum essentiellen Tremor meist schwierig, nicht ganz selten tritt auch eine Koinzidenz zwischen der Parkinson-Krankheit und esentiellem Tremor auf. Rajput und seine Mitarbeiter (1991) stellten fest, dass im Verlauf der Krankheit nahezu jeder Parkinson-Patient einen Tremor entwickelt [vgl. Gerlach, 2003, S. 24].

### 2.1.2.2 Rigor (Muskelsteifheit)

Der Rigor ist eine gleichmäßige Tonuserhöhung (Spannungserhöhung) im Agonisten und Antagonisten (zwei Gegenspieler), was bedeutet, dass die entsprechende Extremität nicht oder nur sehr mühsam bewegt werden kann, auch passiv – durch Bewegung durch eine andere Person – bei völliger Entspannung. Es ist zum Beispiel das typische Zahnradphänomen feststellbar. Wenn man den angewinkelten Arm eines Patienten in eine gestreckte Position bringt, so geschieht diese Bewegung nicht gleichmäßig, wie bei Menschen ohne diesem Symptom, sondern ruckartig, da die Muskelspannung nicht gleichmäßig nachgibt. Die Patienten haben Probleme mit der Unfähigkeit, sich rasch bewegen zu können oder sich nachts im Bett zu wenden. Weiters können Tätigkeiten, die hohe Anforderungen an die Feinmotorik stellen, wie zum Beispiel das Zuknöpfen der Kleidung oder Bastel- und Reparaturarbeiten, Schreiben und so weiter, nicht mehr in der gewohnten Weise durchgeführt werden [vgl. Gerlach, 2003, S. 24].

### 2.1.2.3 Bradykinesie

Unter Bradykinesie versteht man die Verlangsamung der Initiation von Willkürbewegungen mit progressiver Abnahme der Geschwindigkeit und Amplitude bei repetitiven Aktionen. In der Literatur werden die Begriffe der Hypokinesie, die auf der Reduktion von Bewegungsamplituden und Spontanbewegungen beruht, und der Akinese, die einen Verlust sämtlicher willkürlicher motorischer Entäußerungen impliziert, mit der Bradykinesie vermischt und diese als Oberbegriff für alle drei Symptome verwendet. Weiters tritt nach einiger Zeit im Gesicht die Hypomimie<sup>1</sup> auf, welche zuerst einseitig beginnt, die Mundwinkel stehen still und die Lidschlusshäufigkeit nimmt ab. Scheinbar regungslos verfolgen Parkinson-Patienten auch psychisch aufwühlende Ereignisse, sodass der oberflächliche Betrachter den Eindruck gewinnen kann, dass der Patient teilnahms- und interesselos ist. Unter dieser Fehleinschätzung leiden die Patienten sehr, zumal die Mimik ein wichtiges Kommunikationsmittel darstellt [vgl. Gerlach, 2003, S. 22–24].

Ein weiteres Symptom ist das Freezing, welches während des Gehens eines Patienten auftreten kann und bei ca. einem Drittel der Patienten beobachtet wird. Der Patient bleibt, oftmals beim Durchschreiten eines engen Durchganges, für mehrere Sekunden bis Minuten stehen und ist momentan unfähig, die Beine zu bewegen. Die Patienten beschreiben dieses Gefühl, als wären die Füße am Boden festgeklebt [vgl. Hauser, 2003, S. 71].

### 2.1.2.4 Posturale Instabilität

Die posturale Instabilität äußert sich in Gleichgewichtsproblemen, dies wirkt sich zum Beispiel auch auf das Gangbild der betroffenen Personen aus. Sie lässt sich dadurch nachweisen, dass der aufrecht stehende Patient auf ein Schubsen nach vorne, hinten oder zur Seite unsicher reagiert, nicht auspendeln kann, sondern durch einen Ausfallschritt das Fallen verhindern muss oder sogar zu Fall kommt. Es ist noch nicht geklärt, ob die Symptome durch Tiefe Hirnstimulation gebessert werden können, sie sprechen jedenfalls nicht gut auf Dopaminergika<sup>2</sup> an [vgl. Gerlach, 2003, S. 25].

### 2.1.2.5 Weitere Symptome

Sekundäre Merkmale beinhalten Fehlfunktionen des autonomen Nervensystems (Darmträgheit, Schwitzen), kognitiven Abbau (Demenz), emotionelle Störungen (Depression) und sensorische Beschwerden, wie zum Beispiel Schmerzen in den Muskeln [vgl. Gerlach, 2003, S. 25–28].

---

<sup>1</sup> Vermindertes Muskelspiel, welches den Gesichtsausdruck bestimmt („Maskengesicht“)

<sup>2</sup> Substanzen, welche Dopaminrezeptoren stimulieren

### 2.1.3 Medikamentöse Therapie

L-Dopa (L-3,4-Dihydroxyphenylalanin), ein Vorläufer von Dopamin, ist bis jetzt die beste Medikation zur Behandlung der Symptome von Morbus Parkinson. Dopamin selbst kann die Blut-Gehirn-Barriere nicht überwinden, L-Dopa hingegen wird mit Hilfe des Transportsystems für große neutrale Aminosäuren vom Blut aus in das Gehirn transportiert. Es wird zusätzlich ein Decarboxylasehemmer, zum Beispiel Carbidopa, verabreicht, um den frühzeitigen Abbau des L-Dopas vor Erreichen des Gehirns zu hemmen. Die Halbwertszeit von L-Dopa in Verbindung mit Carbidopa beträgt ungefähr 90 Minuten. Eine kurzzeitige Nebenwirkung besteht in Übelkeit und Erbrechen. Bei L-Dopa kommt es allerdings auch zu Langzeit-Nebeneffekten (siehe Abbildung 2.2), im speziellen motorischen Schwankungen (Fluktuationen) und – schwerwiegender – zu Dyskinesien, welche auf das Fortschreiten der Krankheit und die kurze Halbwertszeit von L-Dopa zurückzuführen sind. Die motorischen Schwankungen werden immer stärker, je mehr die Krankheit voranschreitet, da immer weniger Neuronen zur Speicherung des vom L-Dopa abgeleiteten Dopamins zur Verfügung stehen. Es kommt innerhalb eines Tages zu Perioden mit der typischen Parkinson-Symptomatik (Akinesie, Rigidität, Immobilität), welche als OFF-Phasen bezeichnet werden und zu Perioden abwesender Symptomatik, welche als ON-Phasen bezeichnet werden.

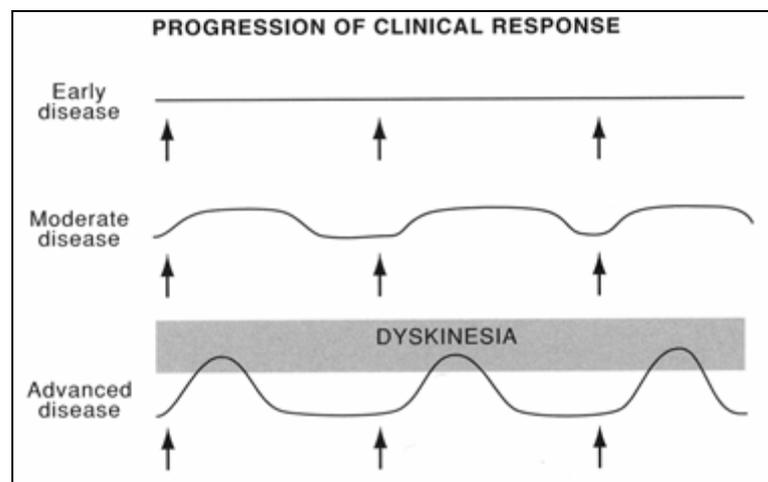


Abbildung 2.2: Fortschreiten der klinischen Auswirkungen [Hauser, 2003, S. 62]

Dyskinesien hingegen sind abnormale, ungewollte Bewegungen, welche mit Schmerzen verbunden sein können [vgl. Hauser, 2003, S. 62–64].

Genauer wird nach einer Phase der guten klinischen Wirkung des L-Dopa (Phase I, „Honey-Moon“) im Verlauf der Krankheit die Wirksamkeit des L-Dopa-Präparates unter der bisherigen Dosierung nachlassen, man wird höher dosieren und/oder ein weiteres Medikament hinzufügen müssen. Danach wird die Wirkung der Medikation vor dem jeweils nächsten Einnahmezeitpunkt abfallen (Wearing-off, End-of-dose; siehe Abbildung 2.3), später kommt es dann zu Dyskinesien

beim Anfluten und Abfluten des L-Dopa-Spiegels (biphasische Dyskinesien; siehe Abbildung 2.4). Zu dem Zeitpunkt wird auch erstmals Freezing feststellbar sein.

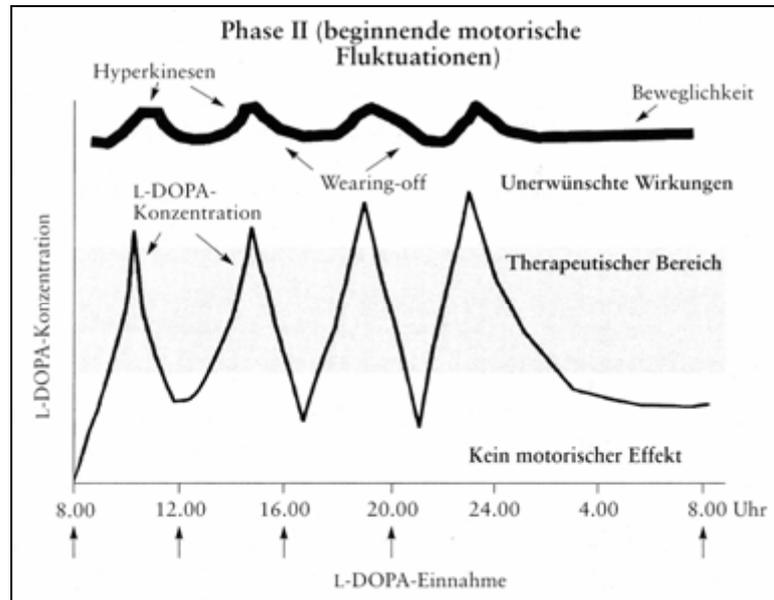


Abbildung 2.3: Schema der Wirkung von L-Dopa in Phase II [Gerlach, 2003, S. 235]

In der dritten Phase kommt es zu unverhofften, pharmakokinetisch nicht mehr nachvollziehbaren ON-OFF-Fluktuationen (Yo-Yoing, rapid oscillation).

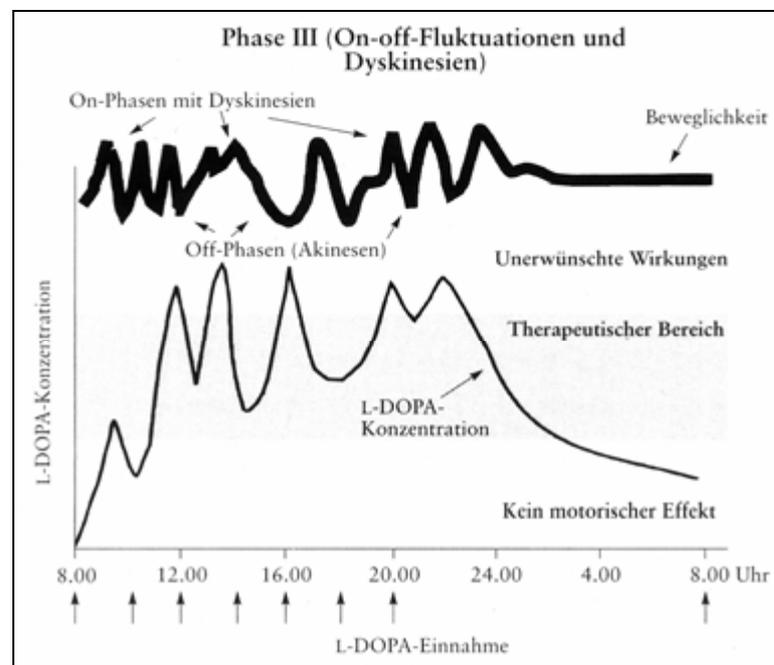


Abbildung 2.4: Schema der Wirkung von L-Dopa in Phase III [Gerlach, 2003, S. 235]

Das sogenannte „Therapeutische Fenster“, in welchem die L-Dopa Dosis gerade wirkt und keine Dyskinesien auftreten, wird mit längerer Anwendung immer schmaler (siehe Abbildung 2.5).

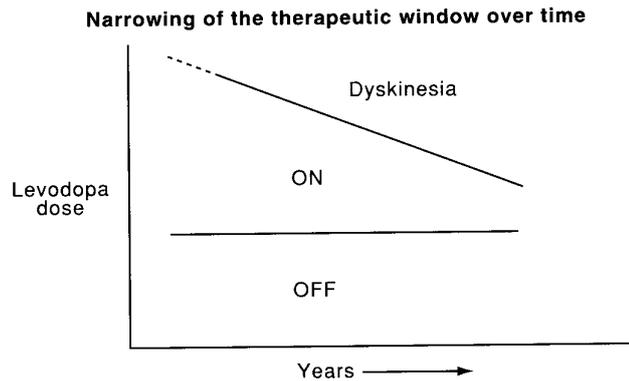


Abbildung 2.5: Schmälerung des therapeutischen Fensters über die Zeit [Hauser, 2003, S. 66]

Catechol-O-methyltransferase-Hemmer (COMT-H.) schwächen den peripheren L-Dopa Metabolismus, welches zusätzlich die Verfügbarkeit des L-Dopas erhöht [vgl. Hauser, 2003, S. 80, 104].

Dopamin-Agonisten wirken direkt als Dopamin-Ersatzstoffe, indem sie die postsynaptischen Dopaminrezeptoren stimulieren. Nebenwirkungen können jedoch Übelkeit, Erbrechen und orthostatische Hypotonie (plötzlicher Blutdruckabfall, der Schwindel verursachen kann) sein. Weiters können auch zentrale, von Dopamin erzeugte Nebeneffekte, wie Alpträume, Halluzinationen und andere psychiatrische Symptome auftreten [vgl. Hauser, 2003, S. 86].

Monoaminoxidase Typ B-Hemmer (MAO-B-H.) verlangsamen den Abbau des körpereigenen und des von L-Dopa abgeleiteten Dopamins und erhöhen so die Verfügbarkeit von Dopamin im Gehirn [vgl. Hauser, 2003, S. 95].

Neuroprotektive Therapien, welche die neuronale Degeneration verlangsamen sollen, sind bis jetzt leider noch nicht bestätigt worden [vgl. LeWitt, 1999, S. 108; Hauser, 2003, S. 74].

Restaurative Therapien sind Therapien, welche darauf abzielen, die verlorenen Neuronen zu ersetzen. Dies kann zum Beispiel durch Transplantation von Zellen geschehen. In Versuchen haben solche Zellen schon überlebt, neuronale Verbindungen wiederhergestellt und die Dopaminkonzentration erhöht, jedoch ergab sich leider keine Verbesserung für die Patienten. Derzeit werden Stammzellen, embryonale Zellen, gentechnisch veränderte Zellen und Zellen aus anderen Körperteilen intensiv auf einen möglichen Einsatz hin untersucht [vgl. Hauser, 2003, S. 74].

Eine vor kurzem durchgeführte Recherche auf Pubmed nach den Schlagworten „Parkinson“ und „Therapy“ hat ergeben, dass neurorestaurative Therapien bis jetzt Gegenstand der Forschung und noch nicht etabliert sind. Als Beispiel einer der neuesten Arbeiten soll hier ein Auszug aus dem Abstract einer Arbeit, welche im Mai dieses Jahres veröffentlicht wird, wiedergegeben werden:

“When F3 human NSCs (Anm.: neural stem cells) were implanted into the brain of murine models of lysosomal storage diseases, stroke, Parkinson disease, Huntington disease or stroke, implanted F3 NSCs were found to migrate to the lesion sites, differentiate into neurons and glial cells, and restore functional deficits found in these neurological disorders. In animal models of brain tumors, F3 NSCs could deliver a bioactive therapeutically relevant molecules to effect a significant anti-tumor response intracranial tumor mass. Since these genetically engineered human NSCs are immortalized and continuously multiplying, there would be limitless supply of human neurons for treatment for patients suffering from neurological disorders including stroke, Parkinson disease, Huntington disease, ALS, multiple sclerosis and spinal cord injury. The promising field of stem cell research as it applies to regenerative medicine is still in infancy, but its potential appears limitless, and we are blessed to be involved in this exciting realm of research”<sup>1</sup>.

## 2.2 Operationen

### 2.2.1 Allgemeines

Wir können derzeit drei verschiedene Operationstechniken unterscheiden: Zufügen einer Läsion im Gehirn, Transplantation und Tiefe Hirnstimulation. Bei läsionschirurgischen Eingriffen werden kleine Gehirnareale zerstört. Dies wird normalerweise dadurch erreicht, dass eine lange, schmale Elektrode in das Gehirn mit der Spitze bis in den Thalamus (Thalamotomie), Global Pallidus (Pallidotomie) oder Subthalamus (Subthalamotomie) eingeführt wird. Danach wird die Läsion üblicherweise dadurch erreicht, dass die Spitze der Elektrode für ca. 1 Minute erhitzt wird, dieses Verfahren wird Thermokoagulation genannt [vgl. Hauser, 2003, S. 125].

Bei der Transplantation wird versucht, den Verlust von Dopamin-Neuronen auszugleichen. Da die Zellen in Lokalisation und Art sehr spezifisch sind, ist es eine attraktive Möglichkeit, diese Zellen zu ersetzen, anstatt zu versuchen, die defekten Neuronen wiederherzustellen. Diese Methoden befinden sich noch im Erprobungsstadium, hauptsächlich werden hierfür Dopamin-Neuronen aus der Substantia nigra von menschlichen Embryonen, 3 bis 5 pro Gehirnhälfte, beziehungsweise menschliche Stammzellen verwendet. Es gibt aber alternative Ansätze, Dopaminzellen von Schweinen und auch andere Zellen, wie zum Beispiel aus der menschlichen Retina, zu verwenden,

---

<sup>1</sup> Su, Kim: Genetically engineered human neural stem cells for brain repair in neurological diseases. (Abstract) In: Brain & Development, Vol. 29 No. 4, 2007, S. 193

welche bisher aber keine nennenswerten Resultate geliefert haben oder noch nicht ausreichend überprüft wurden [vgl. Hauser, 2003, S. 125–126].

## **2.2.2 Deep Brain Stimulation (DBS)**

### **2.2.2.1 Einführung in die DBS**

Während durchgeführter Thalamotomien hat sich schon sehr lange herausgestellt, dass Stimulationen des Zieles der Operation, genauer des Nucleus ventralis intermedius (ventral intermedius nucleus, VIM), den gleichen Effekt wie dessen Zerstörung hatten [vgl. Benabid1, 1995, S. 381]. Dieser Effekt ist noch weitgehend ungeklärt, eine Theorie besagt, dass durch die Stimulation eine Überaktivität der stimulierten Zentren reduziert wird.

Eine zweite Theorie geht davon aus, dass bei Parkinson die elektrischen Entladungen dieser Kerne abnormal und unregelmäßig werden. Durch die Stimulation werden die elektrischen Entladungen zurückgesetzt und normalisiert [vgl. Hauser, 2003, S. 142].

Die Mechanismen zur Unterdrückung von Tremor durch VIM-Stimulation könnten durch kontinuierliche Störung des neuronalen Netzwerkes, welches die propriozeptiven Informationen vom Rückenmark zum Kortex leitet, dort einen Reflex auslöst und zum Vorderhorn weiterleitet, erklärt werden. Vom Vorderhorn aus triggern motorische Nerven eine Muskelkontraktion und erzeugen dadurch wieder einen propriozeptiven Reiz. Wenn man dieses System mit künstlichem neuronalen Rauschen speist, könnte dies dieses zyklische Phänomen deaktivieren und den Tremor stoppen. Man könnte bei Beobachtung dieses Effektes der Neurostimulation überlegen, dass neuronale Netzwerke auch von einem kybernetischen Prozess beeinflusst werden könnten. Dabei basiert die Datenverarbeitung eines Netzwerkes auf der Erkennung von sinnvollen Informationen, welche dann in einen Verarbeitungsalgorithmus gespeist werden, der von der funktionellen Architektur des Netzwerkes bestimmt wird. Schließlich führt dies zu einer Ausgabe des Netzwerkes, die ein bestimmtes Verhalten auslöst. Wenn dieses Netzwerk nun mit einem zusätzlichen Signal gespeist wird, dessen Charakteristik, in diesem Fall hauptsächlich in der Frequenz, reinem (neuralem) Rauschen entspricht, wird es dem Netzwerk unmöglich, sinnvolle Informationen aus dem gesamten Fluss an Informationen zu extrahieren und so ein entsprechendes Verhalten zu generieren. Dies ist ein bekanntes Konzept der Kybernetik und der automatisierten Systeme, wo hochfrequente Signale am Eingang dazu benutzt werden können, ein oszillatorisches Verhalten eines solchen rückgekoppelten Systems zu unterbrechen [vgl. Benabid1, 1995, S. 396–397].

Die Vorteile der Tiefen Hirnstimulation im Vergleich zur Thermokoagulation liegen in der Möglichkeit der bilateralen Platzierung der Elektroden, der leichteren Korrigierbarkeit der

beeinflussten Gehirnregion, der Variabilität der Stimulationsparameter und im Vermeiden irreversibler Läsionen des Gehirnes [vgl. Gerlach, 2003, S. 214].

Für die Neurostimulation stehen prinzipiell drei Zielpunkte zur Wahl. Am häufigsten wird der Nucleus subthalamicus (STN) für die medikamentös therapierten Patienten mit dem L-Dopa Spätsyndrom gewählt. Die Syndrome Rigor, Tremor und der Akinese (Verlust von Bewegung) können durch die STN-Stimulation sehr gut gelindert werden. Bei der Stimulation des Globus pallidus (GPi) werden hauptsächlich Dyskinesien gebessert, bei der Stimulation des VIM wird hauptsächlich der Tremor reduziert [vgl. Hauser, 2003, S. 135–139].

Die Wirkungen einer Tiefen Hirnstimulation bei Morbus Parkinson sind in [Przuntek, 2003, S. 53] wie folgt übersichtlich dargestellt:

Tabelle 2-1: Wirkungen einer Tiefen Hirnstimulation bei Morbus Parkinson [Przuntek, 2003, S. 53]

	STN	GPi	VIM
Tremor	+++	++	+++
Akinese	+++	++	0
Rigor	+++	++	+
Gang	+++	++	0
Dyskinesien	++	+++	+
Off-Dystonie	+++	++	0
L-Dopa-Reduktion	+++	+/0	+/0
Aufwand der postoperativen Versorgung	groß	gering	gering

Die originale Arbeit gibt Aufschluss über die Bedeutung der Kürzel: +, ein wenig effektiv; ++, mäßig effektiv; +++, sehr effektiv; 0, kein Effekt; -, könnte schädlich sein [vgl. Pollak, 2002, S. 80].

Die Tiefe Hirnstimulation ist reversibel, im Gegensatz zur Zufügung von Läsionen. Sie ist eine stereotaktische Methode, das bedeutet, dass sie sich ein dreidimensionales Koordinatensystem zunutze macht, um kleine Orte im menschlichen Körper aufzufinden. Die Qualität und Stabilität der Ergebnisse hängen hauptsächlich von der Positionierung der Elektrode(n) ab, daher wird ein hoher Aufwand betrieben, diese Genauigkeit sicherzustellen. Die gesamte Prozedur kann zwar in einem Schritt geschehen, wird aber oftmals in mehreren Schritten, häufig auf drei Tage verteilt, durchgeführt [vgl. Benabid1, 1995, S. 381].

Der große Erfolg der Tiefen Hirnstimulation ist stark von der engen Kooperation zwischen der Neurologie und der Neurochirurgie abhängig. Im Allgemeinen hat jede der Disziplinen ihre Zuständigkeiten. Der Neurologe ist üblicherweise für die Selektion der Patienten und das postoperative Management der Patienten inklusive der Programmierung, Medikamenteneinstellung und Behandlung von nichtchirurgischen Komplikationen zuständig. Der Neurochirurg ist zuständig für die Platzierung des DBS-Systems, die perioperative Behandlung der Patienten und das Management von chirurgischen Langzeitkomplikationen. Das bedeutet aber nicht, dass der Neurochirurg nicht in die Patientenselektion und das postoperative Management oder der Neurologe nicht in die Operation involviert ist [vgl. Benabid2, 2006, S. 169].

### **2.2.2.2 Patientenselektion für die DBS**

Die Eignung eines Patienten für die Tiefe Hirnstimulation wird vom jeweiligen behandelnden Neurologen festgestellt. Kriterien hierfür kann man den Konsensusrichtlinien der Österreichischen Parkinson Gesellschaft zur Tiefen Hirnstimulation bei Morbus Parkinson [ÖPG, 2003] entnehmen. Diese beinhalten unter anderem ein ausgezeichnetes Ansprechen auf L-Dopa beziehungsweise Apomorphin, einem anderen Dopamin-Agonisten, und die früher beschriebenen Wirkfluktuationen und L-Dopa-induzierten Dyskinesien.

Weiters kann die mittels Tiefer Hirnstimulation optimal erreichbare Symptomkontrolle nicht besser sein, als die beim jeweiligen Patienten optimal induzierbare, medikamentöse „ON“-Phase. Für die Operation müssen also die Symptome der Wirkfluktuationen und Dyskinesien trotz optimaler Medikation gegeben sein. Die Einschlusskriterien sind wie folgt zusammengefasst:

- idiopathische Parkinson-Patienten (IPS)

- ausgezeichnetes Ansprechen auf L-Dopa/Apomorphin
- Hoehn und Yahr-Stadium im ON nicht schlechter als III<sup>1</sup>
- Wirkfluktuationen und L-Dopa induzierte Dyskinesien
- Parkinson-Patienten mit therapierefraktärem Tremor

Der Nucleus subthalamicus, der gegenwärtig als Zielgebiet favorisiert wird, greift auch in limbische und kognitive Regelkreise ein. Daher ist ein entsprechender Eingriff bei depressiven oder dementen Patienten nicht indiziert [vgl. Hausotter, 2003, S. 77].

Patienten mit Demenz sind wegen des hohen Risikos von postoperativen Verwirrheitszuständen oder Psychosen keine geeigneten Operationskandidaten. Auch Patienten mit schweren präoperativen Persönlichkeits- und Verhaltensstörungen beziehungsweise auch mit medikamentös induzierten, schwereren Graden der Lust- und Antriebsverhaltensstörung kommen für eine Operation nicht in Frage, da diese als mögliche Nebenwirkung der Tiefen Hirnstimulation verstärkt werden können. Die Ausschlusskriterien sind wie folgt zusammengefasst [vgl. ÖPG, 2003]:

- Demenz: MMSE<sup>2</sup> < 24 Punkte, DSM IV Kriterien<sup>3</sup>
- Major Depression mit akuter Suizidalität
- schwere Persönlichkeits-/Verhaltensstörung (homeostatische hedonistische Dysregulation)
- non IPS
- strukturelle Hirnläsionen: Hirnatrophie, hypertensive Vaskulopathie, Tumor, Fehlbildungen, AV-Malformationen, Aneurysmen, Hydrocephalus
- internistische Kontraindikationen für eine OP

---

<sup>1</sup> Die Krankheitsstadienskala nach Hoehn und Yahr ist in der Praxis auf Grund seiner einfachen Anwendbarkeit das am häufigsten verwandte Einteilungsschema [vgl. Hausotter, 2003, S. 39].

Stadium III: Geringe bis mäßige Behinderung, gestörte Stellreflexe mit Unsicherheit beim Umdrehen und bei Außenreizen, posturale Instabilität, Arbeitsfähigkeit in Abhängigkeit vom Beruf noch zum Teil erhalten.

<sup>2</sup> Mini Mental State Examination (1975 durch Marshal F. Folstein u.a.); eine ohne großen Zeitaufwand durchführbare Erhebung zur Demenzerkennung.

<sup>3</sup> Diagnostisches und statistisches Manual psychischer Störungen in der 4. Auflage (DSM-IV) der American Psychiatric Association, [vgl. Saß, 2001, S. 174].

Die Grenzwerte für die Diagnose von Demenz für die MMSE und „Dementia Rating Scale“ (DRS) sollten bei Personen höheren Alters oder mit geringerer als durchschnittlicher Bildung mit Vorsicht interpretiert werden. Die Standardgrenzwerte für die Alzheimer'sche Krankheit (MMSE 23–24) könnten inadäquat sensibel oder spezifisch für die Erkennung von geringer Demenz bei der Parkinson'schen Krankheit, charakterisiert durch das frontale dysexekutive Syndrom, sein. Diese Score-Werte sollten in Zusammenhang mit einer qualitativen Analyse von Fehlern interpretiert werden. Bei der DRS kann ein Patient mit Dysarthrie viele Punkte beim Test des Redeflusses verlieren, obwohl dieser trotzdem für eine Tiefe Hirnstimulation in Frage kommt und deswegen einer vollen neuropsychologischen Evaluierung unterzogen werden sollte. Eine Demenz sollte von kognitiven Sekundärsymptomen verursacht durch Medikation oder Depressionen, niedrigem Intelligenzniveau und schlechter Testergebnisse aufgrund unterdurchschnittlicher Bildung differenziert werden. Neuropsychologische Evaluierung ist essentiell, wenn die kognitiven Funktionen eines Patienten abnormal erscheinen, aber die Ursache dafür unbekannt ist [vgl. Lang2, 2006, S. 177].

In [Gerlach, 2003, S. 214–215] sind weitere motorische Beeinträchtigungen, die eine Operation rechtfertigen, angeführt:

- ein hochamplitudiger Tremor, der zu einer funktionellen Beeinträchtigung führt
- eine Akinese von mindestens drei Viertel in den repetitiven Bewegungen der UPDRS, Unfähigkeit aufzustehen oder sich im Bett zu drehen
- ein kleinschrittiges Gangbild, L-Dopa-sensibles Freezing (mit Stürzen), Unfähigkeit ohne Hilfe zu gehen oder
- Dyskinesien mit schwerer funktioneller Beeinträchtigung oder schmerzhaftes OFF-Phasen-Dystonien<sup>1</sup>

Weiters sind von der österreichischen Parkinsongesellschaft [ÖPG, 2003] abweichend auch folgende allgemeinere und auch zusätzliche Kontraindikationen für eine operative Therapie beim IPS zusammengefasst, welche auch das Alter der Patienten inkludiert:

---

<sup>1</sup> Dystonien gehören zur Gruppe der Bewegungsstörungen, deren Ursache in Störungen der motorischen Zentren im Gehirn liegt und sich durch unwillkürliche Verkrampfungen und Fehlhaltungen äußern.

- allgemein:
  - Alter > 75 Jahre
  - schwere zerebrale Makroangiopathie
  - schwere internistische Begleiterkrankungen
  - Antikoagulanzen-therapie
  - Immunsuppression
  - Pseudobulbärparalyse
  
- neuropsychologisch und psychiatrisch:
  - Demenz (wie bei [ÖPG, 2003] auch MMSE  $\leq$  24, zusätzlich Mattis-Demenz-Skala<sup>1</sup>  $\leq$  130)
  - schwere frontale exekutive Störung
  - paranoide oder halluzinatorische Psychose
  - fehlende Kooperationsfähigkeit
  - schwere manische (oder depressive) Störung

Ein multidisziplinärer Ansatz durch Neurochirurgen, Neurologen und Neuropsychologen ist wichtig, um festzustellen, ob der Patient ein geeigneter Kandidat für die Operation ist. Das Alter ist einer der wichtigsten Faktoren, wie der Patient mit der Operation zurechtkommt und sich postoperativ verhält. Ein fortgeschrittenes Alter ist verbunden mit einer höheren Inzidenz von kognitiven Veränderungen nach der Operation [vgl. Machado, 2006, S. 248].

In einer neueren Publikation, welche 547 Artikel, die als Suchtreffer in Zusammenhang mit Tiefer Hirnstimulation und Parkinson'scher Krankheit in Medline erschienen sind, und zwischen 1965 und Mai 2004 publiziert wurden, untersucht hat, wurde unter anderem die Frage nach dem besten

---

<sup>1</sup> Mattis S.: Mattis dementia rating scale (1976). In: Burns, A; Lawlor, B; Craig, S.: Assessment scales in old age psychiatry. – London: Martin Dunitz. S. 186, 1999.

Alter für eine Operation gestellt. Einige Publikationen nannten ein Alterslimit von 70 Jahren und andere wiederum von 75 Jahren, ohne dieses jedoch ausreichend zu begründen. Die Aussage dieser Untersuchung ist, dass zu wenig Evidenz von verfügbaren klinischen Studien existiert, um den Einfluss des Alters auf das postoperative Ergebnis nach einer Tiefen Hirnstimulation bestimmen zu können [vgl. Lang2, 2006, S. 172–172].

Auch die kürzeste Krankheitsdauer wurde in dieser Publikation untersucht. Als Ergebnis ist herausgekommen, dass diese unbekannt ist. Verfügbare Studien haben Levodopa-sensitive Patienten mit einer mittleren Krankheitsdauer von ungefähr 12 bis 15 Jahren ausgewählt. Es wird vorgeschlagen, dass ein Patient eine Krankheitsdauer von mindestens 5 Jahren hat, bevor er für eine Tiefe Hirnstimulation in Erwägung gezogen wird, damit atypische Formen von Parkinson evident und ausgeschlossen werden können [vgl. Lang2, 2006, S. 183].

Für jeden Patienten wird unter Kenntnis der sozialen, familiären und beruflichen Situation und unter Miteinbeziehung des neurologischen Befundes eine individuelle Nutzen-Risiko-Analyse aufgestellt. Es ist wichtig, zu berücksichtigen, dass durch die DBS keine Heilung möglich ist, sondern dass die Erkrankung weiterhin fortschreitet. Auch ist es ein verbreiteter Irrtum, dass nach einer Operation keinerlei Medikamenteneinnahme mehr notwendig ist [vgl. Gerlach, 2003, S. 214–215].

### **2.2.2.3 Durchführung der DBS-Operation**

Nach der präoperativen standardisierten Diagnostik folgt die perioperative Pharmakotherapie. Für die Operation müssen die Patienten frei von motorischen Effekten der bisherigen medikamentösen Therapie sein. Die Medikamente haben verschiedenste Ausschleichzeiten. Diese betragen bis zu 48 Stunden. Patienten mit einer „long duration response“ können unter Umständen länger benötigen, damit die motorischen Effekte dem „OFF“-Status entsprechen [vgl. ÖPG, 2003].

In der radiologischen Session wird der Kopf des Patienten in einen CT-kompatiblen stereotaktischen Rahmen unter der Wirkung von generellen Anästhetika eingespannt. Vier temporäre Tantal-Schrauben, die als Positionierungshilfen fungieren, indem sie zu entsprechenden Gegenstücken an stereotaktischen Rahmen passen, werden hierzu in den Schädel des Patienten geschraubt. Danach folgt die Ventrikulographie, es werden zwei Löcher in den Schädel gebohrt und eine 65 mm lange Kanüle in das vordere Horn von beiden seitlichen Hirnkammern (lat. ventriculus lateralis) eingeführt und zur Bestätigung der Position eine Luftblase von 5 ml injiziert, welche auf einer Röntgenaufnahme gut dargestellt werden kann. Nach der Bestätigung durch die Röntgenkontrolle wird ein Kontrastmittel durch die Kanüle gespritzt und eine vordere und seitliche Röntgenaufnahme gemacht. Um die dritte Hirnkammer (lat. ventriculus tertius) mit der vorderen (AC, anterior commissure) und der hinteren Querverbindung (PC, posterior commissure), zwei

wichtigen Referenzpunkten bei der Vermessung des Gehirnes, darzustellen, sind weitere Röntgenaufnahmen notwendig [vgl. Benabid1, 1995, S. 382].

In einer zunehmenden Zahl von Zentren in Österreich wird auf die Ventrikulographie verzichtet und stattdessen werden mittels Bildfusionstechnik und spezieller Software die Vorteile von MRT (Magnetresonanztomographie; gute Visualisierung von Gewebe, z.B. des STN) und des CT (Computertomographie; geometrische Genauigkeit) kombiniert [vgl. ÖPG, 2003].

Am nächsten Tag folgt eine MRT-Session, in der das gesamte Gehirn, eingespannt in einen MRT-tauglichen, stereotaktischen Rahmen, welcher z. B. mit Kupfersulfat, einem im MRT gut sichtbaren Material, gefüllt ist, aufgenommen wird [vgl. Benabid1, 1995, S. 382]. Anhand dieser Aufnahmen wird die gesamte Operation geplant, das bedeutet, es werden Zielpunkte, Bohrlöcher, Eintrittswinkel und vieles mehr in Bezug auf den stereotaktischen Rahmen berechnet [vgl. ÖPG, 2003]

Am dritten Tag wird der nicht anästhetisierte Patient nochmals in den CT-kompatiblen stereotaktischen Rahmen mit Hilfe der implantierten Schrauben platziert [vgl. Benabid1, 1995, S. 382]. Die Verwendung von Mikroelektroden-Rekording sollte heutzutage obligatorisch bei der Operation sein. Die Methode zur Aufnahme von neuronalen Signalen wurde weiterentwickelt und steht heute in kommerziellen Geräten, wie zum Beispiel Neurotrek von Alpha Omega oder Leadpoint von Medtronic, zur Verfügung, welche mit einer leistungsfähigen Analysesoftware ausgestattet sind [vgl. ÖPG, 2003].

Elektrophysiologisches Mapping ist ein essentieller Teil bei der Implantierung einer DBS-Elektrode bei der Parkinson'schen Krankheit und Tremor. Es sind einige Technologien verfügbar, unter anderem Mikroelektroden- und Semimikroelektroden-Rekording, Mikro- und Makrostimulation. Jedes dieser Verfahren in professioneller Anwendung scheint befriedigende Ergebnisse zu liefern, aber aufgrund von fehlender Evidenz ist es unmöglich, die relativen Risiken und den Gewinn der einzelnen Technologien zu bewerten. Den meisten Studien fehlen Informationen, wie zum Beispiel die Anzahl der verwendeten Spuren, um diese vergleichbar zu machen [vgl. Gross, 2006, S. 280].

Ein Mikroelektroden-Rekording System besteht aus einer Spannvorrichtung mit einem Führungsteil, einem manuellen oder automatisch getriebenen Elektrodenvorschub, den Mikroelektroden, den Kabelverbindungen zu den Mikroelektroden, einem Verstärkerteil zur Verstärkung und Filterung der von den Mikroelektroden aufgenommenen Potentiale, einem A/D-Wandlerteil, mit dem die verstärkten Signale digitalisiert werden, um danach vom Computer verarbeitet werden zu können, einem Stromstimulationsteil, mit dem eine Stromstimulation von Gehirnarealen erfolgen kann und der Verarbeitungseinheit mit Computer und darauf laufender

Analysesoftware und Lautsprecher, wo die aufgenommenen Signale auf dem Bildschirm optisch und auf dem Lautsprecher akustisch ausgegeben werden.

Die Spannvorrichtung, in der bis zu 5 Mikroelektroden eingespannt werden können, wird auf den stereotaktischen Rahmen montiert. Die 1–5 Elektroden werden in einen zylinderförmigen Aufnahmeteil mit fünf Bohrlöchern, welche die Elektroden aufnehmen können, eingespannt, der auf einem Schlitten ruht. Dieser Schlitten wird entweder über eine Mikrometerschraubenmechanik, ähnlich wie sie bei Mikrometern üblich ist (manueller Sondenvorschub), oder einen Motor vor- und rückwärts getrieben (motorgetriebener Sondenvorschub). Dabei werden die Mikroelektroden üblicherweise in Führungskanülen geführt, aus denen sie maximal wenige Zentimeter herausragen. Da die Spitze der Mikroelektroden mit einem Durchmesser von ungefähr 4 µm bei Berührung mit einem härteren Material sofort defekt werden, dienen die Kanülen einerseits als Schutz, andererseits wird das Einspannen wesentlich erleichtert und der Weg der Mikroelektroden außerdem präziser. Die Kanülen ragen nach vorne aus einem Führungsteil heraus, der eine weitere Stabilisierung der Elektrodenführung ermöglicht. Der Mikroelektrodenvorschub wird üblicherweise vom Neurochirurgen bedient. Die von den Mikroelektroden aufgenommenen Signale werden über die Verbindungskabel zum Verstärker geleitet, welcher die Signale filtert und verstärkt<sup>1</sup>.

Es wird zuerst ein ca. 5 cm langer Schnitt in die Kopfhaut vorgenommen und die Kopfhaut vom Schädel getrennt. Dadurch bildet sich eine Tasche, wo in weiterer Folge die externen Komponenten der DBS-Elektrode und der Anschluss des Verlängerungskabels Platz finden werden [vgl. Benabid1, 1995, S. 386].

Durch die Verwendung von mehreren Elektroden kann ein größeres Gebiet beurteilt werden, wodurch das Auffinden der optimalen DBS-Elektrodenposition erleichtert wird. Hierzu wird ein ca. 6 mm großes Loch in den Schädel gebohrt und ein 5-löchiger Führungsteil bis zur Gehirnhaut eingeführt. Dann werden 5 Kanülen mit Hilfe eines Mikromanipulators durch den Führungsteil ins Gehirn eingeführt. Die Kanülen enthalten jeweils eine Mikroelektrode, die 15 mm aus der Kanüle herausragt. Es werden dann die Signale aller 5 Mikroelektroden aufgezeichnet und anschließend die beste der 5 Trajektorien basierend auf dem klinischen Effekt gewählt. Offensichtlich ist das Risiko, ein Blutgefäß zu treffen, fünfmal höher, als wenn man nur eine Elektrode verwenden würde. Es ist aber das gemessene Areal größer, was die Chance zur Findung der besten Stimulationsposition

---

<sup>1</sup> Erfahrung des Verfassers

erhöht. Dies ist vor allem für kleine Zielareale, wie zum Beispiel beim STN, sehr hilfreich [vgl. Benabid1, 1995, S. 387].

Die Zelleitung wird ungefähr 10 mm vor dem berechneten Ziel begonnen. Die Elektroden werden dann jeweils in Schritten von 1 mm und ungefähr 3 mm vor dem Ziel in Schritten von 0,1 mm weiterbewegt und die Zellaktivität mit dem Computer aufgezeichnet [vgl. ÖPG, 2003].

Die exakte Position jeder Stimulations- und Aufnahme Stelle wird mittels Röntgenaufnahme dokumentiert. Diese Aufzeichnungen bieten ein Daten-Set, mit dessen Hilfe die exakte Lokalisation des Zielortes (z. B. VIM) für die Platzierung der chronischen Stimulationselektrode ermittelt werden kann. Abweichungen von einigen wenigen Millimetern zwischen dem theoretischen (Ventriculographie-basierten) und dem korrigierten (Elektrophysiologie-basierten) Ziel können beobachtet werden. Deswegen ist die elektrophysiologische Evaluierung ein wichtiges Mittel zur Lokalisationsfindung.

Mit einer Mikroelektrode kann man einzelne Einheiten aufnehmen, die Zellen des VIM feuern Bursts mit 5–10 großen Spitzen. Mit einer größeren Makroelektrode kann man nur mehrere Einheiten auf einmal aufnehmen, die Amplituden dieser Spitzen variieren entlang der Elektrodenbahn und geben Auskunft über die verschiedenen Gewebsgrenzen [vgl. Benabid1, 1995, S. 385].

Die typische Entladungsaktivität des STN liegt bei einer Frequenz von 20–60 Hz, hat wenig Hintergrundaktivität und eine ausgeprägte Responsivität des Burstverhaltens auf passive Bewegungen der Extremitäten und eventuell tremorsynchrone Zellaktivität [vgl. ÖPG, 2003].

Nach der Messung der Zellenaktivität erfolgt die klinische Testung, welche einen wesentlichen Faktor für die Zielpunktfindung darstellt. Wenn man eine kombinierte Mikro-Makro-Elektrode zur Ableitung verwendet, kann man den Mikroelektrodenteil beim Hineinfahren der Elektrode zur Messung und danach den Makroelektrodenteil zur Probestimulation beim Herausfahren der Elektrode verwenden<sup>1</sup>. Als verlässlicher Indikator dient der Rigor, da er einerseits wenig vom Patienten beeinflusst wird und andererseits innerhalb von Sekunden auf die Stimulation anspricht, wogegen der Tremor und die Bradykinesie zeitlich verzögert reagieren können. Die Prüfung des Rigors erfolgt kontinuierlich am Handgelenk, während gleichzeitig die Stimulationsintensität

---

<sup>1</sup> Erfahrung des Verfassers

stufenweise bis zum Einsetzen von Nebenwirkungen erhöht wird. Die Reduktion des Rigors während der Stimulation kann zum Beispiel in % Reduktion (Auflösung 10 %) im Vergleich zur Basislinie angegeben werden. Durch die Stimulation des STN kann bereits bei niedrigen Spannungen eine nahezu 100 %ige Rigorreduktion erwartet werden [vgl. ÖPG, 2003].

Neben der Testung der Parkinson-Symptome wird von dem der Operation beiwohnenden Neurologen besonders das Auftreten von Nebenwirkungen beobachtet, da dies einen Hinweis auf die Lage der Elektrode in Bezug zum STN geben könnte. Das Auftreten von Dyskinesien während der Stimulation kann nur durch Stimulation des STN induziert und als weiterer Faktor für die Zielpunktfindung herangezogen werden [vgl. ÖPG, 2003].

Eine DBS-Elektrode mit freiliegenden Kontakten wird anschließend an die Evaluierung in das Zielgebiet im Gehirn geführt und nach einer Überprüfung der Elektrodenlokalisierung mittels Röntgenaufnahme fixiert. Die DBS-Elektrode wird mit dem Verlängerungskabel verbunden, die zu diesem Zeitpunkt nicht benötigte Länge des Verlängerungskabels wird in die perkutane Tasche gefaltet und diese später sorgfältig geschlossen. Wenn die Symptome bilateral auftreten, so muss auch eine Elektrode für die kontralaterale Seite nach der gleichen Prozedur implantiert werden. Die DBS-Elektrode wird später, nach einer Testphase mit einem externen Stimulator, über das Verlängerungskabel, welches unter der Haut zu einem implantierbaren Pulsgenerator (IPG) geführt wird, der üblicherweise unter dem Schlüsselbein platziert wird, verbunden [vgl. Benabid, 1995, S. 386–392].

Zur Zeit ist die Firma Medtronic (Minneapolis, MN) der einzige zertifizierte Hersteller für DBS-Hardware in den Vereinigten Staaten und anderen Ländern. Von Medtronic sind 2 Typen von Elektroden verfügbar. Beide Versionen haben 4, jeweils 1,5 mm lange, Kontakte, bei der einen Version sind diese im Abstand von 1,5 mm (Modell 3387; siehe Abbildung 2.6), bei der anderen Version im Abstand von 0,5 mm (Modell 3389) angeordnet. Beim Modell 3389 wird die räumliche Auflösung der Stimulation erhöht, es ergibt sich eine Ausdehnung von 7,5 mm im Gegensatz zu 10 mm beim Modell 3387, was angemessen erscheint, wenn man bedenkt, dass sowohl im VIM, als auch im STN die Länge der Zielregion kleiner als 7,5 mm ist. Andererseits jedoch könnte man das Modell 3387 bevorzugt verwenden, da nämlich nicht mit Sicherheit bekannt ist, ob sich die Zielregion genau im STN befindet und die größere Länge die Möglichkeit einer Stimulierung eines größeren Areals zulässt. Außerdem ergibt sich eine größere Ausdehnung des Stimulationsareals, was bei einer fehlplatzierten Elektrode hilfreich ist. Es gibt aber keine Untersuchungen darüber, welche die eine über die andere Elektrode bevorzugen würden, sodass jeder Anwender selbst entscheiden muss, welche Elektroden er verwenden möchte [vgl. Rezai, 2006, S. 209].



Abbildung 2.6: DBS-Elektrode [Medtronic, 2007]

Die Stimulationsparameter der Stimulatoren (siehe Abbildung 2.7) können mit einem Handgerät über Telemetrie, wenn das Gerät in die Nähe des Stimulators gehalten wird, programmiert werden. Diese sind Pulsbreite, Amplitude, Stimulationsfrequenz und Position der aktiven Elektrode. Der Patient kann den Pulsgenerator mit Hilfe eines Magneten oder eines Handgerätes ein- und ausschalten. Üblicherweise werden Stimulationsfrequenzen von 135 bis 185 Hz, Pulsbreiten von 60–120  $\mu$ s und Amplituden zwischen 1 und 3 V verwendet [vgl. Hauser, 2003, S. 125–126].



Abbildung 2.7: Impulsgenerator mit Elektrode und Kabel [Medtronic, 2007]

Die Aktivierung des Impulsgebers wird am 1. bis 3. Tag nach der Implantierung, abgestimmt auf das Befinden des Patienten, durch den betreuenden Neurologen durchgeführt. Für die Einstellung der Parameter werden – wie bei der OP im Medikamentenzustand OFF – alle vier Pole sequentiell mit einer Frequenz von 130 Hz und einer Pulsbreite von 60  $\mu$ s als Standardeinstellung getestet, die Amplitude wird dabei jeweils um 0,5 V erhöht. Es wird jener Pol für die chronische Stimulation verwendet, bei dem bei der niedrigsten Spannung der beste klinische Effekt und die geringsten Nebenwirkungen auftreten [vgl. ÖPG, 2003].

Wenn während der OP Zweifel über die Wirksamkeit der DBS auftreten, wird der IPG vorerst nicht implantiert, die klinische Testung mit einem externen Stimulationsgerät vorgenommen und die klinischen Ergebnisse und Nebenwirkungen werden genau dokumentiert. Zeigt sich ein positiver klinischer Effekt, wird der IPG in einer zweiten OP-Sitzung implantiert, bei negativem Stimulationsergebnis kann eine Korrektur der Elektrodenposition durch den Neurochirurgen erfolgen. Nach der Operation werden eine Röntgenkontrolle (siehe Abbildung 2.8 und Abbildung 2.9) und die Feststellung der Lokalisation der Stimulationselektrode durchgeführt [vgl. ÖPG, 2003].

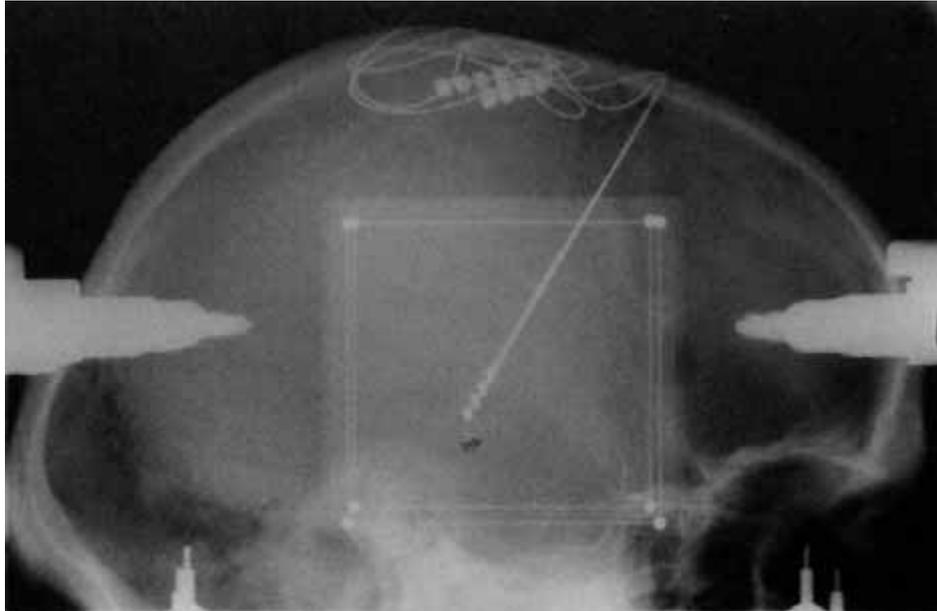


Abbildung 2.8: Röntgenbild mit einer DBS-Elektrode; das Kabel wurde noch nicht verlegt [Benabid1, 1995, S. 389]

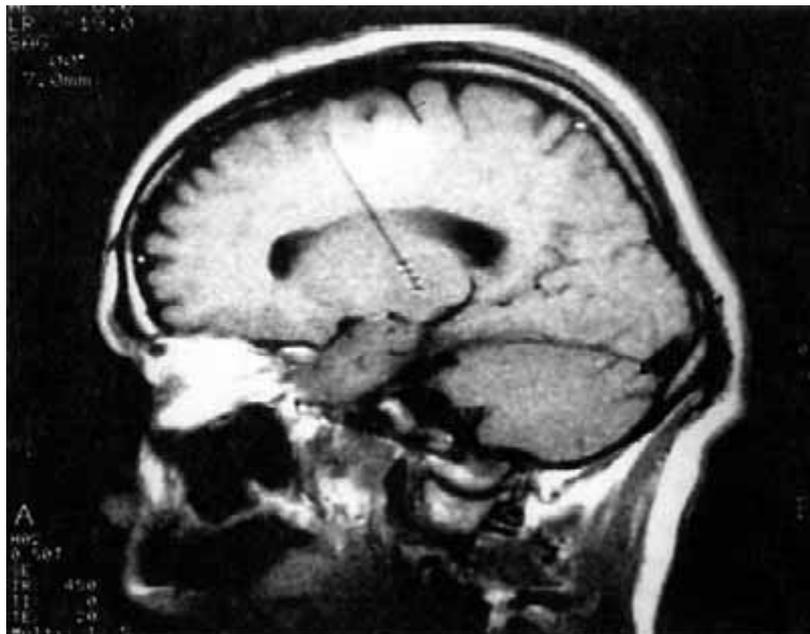


Abbildung 2.9: MRT-Bild mit einer DBS-Elektrode [Benabid1, 1995, S. 391]

#### 2.2.2.4 Risiken der DBS

Risiken der DBS gliedern sich in Risiken aufgrund der Operation, jene die vom Gerät und jene, die von der Stimulation ausgehen können. Operative Risiken treten meist innerhalb von 30 Tagen nach der Operation bei weniger als 5 % der Patienten auf. Als Komplikationen – wie bei anderen neurochirurgischen Operationen – können Hirnblutungen, Schlaganfälle und Infektionen auftreten, sehr selten auch Todesfälle. Weitere postoperative Symptome sind Verwirrung,

Blutungen unter der Haut, Sprechschwierigkeiten, Benommenheit, Schwäche und andere Nervenkrankheiten. Der Großteil dieser Symptome verschwindet innerhalb von 30 Tagen.

Eine schlechte Platzierung der Elektrode kann eine Korrektur der Elektrodenposition mit allen dadurch entstehenden Risiken der erneuten Operation erfordern. Weitere Probleme sind können durch Verschiebung der Elektrode, Erosion der Haut oberhalb des Kabels, Kabelbruch und Fehlfunktion des Systems inklusive des Stimulators entstehen. Nebeneffekte der Stimulation hängen von der exakten Lokalisation der Elektrode ab. Ein Großteil der Nebeneffekte kann behoben werden, wenn man entweder einen anderen Elektrodenkontakt zur Stimulation verwendet oder die Stimulationsintensität verringert. Als Nebeneffekte können Augenlidschluss, Doppelsehen, muskuläre Spasmen, Kribbelgefühl, Benommenheit, Sprechschwierigkeiten, Depressionen, Gemütschwankungen, intensiviertes sexuelles Verlangen, Veränderungen des Sehvermögens, Balanceschwierigkeiten und Schmerzen auftreten. Wenn diese nicht durch eine Änderung der Stimulationsintensität oder des Elektrodenkontaktes behoben werden können, müssen entweder die Elektrodenposition geändert oder die Nebenwirkungen medikamentös behandelt werden [vgl. Hauser, 2003, S. 140–142].

### **2.2.2.5 Nachsorge bei Patienten mit DBS**

In den ersten zwei bis drei Wochen nach der Operation ist in der Regel eine tägliche Untersuchung des Patienten mit Modifizierung der Medikation sowie der Stimulationsparameter erforderlich. Diese sollte stationär erfolgen, danach ist eine ambulante Weiterbetreuung möglich. Drei, 6, 12, 18 und 24 Monate postoperativ sollte der Patient erneut zur Durchführung eines ausführlichen klinischen und apparativen Untersuchungsprogrammes aufgenommen werden. Nach [ÖPG, 2003] sollte der Patient nach einem Monat, nach 3 bis 6 Monaten, nach einem Jahr und anschließend einmal jährlich zur Untersuchung aufgenommen werden.

Patienten mit DBS unterliegen beim Gebrauch von gängigen elektronischen Geräten, wie z. B. Mikrowellenherd, Haarföhn, PC, Küchenmaschinen u.s.w. keinen Einschränkungen. Lediglich der Gebrauch oder die Nähe der folgenden Geräte sollte wegen möglicher Störung des Stimulators eingeschränkt oder gemieden werden: Mobilnetztelefone, Bohrmaschinen, elektrische Bogenschweißgeräte, große Lautsprecherboxen mit Magneten, elektrische Hochöfen, Hochspannungsleitungen, Umspannwerke, Stromgeneratoren und Diebstahldetektoren (z. B. in Kaufhäusern). Bei Sicherheitsprüfsystemen, wie z. B. am Flughafen, sollte der Patient seinen Ausweis vorzeigen und die Geräte möglichst umgehen, um Störungen zu vermeiden. Auch einige diagnostische oder therapeutische Maßnahmen sind entweder nur eingeschränkt möglich, wie zum Beispiel die Kernspintomographie, oder sogar absolut kontraindiziert, wie zum Beispiel die Anwendung von Diathermie [vgl. Gerlach, 2003, S. 216–217].

## 3 UPDRS

### 3.1 Allgemeine Bemerkungen zur UPDRS

Die Unified Parkinson's Disease Rating Scale (1987, siehe Anhang) ist die derzeit am häufigsten verwendete umfangreiche Skala zur Beschreibung der durch die Parkinson'sche Krankheit bedingten Beeinträchtigungen des Patienten. Sie umfasst die Beurteilung der mentalen Situation, die der motorischen Funktionen, der Aktivitäten des täglichen Lebens, sowie die aufgetretenen Komplikationen während der Therapie [vgl. Fuchs, 2002, S. 49–50]. Ergänzend werden oftmals die Skala nach Schwab und England (1969, siehe Anhang) und teilweise auch die Skala nach Hoehn und Yahr (1992, siehe Anhang) eingesetzt.

Die UPDRS ist weder in der Literatur noch im Internet sehr gut und einheitlich dokumentiert. Einerseits sind viele UPDRS-Skalen oft nur unvollständig – teilweise nur die ersten drei statt der vier Teile – angeführt, andererseits wird die Bildung des Score-Wertes oftmals nicht ausreichend erklärt; zum Beispiel wird nicht beschrieben, welche Fragen für die linke und rechte Körperhälfte getrennt beantwortet werden müssen und nicht für beide Seiten gemeinsam, was, wenn man mit der Parkinson'schen Krankheit beziehungsweise der Neurologie nicht sehr vertraut ist, nicht ohne weiteres aus dem Kontext geschlossen werden kann. Weiters sind die Fragen und Antworten verschiedener Quellen teilweise unterschiedlich, es gibt also keinen einheitlichen Standard in Bezug auf die Formulierung der Fragen und Antworten. Deswegen erscheint es ratsam, sich an die originale Quelle zu halten, um eine möglichst hohe Standardisierung und Vergleichbarkeit gewährleisten zu können. In den nächsten Monaten ist die Veröffentlichung einer revidierten Version der UPDRS in Form der MDS-UPDRS geplant, welche sich gerade in einer Testphase befindet [Goetz, 2007], auch diese soll in FuzzyDBSexpert möglichst einfach Eingang finden können.

UPDRS-Skalen in deutscher Übersetzung sind in der Literatur sehr selten zu finden und werden meist für Anwendungsbeobachtungen oder randomisierte Medikamentenstudien verwendet. In [Gerlach, 2003, S. 171–172] ist nur der dritte der vier Teile komplett übersetzt, in [Fuchs, 2002, S. 119–120] ist eine Zusammenfassung der Skala, allerdings ohne Antworten, zu finden. Auch diese Unzulänglichkeit soll mit der neuen MDS-UPDRS [Goetz, 2007] behoben werden, da hier eine Testphase mit standardisierter Übersetzung und Rückübersetzung geplant ist.

Im Internet findet man unter [MünchenIsar, 2007] eine komplette Übersetzung der originalen UPDRS-Skala. Die originale englische UPDRS-Skala [Fahn, 1987] kann man auf der Seite der MD Virtual University [MDVU, 2007], welche von der WE MOVE-Organisation, einer Non-Profit-Organisation zur Wissenssammlung und -vermittlung zu Krankheiten mit Bewegungsstörungen, geschaffen wurde, als .PDF-Datei herunterladen. Weiters ist dort das originale Datenformular

verfügbar, welches ebenfalls sehr weit verbreitet ist und zusammen mit der originalen UPDRS-Skala die Punktevergabe verständlicher macht.

## 3.2 Beschreibung der UPDRS

Die Unified Parkinson's Disease Rating Scale setzt sich aus sechs Teilen zusammen, wobei die letzten beiden Teile für sich selbst Skalen sind und auch eigenständig verwendet werden können:

I. Kognitive Funktionen, Verhalten, Stimmung	4 Fragen
II. Aktivitäten des täglichen Lebens	13 Fragen für ON/OFF getrennt
III. Untersuchung der Motorik	18 Fragen (für versch. Extremitäten), insg. 27 Fragen
IV. Komplikationen während der Therapie (in der vergangenen Woche)	11 Fragen
V. Modifizierte Stadienbestimmung nach Hoehn und Yahr	
VI. Modifizierte Schwab- und England-Skala der Aktivitäten des täglichen Lebens	

Die ersten vier Teile umfassen insgesamt 42 verschiedene Fragen, welche teilweise für mehrere Extremitäten beantwortet werden müssen und so in Summe 55 Fragen ergeben.

### – I. Kognitive Funktionen, Verhalten, Stimmung

Der erste Teil der UPDRS befasst sich mit den geistigen Beeinträchtigungen des Patienten, welche von der Krankheit oder der Therapie hervorgerufen werden können [vgl. Lang1, 1995, S. 27]. Zu den 4 Fragen gibt es jeweils 5 Antwortmöglichkeiten mit den Punkten 0–4, die maximale Punktezahl beträgt 16.

### – II. Aktivitäten des täglichen Lebens

Der zweite Teil beurteilt, wie sehr der Patient aufgrund seiner Krankheit in den Aktivitäten des täglichen Lebens beeinträchtigt ist. Jeder Punkt besteht aus wohlformulierten Fragen und Antwortmöglichkeiten, um den Einfluss der Variabilität zwischen den Untersuchenden zu verringern [vgl. Lang1, 1995, S. 27]. Die insgesamt 13 Fragen müssen getrennt für den ON- und den OFF-Zustand des Patienten beurteilt werden. Der beste ON-Zustand bezeichnet jenen Zustand des Patienten, in dem die medikamentöse Therapie, hauptsächlich mit L-Dopa, die größte Unterdrückung der Parkinsonsymptome bewirkt; der Patient ist beweglicher und in den ADL (activities of daily living, Aktivitäten des täglichen Lebens) selbstständiger. Zu den 13 Fragen gibt es jeweils 5 Antwortmöglichkeiten mit den Punkten 0–4, die maximale Punktezahl beträgt für den ON- und für den OFF-Zustand des Patienten jeweils 52.

– III. Untersuchung der Motorik

Teil drei der UPDRS evaluiert die motorischen Funktionen des Patienten und deren Beeinträchtigung durch die Parkinson'sche Krankheit, anhand von Fragen in Bezug auf Sprache, Gesichtsausdruck, Tremor, Rigidität und Agilität. Als Basis für diese Skala diente die Columbia Scale, von der weniger wichtige Teile gestrichen und zu den wichtigeren Teilen hinzugefügt wurden. Bei der Erhebung ist es wichtig, die Zeit der letzten Einnahme des Medikamentes zu notieren. Obwohl in der originalen Skala nicht vorgesehen, kann eine getrennte Beurteilung im ON- und OFF-Zustand des Patienten die Bewertung erweitern [vgl. Lang1, 1995, S. 27]. Die 18 verschiedenen Fragen werden entweder nur einmal (7 Fragen), für die linke und rechte Körperhälfte getrennt (5\*2 Fragen) beziehungsweise für Gesicht oder Nacken und alle vier Extremitäten (2\*5 Fragen), erhoben. So ergeben sich in Summe 27 Fragen, zu denen es jeweils 5 Antwortmöglichkeiten mit den Punkten 0–4 gibt. Die maximale Punktezahl beträgt 108.

– IV. Komplikationen während der Therapie (in der vergangenen Woche)

Der vierte Teil der UPDRS bewertet die Komplikationen der Therapie, wie Dyskinesien, Fluktuationen, Schlafstörungen und Schwindel. Dyskinesien werden hinsichtlich ihrer Dauer über den Tag, der resultierenden Stärke der Beeinträchtigung des Patienten und der Schmerzhaftigkeit bewertet. Bei 8 von 11 Fragen gibt es 5 Antwortmöglichkeiten mit den Punkten 0–4, die restlichen 3 müssen mit „ja“ (1 Punkt) oder „nein“ (0 Punkte) beantwortet werden. Die maximale Punktezahl beträgt 23.

○ Blutdruck, Puls

Für eine vollständige Bewertung müssen der Blutdruck des Patienten in Rückenlage, im Sitzen und im Stehen, sowie die Herzfrequenz im Sitzen und im Stehen gemessen werden.

– V. Modifizierte Stadienbestimmung nach Hoehn und Yahr

Das modifizierte Stadium nach Hoehn und Yahr ist ein sehr häufig gebrauchter Score-Wert. Er gibt auf sehr schnelle und unkomplizierte Weise Auskunft über die Stärke der Beeinträchtigung des Patienten durch die Parkinson'sche Krankheit. Die Skala ist im Anhang in deutscher Übersetzung zu finden und einfach zu interpretieren. Sie besitzt 8 verschiedene Abstufungen der Stärke der Beeinträchtigung des Patienten.

– VI. Modifizierte Schwab- und England-Skala der Aktivitäten des täglichen Lebens

Die modifizierte Schwab- und England-Skala der Aktivitäten des täglichen Lebens gibt ebenfalls sehr rasch Auskunft über die Stärke der Beeinträchtigung des Patienten in einer

Abstufung von 11 Stufen. Auch diese Skala ist im Anhang in deutscher Übersetzung zu finden und sehr einfach zu interpretieren, wobei das englische Original [vgl. Lang1, 1995, S. 40] oftmals verständlicher ist, als die deutsche Übersetzung.

Bei einer genaueren Analyse der Punktevergabe fällt auf, dass diese in manchen Fällen offensichtlich unausgewogen ist. Exemplarisch sollen hier zwei Fragen angeführt werden (Fragen entnommen aus [MünchenIsar, 2007]):

19. Gesichtsausdruck:

- 0 - Normal.
- 1 - Minimal veränderte Mimik, könnte ein normales "Pokergesicht" sein.
- 2 - Leichte, aber eindeutig abnorme Verminderung des Gesichtsausdruckes.
- 3 - Mäßig verminderte Mimik; Lippen zeitweise geöffnet.
- 4 - Maskenhaftes oder erstarrtes Gesicht mit stark oder völlig fehlendem Ausdruck; Lippen stehen um 7 mm auseinander.

41. Leidet der Patient an Schlafstörungen, z. B. Schlaflosigkeit oder Schläfrigkeit?

- 0 - Nein.
- 1 - Ja.

Hier wird deutlich, dass eine Schlaflosigkeit des Patienten, zum Beispiel wegen der starken Symptome oder wegen vorhandener Depressionen, genauso schwer gewertet wird, wie eine minimal veränderte Mimik; dies ist sicher nicht optimal. Es ist daher sinnvoll, zusätzlich Gewichtungen für die einzelnen Fragen einzuführen, um einer optimalen Entscheidungsfindung für oder gegen eine Tiefe Hirnstimulation näher kommen zu können. Solch eine Gewichtung wird den Experten in FuzzyDBSexpert ermöglicht.

## 4 Systemanforderungen

In diesem Kapitel werden verschiedene Arten von Anforderungen und Bedingungen diskutiert, welche ein System erfüllen muss, damit es erfolgreich in der Medizin eingesetzt werden kann.

### 4.1 Prinzipielle Anforderungen

Folgende Grundanforderungen werden von den medizinischen Experten an ein System, welches erfolgreich für die Entscheidungsunterstützung in Hinblick auf die Tiefe Hirnstimulation eingesetzt werden kann, gestellt:

- Entscheidung, ob die Therapie für einen idiopathischen Parkinson-Patienten in Form einer Operation für die Tiefe Hirnstimulation bei einem Patienten durchgeführt werden soll oder nicht, auf der Basis von Kriterien, die von medizinischen Experten als relevant betrachtet werden.
- Entscheidung über den Implantationsort aufgrund von Kriterien, die von den medizinischen Experten vorgegeben werden.

Eine Entscheidung ist hier nicht im Sinne einer ärztlichen Entscheidung gemeint, sondern als Ergebnis einer Anwendung von Expertenwissen auf die jeweiligen Patientendaten zur Entscheidungsunterstützung für den behandelnden Arzt.

Diese Aufgaben geben die Problemstellung vor, die mit Hilfe von FuzzyDBSexpert gelöst werden soll. Zur Erreichung dieser Ziele und damit dieses System auch in der Praxis benutzt wird, sind eine ganze Reihe weiterer Anforderungen an ein solches System zu stellen. Es hat keinen großen Sinn, ein System zu entwerfen und zu entwickeln, wenn es nachher nicht von den Experten und Anwendern akzeptiert und verwendet wird. Daraus kann man höchstens lernen, wie man es nicht machen sollte beziehungsweise wo vielleicht noch Entwicklungs- und Überzeugungsarbeit zu leisten wäre.

### 4.2 Anforderungen an Expertensysteme im klinischen Einsatz

Die Anforderungen für den Einsatz eines Systems in der Klinik scheinen recht hoch zu sein und jede dieser Anforderungen muss entsprechende Mindestanforderungen übersteigen, damit ein solches System zum Einsatz kommen kann. Schließlich soll das wissensbasierte System nicht nur zum Einsatz in Wissenschaft und Forschung kommen, wo beispielsweise mangels alternativer Lösungen über ein etwas umständliches Benutzerinterface hinweg gesehen wird, wenn damit nur

die Ziele entsprechend erreicht werden können. Es geht vielmehr darum, Benutzer im klinischen Bereich mit Hilfe von Expertenwissen in ihren Entscheidungen zu unterstützen.

## 4.2.1 Elementare Bedingungen

Das System muss folgende elementare Bedingungen erfüllen:

- das System selbst betreffend
  - Die Funktionen des Systems müssen inhaltlich korrekt sein.
- die Wissensakquisition betreffend
  - Es muss eine Analyse des Fachgebietes vorausgehen, welche transparent und nachvollziehbar ist, damit Vertrauen bei den potentiellen Anwendern geschaffen wird, was die Glaubwürdigkeit der Ergebnisse anbelangt [vgl. Schnabel, 1996, S. 195–196]. Dieser Prozess findet auch Eingang in diese Arbeit, um hier dieses Vertrauen schaffen zu können.
  - Meines Erachtens nach sollte das System mit einer grundlegenden Wissensbasis ausgestattet sein, welche auf anerkanntem Wissen beruht, von der man ausgehen kann und die dann aber flexibel veränder- und erweiterbar ist.

## 4.2.2 Notwendige Bedingungen

### 4.2.2.1 Benutzeroberfläche/Datenerfassung

Solange eine manuelle Eingabe von Daten in größerem Umfang nicht komfortabel genug ist, wird sich ein solches System im klinischen Betrieb nicht durchsetzen [vgl. Schnabel, 1996, S. 197].

- Es ist also eine Forderung an FuzzyDBSexpert, die Eingabemasken möglichst übersichtlich und benutzerfreundlich zu gestalten. Weiters sollte ein mehrfacher Nutzen aus der Eingabe der Daten gezogen werden können. Nachdem die UPDRS in FuzzyDBSexpert Verwendung findet, wird als zusätzlicher Nutzen die dazugehörige Tabelle samt „herkömmlichem“ Gesamtscore-Wert zum Kopieren zur Verfügung gestellt. Dieser Wert muss ohnehin bei Parkinson-Patienten erhoben werden, um eine genaue, qualitativ hochwertige und international standardisierte Dokumentation des Zustandes des Patienten zu ermöglichen und komplettiert so die eigene Dokumentation.
- Ein weiterer Schritt wäre die Möglichkeit des Importes der UPDRS-Tabelle aus anderen Programmen, damit die aktuellen Werte, so diese schon in einem anderen Computersystem

erhoben wurden, nicht nochmals eingegeben werden müssen. Nachdem mir kein standardisiertes Format für die Übertragung von UPDRS-Daten in Computersystemen bekannt ist, könnte man mit einer solchen Importfunktion leider nur wenige Anwender unterstützen und müsste viele maßgeschneiderte Module programmieren.

- Eine Überlegung wäre eine Exportfunktion für die UPDRS-Tabelle wert. Diese Daten könnten für statistische Analysen dienen, zum Beispiel in welchen Bereichen sich die Patienten mit einer Tiefen Hirnstimulation und in welchem Ausmaß sich welche Symptome verbessert haben, getrennt nach Stimulationsort. Die Daten, welche auf Papier erhoben wurden, sind für solche Auswertungen nutzlos, sie müssten mühevoll wieder in ein Computersystem eingetippt werden, wohingegen exportierte Werte sofort mit Hilfe einer Tabellenkalkulation oder eines Statistikpaketes visualisier- und analysierbar wären. Als standardisiertes Format würde sich hierzu das CSV-Format (comma separated values) eignen, da dies von allen ernstzunehmenden Tabellenkalkulations- und Statistikprogrammen verarbeitbar ist.

#### **4.2.2.2 Erklärung/Rechtfertigung**

Wenn das System keine Erklärung seines Vorgehens auf ärztlichem Niveau abgeben kann, wird ein solches System auch von keinem Arzt akzeptiert [vgl. Schnabel, 1996, S. 196].

- Hier gilt es, die zugrunde liegende Wissensbasis ausreichend zu recherchieren und mit Hilfe medizinischer Experten abzusichern. Optimalerweise könnten fundierte statistische Auswertungen gemacht werden, die die Akzeptanz der Wissensbasis untermauern würden.
- Weiters ist eine Möglichkeit vorgesehen, die Wissensbasis so ändern zu können, dass sie von Personen mit teilweise nicht übereinstimmenden oder sogar widersprüchlichen fachlichen Meinungen so geändert werden kann, damit sie diesen Ansprüchen gerecht wird. Nachdem es am medizinischen Sektor, nicht nur im ärztlichen Bereich, vorkommt, dass verschiedene Lehrmeinungen und „Schulen“ vertreten werden, wäre es ineffektiv, ein System zu gestalten, welches von den Personen einer Meinung verwendet wird und von allen anderen Personen mit einer abweichenden Meinung nicht, nur weil das wissensbasierte System nicht flexibel genug ist, einen Großteil der vorherrschenden Meinungen vertreten zu können.

#### **4.2.2.3 Flexibilität**

Bei wissensbasierten Systemen, zum Beispiel in Form von Diagnostiksystemen, hat sich gezeigt, dass diese meist nicht flexibel genug sind, zum Beispiel gegenüber der Änderung von aktuell betrachteten Hypothesen [vgl. Schnabel, 1996, S. 197]. Diese könnten nach Änderung der Hypothese nicht mehr zum Einsatz kommen.

Wie oben schon angesprochen, lässt sich die Wissensbasis des wissensbasierten Systems so flexibel ändern, dass aktuelles Wissen beziehungsweise verschiedene Meinungen in dem System berücksichtigt werden können.

#### **4.2.2.4 Arbeitserleichterung**

Die Funktionen des Systems müssen dem Arzt helfen, Zeit zu sparen und die Qualität der Arbeit zu verbessern [vgl. Schnabel, 1996, S. 196].

- Bei vielen Computersystemen für die Medizin ist es häufig so, dass statt einer Zeitersparnis ein Mehraufwand an Zeit zum Beispiel für die Eingabe von Daten entsteht. Im Gegenzug dazu kann aber eine Erhöhung der Qualität der Arbeit stattfinden, welche entsprechend honoriert werden sollte. Manche Personen mit fehlendem Verständnis für einen höheren Aufwand für die Dateneingabe reagieren daraufhin zum Beispiel mit der Weigerung, alle Daten ordnungsgemäß einzugeben oder einer gewissen Nachlässigkeit bei der Genauigkeit der Eingabe der Daten. Im FuzzyDBSexpert wird versucht, diesen Eingabeaufwand möglichst gering zu halten und zum Beispiel – wie vorhin schon beschrieben – eine Kopie der UPDRS-Tabelle zu ermöglichen, um einen Mehrwert durch die Eingabe der Daten in das System zu erhalten und damit einen zusätzlichen Anstoß zu geben, das System ordnungsgemäß zu verwenden.

Wenn nicht alle notwendigen Bedingungen erfüllt sind, wird das System wahrscheinlich nicht im klinischen Betrieb einzuführen sein. Sind diese jedoch erfüllt, so ist das noch keine Garantie dafür, dass dieses System auch erfolgreich eingesetzt werden kann [vgl. Schnabel, 1996, S. 196].

### **4.2.3 Hinreichende Bedingungen**

#### **4.2.3.1 Organisatorische Einbettung in die Klinik- bzw. Praxisroutine**

Nachdem FuzzyDBSexpert ein sehr spezielles, auf einen sehr schmalen Bereich der Neurologie beschränktes, wissensbasiertes System ist, wird es voraussichtlich nicht laufend im klinischen Betrieb benötigt. Der langfristige, breite Einsatz des Systems wird zum Großteil davon abhängen, ob es von medizinischen Experten selbst verwendet und akzeptiert wird.

#### **4.2.3.2 Validierung des Systems**

Ein meiner Meinung nach äußerst wichtiger Punkt besteht in der Validierung der Wissensbasis. Hierzu könnte man die Daten von aktuellen Patienten mit einer bevorstehenden Entscheidung darüber, ob eine Operation für eine Tiefe Hirnstimulation vorgenommen werden soll oder nicht, in die Wissensbasis eintragen und im Nachhinein die Daten daraufhin analysieren, zu welcher Entscheidung das FuzzyDBSexpert und zu welcher Entscheidung die medizinischen Experten

gelangt sind. Mit Hilfe dieser Analysen kann die Wissensbasis auf Fehler hin untersucht und fein eingestellt werden, was auch immer im laufenden Betrieb für alle Zukunft möglich sein sollte. Auf diesem Weg kann man langsam, aber stetig zu einer „optimalen“ Einstellung der Wissensbasis gelangen. Natürlich kann es sich dabei um einen äußerst aufwändigen und langwierigen Prozess handeln, der zusätzlich wegen der notwendigen statistischen Auswertungen den Rahmen dieser Arbeit sprengen würde und zum Beispiel im Rahmen von wissenschaftlichen Publikationen fortgeführt werden könnte.

#### **4.2.3.3 Pflege der Wissensbasis (Erweiterung/Änderung)**

Aufgrund der Beschränktheit auf ein spezielles medizinisches Gebiet muss FuzzyDBSexpert wahrscheinlich nicht so flexibel sein, wie ein System, welches einen größeren Bereich in der Medizin abdeckt, in dem sich laufend zusätzliche medizinische Erkenntnisse ergeben können. Dennoch macht FuzzyDBSexpert es möglich, auch zukünftiges Wissen über die Expertenschnittstelle in die Wissensbasis einzubringen.

Beim Vorliegen der hinreichenden Bedingungen kann das System als ein kommerzielles System bezeichnet werden [vgl. Schnabel, 1996, S. 197].

## 5 Wissensbasierte Systeme

In diesem Kapitel wird der prinzipielle Aufbau eines wissensbasierten Systems gezeigt.

### 5.1 Aufbau eines wissensbasierten Systems

FuzzyDBSexpert soll den wissenschaftlichen Ansprüchen an ein wissensbasiertes System genügen. Der Aufbau eines prototypischen vollständigen Systems ist dazu in Abbildung 5.1 dargestellt, anhand derer die Komponenten in den folgenden Kapiteln beschrieben werden.

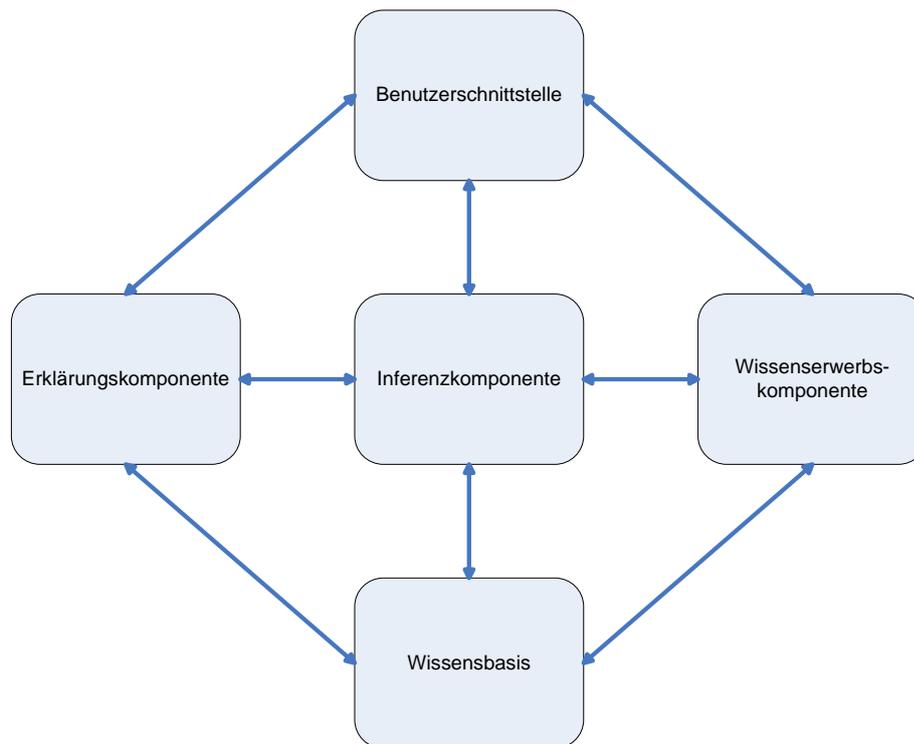


Abbildung 5.1: Komponenten eines wissensbasierten Systems

#### 5.1.1 Wissensbasis

In der Wissensbasis wird das Wissen über die zu lösende Aufgabe gespeichert. Dieses Wissen umfasst sowohl Fakten, als auch Regeln. Fakten sind elementare Sachverhaltsdarstellungen, die den Zustand eines Systems beschreiben. Zum Beispiel stellen die möglichen Stimulationsorte für die Tiefe Hirnstimulation Fakten dar. Regeln beschreiben die Zusammenhänge zwischen Fakten, welche vor allem von allgemein gültiger Natur sind. Die Regel „WENN Tremor DANN VIM-Stimulation“ stellt das Wissen dar, dass bei dominantem Tremor eine Stimulation des VIM indiziert ist.

Das in der Wissensbasis enthaltene Wissen kann man in generisches und fallspezifisches Wissen unterteilen. Das generische Wissen ist von dem spezifischen Fall unabhängig und umfasst allgemeines Hintergrundwissen über den Problembereich. Das fallspezifische Wissen wird zur Bearbeitung eines konkreten Problems verwendet, zum Beispiel UPDRS-Werte des gerade untersuchten Patienten.

### **5.1.2 Inferenzkomponente**

Die Inferenzkomponente verarbeitet das in der Wissensbasis vorhandene Wissen. Dabei wird neues Wissen aus den Fakten und Regeln abgeleitet, zum Beispiel, dass der Patient für die OP kontraindiziert ist. Der Inferenzvorgang ist im Allgemeinen sehr komplex und erfordert ein methodisches Vorgehen. Dazu wird üblicherweise Wissen über die Problemstellung und die Problemlösungsfähigkeit bzw. den Vorgang des Inferenzverfahrens selbst herangezogen. Dieses Wissen wird als Meta-Wissen bezeichnet (Wissen über Wissen).

### **5.1.3 Wissensakquisitionskomponente**

Die Wissensakquisitionskomponente dient der Erweiterbarkeit des wissensbasierten Systems. Diese Änderungen können entweder automatisch oder manuell erfolgen. Im Fall von FuzzyDBSexpert erfolgt die Änderung der Wissensbasis manuell. Das hinzuzufügende und das gesamte resultierende Wissen müssen von dem bearbeitenden Wissensingenieur einer Konsistenzprüfung unterzogen werden.

### **5.1.4 Benutzerschnittstelle**

Die Benutzerschnittstelle ist die Schnittstelle des wissensbasierten Systems nach außen hin. Über diese kommunizieren die Benutzer, aber auch die Wissensingenieure beziehungsweise Experten, mit dem System. Diese ist eine nicht zu unterschätzende Komponente, da von der Benutzerschnittstelle die Akzeptanz gegenüber dem System stark beeinflusst wird.

### **5.1.5 Erklärungskomponente**

Über die Erklärungskomponente erhält der Benutzer Auskunft über die Lösungsfindung für das konkrete Problem. Im Fall von FuzzyDBSexpert ist diese Komponente weniger umfangreich, da keine komplexen Probleme gelöst werden müssen und eine Darstellung der verwendeten Regeln ausreichend ist.

## 5.2 Arten von Wissen

In einer Wissensbasis kann unterschiedlichstes Wissen gespeichert werden, deswegen sollen im Folgenden die verschiedenen „Arten“ von Wissen näher erläutert werden.

### 5.2.1 Faktenwissen

Unter Fakten versteht man die Beschreibung von elementaren Sachverhalten, die in einem System auftreten. Fakten können zum Beispiel die Beschreibung von Eigenschaften eines Objektes darstellen. Auch Ereignisse und Beziehungen zwischen Objekten können durch Fakten dargestellt werden (zum Beispiel, dass sich die Substantia nigra im Gehirn befindet).

### 5.2.2 Zusammenhänge (Beziehungen)

Fakten können untereinander in Beziehung stehen. Diese können zum Beispiel in Form von Regeln, wie „das Pyramidenbahnsystem kann Bewegungen entweder fördern oder hemmen“, statische Zusammenhänge beschreiben. Es gibt auch noch weitere Zusammenhänge, welche durch Regeln beschrieben werden können. Zu den wichtigsten zählen zeitliche Zusammenhänge, welche die Entwicklung von Eigenschaften, wie zum Beispiel „im Verlauf der Parkinson-Krankheit entwickelt nahezu jeder Patient einen Tremor“, und den Ablauf bestimmter Ereignisse, wie zum Beispiel „nach dem Einschalten der Stimulation wird der Tremor deutlich gebessert“, beschreiben. Weiters sind kausale Zusammenhänge zu nennen, welche Ursache-Wirkung-Beziehungen erfassen, die vor allem bei der Suche nach Erklärungen für Phänomene von Bedeutung sind, wie zum Beispiel, dass eine Degenerierung von Dopamin-Neuronen die Ursache für Morbus Parkinson ist.

### 5.2.3 Methodisches Wissen

Fakten und Zusammenhänge beschreiben die Welt des Problembereiches, beinhalten aber kein Wissen darüber, wie konkrete Problemstellungen gelöst werden sollen. Zur Lösung einer konkreten Problemstellung ist methodisches Wissen notwendig, welches Auskunft darüber gibt, wie zur Problemlösung vorgefahren werden kann. Methodisches Wissen kann zum Beispiel in Zerlegungsstrategien oder aber auch in Form einer „prozedural“ festgelegten Abfolge von Schritten vorliegen.

### 5.2.4 Meta-Wissen

Unter Meta-Wissen versteht man Wissen über die Wissensbasis selbst, den Aufbau, die Struktur und die Anwendbarkeit des Wissens. Dieses Wissen wird benötigt, damit das System in der Lage ist, die Möglichkeiten im Umgang mit den gestellten Problemen zu erkennen. Im einfachsten Fall

kann zum Beispiel entschieden werden, ob ein bestimmtes Problem gelöst werden kann oder nicht. Meta-Wissen wird dazu verwendet, die Auswahl von Daten und Strategien zur konkreten Problemlösung zu steuern. Den Vorgang, in dem das System Meta-Wissen verwendet, um die Wissensverarbeitung zu steuern, nennt man Meta-Reasoning.

## 5.3 Wissensrepräsentation

Derzeit ist noch nicht bekannt, warum manche Repräsentationsformen für gewisse Aufgaben geeignet sind und andere nicht. Es sind schon verschiedenste Arten der Wissensrepräsentation erfolgreich in prototypischen Systemen eingesetzt worden.

Unter einer Wissensrepräsentationsform versteht man die Beschreibung von Wissen auf einer syntaktischen und formalen Ebene. Eng verbunden mit der Repräsentation von Wissen durch geeignete Datenstrukturen sind interpretative Prozeduren, die eine Manipulation dieser Datenstrukturen zulassen, um Inferenzen (Schlussfolgerungen) durchzuführen [vgl. Schnabel, 1996, S. 86].

Für die Speicherung von Wissen sind einige verschiedene Ansätze möglich und in Verwendung. Die wichtigsten kann man zur Übersicht unterteilen in

- prozedurale Methoden,
- objektorientierte Methoden und
- logikbasierte Methoden.

Diese Methoden stellen formale Mittel beziehungsweise Sprachen zur Verfügung, in denen Wissen repräsentiert werden kann.

### 5.3.1 Prozedurale Methoden

Beim Ansatz der prozeduralen Methoden wird Wissen in Form von Prozeduren dargestellt. Im Unterschied zu herkömmlichen Programmen unterliegt der Aufruf jedoch der expliziten Kontrolle der Inferenzkomponente. Die Inferenzkomponente ist mit Wissen über die Problemlösungsmöglichkeiten ausgestattet und verwendet dieses, um für die Lösung eines auftretenden Problems eine geeignete Prozedur zu bestimmen und diese anzuwenden. Das Hauptproblem dieses Ansatzes in Hinblick auf FuzzyDBSexpert ist die mangelnde Modularität. Bei der Erweiterung des Wissens muss man explizit in die bestehende Repräsentation eingreifen und diese verändern. Es ist nicht möglich, dieses neue Wissen einfach in der Wissensbasis abzulegen

[vgl. Egly, 2002, S. 3.7–3.9]. Abgesehen von dem vorhin genannten Grund könnte es durchaus möglich sein, FuzzyDBSexpert mit Hilfe von prozeduralen Methoden zu realisieren.

## 5.3.2 Objektorientierte Methoden

Bei diesen Methoden werden die relevanten Objekte, deren Eigenschaften und Beziehungen untereinander ermittelt. Bei objektorientierten Methoden geht man davon aus, dass die Struktur der Objekte bei der Wissensrepräsentation, im Gegensatz zu den prozeduralen und logikbasierten Methoden, beibehalten wird [vgl. Egly, 2002, S. 3.32–3.34].

### 5.3.2.1 Frames

Framesysteme basieren auf einem objektorientierten Ansatz zur Wissensrepräsentation, welcher 1975 von Marvin Minsky vorgestellt worden ist. Die Kernidee des Ansatzes besteht darin, wiederkehrende Situationen durch ein Gerüst beziehungsweise einen Rahmen (Frame) zu beschreiben, welcher die üblichen möglichen Eigenschaften der Situation beinhaltet und so eine abstrakte, generische Beschreibung darstellt, die ein Prototyp für eine Situation ist. Eine „Situation“ ist als ein Objekt zu verstehen, welches nicht notwendigerweise ein materielles Objekt sein muss. Es kann auch ein ideelles Objekt, wie zum Beispiel ein Vorgang, sein. Durch Festlegen der konkreten Eigenschaften wird aus einer wiederkehrenden Situation eine spezielle Situation, die eine Instanz des generischen Konzeptes ist. Dieser spezielle Frame ist gleich strukturiert wie der generische Frame, enthält aber individuelle statt generische Werte. Der generische Frame bzw. Prototyp entspricht einem Konzept der realen oder vorstellbaren Welt, seine Instanzen entsprechen Objekten oder Sachverhalten der realen Welt. Diese objektorientierten Konzepte werden mittels eines Vererbungsmechanismus erweitert [vgl. Egly, 2002, S. 3.38–3.39].

### 5.3.2.2 Semantische Netze

Eine weitere Familie von Formalismen, welche objektorientierte Methoden darstellen, sind semantische Netze. In einem semantischen Netz wird Wissen bildlich gesehen in Form eines gerichteten Graphen gespeichert. Die Knoten des Graphen werden als Konzepte bezeichnet und stellen sowohl Objekte der realen Welt, als auch Verfahren und andere Dinge dar. Beziehungen zwischen den Konzepten werden durch Kanten in dem Graphen dargestellt, welche als konzeptuelle Relationen bezeichnet werden. Die Konzepte von semantischen Netzen sind weit älter als die der Frames [vgl. Egly, 2002, S. 3.51–3.52].

## 5.3.3 Logikbasierte Methoden

Viele Ansätze zur Wissensrepräsentation verwenden Methoden der formalen Logik. Logik dient dazu, aus Formulierung von bestimmten Sachverhalten durch Aussagen beziehungsweise Sätze in

einem logischen System auf das Zutreffen weiterer Sachverhalte, die inhärent wahr oder falsch sind, systematisch, also nach bestimmten Vorschriften, zu schließen. Dadurch versucht man, den Vorgang des „logischen“ Schließens, wie er bei der Lösung von z. B. mathematischen oder logistischen Problemen auftritt, auf adäquate Weise zu beschreiben und zu untersuchen. Dabei muss man zwischen zwei Komponenten unterscheiden:

– Syntax

Die zu untersuchenden Sätze müssen in einer formalen Sprache<sup>1</sup> liegen, die festlegt, welche Ausdrücke (Formeln) zulässige Sätze sind.

– Semantik

Die Semantik legt die Bedeutung der Sätze fest, das bedeutet, was letztlich durch einen Satz beziehungsweise eine Menge von Sätzen ausgedrückt wird. Dies erfolgt üblicherweise durch Konzepte, wie Interpretationen und Modelle.

Mit Hilfe der Semantik können wahre Sätze von solchen, die nicht wahr sind, unterschieden werden. Weiters können logische Folgerungsbeziehungen zwischen den Sätzen betrachtet werden, vor allem von der Form:

Wenn die Sätze  $\varphi_1, \dots, \varphi_n$  gelten, dann gilt auch der Satz  $\psi$ .

Um solche Folgerungsbeziehungen nachvollziehen zu können, werden logische Inferenzsysteme (Kalküle) verwendet, in denen durch syntaktische Manipulation von Sätzen unter Anwendung von Axiomen und Regeln ein „Beweis“ für einen folgerbaren Satz konstruiert wird. Man kann damit eine Verarbeitung semantischer Inhalte vornehmen, ohne vorher über diese Inhalte selbst Bescheid zu wissen. Die verbreitetsten Formen der Logik sind die Aussagenlogik und die Prädikatenlogik, ein Superformalismus der Aussagenlogik [vgl. Egly, 2002, S. 3.9–3.11].

### 5.3.3.1 Aussagenlogik

Die Aussagenlogik ist eine elementare Form der Logik und für die Praxis von großer Bedeutung, da viele Problemstellungen auch in ausdrucksstärkeren Formalismen, wie der Prädikatenlogik, unter bestimmten Voraussetzungen auf die Aussagenlogik zurückgeführt werden können. Gegenstand

---

<sup>1</sup> Als formale Sprache wird jede beliebige – im Allgemeinen unendliche – Menge von Zeichenketten bezeichnet.

der Aussagenlogik sind in sich geschlossene Aussagen, die mit wahr oder falsch bewertet werden können, und deren logische Beziehung zueinander.

Beispiele solcher geschlossenen Aussagen sind:

- A1: Tremor ist vorhanden.
- A2: Rigor ist vorhanden.
- A3: Bradykinesie ist vorhanden.
- A4: Posturale Instabilität ist vorhanden.
- A5: Parkinson ist eine neurologische Erkrankung.
- A6: Posturale Instabilität ist nicht vorhanden.
- A7: Morbus Parkinson liegt vor.

Diese Aussagen werden durch logische Verknüpfungen zu komplexeren Aussagen verbunden, zum Beispiel:

- C1: Wenn Tremor vorhanden ist und Rigor vorhanden ist und Bradykinesie vorhanden ist, dann liegt Morbus Parkinson vor.

Durch iterierte Anwendung können geschachtelte Aussagen gebildet werden. In der Aussagenlogik wird über die Inhalte der elementaren Aussagen abstrahiert und die Aussagen werden auf ihren Wahrheitswert reduziert, das heißt, man betrachtet in obigem Fall abstrakt die Aussagen A1, A2 und A3. Bei komplexen Aussagen ist nur die logische Struktur von Interesse, die durch die Anwendung der Verknüpfungen (WENN... DANN..., UND und ODER) auf elementare Aussagen gegeben ist. Die Verknüpfungen werden durch Symbole dargestellt, komplexe Aussagen werden durch Formeln beschrieben. Im obigen Beispiel erhält man für die Regel C1:  $A1 \wedge A2 \wedge A3 \rightarrow A7$ .

Der Vorteil der Aussagenlogik besteht darin, dass die Gültigkeit eines beliebigen gegebenen Satzes in endlicher Zeit durch automatische Berechnung entschieden werden kann, sie ist also entscheidbar (Alonzo Church<sup>1</sup>) [vgl. Egly, 2002, S. 3.11–3.12].

### 5.3.3.2 Syntax

Der Syntax der Aussagenlogik soll hier, nur soweit für das Verständnis des wissensbasierten Systems notwendig, angerissen werden.

Ausgangspunkt für die Bildung von aussagenlogischen Formeln ist eine Menge  $A$  von elementaren Aussagen (Atome). Die Menge der Aussagen über  $A$  ist die kleinste Menge von Formeln  $F$ , die folgende Punkte erfüllt:

- Jedes Atom  $a \in A$  ist in  $F$ .
- Wenn  $\varphi_1, \varphi_2$  Formeln aus  $F$  sind, dann sind
  - $\varphi_1 \wedge \varphi_2$  (Konjunktion, UND)
  - $\varphi_1 \vee \varphi_2$  (Disjunktion, ODER)
  - $\varphi_1 \rightarrow \varphi_2$  (Implikation; wenn  $\varphi_1$  dann  $\varphi_2$ )
  - $\varphi_1 \leftrightarrow \varphi_2$  (Äquivalenz;  $\varphi_1$  genau dann, wenn  $\varphi_2$ )
  - $\neg\varphi_1$  (Negation)

ebenfalls in  $F$ .

Teilausdrücke werden mit Klammern „(“ und „)“ abgegrenzt. Weiters werden Bindungsregeln für die Operatoren vereinbart, der Stärke der Bindung nach sortiert, von der stärksten bis zur schwächsten lautet die Reihenfolge:

–  $\neg$

–  $\wedge$

---

<sup>1</sup> \*14.06.1903 Washington, † 11.08.1995 Hudson, Ohio. Alonzo Church lieferte wichtige Erkenntnisse und Beiträge auf dem Gebiet der theoretischen Informatik, mathematischen Logik und der Rekursionstheorie.

–  $\vee$

–  $\rightarrow, \leftrightarrow$

Beispielsweise bedeutet  $\neg A1 \wedge A2 \vee A3 \rightarrow A10$  soviel wie  $((\neg A1) \wedge A2) \vee A3 \rightarrow A10$  [vgl. Egly, 2002, S. 3.12–3.13].

### 5.3.3.3 Semantik

Die Semantik von Formeln der Aussagenlogik wird durch Interpretationen und Modelle festgelegt. In der Abstraktion kann dabei jede atomare Aussage  $a$  entweder den Wert wahr (true, t) oder falsch (false, f) annehmen.

**Interpretation:** eine Interpretation ist eine Abbildung

$$I: A \rightarrow \{t, f\}$$

die jedem Atom  $a \in A$  einen Wahrheitswert  $I(a)$  zuordnet. Die Menge aller möglichen Interpretationen wird mit  $Int(A)$  bezeichnet. Der Wahrheitswert einer komplexen Aussage  $\varphi$  bezüglich einer Interpretation  $I$ , geschrieben als  $I(\varphi)$ , lässt sich rekursiv aus den Wahrheitswerten der verknüpften Teilaussagen anhand der folgenden Verknüpfungstabelle ermitteln.

Tabelle 5-1: Logische Verknüpfungen

$I(\varphi_1)$	$I(\varphi_2)$	$I(\varphi_1 \wedge \varphi_2)$	$I(\varphi_1 \vee \varphi_2)$	$I(\varphi_1 \rightarrow \varphi_2)$	$I(\varphi_1 \leftrightarrow \varphi_2)$	$I(\varphi_1)$
t	t	t	t	t	t	f
t	f	f	t	f	f	f
f	t	f	t	t	f	t
f	f	f	f	t	t	t

**Modell:** Eine Interpretation  $I$  ist ein Modell für eine Aussage  $\varphi$  ( $I \models \varphi$ ), wenn  $I(\varphi) = t$  gilt.  $I$  ist ein Modell für eine Menge von Aussagen  $S$  ( $I \models S$ ), wenn  $I$  ein Modell für jedes  $\varphi \in S$  ist (d.h.  $\forall \varphi \in S: I \models \varphi$ ).

**Logische Folgerung:** Eine Aussage  $\psi$  folgt logisch aus einer Aussage  $\varphi$  ( $\varphi \models \psi$ ), wenn jedes Modell von  $\varphi$  ein Modell von  $\psi$  ist.

**Erfüllbare Aussage:** Eine Aussage  $\varphi$  ist erfüllbar, wenn sie (mindestens) ein Modell besitzt.

**Logische Äquivalenz:** Zwei Aussagen  $\varphi$  und  $\psi$  sind logisch äquivalent, wenn  $\varphi \models \psi$  und  $\psi \models \varphi$  gilt.

Weitere Definitionen und Begriffe kommen im Rahmen von FuzzyDBSexpert nicht zur Anwendung und werden deshalb an dieser Stelle nicht weiter ausgeführt [vgl. Egly, 2002, S. 3.13–3.15].

### 5.3.3.4 Elementare Resultate der Aussagenlogik

Folgende, für die späteren Betrachtungen wichtige Formeln sind logisch äquivalent (für beliebige  $\varphi$  und  $\psi$ ) [vgl. Egly, 2002, S. 3.15].

$\neg\varphi \vee \psi$  und  $\varphi \rightarrow \psi$  Darstellung der Implikation als Disjunktion

$\neg(\varphi \vee \psi)$  und  $\neg\varphi \wedge \neg\psi$ ;  $\neg(\varphi \wedge \psi)$  und  $\neg\varphi \vee \neg\psi$  De Morgan'sche Regeln

### 5.3.4 Prädikatenlogik

Die Prädikatenlogik ist wesentlich ausdrucksstärker als die Aussagenlogik. Die Komponenten sind hier Objekte, deren Eigenschaften und Beziehungen untereinander. Aus elementaren Aussagen über Eigenschaften und Beziehungen von Objekten können durch logische Verknüpfungen komplexere Aussagen gebildet werden. Objekte bleiben teilweise unbestimmt und werden durch Platzhalter (Variablen) repräsentiert (z. B.  $\text{mensch}(x)$ ). Über logische Quantoren ( $\forall$  Allquantor,  $\exists$  Existenzquantor) kann die Betrachtung einer Aussage für bestimmte Belegungen der Variablen ausgedrückt werden (z. B.  $\forall x(\text{mensch}(x) \rightarrow \text{sterblich}(x))$ ).

Im Gegensatz zur Aussagenlogik sind die Probleme der Erfüllbarkeit und der Gültigkeit eines Satzes der Prädikatenlogik im Allgemeinen nicht entscheidbar (Alonzo Church), das heißt, es gibt keinen Algorithmus, der für einen gegebenen Satz in endlicher Zeit feststellt, ob dieser Satz erfüllbar beziehungsweise gültig ist [vgl. Egly, 2002, S. 3.17–3.23].

## 6 Regelbasierte Systeme

In diesem Kapitel soll eine weitere Basis für FuzzyDBSexpert erläutert werden, nämlich Systeme, welche auf Regeln basieren.

### 6.1 Was sind Regeln?

Regeln sind formalisierte Konditionalsätze der Form

WENN  $A$  DANN  $B$   
(if  $A$  then  $B$ )

mit der Bedeutung: Wenn  $A$  wahr (erfüllt, bewiesen) ist, dann schliesse, dass auch  $B$  wahr ist.

$A$  und  $B$  sind hierbei Aussagen oder prädikatenlogische Formeln, je nachdem, welche Wissensrepräsentationsform gewählt wurde. Die Formel im „WENN“-Teil der Regel wird als Prämisse oder Bedingungsteil bezeichnet, die Formel im „DANN“-Teil der Regel als Konklusion oder Schlussfolgerungsteil. Wenn die Prämisse einer Regel erfüllt ist, kann man auch sagen, dass die Regel feuert. In den folgenden Abschnitten wird die Betrachtung auf die Aussagenlogik beschränkt, da diese auch das Mittel der Wahl ist. Regeln stellen einen guten Kompromiss zwischen Verständlichkeit der Wissensdarstellung und formalen Ansprüchen dar.

Häufig legt man bei der Modellierung der Wissensbasis auf eine syntaktisch möglichst einfache Form der Regeln Wert, einerseits, um eine effiziente Abarbeitung der Regeln zu erlauben und andererseits verbessert es die Übersichtlichkeit der Wirkung der einzelnen Regeln, diese sind dann einfacher für Benutzer lesbar. Im Allgemeinen stellt man an die syntaktische Form der Regeln folgende Bedingungen:

- Die Verknüpfung  $\vee$  darf nicht in der Prämisse einer Regel auftreten.
- Die Konklusion einer Regel soll nur aus einem Literal, also einem positiven oder negativen Atom bestehen.

Regeln, die diese beiden Bedingungen nicht erfüllen, müssen vereinfacht werden. Durch wiederholte Anwendung der folgenden Schritte kann man komplexe in syntaktisch einfachere Formeln umformen, die beiden oben angegebenen Bedingungen genügen:

1. Ersetze die Regel

$$\text{if } K_1 \vee \dots \vee K_n \text{ then } D_1 \wedge \dots \wedge D_m$$

durch die  $n \cdot m$  Regeln

$$\text{if } K_i \text{ then } D_j, i \in \{1, \dots, n\}, j \in \{1, \dots, m\}.$$

2. Ersetze die Regel

$$\text{if } K \text{ then } L_1 \vee \dots \vee L_p$$

( $K$  ist eine Disjunktion von Literalen  $L_p$ ) durch die  $p$  Regeln

$$\text{if } K \wedge \left( \bigwedge_{k \neq k_0} \neg L_k \right) \text{ then } L_{k_0}, k_0 \in \{1, \dots, p\}.$$

Im Allgemeinen lassen sich alle Regeln der Form „if  $A$  then  $B$ “ mit Hilfe der Distributivgesetze, der De Morgan'schen Regeln und der Darstellung der Implikation als Disjunktion (siehe Kapitel 5.3.3.4) in eine Form umwandeln, dass die Prämisse  $A$  eine Disjunktion von Konjunktionen  $K_i$  und die Konklusion  $B$  eine Konjunktion von Disjunktionen  $D_j$  ist.

Diese Beschränkung der Formeln auf syntaktisch einfache Regeln vergrößert zwar die Zahl der Regeln, aufgrund ihrer einfachen Struktur lassen sie sich jedoch viel schneller auswerten. Regeln in einer solchen Darstellung zeigen das darzustellende Wissen aber auch in einer sehr verständlichen Form und können wegen der Übersichtlichkeit dazu beitragen, Inkonsistenzen zu vermeiden. An dieser Stelle ist aber auch der Wissensingenieur gefordert, denn bevor Regeln in die Wissensbasis aufgenommen werden können, müssen diese erst in die dargestellte Form gebracht werden, was eine gewisse Vertrautheit mit der Aussagenlogik voraussetzt [vgl. Beierle, 2006, S. 71–73].

## 6.2 Die Wissensbasis eines regelbasierten Systems

In den Regeln in abstrakter Form, wie wir sie im vorigen Kapitel kennen gelernt haben, steckt noch kein Wissen, denn weder die Prämissen, noch die Konjunktionen, werden näher spezifiziert. Die Wissensbasis eines regelbasierten Systems enthält zunächst Objekte und deren Beschreibungen mittels einer endlichen Menge diskreter Werte. Die Regeln repräsentieren Zusammenhänge zwischen Objekten oder Mengen von Objekten. Objekte und Regeln zusammen bilden das generische Wissen. Wenn man das regelbasierte System auf einen speziellen Fall anwendet, so kommt das fallspezifische Wissen hinzu [vgl. Beierle, 2006, S. 76–77].

## 6.3 Inferenz in einem regelbasierten System

Die grundlegende Inferenzregel in einem regelbasierten System ist der sogenannte modus ponens:

if $A$ then $B$	(Regel)	
$A$ wahr	(Faktum)	modus ponens
$B$ wahr	(Schlussfolgerung)	

Die Regel ist Teil des generischen Wissens, das Faktum basiert im Allgemeinen auf einer Beobachtung und ist fallspezifisches Wissen, durch Anwendung der Regel entsteht neues fallspezifisches Wissen. Mit dem modus ponens, welcher eine Regel zur Inferenz benutzt, ist die Leistungsfähigkeit von regelbasierten Systemen keineswegs erschöpft, diese entsteht durch die Verkettung von mehreren Regeln und ermöglicht es so, komplexe Informationen auf relativ einfache Weise zu verarbeiten. Bei der verketteten Inferenz spielt die Richtung eine große Rolle, nach ihr kann man Vorwärts- und Rückwärtsverkettung unterscheiden, welche in den folgenden Kapiteln umrissen werden sollen.

Man muss sich aber bewusst sein, dass die Anwendung dieser Regeln gerichtet erfolgt, das bedeutet, dass sie erst feuern können, wenn ihre Prämisse erfüllt ist. Es können viele – jedoch nicht alle – Elemente der Aussagenlogik für regelbasierte Systeme angewendet werden, wie folgendes Beispiel zeigt. Laut Aussagenlogik sind folgende Regeln äquivalent:

$$\text{if } A \text{ then } B \quad \text{und} \quad \text{if } \neg A \text{ then } \neg B$$

Nach der klassischen Aussagenlogik haben die beiden Regeln das Äquivalent  $\neg A \vee B$ . Wenn man nun aber den modus ponens zur Inferenz verwendet, dann feuert entweder nur die linke (wenn  $A$  wahr ist) oder nur die rechte (wenn  $A$  falsch ist) Regel, man kommt so zu einem von der Aussagenlogik verschiedenen Ergebnis. Um die klassisch-logische Äquivalenz zu erhalten, falls dies notwendig ist, muss man entweder beide Regeln in die Wissensbasis aufnehmen, oder den modus tollens zusätzlich zum modus ponens implementieren:

if $A$ then $B$	(Regel)	
$B$ falsch	(Faktum)	modus tollens
$A$ falsch	(Schlussfolgerung)	

[vgl. Beierle, 2006, S. 78–80]

## 6.4 Vorwärtsverkettung

Die Vorwärtsverkettung (siehe Abbildung 6.1) wird auch datengetriebene Inferenz genannt. Bei diesem Verfahren wird das fallspezifische Wissen als Ausgangsposition für den Schlussfolgerungsprozess verwendet. Aus erfüllten Prämissen wird auf die Wahrheit der Konklusionen geschlossen und dieses neue faktische Wissen weiter im Inferenzprozess benutzt. Dieses Verfahren endet, wenn keine neuen Fakten mehr abgeleitet werden können [vgl. Beierle, 2006, S. 81–82].

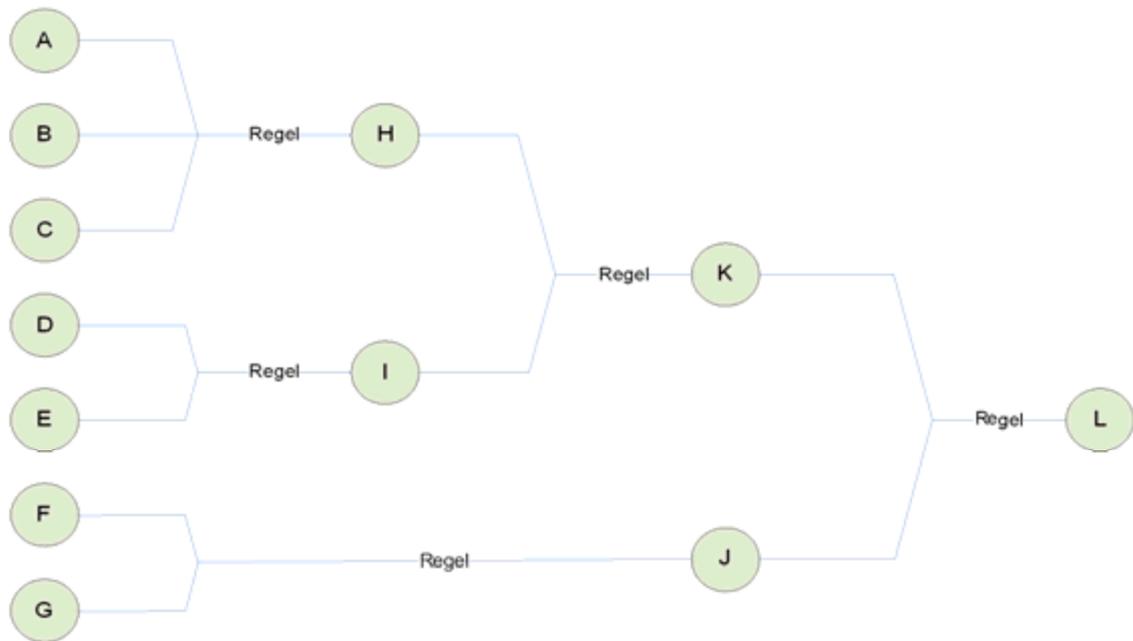


Abbildung 6.1: Beispiel einer Vorwärtsverkettung [vgl. Beierle, 2006, S. 82]

## 6.5 Rückwärtsverkettung

Die Rückwärtsverkettung wird auch zielorientierte Inferenz genannt. Hier geht man nicht von den gegebenen Daten aus, sondern von einem Zielobjekt, über dessen Zustand der Benutzer Informationen erhalten möchte. Das System durchsucht dann die Regelbasis nach geeigneten Regeln, die das Zielobjekt in der Schlussfolgerung enthalten. Die Objekte der Prämissen werden zu Zwischenzielen und so fort [vgl. Beierle, 2006, S. 83–84].

## 6.6 Das Problem der Widersprüchlichkeit

Wenn in der Wissensbasis sehr viele Regeln gespeichert sind, kann es leicht passieren, dass widersprüchliches formales Wissen in der Wissensbasis gespeichert ist. Experten benutzen zur Formulierung ihrer Schlüsse oft nicht genannte Annahmen oder vergessen auf die Nennung von Ausnahmen. Folgendes Beispiel verdeutlicht einen Widerspruch:

$$\text{if } A \text{ then } C$$
$$\text{if } A \wedge B \text{ then } \neg C$$

Sind bei einem konkreten Fall  $A$  und  $B$  wahr, dann kommt das System zu den Schlüssen  $C$  und  $\neg C$ , was einen Widerspruch darstellt. Wir werden später eine Logik kennen lernen, wo trotz solcher Regeln oder sogar mit solchen, nach der klassischen zweiwertigen Logik widersprüchlichen, Aussagen gearbeitet werden kann. Regelbasen können in zwei unterschiedlichen Weisen widersprüchlich sein:

- Die Regelbasis ist klassisch-logisch inkonsistent, es gibt also keine Belegung der Objekte mit Werten, sodass alle Regeln erfüllt sind.
- Die Regelbasis führt zu widersprüchlichen Ableitungen (wie im Beispiel oben).

In der Praxis kommt letzterer Widerspruch weit häufiger vor. Man sollte deswegen eine Konsistenzprüfung in solchen regelbasierten Systemen integrieren, damit Widersprüchlichkeiten vermieden werden können, welche zu einem Fehlverhalten des Systems, zumindest aber zu einem ungeplanten Verhalten führen [vgl. Beierle, 2006, S. 85].

## 6.7 Die Erklärungskomponente

Bei regelbasierten Systemen ist es einfach, dem Benutzer eine Erklärung für die gezogenen Schlüsse aufzuzeigen, indem man die verwendeten Regeln auflistet [vgl. Beierle, 2006, S. 86].

## 6.8 Modularität und Effizienz regelbasierter Systeme

In einem „idealen“ regelbasierten System, wie es skizziert wurde, wird jede Regel als eine Wissensseinheit gespeichert, welche unabhängig von der restlichen Wissensbasis ist. Diese haben die in Kapitel 6.1 einfache Form und können somit sehr einfach in Computern, beispielsweise in einer Datenbank, gespeichert werden. Dieses Wissen ist auch sehr modular, die Wissensbasis kann sehr einfach an neues Wissen angepasst werden, indem man die neuen Regeln einfach dazu speichert [vgl. Beierle, 2006, S. 95–96].

## 7 Quantitative Methoden

Um unsicheres Wissen zu repräsentieren, wurden von Anfang an auch quantitative Ansätze zur Repräsentation und Verarbeitung von Wissen verwendet. Es sollen an dieser Stelle die bekanntesten Verfahren dargestellt werden, um schließlich zur Fuzzy-Logik zu gelangen.

### 7.1 Probabilistische Netzwerke

Ein äußerst bekannter Vertreter von wissensbasierten Systemen, welcher mit probabilistischen Netzwerken arbeitet, ist MYCIN, eines der ersten namhaften Expertensysteme. Im Allgemeinen werden hier den Formeln und Aussagen numerische Größen zugeordnet, die den Grad ihrer Gewissheit, die Stärke der Einflussnahme, ihren Zugehörigkeitsgrad zu einer gewissen Menge und dergleichen ausdrücken. Die Repräsentation dieser Größen durch Zahlen ist auch ein sehr großer Vorteil von probabilistischen Netzwerken für die Verarbeitung im Computer [vgl. Beierle, 2006, S. 89–92].

Durch die Komplexität vieler praktischer Probleme kommt man aber leicht an die Grenzen der Berechenbarkeit, jedoch kann man mit Vereinfachungen und Annahmen die Komplexität reduzieren. Ein weiteres Problem tritt bei der Konzeption eines solchen Systems auf: Für den Aufbau der Wissensbasis benötigt man eine ganze Menge von exakten Zahlen. Wenn gesicherte statistische oder natürlich berechenbare Daten vorliegen, stellt die exakte Formulierung in Zahlen kein Problem dar. Es wäre aber auch eine Verwendung von subjektiven Quantifizierungen möglich. Dies kommt einer mehr oder weniger genauen Schätzung gleich, aus der aber ein exakter Zahlenwert resultiert, welcher für exakte Berechnungen verwendet wird. Wenn dieser Wert noch dazu eine exakte Grenze, zum Beispiel für oder gegen eine Operation zur Tiefen Hirnstimulation darstellt, dann kann man sich leicht vorstellen, dass eine Verrückung dieser Grenze große Auswirkungen auf die Patienten im Grenzbereich hat.

Man muss sich bei der Verwendung von quantitativen Methoden auch eine funktionierende Semantik überlegen, um einerseits eine Modellierung der konkreten Probleme zu ermöglichen und andererseits den berechneten Zahlen die richtige Bedeutung beimessen zu können. Umgekehrt muss man sich überlegen, ob Unsicherheiten bei der Quantifizierung aus unvollständigen Informationen oder durch problemspezifische Unbestimmtheit entstehen [vgl. Beierle, 2006, S. 359].

Die verbreitetste und am weitesten entwickelte Methode zur quantitativen Darstellung von Unsicherheit ist die Wahrscheinlichkeitstheorie. Im Allgemeinen ist probabilistisches Schließen jedoch nur in einem sehr begrenzten Ausmaß möglich. So lässt sich zum Beispiel die Wahrscheinlichkeit einer Konjunktion  $P(A \wedge B)$  im Allgemeinen nicht aus den

Wahrscheinlichkeiten der einzelnen Konjunkte  $P(A)$  und  $P(B)$  berechnen. Daher wird einem probabilistischen System in der Regel Wissen in Form von Unabhängigkeitsannahmen hinzugefügt. Unabhängigkeiten lassen sich gut mittels Graphen veranschaulichen und ermöglichen effiziente Verfahren zur Berechnung von Wahrscheinlichkeiten. Für wissensbasierte Systeme werden vor allem Markov-Netze und Bayes-Netze verwendet, welche in [Beierle, 2006] ausführlich vorgestellt werden [vgl. Beierle, 2006, S. 359–410].

## 7.2 Übergang zu anderen Methoden

Ein Nachteil des probabilistischen Ansatzes ist die Erfordernis präziser Wahrscheinlichkeitswerte. Einerseits ist die Spezifikation solcher exakten Werte problematisch, andererseits lässt sich probabilistische Unsicherheit nicht von der durch fehlendes Wissen bedingten Unsicherheit unterscheiden. Mit der Dempster-Shafer-Theorie wurde ein Konzept zur expliziten Modellierung von Unwissenheit entwickelt. Das Ziel der Fuzzy-Theorie ist hingegen die adäquate Beschreibung ungenauer Begriffe. Die Anwendung der Fuzzy-Theorie auf unscharfe Wissenszustände führt zur Probabilistik. Sowohl die Dempster-Shafer-, als auch die Fuzzy-Theorie, lassen sich als Verallgemeinerung des wahrscheinlichkeitstheoretischen Ansatzes auffassen [vgl. Beierle, 2006, S. 410].

### 7.2.1 Verallgemeinerte Wahrscheinlichkeitstheorie

Um bei der Modellierung von Wahrscheinlichkeiten etwas flexibler zu sein, kann man anstatt eines präzisen Wahrscheinlichkeitswertes ein Wahrscheinlichkeitsintervall annehmen. Die Verwendung von Wahrscheinlichkeitsintervallen macht aber das schon komplexe probabilistische Kalkül noch aufwändiger und außerdem erhält man in vielen Fällen nicht aussagekräftige, weil zu große, Wahrscheinlichkeitsintervalle als Ergebnis. Eine Alternative zur Anwendung der bisher vorgestellten Methoden wäre, das Konzept der Wahrscheinlichkeit selbst zu verallgemeinern, um insbesondere den Additionssatz für disjunkte Ereignisse  $P(A \cup B) = P(A) + P(B)$  zu umgehen. Dies erreicht man durch die folgenden Definitionen:

**Kapazität:** Sei  $F:2^\Omega \rightarrow [0,1]$  eine Funktion von der Potenzmenge einer Menge  $\Omega$  ins Einheitsintervall.  $F$  heißt normalisierte Kapazität, wenn sie die folgenden Bedingungen erfüllt:

- Normalisierung:  $F(\Omega) = 1$
- Monotonie: Für alle  $A, B \subseteq \Omega$ ,  $A \subseteq B$  gilt:  $F(A) \leq F(B)$

Kapazitäten werden in der Literatur auch als Fuzzy-Maße bezeichnet. Wahrscheinlichkeitsfunktionen sind auch Kapazitäten, aber nicht umgekehrt. Es wird nicht gefordert, dass Kapazitäten additiv sind, insbesondere gilt:  $F(\bar{A}) \neq 1 - F(A)$

**Dualität:** Zwei reellwertige Funktionen  $F, G$  auf  $2^\Omega$  heißen dual zueinander, wenn für jedes  $A \subseteq \Omega$  gilt

$$F(A) = 1 - G(\bar{A}) \text{ und } G(A) = 1 - F(\bar{A}).$$

Eine Wahrscheinlichkeitsfunktion ist offensichtlich dual zu sich selbst. Statt Wahrscheinlichkeitsintervallen, also Mengen möglicher Wahrscheinlichkeitswerte, kann man auch Mengen von Wahrscheinlichkeitsfunktionen betrachten.

**Untere und obere Wahrscheinlichkeitsschranke:** Sei  $\Pi$  eine nichtleere Menge von Wahrscheinlichkeitsfunktionen über derselben Menge  $\Omega$ . Die untere Wahrscheinlichkeitsschranke  $\Pi_u$  und die obere Wahrscheinlichkeitsschranke  $\Pi_o$  werden für alle  $A \subseteq \Omega$  definiert durch

$$\Pi_u(A) = \inf\{P(A) \mid P \in \Pi\} \text{ und}$$

$$\Pi_o(A) = \sup\{P(A) \mid P \in \Pi\}$$

Wobei  $\inf(M)$  das Infimum und  $\sup(M)$  das Supremum einer Menge  $M$  bezeichnet [vgl. Beierle, 2006, S. 410–412].

## 7.2.2 Die Dempster-Shafer-Theorie

### 7.2.2.1 Grundlagen

Die Dempster-Shafer-Theorie (DS-Theorie) ist eine Theorie des plausiblen Schließens, die auf dem Begriff der Evidenz beruht, deswegen wird sie auch Evidenztheorie genannt. Das zentrale Konzept der DS-Theorie ist das der Glaubensfunktion, die eine explizite Repräsentation von Unwissenheit erlaubt. Der zweite Vorteil dieser Theorie ist die Verfügbarkeit einer Regel zur Kombination von Evidenzeinflüssen, welche die Wissensverarbeitung vereinfachen kann [vgl. Beierle, 2006, S. 412].

### 7.2.2.2 Basismaße und Glaubensfunktion

Das Universum  $\Omega$  aller möglichen und sich gegenseitig ausschließenden Ereignisse wird in der DS-Theorie als Wahrnehmungsrahmen bezeichnet. Jedes Ereignis lässt sich als eine Teilmenge von  $\Omega$  darstellen. Wie in der Wahrscheinlichkeitstheorie lassen sich Ereignisse mit aussagenlogischen

Formeln identifizieren, es wird in dieser Theorie jedoch im Allgemeinen eine mengentheoretische Schreibweise bevorzugt.

**Basismaß:** Ein Basismaß (basic probability assignment, bpa) über dem Rahmen  $\Omega$  ist eine Funktion  $m : 2^\Omega \rightarrow [0,1]$ , die den folgenden Bedingungen genügt:

$$m(\emptyset) = 0$$

$$\sum_{A \subseteq \Omega} m(A) = 1$$

Die numerische Größe  $m(A)$  ist das Maß an Glauben, das man exakt der Menge  $A$  zuweist. Ein Basismaß ist weder ein Wahrscheinlichkeitsmaß, noch eine Kapazität, da es im Allgemeinen weder monoton, noch additiv ist. Die oben angegebenen Bedingungen stellen sicher, dass man der leeren Menge keinen Glauben schenkt und dass die Gesamtmenge des Glaubens 1 beträgt.

**Glaubensfunktion:** Sei  $m$  ein Basismaß über den Rahmen  $\Omega$ . Die durch  $m$  induzierte Glaubensfunktion  $Bel$  wird definiert durch

$$Bel : 2^\Omega \rightarrow [0,1], \quad Bel(A) := \sum_{B \subseteq A} m(B).$$

$Bel(A)$  ist also das totale Maß an Glauben, das mit Sicherheit der Menge  $A$  zugewiesen wird. Demgegenüber kann man auch das totale Maß an Glauben betrachten, das möglicherweise  $A$  zugewiesen werden könnte, wenn nur die Umstände entsprechend günstig wären, was als Plausibilität bezeichnet wird. Auf die mathematische Definition soll hier nicht näher eingegangen werden [vgl. Beierle, 2006, S. 412–413].

Dempster's Kombinationsregel stellt noch ein Werkzeug zur Verfügung, damit man mehrere gewichtete Evidenzen bei der Bewertung einer Hypothese in Betracht ziehen kann. Diese soll aber hier nicht weiter ausgeführt werden, man findet weitere Details in [Beierle, 2006].

## 8 Fuzzy-Logik

„Die Theorie der Fuzzy Sets ist nicht innerhalb der Mathematik als Verallgemeinerung der Mengentheorie entstanden. Zadeh war und ist kein Mathematiker, der daran interessiert gewesen wäre, ein abstraktes Theoriegebäude zu begründen. Ihn interessierten die mathematischen Prinzipien für technische Anwendungen und dort hatte er schon bald die große Schere bemerkt, die sich zwischen Theorie und Praxis auftat. Als er die Fuzzy Set Theorie schuf, war dies ein Versuch, diese Kluft zwischen reiner Mathematik und technischer Anwendung zu schließen“ [Seising, 2005, S. 3].

Das Kapitel Fuzzy-Logik schließt nahtlos an das Kapitel „Übergang zu anderen Methoden“, wie die Dempster-Shafer-Theorie, an. Nachdem es für diese Arbeit aber von so herausragender Bedeutung ist, wird ihr ein eigenes zugeordnet, sie kann aber nicht ausschließlich der Erläuterung der Fuzzy-Logik dienen. Es werden im Folgenden nur die Grundkonzepte und die notwendigen Details und Erweiterungen der vorangegangenen Kapitel dargestellt, um die Ausführungen im methodischen Teil nachvollziehen zu können.

### 8.1 Einführung in die Fuzzy-Logik

„Hier geht es keineswegs in erster Linie darum, Anwendungsgebiete zu erschließen, deren spezifische Problematik mit anderen Methoden nicht lösbar wären. Wenn der Einsatz von Methoden der Fuzzy-Logik zu neuartigen Produkten führt, dann vielmehr deshalb, weil sich die Produktentwicklung vereinfacht und die Entwicklungszeit erheblich verkürzt“ [Bothe, 1995, S. 1].

Die Fuzzy-Logik beschreibt Wahrheitswerte nicht nur mit „wahr“ oder „falsch“, wie dies bei der zweiwertigen Logik der Fall ist, oder mit Zahlenwerten, sondern mit Hilfe einer linguistischen Werteskala. Die Werte liegen dabei in Form verbaler Ausdrücke vor, wie zum Beispiel *sehr falsch*, *falsch*, *wahr*, *sehr wahr*. Damit ergibt sich die Möglichkeit, verbale Ausdrücke in einen mathematisch fassbaren Bereich zu transformieren und dort automatisch weiter zu verarbeiten [vgl. Bothe, 1995, S. 2].

„Uncertainty, with variations such as vagueness and imprecision, can be found at almost every step in medical reasoning“ [Boegl, 2004, S. 3].

Den Begriff „Fuzzy-Logik“ kann man mit „unscharf begrenzte, fusselige Logik“ übersetzen. Mit Hilfe der Fuzzy-Logik können auch Beschreibungen verarbeitet werden, die auf der Basis von unscharfen umgangs- und fachsprachlichen Aussagen (Regeln) gegeben sind. Wesentliche Einflussparameter sind oft unbekannt oder wurden bei der Regelerstellung nicht erfasst. Bei komplexen Vorgängen können dadurch „Unschärfen“ entstehen. Die Kombination bestehender

unscharfer Einzelaussagen über ein System trägt aber einen hohen Wahrheitswert und repräsentiert ein spezielles Expertenwissen, aufgrund dessen Entscheidungen getroffen werden können. Da die Voraussetzungen der Wahrscheinlichkeitstheorie nicht notwendigerweise erfüllt sein müssen, weil zum Beispiel der Einfluss bestimmter Parameter nicht erfasst wurde, besteht ein großer Unterschied zwischen Unschärfe und Wahrscheinlichkeit einer unscharfen Aussage. Unschärfe kann auch als „Möglichkeit“ interpretiert werden [vgl. Bothe, 1995, S. 3].

## 8.2 Grundlegende Konzepte

Der Begriff der „Fuzzy-Logik“ kann folgendermaßen in einer engen und einer weiten Deutung gebraucht werden.

### 8.2.1 Enge Deutung der Fuzzy-Logik

„Die enge Deutung beschreibt ein im mathematischen Sinne logisches System mit dem Ziel, Modelle für die Erscheinungsformen der menschlichen Beweis- und Entscheidungsfindung aufzustellen. Diese Erscheinungsformen werden in symbolischer Form eher annähernd als exakt festgelegt“ [Bothe, 1995, S. 6].

### 8.2.2 Weite Deutung der Fuzzy-Logik

Die weite Deutung beschreibt die Theorie unscharfer Mengen, das heißt, die Lehre von Mengen mit unscharfen Begrenzungen. Die Bedeutung dieser Theorie ergibt sich daraus, dass

- viele natürliche Mengen eher unscharf als scharf begrenzt sind,
- die Gesetze der klassischen Mengenlehre über die Boole'sche Algebra auf die klassische Logik und Schaltalgebra übertragbar sind und sich die Gesetze der Theorie unscharfer Mengen daraus ableiten lassen [vgl. Bothe, 1995, S. 6].

### 8.2.3 Unscharfe Mengen

Ein zentrales Konzept der Fuzzy-Logik sind die unscharfen Mengen. Die Definition einer bestimmten unscharfen Menge  $M$  erfolgt mit Hilfe einer charakteristischen Funktion  $\mu_M$ , die als „Zugehörigkeitsfunktion“ bezeichnet wird. Die Werte von  $\mu_M$  liegen im Allgemeinen zwischen 0 (keine Zugehörigkeit) und 1 (vollständige Zugehörigkeit). Damit können im Gegensatz zur klassischen Mengenlehre auch weiche Übergänge der Elementzugehörigkeit zu einer Menge beziehungsweise des Zutreffens einer Aussage, beschrieben werden. So werden zum Beispiel die meisten Menschen zustimmen, wenn behauptet wird, dass man den Begriff „lauwarm“ weder

vollständig der Menge „warm“, noch „kalt“ zuordnen kann, sondern zu einem gewissen Anteil zu beiden Mengen. Scharf begrenzte Mengen können jedoch insofern in unscharfen Mengen enthalten sein, als dass die Zugehörigkeitsfunktionen nur die Werte 0 und 1 annehmen können [vgl. Bothe, 1995, S. 6].

## 8.2.4 Linguistische Variable

„In medical diagnosis, which is strongly influenced by human perception and judgement, in many cases it is more appropriate to describe the underlying knowledge by means of linguistic variables than by quantitative descriptions“ [Boegl, 2004, S. 5].

Die Werte einer linguistischen Variablen sind Worte beziehungsweise Terme einer natürlichen oder synthetischen Sprache. Sie werden durch unscharfe Mengen  $A_i$  beziehungsweise deren Zugehörigkeitsfunktionen  $\mu_i(x)$  in Form von Verteilungsfunktionen über einer Basisvariablen  $x$  eines Grundbereichs  $X$ , repräsentiert. Diese Zugehörigkeitsfunktionen bilden eine linguistische auf eine numerische Werteskala, also den Bereich von 0 bis 1, ab [vgl. Bothe, 1995, S. 9].

## 8.2.5 Fuzzy-Inferenzsysteme

Fuzzy-Inferenzsysteme dienen dazu, aus bestimmten Eingangsgrößen bestimmte Ausgangsgrößen zu ermitteln, man könnte dies auch als Fuzzy-Regelung bezeichnen. Fuzzy-Inferenzsysteme funktionieren in folgenden 5 Schritten, welche in Abbildung 8.1 dargestellt sind:

- Fuzzifizierung der Eingangssignale
- Anwendung der Fuzzy-Operatoren
- Anwendung der Fuzzy-Implikation
- Aggregation der Ausgangsgrößen
- Defuzzifizierung der Ausgangsgrößen

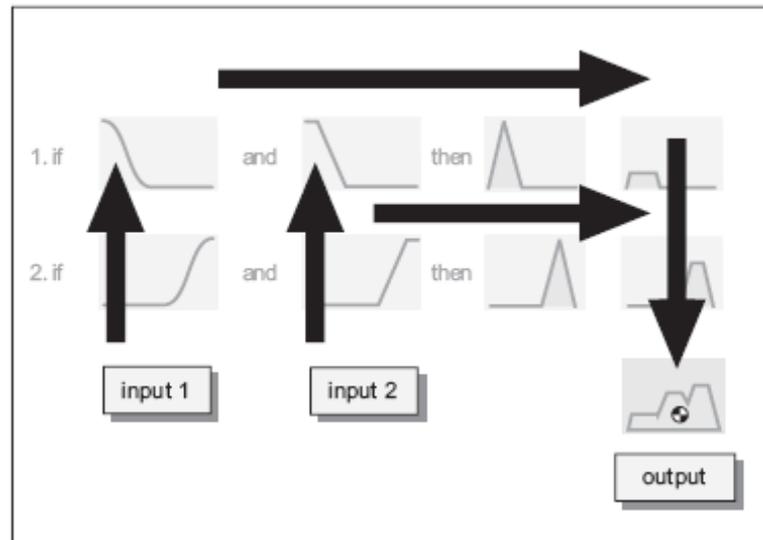


Abbildung 8.1: Schema einer Fuzzy-Inferenz [MathWorks, 2004, S. 2-27]

### 8.2.5.1 Fuzzifizierung der Eingangssignale

Viele Eingangswerte  $x_i$  eines Systems sind scharfe Zahlen. Diese müssen zunächst auf die linguistische Werteskala abgebildet werden, auf der das System dann mittels (WENN-DANN-) Regeln operiert, da die Ausgangswerte  $y_i(x)$  auf der Basis dieser Regeln hergeleitet werden. Diese  $x_i$  sollten zunächst in den Zugehörigkeitsraum der verwendeten linguistischen Terme transformiert werden. Wenn eine linguistische Variable durch  $n$  Terme beschrieben wird (siehe Abbildung 8.2), entsteht als fuzzifiziertes Signal ein  $n$ -dimensionaler Vektor  $s(x)$  mit den Elementen  $\mu_i(x) \in [0,1]$ ,  $i=1, \dots, n$ . Man spricht auch vom „Sympathievektor“ des Eingangswertes. Dieser Vektor wird beim Herleiten der Folgerungen weiterverarbeitet. Die Transformation wird „Fuzzifizierung der Eingangssignale“ genannt [vgl. Bothe, 1995, S. 10].

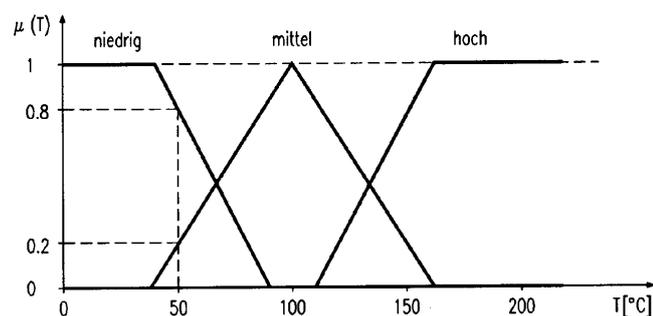


Abbildung 8.2: Drei Terme der linguistischen Variable „Temperatur“ werden durch Zugehörigkeitsfunktionen über den numerischen Temperaturwerten beschrieben (50°C sind hier zu 0,8 „niedrig“, zu 0,2 „mittel“ und zu 0,0 „hoch“) [Bothe, 1995, S. 11]

### 8.2.5.2 Anwendung der Fuzzy-Operatoren

Bei den Fuzzy-Regeln (z. B. WENN ... DANN ...) muss, bevor die Regel ausgewertet werden kann, erst der Ausdruck des Bedingungssteils berechnet werden, wenn dieser aus mehreren mit UND oder

ODER verknüpften Teilen besteht. Dazu müssen erst die logischen Verknüpfungen der zweiwertigen Logik so erweitert werden, dass sie auch in der Fuzzy-Logik anwendbar sind. Die logischen Fuzzy-Operatoren werden in Kapitel 8.3.2 näher vorgestellt.

### 8.2.5.3 Anwendung der Fuzzy-Implikation

Mit Hilfe von linguistischen Variablen werden zur Beschreibung des Systemverhaltens eine Reihe von Aussagen getroffen, die für bestimmte Eingangswertkombinationen die dazugehörigen Ausgangswertkombinationen in Form von WENN-DANN-Regeln festlegen. Da die Eingangswerte im Allgemeinen nicht genau die Bedingungen der Regeln erfüllen werden, können nur unscharfe Schlüsse beziehungsweise Folgerungen gezogen werden. Der dazu erforderliche Vorgang wird als „Approximatives Schließen“ bezeichnet [vgl. Bothe, 1995, S. 11–12].

Der Aktionsteil besteht also aus einem Fuzzy-Set, das abhängig vom Bedingungsteil umgeformt wird. Typische Beispiele für die Implikation sind auch jene, die für das Fuzzy-UND verwendet werden, also der MIN (Minimum)-Operator, welcher die Ausgangsfunktion beschneidet und der PROD (Produkt)-Operator, welcher die Ausgangsfunktion reskaliert [vgl. MathWorks, 2004, S. 2-24].

Wenn man die Regel

if  $u$  is  $A$  then  $v$  is  $C$

betrachtet (siehe Abbildung 8.3), wobei  $A$  und  $C$  die Fuzzy-Sets für die Bedingung (antecedent) beziehungsweise den Aktionsteil (Schlussfolgerungsteil, consequent) darstellen und  $u \in U$  und  $v \in V$  Werte aus dem Eingangs- beziehungsweise Ausgangsuniversum sind, dann sind die beiden bekanntesten Funktionen für die Implikation definiert als:

- Minimum  $C' = \min(\mu_A(x), C)$
- Produkt  $C' = \mu_A(x) * C$

Die Implikation  $A \rightarrow C$  für einen Wert  $x$  ergibt sich zu dem Fuzzy-Set  $C' = \mu_{A \rightarrow C}(x)$ , welches auf dem Ausgangsuniversum  $V$  definiert ist.

Eine Regel, die nicht feuert, also wo die Folgerung absolut falsch ist ( $\mu_A(x)=0$ ), ergibt eine leere Implikation. Diese Regel nimmt auch nicht mehr an der Inferenz des Systems teil [vgl. Pena, 2004, S. 10].

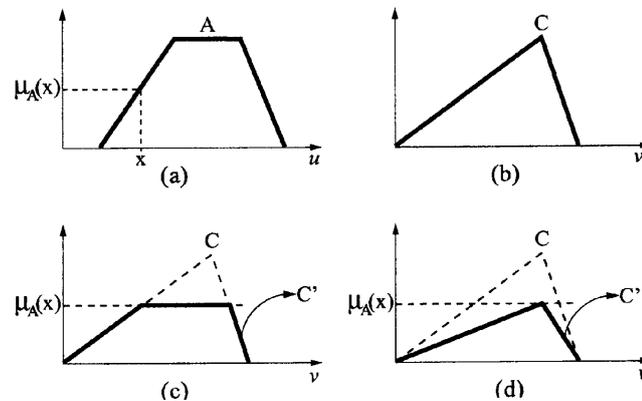


Abbildung 8.3: Bekannte Implikationsoperatoren bei gegebener Regel „if u is A then v is C“:

- (a) Fuzzy-Eingangsvariable A mit einem gegebenen Element x
- (b) Fuzzy-Ausgangsgröße C der Regel
- (c) Minimum-Implikation von  $C' = \mu_{A \rightarrow C}(x)$
- (d) Produkt-Implikation von  $C' = \mu_{A \rightarrow C}(x)$  [Pena, 2004, S. 11]

#### 8.2.5.4 Aggregation der Ausgangsgrößen

Alle resultierenden Ausgangsgrößen müssen nun zu einem einzigen Fuzzy-Set zusammengefasst werden. Hierfür dient die Aggregation, welche alle Ausgangsgrößen, repräsentiert durch jeweils ein eigenes Fuzzy-Set, zu einem einzigen resultierenden Fuzzy-Set zusammenfasst. Die Aggregationsfunktion sollte jedenfalls kommutativ sein, damit die Ausgangsgrößen unabhängig von ihrer Reihenfolge immer zum gleichen Resultat führen. Beispiele für Aggregationsfunktionen sind MAX (Maximum), PROBOR (Probabilistisches ODER) und SUM (Summe) [vgl. MathWorks, 2004, S. 2-24-2-25].

#### 8.2.5.5 Defuzzifizierung der Ausgangsgrößen

Damit aus den unscharfen Ausgangsmengen für die einzelnen Ausgangsgrößen scharfe Werte, zum Beispiel Einstellwerte oder ja/nein-Entscheidungen, erzeugt werden können, müssen diese wieder defuzzifiziert werden. Hierzu wird oft die Fläche unter den Kurvenverläufen der Zugehörigkeitsfunktionen herangezogen. Häufig wird der Abszissenwert des Flächenschwerpunkts als resultierende scharfe Ausgangsgröße gewählt. Dieser Vorgang wird als „Defuzzifizierung der Ausgangsgröße“ bezeichnet [vgl. Bothe, 1995, S. 12–13].

## 8.3 Spezielle Konzepte der Fuzzy-Logik

### 8.3.1 Zugehörigkeitsfunktionen

#### 8.3.1.1 Grundlagen der Zugehörigkeitsfunktionen

Eine Zugehörigkeitsfunktion  $\mu(x)$  sollte die folgenden Eigenschaften haben:

- $\mu(x) \in X: \mu(x) \geq 0$ ,
- $\mu(x)$  soll umso größer sein, je besser  $x$  das Bewertungskriterium des Experten erfüllt.

Der Wertebereich der bei abgeschlossenen Problemstellungen auftretenden Zugehörigkeitsfunktion  $\mu(x)$  sollte, muss aber nicht, normalisiert werden. Hierfür stehen auch die unterschiedlichsten Transformationen zur Verfügung, wie zum Beispiel die folgenden:

- Einheitsintervall-Normierungen für  $\mu(x) \rightarrow [0,1]$ ,
- Normalisierung auf ein Bezugselement  $x_0$  mit  $\mu_A(x_0) = 1$ .

Zur Bestimmung der Zugehörigkeit eines Elements zu einer Menge liegt immer ein definiertes Bewertungskriterium zugrunde. Je nach Genauigkeit des Bewertungskriteriums ergibt sich die Schärfe der Menge [vgl. Bothe, 1995, S. 27].

#### 8.3.1.2 Konstruktion einer Zugehörigkeitsfunktion

Die Festlegung von Zugehörigkeitsfunktionen kann auf zwei unterschiedliche Arten erfolgen. In beiden Fällen werden die Elemente  $x \in X$  bezüglich eines festgelegten unscharfen Kriteriums von einem Experten nach Zugehörigkeit bewertet.

- unbestimmte Methode (siehe Abbildung 8.4)

Ausgehend von einer scharfen Teilmenge  $A' \subseteq X$  wird angenommen, dass die Zugehörigkeit der Elemente am Rand der Menge  $A$  geringer als die der Elemente im Zentrum der Menge  $A$  ist und die Randelemente in außerhalb liegende Nachbarmengen ausstrahlen und hier ebenfalls eine gewisse Zugehörigkeit haben. Zur Modellierung von  $\mu_A(x)$  kann zuerst ein Randniveau  $\alpha$  (oder auch ein linkes Randniveau  $\alpha$  und ein rechtes  $\beta$ ) festgelegt werden, welches am linken und am rechten Rand erreicht wird. Der Verlauf von  $\mu_A(x)$  wird mit Hilfe von bekannten und geschätzten Systemeigenschaften vom Experten modelliert [vgl. Bothe, 1995, S. 27–28].

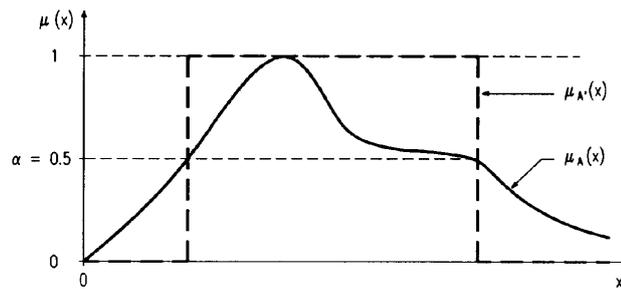


Abbildung 8.4: Ausgehend von einer geschätzten scharfen Menge  $A$  kann eine realistische Zugehörigkeitsfunktion  $\mu_A(x)$  dadurch konstruiert werden, dass Nachbarschaftsbeziehungen zwischen den einzelnen Elementen  $x$  den scharfen Kurvenverlauf deformieren [Bothe, 1995, S. 28]

- bestimmte Methode (siehe Abbildung 8.5)

Alle Elemente beziehungsweise Objekte  $x \in X$  sind a priori bekannt. Zum Beispiel kennt man die Eigenschaften des Systems für bestimmte Werte  $x \in X$  und hat somit einen diskretisierten Verlauf der Zugehörigkeitsfunktion  $\mu_A(x)$  vorgegeben. Falls die nicht bekannten Zwischenwerte  $\mu_A(x)$  bei  $x \in X$  benötigt werden, so können diese interpoliert werden [vgl. Bothe, 1995, S. 28].

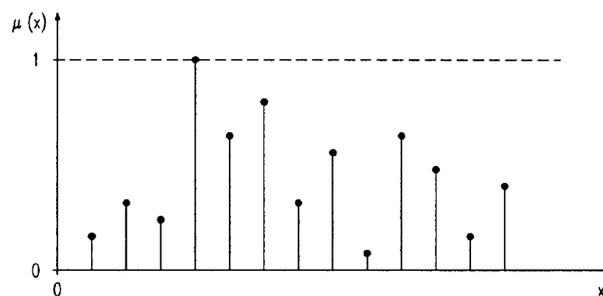


Abbildung 8.5: Konstruktion einer Zugehörigkeitsfunktion durch eine repräsentative Auswahl diskreter Elemente  $x$ , welche bekannt sind und direkt bewertet werden können [Bothe, 1995, S. 28]

### 8.3.1.3 Verschiedene Zugehörigkeitsfunktionen

Wenn die Zugehörigkeitsfunktionen von einem Experten festgelegt worden sind, dann müssen diese auch im Computer gespeichert werden. Würde man nun eine Kurve aus den Beispielen des vorangehenden Kapitels speichern wollen, so müsste man den gesamten Verlauf punktweise speichern, je genauer die Kurve eingehalten werden soll, desto mehr Datenpunkte müssen gespeichert werden, um den Kurvenverlauf wiederzugeben. Nun lassen sich aber sehr viele Kurvenverläufe durch bekannte Formen und Formeln darstellen, was den Vorteil hätte, dass man nur die Bezeichnung der Kurvenform und deren Parameter speichern müsste.

Im Folgenden sind einige Kurvenverläufe von Zugehörigkeitsfunktionen aus MATLAB (© The MathWorks, Inc.) dargestellt:

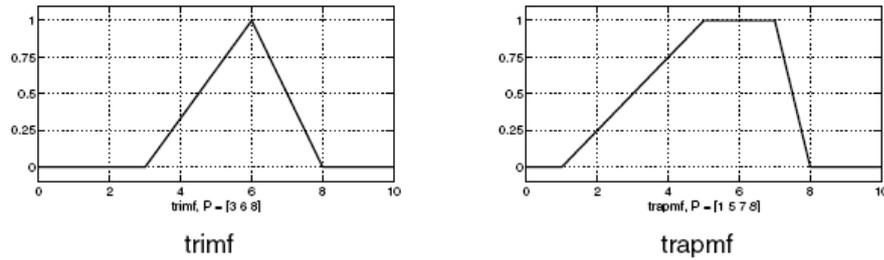


Abbildung 8.6: Dreieck- und Trapezoidalfunktion aus MATLAB [MathWorks, 2004, S. 2-10]

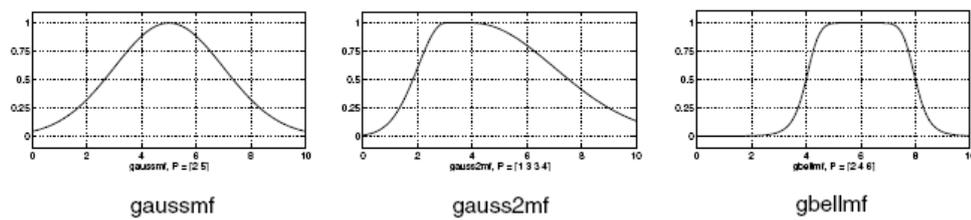


Abbildung 8.7: Gauß'sche Glockenkurve, zwei überlagerte Gauß'sche Glockenkurven und verallgemeinerte Glockenkurve aus MATLAB [MathWorks, 2004, S. 2-11]

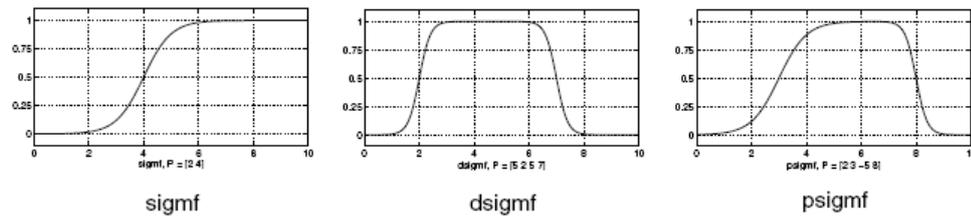


Abbildung 8.8: Sigmoidale Kurve, Differenz und Produkt zweier sigmoidaler Kurven aus MATLAB [MathWorks, 2004, S. 2-11]

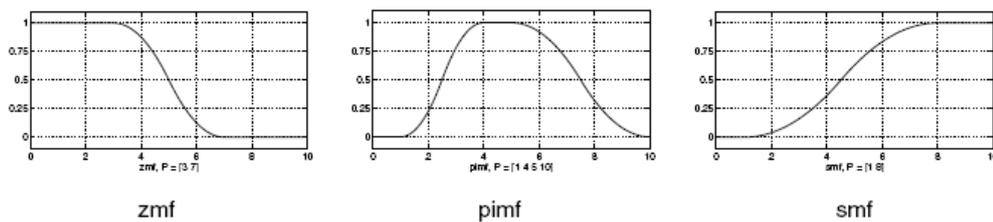


Abbildung 8.9: Z-, Pi- und S-Kurve als Beispiele polynomieller Kurven aus MATLAB [MathWorks, 2004, S. 2-13]

## 8.3.2 Logische Operatoren

### 8.3.2.1 Fuzzy-UND, -ODER und -Komplement

In Kapitel 5.3.3.3 wurden logische Operatoren für eine zweiwertige Logik vorgestellt. Es gilt nun, diese zu erweitern, sodass sie auch für die Fuzzy-Logik verwendet werden können. In der

Abbildung 8.10 sind oben die Graphen für eine zweiwertige Logik mit den Zuständen 0 und 1 (falsch und wahr) und unten für einen kontinuierlichen Wertebereich dargestellt.

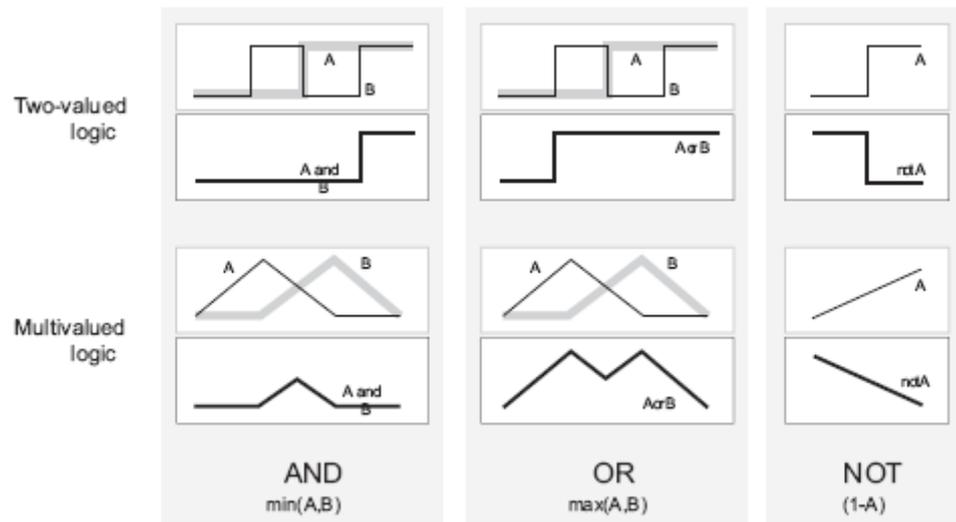


Abbildung 8.10: Logische Operatoren in zwei- und mehrwertiger Logik [MathWorks, 2004, S. 2-14]

Diese Fuzzy-Operatoren sind aber keineswegs so definiert, sondern spiegeln nur den Konsens wider, wie er in den meisten Anwendungsfällen Verwendung findet. Dies sind für das Fuzzy-UND (Konjunktion, Durchschnitt) das Minimum, für das Fuzzy-ODER (Disjunktion, Vereinigung) das Maximum und für das Fuzzy-Komplement das additive Komplement [vgl. MathWorks, 2004, S. 2-14].

### 8.3.2.2 t-Norm und s-Norm

Der Durchschnitt (Fuzzy-UND) und die Vereinigung (Fuzzy-ODER) können aber auch allgemeiner betrachtet werden.

Eine zweistellige Funktion  $t: [0,1] \times [0,1] \rightarrow [0,1]$  heißt t-Norm, wenn  $\forall x \in X$  die folgenden Bedingungen gelten:

- $t(0,0) = 0$  und  $t(\mu_A, 1) = t(1, \mu_A) = \mu_A$  (Begrenztheit),
- $t(\mu_A, \mu_B) \leq t(\mu_C, \mu_D)$ ,  $\forall \mu_A \leq \mu_C, \mu_B \leq \mu_D$  (Monotonie),
- $t(\mu_A, \mu_B) = t(\mu_B, \mu_A)$  (Kommutativität),
- $t(\mu_A, t(\mu_B, \mu_C)) = t(t(\mu_A, \mu_B), \mu_C)$  (Assoziativität) [vgl. Bothe, 1995, S. 42].

Die bekanntesten t-Norm-Operatoren sind *Minimum*, *Produkt* und *Beschränktes Produkt*, welche wie folgt definiert sind:

- Minimum:  $\min(\mu_A(x), \mu_B(x))$
- Produkt:  $\mu_A(x) * \mu_B(x)$
- Beschränktes Produkt:  $\max(0, \mu_A(x) + \mu_B(x) - 1)$

Die folgende Abbildung visualisiert die gebräuchlichsten t-Norm-Operatoren [vgl. Pena, 2004, S. 7].

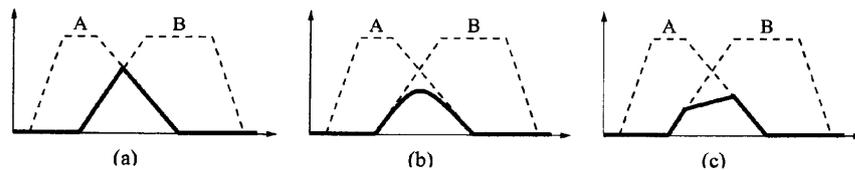


Abbildung 8.11: Gebräuchliche t-Norm Operatoren: (a) Minimum, (b) Produkt und (c) Begrenztes Produkt [Pena, 2004, S. 8]

Eine zweistellige Funktion  $t: [0,1] \times [0,1] \rightarrow [0,1]$  heißt s-Norm (t-Conorm), wenn  $\forall x \in X$  die folgenden Bedingungen gelten:

- $s(1,1) = 1$  und  $t(\mu_A, 0) = t(0, \mu_A) = \mu_A$  (Begrenztheit),
- $s(\mu_A, \mu_B) \leq s(\mu_C, \mu_D)$ ,  $\forall \mu_A \leq \mu_C, \mu_B \leq \mu_D$  (Monotonie),
- $s(\mu_A, \mu_B) = s(\mu_B, \mu_A)$  (Kommutativität),
- $s(\mu_A, s(\mu_B, \mu_C)) = s(s(\mu_A, \mu_B), \mu_C)$  (Assoziativität) [vgl. Bothe, 1995, S. 42].

Die bekanntesten s-Norm-Operatoren sind *Maximum*, *Probabilistische Summe* und *Beschränkte Summe*, welche wie folgt definiert sind:

- Maximum:  $\max(\mu_A(x), \mu_B(x))$
- Probabilistische Summe:  $\mu_A(x) + \mu_B(x) - \mu_A(x) * \mu_B(x)$
- Beschränktes Produkt:  $\min(1, \mu_A(x) + \mu_B(x))$

Die folgende Abbildung visualisiert die gebräuchlichsten s-Norm-Operatoren [vgl. Pena, 2004, S. 8].

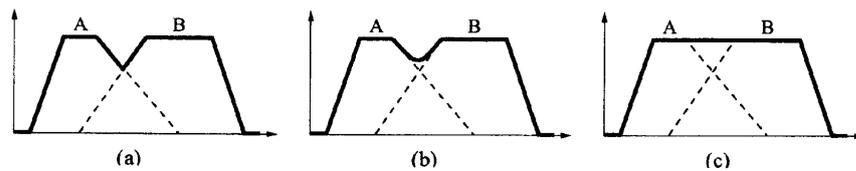


Abbildung 8.12: Gebräuchliche s-Norm Operatoren: (a) Maximum, (b) Probabilistische Summe und (c) Beschränkte Summe [Pena, 2004, S. 8]

### 8.3.3 Defuzzifizierungsmethoden

Es sind sehr viele verschiedene Methoden verfügbar, welche aus den unscharfen Ausgangsgrößen scharfe Stellwerte berechnen. In [Bothe, 1995, S. 146–147] werden zum Beispiel die Max-Methode, Links- und Rechts-Max-Methode, Mittelwert-Max-Methode, Centroiden-Methode, Modifizierte-Centroiden-Methode und Lineare DeFuzzifizierung genannt, die meisten davon sind in MATLAB [MathWorks, 2006] implementiert.

An dieser Stelle sollen die beiden gebräuchlichsten Methoden, die Centroiden-Methode (Center of Areas, COA) und die Mittelwert-Max-Methode (Mean of Maxima, MOM), näher beschrieben werden.

#### 8.3.3.1 Center of Area

Als Ausgangswert wird der Abszissenwert des Schwerpunktes der Fläche unterhalb der Zugehörigkeitsfunktion der Ausgangsgröße  $Y = \mu_Y(v)$  im Universum  $V$  der Variablen  $v$  berechnet:

$$y_{COA} = \frac{\int v \cdot \mu_Y(v) dv}{\int \mu_Y(v) dv}$$

Wenn das Ausgangsuniversum diskret ist, müssen die Integrale nur durch Summen ersetzt werden [vgl. Pena, 2004, S. 11].

#### 8.3.3.2 Mean of Maxima

Die Methode Mean of Maxima ergibt den mittleren Abszissenwert des Maximum-Plateaus der Zugehörigkeitsfunktion. Diesen kann man folgendermaßen berechnen [vgl. Pena, 2004, S. 11]:

$$y_{\inf} = \min(z \mid \mu_Y(z) = \max(\mu_Y(v)))$$

$$y_{\sup} = \max(z \mid \mu_Y(z) = \max(\mu_Y(v)))$$

$$y_{MOM} = \frac{y_{\text{inf}} + y_{\text{sup}}}{2}$$

### 8.3.4 Fuzzy-Systeme

Es existieren zur Zeit drei verbreitete Typen von Fuzzy-Systemen, welche sich darin unterscheiden, wie die Auswirkungen der Regeln definiert sind: Mamdani-, TSK- und Singleton-Fuzzy-Systeme [vgl. Pena, 2004, S. 15].

#### 8.3.4.1 Mamdani-Fuzzy-Systeme

Diese Fuzzy-Systeme haben Fuzzy-Sets als Ausgangsgrößen, wie sie bis jetzt beschrieben wurden. Sie werden Mamdani Fuzzy-Systeme genannt, da sie erstmals von Mamdani eingeführt wurden [vgl. Pena, 2004, S. 15–16].

Mamdani-Systeme sind in MATLAB implementiert und haben folgende Vorteile:

- hohe Intuitivität durch Verwendung von linguistischen Variablen
- hohe Akzeptanz
- gute Verarbeitbarkeit von menschlichen Angaben [vgl. MathWorks, 2004, S. 2-78]

#### 8.3.4.2 TSK-Fuzzy-Systeme

Diese Systeme leiten ihren Namen von Takagi, Sugeno und Kang ab, welche einen alternativen Typ von Fuzzy-Systemen entwickelt haben. Die Ausgangsgröße ist hier als Funktion definiert, welche üblicherweise linear ist. Dies wirkt sich vorteilhaft auf die Genauigkeit des Systems aus, die im Allgemeinen höher als die von Mamdani-Systemen ist. Die Interpretierbarkeit der Ergebnisse ist signifikant reduziert, da diese nicht länger linguistische Konzepte repräsentieren [vgl. Pena, 2004, S. 16].

Im Gegensatz zu den bisher behandelten Regeln lassen sich die Regeln folgendermaßen darstellen:

$$\text{if } u = A_i \text{ then } y_i = s_{0,i} + s_{1,i} \cdot x_{IN},$$

wobei  $s_0$  und  $s_1$  die Parameter des linearen Gleichungssystems sind und  $x_{IN}$  der Eingangswert.

Am Ausgang werden also direkt numerische Werte produziert, die Linearkombinationen der Eingangswerte sind. Die Aktivierungsgrade  $\beta_i$  der  $i$  Regeln werden berechnet durch die Fuzzifizierung der Eingangswerte mit

$$\beta_i = \mu_{A_i}(x_0).$$

Die Teilergebnisse der einzelnen Regeln sind bereits im Konsequenzteil der Formel gegeben mit

$$y_i = \sum_k (s_{k,i} \cdot x_k),$$

wobei der Laufindex  $k$  die Anzahl der zu berücksichtigenden Eingangsgrößen im Bedingungsteil der Regel ist. Das Gesamtergebnis ergibt sich, wie bei der hier nicht betrachteten Tsukamoto-Inferenz-Methode (siehe [Bothe, 1995, S. 152–153]), durch gewichtete Mittelwertbildung mit der Formel

$$y_{OUT} = \frac{\sum_i (\beta_i \cdot k_i)}{\sum_i (\beta_i)}.$$

Das Ergebnis liegt hier bereits numerisch vor und muss nicht erst defuzzifiziert werden, um eine scharfe Stellgröße zu erhalten [vgl. Bothe, 1995, S. 154].

In MATLAB sind Sugeno-(Takagi-Sugeno-Kang)-Methoden (in [Bothe, 1995, S. 153] wird auch von der Sugeno-Methode gesprochen und Takagi und Kang nicht erwähnt) implementiert. Diese weisen folgende Vorteile auf:

- effiziente Berechnung
- arbeiten sehr gut mit linearen Techniken (z. B. Regler)
- lassen sich gut adaptieren und optimieren (zum Beispiel mit Hilfe von genetischen Algorithmen oder in einem Lernprozess mittels neuronaler Netze [vgl. Bothe, 1995, S. 154])
- garantierte Stetigkeit der Ausgangsgrößen
- gute mathematische Analysierbarkeit [vgl. MathWorks, 2004, S. 2-78].

### 8.3.4.3 Singleton-Fuzzy-Systeme

Bei den Singleton-Fuzzy-Systemen sind die Ausgangsgrößen konstante Werte. Diese können als Spezialfall sowohl der Mamdani- als auch der TSK-Systeme aufgefasst werden. Die Singleton-Fuzzy-Systeme stellen einen Kompromiss zwischen der Interpretierbarkeit der Mamdani-Systeme mit den aussagekräftigen Variablennamen und der Genauigkeit der TSK-Systeme mit den linearen Funktionen dar [vgl. Pena, 2004, S. 17].

## 9 Projektbeschreibung

FuzzyDBSexpert dient dazu, eine Unterstützung des Arztes beziehungsweise eine Nachvollziehbarkeit von Ergebnissen für den Patienten, für folgende Entscheidungen zu bieten:

- Entscheidung, ob eine Operation für eine Tiefe Hirnstimulation bei einem Patienten durchgeführt werden soll oder nicht, auf der Basis von Kriterien, die von den medizinischen Experten als relevant betrachtet werden
- Entscheidung über den Implantationsort aufgrund von Kriterien, die von den medizinischen Experten vorgegeben werden

Hierfür werden nicht nur die Kriterien der UPDRS, sondern auch andere für die Entscheidung relevante Patientendaten miteinbezogen. Als Ergebnis wird eine Aussage über den Grad der Eignung für eine Tiefe Hirnstimulation getroffen, die jedoch nur eine Hilfestellung für den Arzt darstellen soll und keinesfalls eine ihm vorweggenommene Entscheidung.

### 9.1 Realisierungsmodell

Die Entscheidung über die Realisierungsform von FuzzyDBSexpert hinsichtlich der Programmausführung fiel auf ein webbasiertes System mit Datenbankbindung, da dies folgende Vorteile bietet:

- sofortige weltweite Verfügbarkeit, sobald das Programm der Öffentlichkeit zugänglich gemacht wird
- keine Notwendigkeit der Installation von Software auf dem Client, was die Hemmschwelle zum Testen der Software, aber auch die Anzahl der möglichen clientseitigen Probleme herabsetzt
- bei Verwendung von plattformübergreifenden Programmiersprachen, wie es für ein webbasiertes System üblich ist, ist die Software auf unterschiedlichen Computerplattformen mit verschiedenen Betriebssystemen lauffähig
- bei Entwicklung einer neuen Version des Programmes kann diese sofort nach Installation weltweit verfügbar gemacht werden
- die Bearbeitung und Erweiterung der Wissensbasis ist weltweit von autorisierten Experten möglich, was eine Konsultierung der Experten vor Ort oder eine komplizierte Kommunikation hinfällig macht

Es ist natürlich möglich, die Software in einem LAN (local area network, Lokales Netzwerk) oder auf einem Client laufen zu lassen. Dazu sind ein Webserver mit Servlet/JSP-Engine, welche für JSF konfiguriert werden kann und eine Datenbank notwendig, welche nicht immer in einem Betrieb zur Verfügung gestellt werden können, da es zu Problemen mit Berechtigungen oder fehlender Infrastruktur kommen kann. Privat wäre dies kein Problem, es ist aber ein für einen Privatanwender hoher Installations- und Konfigurationsaufwand und ein gewisser Speicherplatzbedarf notwendig.

Die Realisierung von FuzzyDBSexpert als webbasiertes Programm bringt aber auch folgende Probleme mit sich:

- Übermittlung und Speicherung von sensiblen Daten im Internet

Die Übermittlung und Speicherung von sensiblen Daten im Internet ist sehr problematisch, da sehr viele Webserver ständigen Attacken ausgesetzt sind und eine absolute Sicherheit nicht garantiert werden kann. Außerdem sieht das Datenschutzgesetz sehr aufwändige Sicherheitsvorkehrungen für die Übermittlung und Speicherung von personenbezogenen Daten vor, die Realisierung dieser Sicherheitsvorkehrungen würde den Rahmen dieses Projektes sprengen. Da personenbezogene Daten nicht unbedingt für die Funktion von FuzzyDBSexpert notwendig sind, wird im Rahmen dieses Projektes auf die Übermittlung und Speicherung von personenbezogenen Daten und somit auf die Realisierung komplizierter Sicherheitsmechanismen verzichtet. Die eingegebenen Daten können dann indirekt personenbezogen gespeichert werden. Das bedeutet, dass der betreffende Patient einen Identifikationscode erhält, welcher aber keinen Aufschluss auf die Person seitens der Öffentlichkeit geben darf, also zum Beispiel nicht aus der Sozialversicherungsnummer oder dem Namen in Kombination mit dem Geburtsdatum bestehen darf.

- Sicherung der Daten

Damit die Daten im Falle eines Serverproblems gesichert sind und nicht zur Gänze verloren gehen können, müssen sie gesichert werden. Wird mit dem Serviceprovider ein entsprechender Vertrag abgeschlossen, liegt die Verantwortung über die Sicherheit und Verfügbarkeit der Daten in dessen Bereich.

## 9.2 Prinzipielle Struktur

Die prinzipielle Struktur von FuzzyDBSexpert (siehe Abbildung 9.1) entspricht der eines wissensbasierten Systems, dessen Benutzerschnittstelle und Expertenschnittstelle über das Internet transportiert werden.

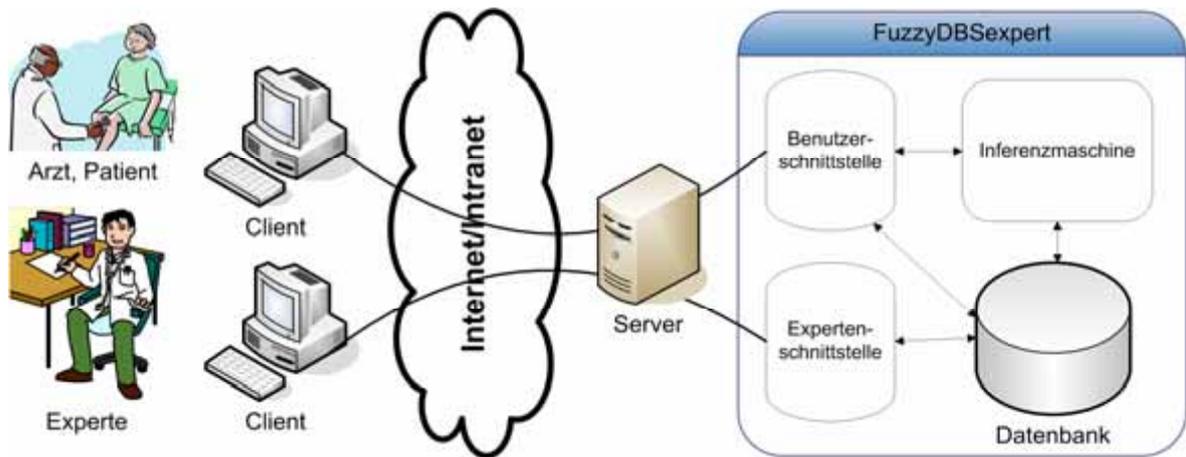


Abbildung 9.1: Struktur von FuzzyDBSexpert

## 9.2.1 Benutzerschnittstelle

Die Benutzerschnittstelle stellt die Schnittstelle zwischen dem Benutzer, beispielsweise dem Arzt oder Patienten, und FuzzyDBSexpert dar. Sie dient im Wesentlichen zur Eingabe der Patientendaten und nach der Ermittlung der Eignung des Patienten für die Tiefe Hirnstimulation zur Ausgabe der Ergebnisse.

### 9.2.1.1 Dateneingabe

Bei der Eingabe werden hauptsächlich die Fragen der UPDRS (siehe Kapitel 3) zu beantworten sein, dies stellt den größten Anteil an der Dateneingabe dar. Für dieses Projekt sollen die Fragen und Antworten der UPDRS beibehalten und in vollem Umfang übernommen werden, da die UPDRS in der Wissenschaft schon gut etabliert und international anerkannt ist. Sehr viele Studien, die Parkinson betreffen, verwenden den UPDRS-Score-Wert, um einen Vorher-Nachher-Vergleich anzustellen. Es ergibt sich daraus der zusätzliche Nutzen, die Antworten durch Ankreuzen am Computer auswählen und diese nachher zum Beispiel ausdrucken oder in einen Arztbrief kopieren zu können. Nicht zuletzt wird der Score-Wert automatisch berechnet. Bei der Verwendung eines Formulars auf Papier muss die Score-Wert-Bildung manuell erfolgen, daher sind Rechenfehler möglich, was für eine weitere Überprüfung durch nochmalige Berechnung zusätzlich Zeit in Anspruch nimmt. Die Ermittlung des herkömmlichen UPDRS-Score-Wertes stellt einen zusätzlichen Nutzen für die langwierige Dateneingabe dar, im Hintergrund werden für die eigentliche Aufgabe des FuzzyDBSexpert die Antworten gewichtet und dann weiter verarbeitet.

Andere Parameter, welche am Eingabeschirm abgefragt werden, können, je nach Expertenmeinung, zum Beispiel das Patientenalter oder die Dauer der Krankheit sein. Der Patienteneingabeschirm ist intuitiv, wie die im Internet verfügbaren UPDRS-Tabellen (siehe [MDVU, 2007] und [MünchenIsar, 2007]), aufgebaut, also in Form einer Liste von Fragen mit Antworten, welche mittels einander ausschließenden Optionsschaltflächen angewählt werden können. Es kann

natürlich dazu kommen, dass einige Patientendaten nicht ermittelt werden können oder nicht bekannt sind. Trotzdem liefert FuzzyDBSexpert ein Ergebnis. Dies ist wegen des regelbasierten Ansatzes einfach möglich, da die Regeln, welche die fehlenden Daten benötigen, einfach nicht feuern.

### **9.2.1.2 Datenausgabe**

Wie im Kapitel 3 schon angesprochen wurde, kann wahlweise auch die herkömmliche UPDRS-Tabelle mit berechnetem Score-Wert angezeigt, ausgedruckt und kopiert werden.

Die Hauptaufgabe von FuzzyDBSexpert soll die Ermittlung und Ausgabe des Grades der Eignung des Patienten für eine Tiefe Hirnstimulation sein. Prinzipiell wird hier ein eigener Score-Wert ausgegeben, welcher zwischen 0 und 1 liegt, wobei 0 bedeutet, dass der Patient überhaupt nicht für eine Tiefe Hirnstimulation geeignet ist und 1, dass der Patient optimal für eine Tiefe Hirnstimulation geeignet ist. Da in der Medizin hauptsächlich ganzzahlige Werte als Score-Werte verwendet werden, könnte man den Bereich auch von 0 bis 100 definieren, dies würde nur eine Änderung der Fuzzy-Variable und deren Fuzzy-Sets notwendig machen.

## **9.2.2 Expertenschnittstelle**

Die Expertenschnittstelle dient dazu, dass jeder Experte, auch wenn er keine Programmiersprache beherrscht, die Wissensbasis verändern und erweitern kann und zwar jenen Teil, für den er autorisiert ist. Dies wird mit Hilfe einer Authentifizierung mittels Benutzernamen und Passwort des jeweiligen Experten erreicht. Dieser Ansatz ermöglicht es dem Experten, auch ohne prüfende Instanz eines Wissensingenieurs, die Wissensbasis zu verändern, wenn auch nur jene Teile, für die er autorisiert ist. Hier könnte es zu unerwarteten Ergebnissen und Problemen kommen, wenn auch die Fuzzy-Logik sehr viele Fehler und Inkonsistenzen verzeiht.

Die Experten müssen sich ausreichend Gedanken bei der Veränderung beziehungsweise Erstellung der Wissensbasis machen, was bei dem für die breite Öffentlichkeit eher neuen und auch für den Wissensingenieur umfangreichen Gebiet der Fuzzy-Logik sicher etwas schwieriger ist, als bei einem System, welches nur auf einfachen WENN-DANN-Regeln beruht. Es wäre anzuraten, bei Erstellung einer neuen Wissensbasis oder bei geplanten Veränderungen der bestehenden Wissensbasis ein Prototyping auf dem Papier oder besser in einem externen Programm wie zum Beispiel MATLAB [MathWorks, 2006] durchzuführen, damit man sich die Ergebnisse der Inferenz besser vorstellen kann. Dies macht es natürlich sehr schwierig für Experten, welche nicht mit der Fuzzy-Logik und verfügbaren Tools vertraut sind, selbst die Wissensbasis zu erweitern, sie werden auf die Hilfe von Wissensingenieuren angewiesen sein, welche die Vorgaben der Experten in Fuzzy-Logik formalisieren, prototypisch testen und in der Wissensbasis implementieren können. Das

Testen kann natürlich wieder von den Experten selbst übernommen werden. Jeder Experte kann seine eigenen Wissensbasen im System speichern und gegebenenfalls anderen Benutzern zugänglich machen.

## 9.2.3 Wissensverarbeitung

In diesem Kapitel wird auf die prinzipielle Verarbeitung des Wissens, wie es in FuzzyDBSexpert geschieht, eingegangen. Wie in Kapitel 9.2.1.1 beschrieben, sind nicht nur die Antworten zu den Fragen der UPDRS, sondern auch Daten, wie das Patientenalter oder die Dauer der Krankheit, Eingangsgrößen.

### 9.2.3.1 Gewichtung der UPDRS-Antworten

Da die Gewichtung der Fragen, wie in Kapitel 3.2 schon beschrieben, für den einen oder anderen Experten unzulänglich erscheinen kann, wird auf dieser Ebene eine Gewichtung eingeführt. Das bedeutet, dass die einzelnen Fragen der UPDRS ein unterschiedliches Gewicht zur Bildung des Teil-Score-Wertes einer der vier Teilbereiche der UPDRS (Kognitive Funktionen, Verhalten, Stimmung; Aktivitäten des täglichen Lebens; Untersuchung der Motorik; Komplikationen während der Therapie) hat. Es werden aus den gewichteten Einzelfragen vier einzelne Score-Werte entstehen, jeweils einer für die verschiedenen Teile des UPDRS.

Es sollte in der Praxis nicht häufig vorkommen, dass eine oder mehrere Fragen nicht beantwortet werden, denn damit würde auch der herkömmliche UPDRS-Wert verfälscht werden und man müsste von einem Mindest-UPDRS-Wert sprechen. Es soll hier auch so gehandhabt werden, dass eine Frage, welche nicht beantwortet wird, vom Benutzer als 0 eingegeben werden soll, das bedeutet, dass keine Einschränkungen für den Patienten in dem betreffenden Punkt gegeben sind. Es gibt in [Eitel, 2003, S. 90] die Vorgehensweise, die Gewichte der restlichen Antworten so zu verändern, dass sie insgesamt, ohne Berücksichtigung der Gewichte der Fragen der nicht bekannten Antworten, 1 ergeben. Wenn bei der UPDRS eine Frage nicht beantwortet werden kann, dann könnte dies daran liegen, dass diese Daten nicht bekannt sind. Um jedoch die Erhebung des UPDRS-Score-Wertes durchführen zu können, auf den auch viele wissenschaftliche Studien zurückgreifen, müssen die Daten des Patienten so gewissenhaft erhoben werden, dass sie vollständig sind. Die Erhebung aller dieser Daten dient nicht zuletzt einer qualitativ hochwertigen Dokumentation der Krankheit des Patienten.

Es könnte natürlich überlegt werden, ob man anstatt der Erhebung der UPDRS die ungenauere Skala nach Hoehn und Yahr (nach [ÖPG, 2003] als Einschlusskriterium definiert) und einige andere Kriterien, wie die Möglichkeit der Durchführung von repetitiven Bewegungen, die Unfähigkeit, sich im Bett zu drehen, aufzustehen, das Vorhandensein von Dyskinesien und so

weiter, als Alternative benutzt, doch erstens sind diese Fragen schon in der UPDRS enthalten und zweitens ist eine selbst erstellte Fragenliste nicht standardisiert. Wenn die Fragen nicht standardisiert sind, könnte es zu einer geringeren Akzeptanz unter den verschiedenen Experten und Benutzern kommen.

### **9.2.3.2 Ausschlusskriterien**

In der Literatur [ÖPG, 2003; Gerlach, 2003], siehe Kapitel 2.2.2.2, sind vielfältige Ausschlusskriterien für eine operative Therapie bei Morbus Parkinson definiert. Vom Arzt wird natürlich sichergestellt, dass keines der Ausschlusskriterien zutrifft, sollte sich der Patient sonst für eine Tiefe Hirnstimulation eignen. Eine Abfrage der Ausschlusskriterien hätte auch in FuzzyDBSexpert stattfinden können. Dies wäre in Form von äußerst einfachen WENN-DANN-Regeln möglich gewesen, wofür aber ein eigenes Modul erstellt werden hätte müssen, da FuzzyDBSexpert ja eigentlich auf Fuzzy-WENN-DANN-Regeln basiert. Der Aufwand hätte sich für eine unnötig komplizierte Einbringung eines solchen „Fragebogens“ nicht gerechnet. Man kann die Kontraindikationen auch in Form von Fuzzy-Variablen einbringen, welche abgefragt werden und man müsste mit der Wahl von geeigneten Fuzzy-Sets und einer geeigneten Defuzzifizierungsmethode sicherstellen, dass, wenn auch nur eine einzige Kontraindikation zutrifft, als Ergebnis die Eignung für eine Tiefe Hirnstimulation sehr schlecht sein müsste. Diese Kontraindikationen würden aber das System unnötig kompliziert machen und könnten außerdem die Genauigkeit des Ergebnisses negativ beeinflussen.

### **9.2.3.3 Fuzzy-Inferenz**

Wenn die vier Score-Werte der UPDRS berechnet wurden, werden diese fuzzifiziert und zusammen mit den anderen fuzzifizierten Eingangsgrößen, zum Beispiel dem Patientenalter oder der Krankheitsdauer, einer Fuzzy-Inferenz zugeführt. Wie in Kapitel 8.3.4 beschrieben, gibt es verschiedene Methoden der Inferenz. Sehr verbreitet ist die MAX-MIN-Methode, wo die Bedingungen mit ODER (MAX) verknüpft werden und die Implikation durch die Anwendung des UND-Operators (MIN) durchgeführt wird.

### **9.2.3.4 Defuzzifizierung der Ausgangsgrößen**

Für die Defuzzifizierung der Ausgangsgrößen kommt intuitiv die Centroiden-Methode in Frage, wie sie in Kapitel 8.3.3.1 vorgestellt wurde. Mit dieser Methode wird sich aber niemals ein Wert von 0 oder 1 für die Ausgangsgröße einstellen können, denn sonst müssten die Zugehörigkeitsfunktionen, die bei optimaler Konstellation zur Ausgangsgröße beitragen, einen Schwerpunkt genau bei 0 beziehungsweise 1 haben und somit das gleiche „Gewicht“ links und rechts von 0 beziehungsweise 1 haben, was aber in der Realität unmöglich ist.

## 9.2.4 Wissensspeicherung

Im FuzzyDBSexpert muss Wissen in verschiedenster Form gespeichert werden. Wir können fallspezifisches Wissen, also im Prinzip die Daten über einen Patienten, von dem regelhaften Wissen trennen, welches die Wissensbasis darstellt. Für die Speicherung jeglichen Wissens dient eine Datenbank, welche geradezu prädestiniert zum Speichern von Daten für webbasierte Applikationen ist.

Die Vorteile einer Datenbank für eine Webapplikation sind folgende [vgl. Kemper, 2006, S. 17–21]:

- Ermöglichung beziehungsweise Vereinfachung des Mehrbenutzerbetriebes
- Vermeidung von Datenverlust
- integriertes Sicherheitsmanagement durch Benutzerverwaltung
- geringe Entwicklungskosten für die Datenspeicherung
- Abstraktion von der physischen Ebene der Daten

## 9.3 Programmmodule

Das Programm ist, wie für ein zeitgemäßes Softwareprojekt üblich, modular aufgebaut (siehe Abbildung 9.2), damit man Komponenten zum Beispiel anderen Orts fertigen lassen kann beziehungsweise ganze Komponenten, wie etwa die Benutzerschnittstelle, leichter ausgetauscht werden können.

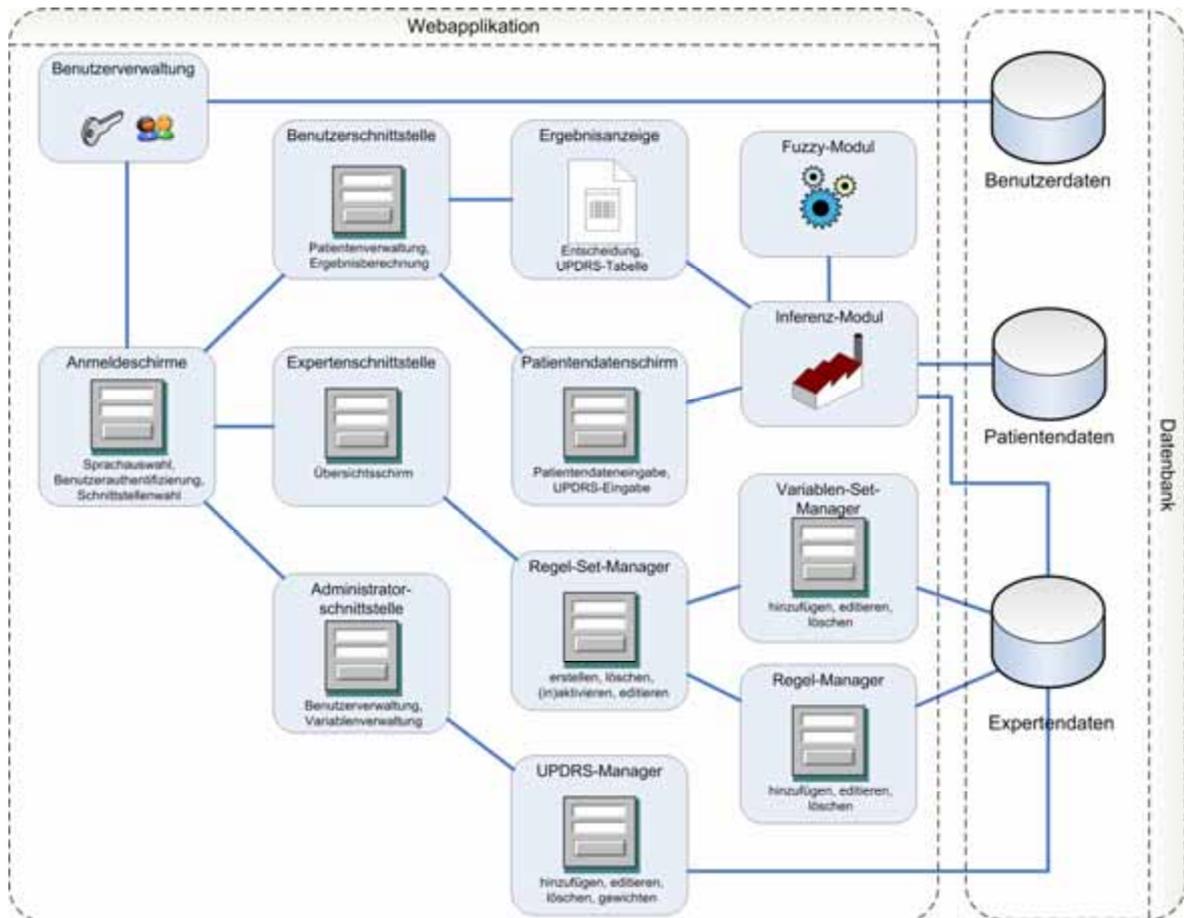


Abbildung 9.2: Schema der Programmstruktur von FuzzyDBSexpert

#### – Anmeldeschirme

Wenn Benutzer die Startseite von FuzzyDBSexpert erreichen, können sie zunächst die gewünschte Sprache wählen und sich dann an einem Bildschirm zur Authentifizierung mit Benutzername und Passwort anmelden.

#### – Benutzerschnittstelle

Die Benutzerschnittstelle dient, wie in Kapitel 9.2.1 schon beschrieben, als Schnittstelle zwischen den Benutzern und FuzzyDBSexpert. Hier werden Daten in Eingabemasken eingegeben beziehungsweise als Ausgaben präsentiert, zum Beispiel das berechnete Ergebnis oder die UPDRS-Tabelle. Die Eingabemasken und Ausgabebildschirme sind durchwegs anhand des im System gespeicherten Wissens dynamisch generiert, haben also keine feste Form, nur ein fixes Darstellungsschema.

- Expertenschnittstelle

Mit Hilfe der Expertenschnittstelle kann das relationale Wissen der Wissensbasis verändert werden, wie schon in Kapitel 9.2.2 beschrieben wurde. Auch diese Inhalte werden dynamisch anhand des in der Wissensbasis gespeicherten Wissens gestaltet.

- Administratorschnittstelle

Die Administratorschnittstelle dient einerseits hauptsächlich zur Verwaltung der Benutzer und andererseits zur Verwaltung der UPDRS.

- Benutzerverwaltung

Das Benutzerverwaltungsmodul ist für die Abwicklung der Authentifizierung der Benutzer.

- Regel-Set-Manager

Dieses Modul ist zuständig für die Verwaltung der Wissensbasis, mit dessen Hilfe man Regeln aus der Datenbank laden oder in die Datenbank speichern kann. Es setzt sich aus verschiedenen Modulen zusammen, unter anderem aus dem Variablen-Set-Manager zur Verwaltung der Fuzzy-Sets und dem Regel-Manager zur Verwaltung der Regeln.

- Patientendatenschirm

In diesen Schirmen werden die UPDRS-Antworten und auch andere Patientendaten, welche als Basis zur Berechnung dienen sollen, eingegeben.

- Inferenz-Modul

Dieses umfangreiche Modul wendet das Regelwissen auf das fallspezifische Wissen an, welches in der Datenbank gespeichert ist und führt den gesamten Inferenzprozess, angefangen von der Gewichtung der UPDRS-Antworten über die Fuzzifizierung der Eingangsgrößen, der Durchführung der Fuzzy-Inferenz bis hin zur Berechnung und Defuzzifizierung der Ausgangsgröße durch.

- Fuzzy-Modul

Es bietet sich natürlich an, die Fuzzy-Methoden in ein eigenes Modul zu fassen, sodass dieses einfach von mehreren Personen weiterverwendet oder weiterentwickelt oder eine bestehende Bibliothek verwendet werden kann und eine von der Funktion FuzzyDBSexperts komplett unabhängige Funktionalität realisiert, welche nur die „Welt“ der Fuzzy-Logik umfasst. Dieses Modul sollte möglichst leicht austauschbar sein.

## 9.4 Testverfahren

Die Ergebnisse des Systems, also des von den Experten formulierten und vom Wissensingenieur umgesetzten Wissens sollte möglichst früh einer Überprüfung auf Lieferung von korrekten Ergebnissen unterzogen werden. Hierfür bietet sich an, die Inferenzmaschine in einer Testumgebung aufzubauen, welche Fuzzy-Inferenz unterstützt, wie zum Beispiel MATLAB mit der „Fuzzy Logic Toolbox“ [MathWorks, 2006]. Wenn mit diesem oder einem anderen Tool die Ergebnisse vor der Implementierung überprüft werden, kann dies viel Zeit sparen, wenn die sich ergebenden Resultate von den erwarteten stark abweichen, denn in einer Entwicklungsumgebung für Fuzzy-Inferenzsysteme kann man mit einem Mausklick zum Beispiel die Methode für die Aggregation oder Defuzzifizierung der Ausgangsgröße oder für die Fuzzy-Implikation umstellen und das Ergebnis sofort neu berechnen lassen.

Natürlich muss man nicht die gesamte UPDRS prototypisch realisieren, es genügt, wenn man für eine repräsentative Menge an Patienten die UPDRS zum Beispiel mit Hilfe einer Tabellenkalkulation oder händisch die in Kapitel 9.2.3.1 genannte Gewichtung und Zusammenfassung der einzelnen Antworten auf die Fragen der UPDRS zu vier Score-Werten durchführt und mit diesen Ergebnissen die Inferenzmaschine speist. Auf diese Weise kann man eine Feineinstellung der Parameter vornehmen, die bei einem webbasierten System nicht so komfortabel möglich ist, da schnell und einfach die meisten Parameter verändert und die Fuzzy-Sets der Eingangs- und Ausgangsgrößen adäquat grafisch dargestellt werden können.

Erst wenn die oben angegebene Überprüfung der Ergebnisse des Fuzzy-Inferenzsystems durchgeführt wurde, sollte mit der Eingabe des Regelwissens seitens des Experten oder Wissensingenieurs begonnen werden.

# 10 Methode

Dieses Kapitel hat eine detaillierte Beschreibung von FuzzyDBSexpert, dessen Strukturen in der Datenbank und im Speicher und nicht zuletzt der Funktionsweise als Ziel. Im ersten Kapitel erfolgt eine Beschreibung der Grundlagen, welche zum Verständnis von FuzzyDBSexpert notwendig sind. Im zweiten Kapitel wird das Programm beschrieben und in den folgenden Kapiteln auf die technischen Details eingegangen.

## 10.1 Grundlagen

In diesem Kapitel wird die prinzipielle Struktur von FuzzyDBSexpert im Kontext der vorangegangenen Kapitel über Fuzzy-Logik und wissensbasierte Systeme erläutert. Dies ist sehr hilfreich zum Verständnis der nachfolgenden Kapitel über das Programm und dessen technische Realisierung.

### 10.1.1 Wissensbasis

Die Wissensbasis von FuzzyDBSexpert enthält fallspezifisches und generisches Wissen. Fallspezifisches Wissen liegt in FuzzyDBSexpert in Form von Patientendaten vor, welche zu verschiedenen Zeitpunkten erhoben werden können. Die Patientendaten setzen sich aus Variablen und der UPDRS zusammen, diese werden in Form eines Sets, dem Patientendaten-Set, gespeichert. Generisches Wissen wird in Form von Regeln, Gewichten, Fuzzy-Sets zu Fuzzy-Variablen und weiteren Details gespeichert, welche zu einem Regel-Set zusammengefasst werden.

#### 10.1.1.1 Fallspezifisches Wissen (Patientendaten)

Die Patientendaten setzen sich in FuzzyDBSexpert aus zwei Komponenten zusammen:

- Variablen
- UPDRS-Datensatz

Diese Daten sind in FuzzyDBSexpert in einem Patientendaten-Set zusammengefasst. Folgende Abbildung soll diesen Sachverhalt veranschaulichen:

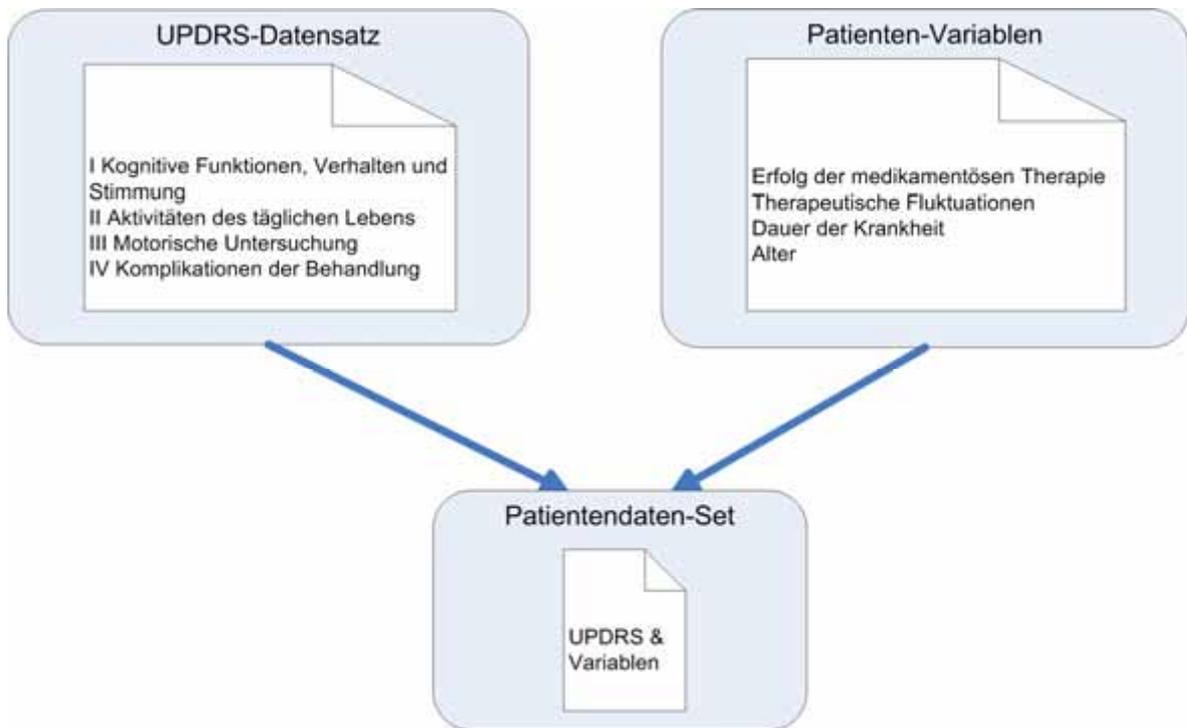


Abbildung 10.1: Patientendaten-Set

Ein Patientendaten-Set ist eine Momentaufnahme des Zustandes eines Patienten und wird dementsprechend auch mit einem Zeitstempel versehen.

#### 10.1.1.1 Variablen

Variablen sind in FuzzyDBSexpert alle Werte, welche einem Patienten zugeordnet werden können. Sinn macht es zum Beispiel, die Dauer der Krankheit oder den Erfolg der medikamentösen Therapie mit L-Dopa als Variable anzugeben. Bei allen Variablen handelt es sich um Fuzzy-Variablen, welche in FuzzyDBSexpert folgende Attribute, die für alle Benutzer gleich sind, besitzen (siehe Abbildung 10.2):

- Name
- UODLower: Die untere Grenze des „Universe of Discourse“
- UODUpper: Die obere Grenze des „Universe of Discourse“
- Einheit
- Typ (siehe Tabelle 10-1)



Abbildung 10.2: Beispiel einer Variablen in FuzzyDBSexpert

In FuzzyDBSexpert können Variablen folgendermaßen typisiert werden:

Tabelle 10-1: Variablentypen in FuzzyDBSexpert

Variablentyp	Beschreibung
Ausgangsgröße	Eine Variable, welche als Ausgangsgröße fungiert, also nur in den Schlussfolgerungsteilen von Regeln auftreten darf
Eingangsgröße	Werte, die für den Patienten erfasst werden müssen, wie zum Beispiel Alter oder Erfolg der medikamentösen Therapie
UPDRS-automatische-Eingangsgröße	Die UPDRS-automatischen-Eingangsgrößen werden aus der UPDRS als die gewichteten Summen der UPDRS-Sektionen UPDRS-I bis UPDRS-IV automatisch berechnet (siehe nächstes Kapitel)
UPDRS-Summenwert	Der UPDRS-Summenwert stellt den gewichteten Summenwert über die gesamte UPDRS dar und wird automatisch aus dem UPDRS-Datensatz berechnet. Dieser Variablentyp muss getrennt von den anderen UPDRS-Eingangsgrößen geführt werden, damit das System weiß, welcher Variable es den UPDRS-Summenwert zuweisen kann.

Im Zuge der Aufnahme der Patientendaten werden diese Variablen mit einem Wert versehen. Folgende exklusiv anzugebende Werte können Variablen zugewiesen werden:

– Zahlenwert

Ein Zahlenwert bietet sich zum Beispiel bei der Angabe des Patientenalters oder der Dauer der Krankheit an.

– linguistischer Ausdruck

Statt des Zahlenwertes kann auch ein linguistischer Ausdruck angegeben werden. Dies bietet sich vor allem bei Werten an, welche nicht einfach mit Hilfe eines Zahlenwertes repräsentiert werden können. Ein linguistischer Ausdruck kann aus den Bezeichnungen der Fuzzy-Sets der jeweiligen Fuzzy-Variable, Modifikatoren und Operatoren bestehen. Hierbei ist zu beachten, dass die Fuzzy-Sets vom gewählten Regel-Set, also vom Experten abhängen, da sie dort definiert werden. Modifikatoren (siehe Tabelle 10-2) und Operatoren (siehe Tabelle 10-3) werden in Kapitel 10.1.1.2.2 detailliert besprochen.

- kein Wert

Ein Spezialfall liegt vor, wenn keiner der beiden Werte für eine Variable angegeben werden kann, da dieser zum Beispiel unbekannt ist oder nicht erhoben werden kann. In diesem Fall wird die Variable ignoriert und die Inferenz mit den verbleibenden Variablen durchgeführt.

### 10.1.1.1.2 Patienten-UPDRS-Datensatz

Bestandteil eines Patientendaten-Sets ist auch die UPDRS, welche bereits im Kapitel 3 beschrieben wurde. Nachdem, wie schon erwähnt, die Gewichtung bei manchen Fragen ungünstig gewählt ist beziehungsweise die UPDRS in Hinsicht auf eine Bewertung der Eignung für eine Tiefe Hirnstimulation spezifisch gewichtet werden soll, wurde in FuzzyDBSexpert die Möglichkeit geschaffen, jede einzelne Frage mit einem Gewicht zu versehen (siehe Abbildung 10.3).

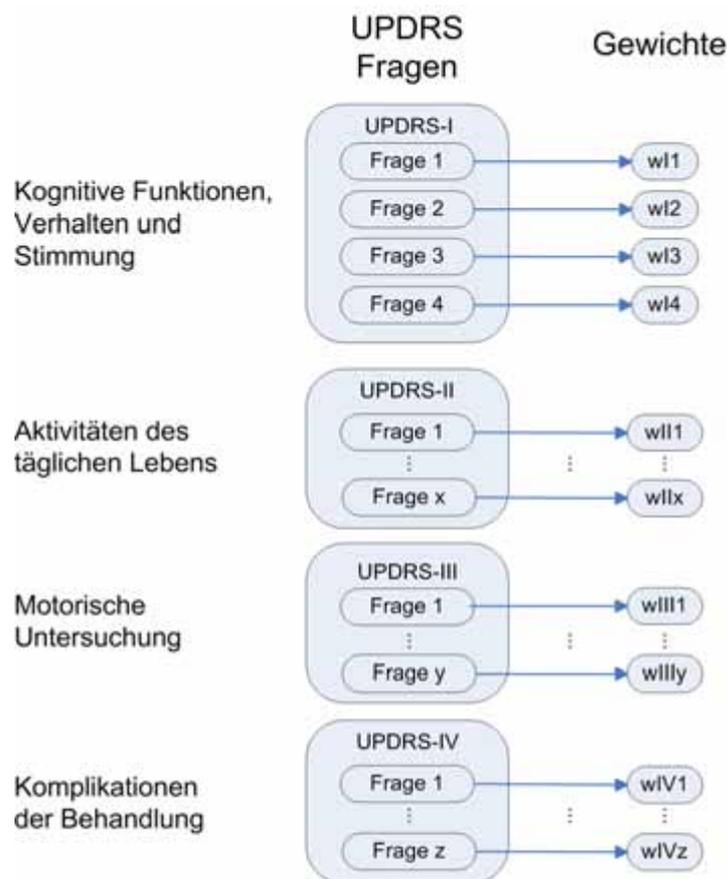


Abbildung 10.3: Patienten-UPDRS-Datensatz

Die UPDRS besteht aus vier Sektionen (siehe Kapitel 3.1). Nachdem diese vier Bereiche häufig getrennt betrachtet werden, eine Beurteilung des Patienten zum Beispiel nur nach Teil III der UPDRS erfolgen kann, wurde auch in FuzzyDBSexpert diese Trennung der Sektionen eingeführt. Dies bringt den Vorteil, dass nur, wie vorhin erläutert, eine der vier Sektionen ausgefüllt werden muss und getrennt gehandhabt werden kann.

Die Daten werden kompatibel zur standardisierten UPDRS erfasst, was den Vorteil der Vergleichbarkeit bietet. Außerdem kann die UPDRS in diesem Fall für eigene Dokumentationszwecke abgelegt werden. Zu diesem Zweck werden auch die ungewichteten Teilsommen und die einfache Gesamtsumme berechnet. Die erfassten Daten werden anschließend mit den vom Experten im jeweiligen Regel-Set angegebenen Gewichte gewichtet (siehe Abbildung 10.4) und zwar so, dass die erreichbare Gesamtsumme für alle vier Bereiche der UPDRS gleich jener der ungewichteten UPDRS bleibt:

$$score = \frac{\sum_{i=1}^n v_i^{max}}{\sum_{i=1}^n v_i^{max} \cdot w_i} \sum_{i=1}^n v_i w_i$$

$n$  ..... Anzahl der Fragen der jeweiligen Sektion  
 $v_i$  ..... Punktezahl, die der Antwort auf Frage  $i$  entspricht  
 $v_i^{max}$  ..... Maximale Punktezahl für Frage  $i$   
 $w_i$  ..... Gewicht der Frage  $i$

Durch dieses Verfahren der Gewichtung bleibt die maximale Punktezahl für jede UPDRS-Sektion konstant, was Voraussetzung für die nachfolgende Umwandlung in eine Fuzzy-Variable ist, da diese ein bestimmtes UOD haben müssen.

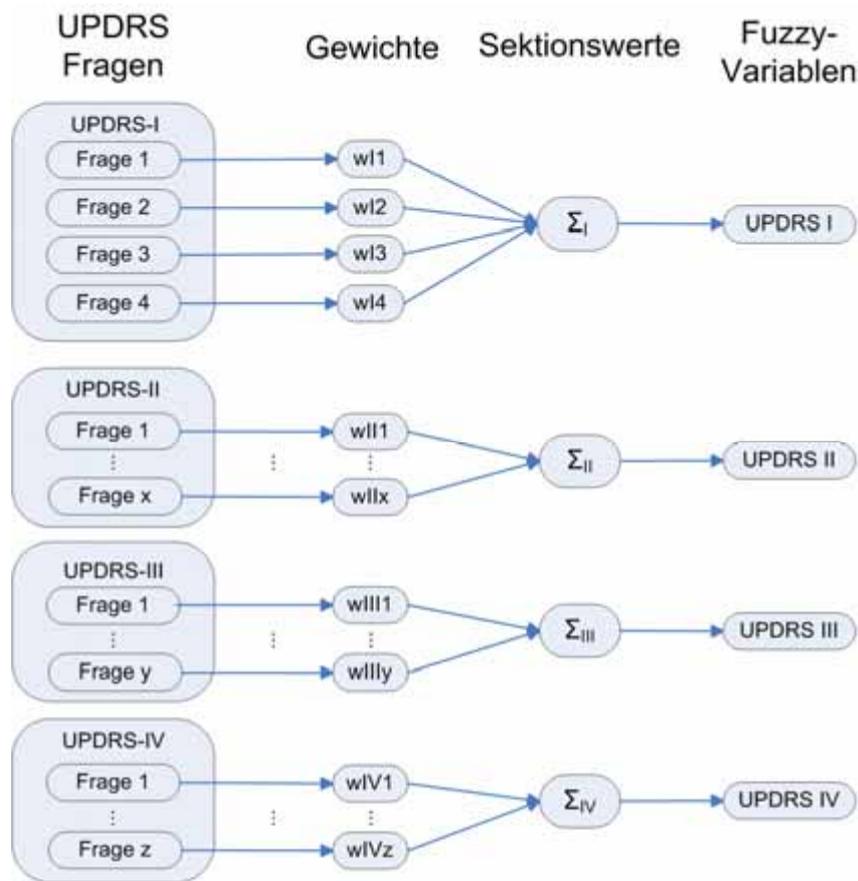


Abbildung 10.4: Gewichtung und Summenbildung der UPDRS

Die Umwandlung dieser Sektions-Scores in Fuzzy-Variablen wird bei der Inferenz in Kapitel 10.3.3.7 beschrieben.

### 10.1.1.2 Generisches Wissen (Regel-Set)

Das generische Wissen wird in FuzzyDBSexpert in Regel-Sets zusammengefasst. Regel-Sets beinhalten folgende Komponenten (siehe Abbildung 10.5):

- Fuzzy-Sets für die ausgewählten Variablen
- Regeln, welche die Variablen benutzen
- Gewichte für die UPDRS

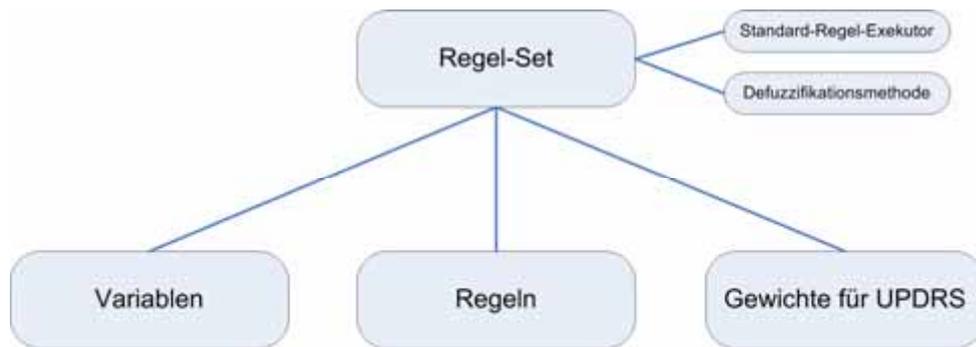


Abbildung 10.5: Aufbau eines Regel-Sets

### 10.1.1.2.1 Regel-Set-Variablen

Für ein Regel-Set werden die zu verwendenden Variablen definiert, um sie später in den Regeln verwenden zu können. Dies dürfen Eingangsvariablen, automatisch generierte Eingangsvariablen (UPDRS-Sektionswerte oder der UPDRS-Summenwert) und Ausgangsvariablen sein (siehe Tabelle 10-1). Einer Variable können, nachdem es sich um Fuzzy-Variablen handelt, ein oder mehrere Fuzzy-Sets zugeordnet sein, welche unter Angabe der Form und entsprechend dieser, einem oder mehreren Parametern, definiert werden kann (siehe Abbildung 10.6).

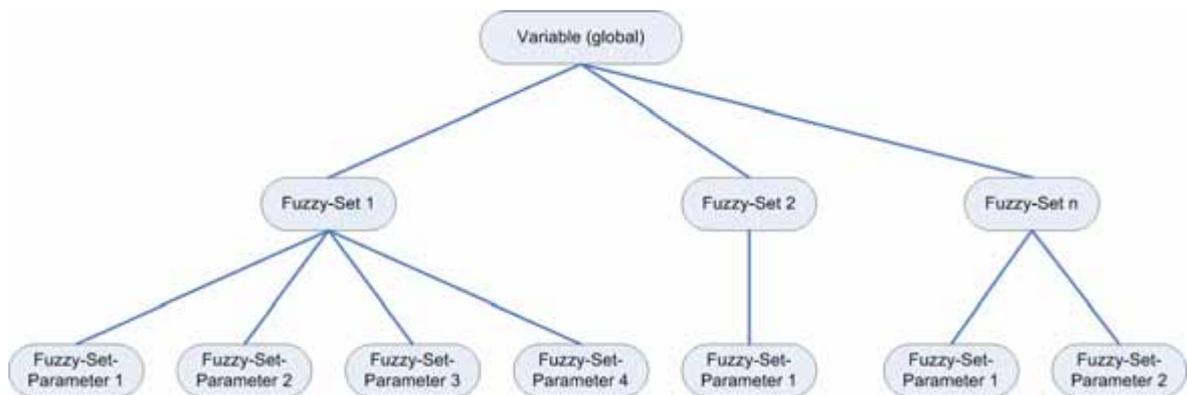


Abbildung 10.6: Aufbau einer Variablen in FuzzyDBSexpert

Anhand folgenden Beispiels in Abbildung 10.7 soll der Zusammenhang zwischen einer Variablen, deren Form und den Parametern erklärt werden:

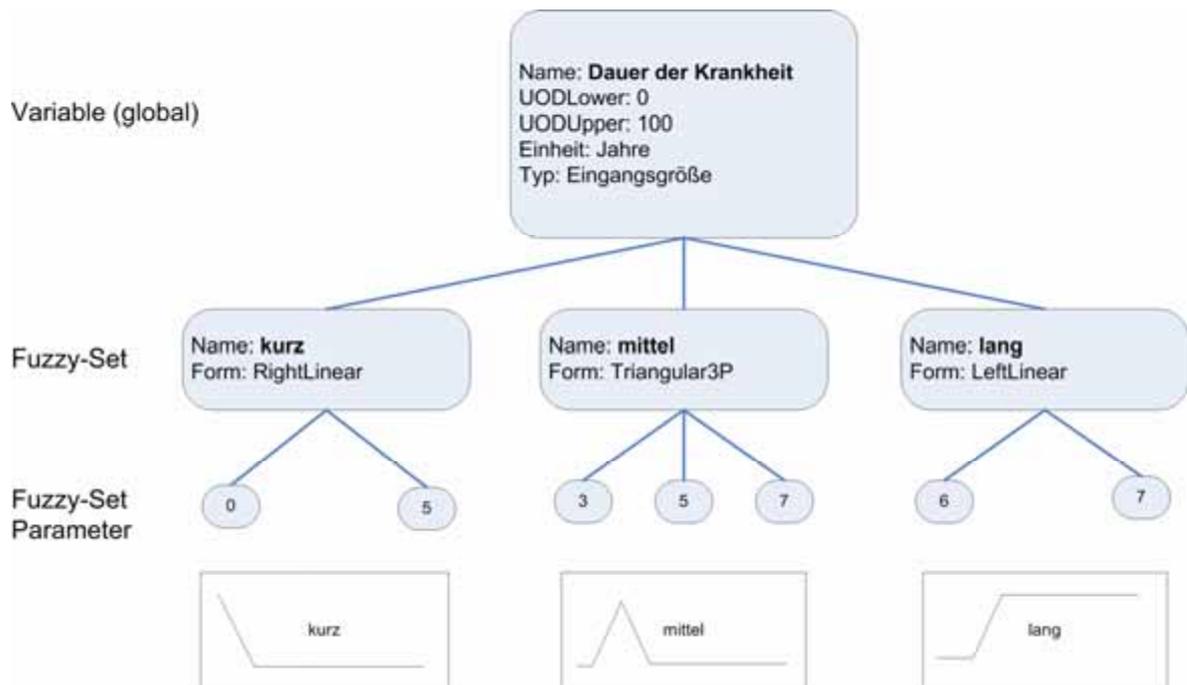


Abbildung 10.7: Beispiel einer Regel-Set-Variable

Eine Regel-Set-Variable basiert auf einer Variable, wie sie in Kapitel 10.1.1.1.1 definiert wurde. Diese Variablen besitzen zwar schon Eckdaten, die sie beschreiben, unter anderem den Namen und das UOD definieren, jedoch fehlen ihnen noch die Fuzzy-Sets, welche sie verwendbar machen. An dieser Stelle geht es nun darum, dass jeder Experte seine eigenen Vorstellungen der Ausprägung dieser Variablen ins System einbringen können soll und demnach selbst die Fuzzy-Sets für sie definiert.

Im Beispiel in der Abbildung 10.7 besitzt das erste Fuzzy-Set den Namen „kurz“ und ist durch ein rechts-lineares Fuzzy-Set definiert, welches bei 0 (Jahren) mit dem Wert 1 beginnt und ab 5 (Jahren) den Wert 0 hat.

### 10.1.1.2.2 Regel-Set-Regeln

Die zweite Hauptkomponente der Regel-Sets sind die Regeln. Diese sind hierarchisch folgendermaßen aufgebaut:

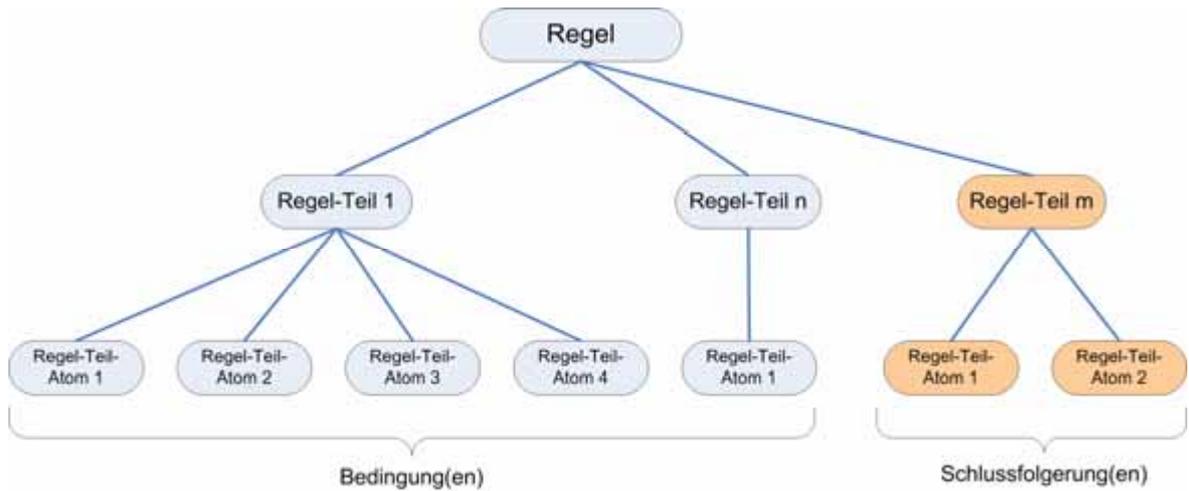


Abbildung 10.8: Aufbau einer Regel-Set-Regel

Eine Regel kann aus einem oder mehreren Bedingungs-, sowie einem oder mehreren Schlussfolgerungsteilen bestehen. Jeder dieser Teile bezieht sich auf eine ganz bestimmte Variable, die in den einzelnen Regel-Teilen definiert wird. Die Regel-Teile bestehen wiederum aus einem oder mehreren Regel-Teil-Atomen. Die Struktur wurde so gewählt, um es zu ermöglichen, auch andere Fuzzy-Bibliotheken, neben dem NRC Fuzzy-Toolkit [NRCFuzzy], 2007] verwenden zu können. Da nicht die Struktur von allen beziehungsweise zukünftig erscheinenden Bibliotheken bekannt sein kann, wurde eine möglichst flexible Struktur gewählt. Folgendes Beispiel soll die Struktur einer Regel in FuzzyDBSexpert näher erläutern:

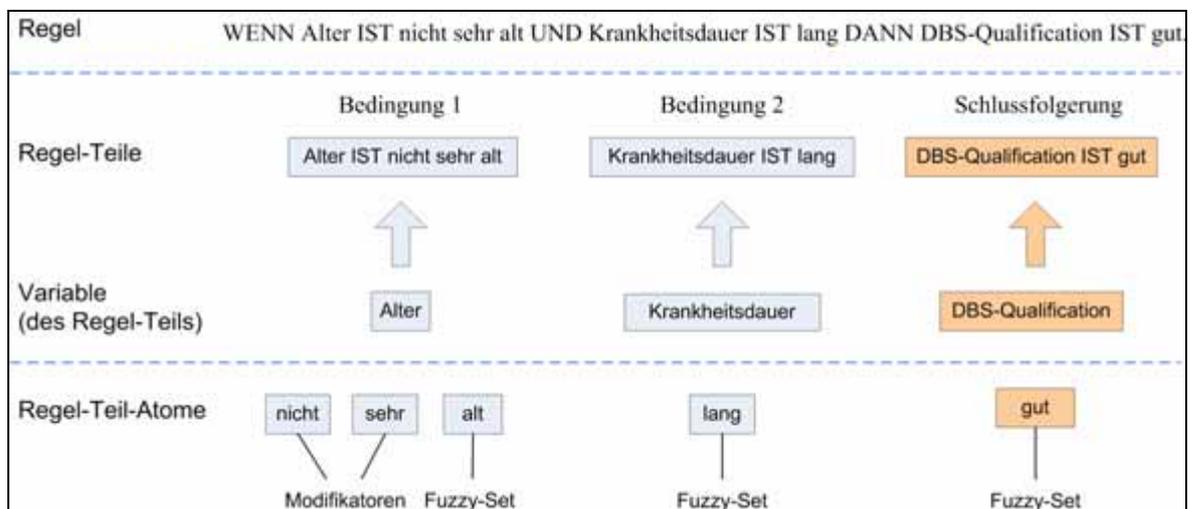


Abbildung 10.9: Beispiel einer Regel in FuzzyDBSexpert

Diese Regel besteht aus zwei Bedingungs- und einem Schlussfolgerungsteil. Unter dem Regel-Teil ist die jeweilige Variable, auf welche sich der Regel-Teil bezieht, dargestellt. Die Variable alleine nützt jedoch noch nichts, es muss definiert werden, auf welches Fuzzy-Set sich dieser Regel-Teil bezieht, was mit Hilfe der Regel-Teil-Atome geschieht. Für jeden Regel-Teil muss es also mindestens ein Atom geben, welches mindestens ein Fuzzy-Set der entsprechenden Fuzzy-Variable

enthält, damit diese Regel Sinn macht. Zusätzlich können jedoch auch beliebig viele Modifikatoren die Form der Fuzzy-Sets verändern, wie man anhand des ersten Regel-Teils in der Abbildung 10.9 erkennen kann. Folgende Modifikatoren sind in FuzzyDBSexpert implementiert:

Tabelle 10-2: Fuzzy-Modifikatoren

Fuzzy-Modifikatoren	Mathematische Formulierung
NOT	$1-x$
MORE OR LESS	$x^{1/3}$
SOMEWHAT	$x^{1/2}$
VERY	$x^2$
EXTREMELY	$x^3$
ABOVE	ABOVE identifiziert den ersten Wert $x$ , wo die Zugehörigkeitsfunktion das Maximum hat. Alle Werte darunter werden auf 0 gesetzt, alle Werte darüber auf $1-x$ .
BELOW	BELOW identifiziert den ersten Wert $x$ , wo die Zugehörigkeitsfunktion das Maximum hat. Alle Werte darüber werden auf 0 gesetzt, alle Werte darunter auf $1-x$ .

Jeder dieser Modifikatoren ändert die Form des Fuzzy-Sets nach den oben angegebenen mathematischen Prinzipien.

Damit man für einen Regel-Teil auch mehrere Fuzzy-Sets verwenden kann, benötigt man Operanden, um diese entweder konjunktiv (UND) oder disjunktiv (ODER) verknüpfen zu können. Weiters werden auch Klammern-Ausdrücke unterstützt, um kompliziertere Konstruktionen zu ermöglichen. Folgende Tabelle fasst die Operatoren, welche in FuzzyDBSexpert verwendet werden können, zusammen:

Tabelle 10-3: Operatoren in FuzzyDBSexpert

Operator	Bedeutung
AND	Konjunktion (UND)
OR	Disjunktion (ODER)
(	öffnende Klammer, beeinflusst Berechnungsreihenfolge
)	schließende Klammer, beeinflusst Berechnungsreihenfolge

### 10.1.1.2.3 Regel-Set-UPDRS-Gewichte

Wie in Kapitel 10.1.1.1.2 schon erwähnt, kann jede einzelne Frage der UPDRS gewichtet werden, um so ungünstig verteilte Gewichtungen, welche sich aus der UPDRS ergeben, ausgleichen und abweichende Meinungen von Experten, was die Wichtigkeit der einzelnen Fragen angeht, einbringen zu können. Zu diesem Zweck wurde die Möglichkeit geschaffen, dass jeder Experte seine eigenen Gewichte definieren und verwalten kann. Es hat sich in einer früheren Arbeit [Eitel,

2003, S. 100] gezeigt, dass eine symbolische und in ihren Ausprägungen begrenzte Gewichtung sinnvoll ist.

Den Fragen der UPDRS werden zunächst Gewichte zugewiesen. Bei der Auswertung der Daten wird jede einzelne Frage der UPDRS mit dem Gewicht multipliziert (siehe Kapitel 10.1.1.1.2).

## 10.1.2 Inferenz

Wie in Kapitel 8.2.5 beschrieben, läuft die Fuzzy-Inferenz meist in folgenden Schritten ab:

- Fuzzifizierung der Eingangsgrößen
- Anwendung der Fuzzy-Operatoren
- Anwendung der Fuzzy-Implikation
- Aggregierung der Ausgangsgrößen
- Defuzzifizierung der Ausgangsgrößen

Nach genau diesem Prinzip wird auch die Fuzzy-Inferenz in FuzzyDBSexpert durchgeführt, wobei die Aufgaben verschieden verteilt sind.

### 10.1.2.1 Fuzzifizierung der Eingangssignale

Alle Eingangsvariablen werden für die Inferenz fuzzifiziert. Werte für diese können entweder in linguistischer Form oder als Zahlenwert vorliegen oder unbekannt sein. Liegt der Wert als linguistischer Ausdruck vor, wird dieser direkt der Fuzzy-Variable zugewiesen. Bei Vorliegen eines numerischen Wertes wird dieser zuerst in ein Singleton-Fuzzy-Set (eine ein-elementige Menge) umgewandelt und dann der Fuzzy-Variable zugewiesen. Sollte der Wert einer Eingangsvariablen unbekannt sein, das bedeutet, dass weder ein linguistischer Ausdruck, noch ein numerischer Wert angegeben wurde, so wird diese Variable einfach weggelassen und der Inferenzprozess läuft mit den übrigen Variablen weiter. Variablen für Patienten sind auch in Form von Sets organisiert, in welche sie gespeichert und auch wieder entfernt werden können.

### 10.1.2.2 Anwendung der Fuzzy-Operatoren und -Implikation

Zuerst wird für jede Regel geprüft, ob die Werte für alle Variablen bekannt sind und sich diese daher im Variablen-Set für den Patienten-Datensatz befinden (siehe voriges Kapitel). Falls mindestens eine Variable zu dieser Regel fehlt, so wird sie übergangen und mit der nächsten Regel fortgesetzt.

Bevor eine Regel zur Anwendung kommt, also der Regel-Exekutor ausgeführt wird, wird überprüft, ob sich eine signifikante Ausgangsgröße ergeben würde; wenn nicht, unterbleibt die Auswertung der Regel, womit Rechenzeit gespart wird. Zur Zusammenfassung der Bedingungssteile jeder einzelnen Regel wird die Fuzzy-Operation UND verwendet, konkret der MIN-Operator.

Für das gesamte Regel-Set wird ein standardmäßiger Regel-Exekutor definiert, der immer zur Anwendung kommt, außer wenn für einzelne Regeln eigene Regel-Exekutoren angegeben wurden. Regel-Exekutoren definieren, wie die Regeln feuern und wie demnach die Fuzzy-Ausgangswerte zustande kommen. Im folgenden Abschnitt werden die Regel-Exekutoren näher beschrieben.

### 10.1.2.2.1 Mamdani-MIN-MAX-MIN-Operator

Dieser Regel-Exekutor verwendet für die Inferenz den Mamdani-MIN-Inferenz-Operator und für die Komposition den MAX-MIN-Operator. Die Ausgangsgröße wird so erzeugt, dass das Fuzzy-Set des Schlussfolgerungsteiles bei dem berechneten Wert (Grad der Erfüllung der Regel) abgeschnitten wird und zwar beim Minimum (MIN) der maximalen (MAX) Zugehörigkeitswerte der Schnittmenge der Fuzzy-Wert-Bedingungsteil-Fuzzy-Set- und Eingangswert-Paare (siehe Abbildung 10.10) [vgl. NRCFuzzy], 2007].

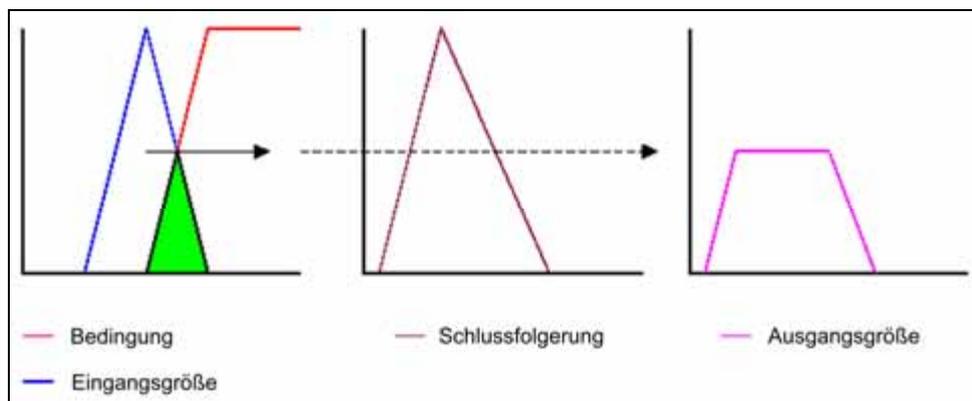


Abbildung 10.10: Beispiel der Anwendung des Mamdani-MIN-MAX-MIN-Operators [NRCFuzzy], 2007]

### 10.1.2.2.2 Larsen-PRODUCT-MAX-MIN-Operator

Der Larsen-PRODUCT-MAX-MIN-Operator berechnet sich ähnlich wie der Mamdani-MIN-MAX-MIN-Operator; hier wird die Ausgangsgröße nicht durch den MIN-Operator beschnitten, sondern mit dem PROD-Operator skaliert (siehe Abbildung 10.11) [vgl. NRCFuzzy], 2007].

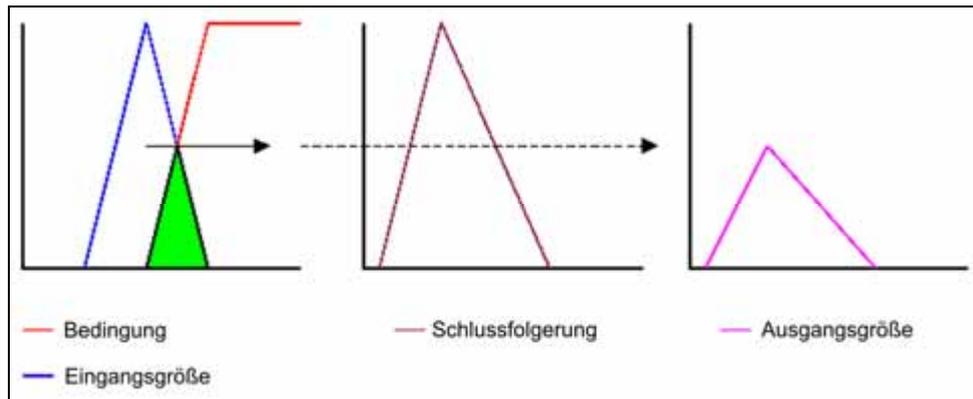


Abbildung 10.11: Beispiel der Anwendung des Larsen-PRODUCT-MAX-MIN-Operators [NRCFuzzy], 2007]

### 10.1.2.3 Aggregation der Ausgangsgrößen

Für jede Regel, welche ausgeführt wird, erhält man eine Ausgangsgröße pro Ausgangsvariable. Diese Ausgangsgrößen müssen noch miteinander verbunden werden, um so zu einer einzigen globalen Ausgangsgröße beizutragen. Dies geschieht mit Hilfe der Fuzzy-Vereinigung (Fuzzy-Union).

### 10.1.2.4 Defuzzifizierung der Ausgangsgrößen

Um einen einzigen Wert aus dem Ausgangs-Fuzzy-Set zu erhalten, welcher maschinell oder vom Menschen weiter verwendet werden kann, wird eine Defuzzifizierungsmethode angewendet. Folgende Defuzzifizierungsmethoden werden in FuzzyDBSexpert unterstützt:

- Center of Area
- Mean of Maxima
- Moment
- Weighted Average

## 10.1.3 Ergebnisse

Im folgenden Abschnitt werden die Ergebnisse und Ausgaben, welche in FuzzyDBSexpert möglich sind, erläutert.

### 10.1.3.1 DBS-Eignung

Hauptresultat ist die Angabe eines Score-Wertes für die Eignung zur Tiefen Hirnstimulation (DBS-Qualification). Dieser Wert ist eine Fuzzy-Variable vom Typ Ausgangsgröße, welche im Programm aber gesondert behandelt wird. So wird der resultierende Score-Wert zum Beispiel innerhalb eines

Textes als Baustein eingefügt. Man könnte durch einen einfachen Vergleich mit statistisch berechneten oder auch anders ermittelten Bereichen ein textuelles Ergebnis liefern, beispielsweise wenn dieser Wert über 0,8 liegt, so soll die Eignung als sehr gut angegeben werden.

### 10.1.3.2 Variablen

In FuzzyDBSexpert können alle Variablen, sowohl Eingangs-, berechnete Eingangs- als auch alle Ausgangsvariablen mit und ohne Beschreibung in der jeweils gewählten Sprache angezeigt werden. Abgesehen von der DBS-Eignung können auch andere Ausgangsvariablen definiert und deren Wert berechnet werden, diese Ergebnisse findet man in dieser Übersicht.

### 10.1.3.3 UPDRS-Ergebnisse

Die UPDRS ist ein wichtiger Bestandteil der Patientendaten. Nachdem die UPDRS häufig bei Parkinson-Patienten zur Dokumentation verwendet wird, ergibt sich ein Mehrwert für die Benutzer, indem sie die UPDRS für ihre eigene Dokumentation weiter verwenden können. Dazu werden nicht nur die gewichteten Sektionssummen und die Gesamtsumme, sondern auch die ungewichteten angezeigt, welche direkt der standardisierten UPDRS entsprechen.

### 10.1.3.4 Regeln

In FuzzyDBSexpert können die Regeln des Regel-Sets, welche zu den entsprechenden Ergebnissen geführt haben, angezeigt werden und zwar in einer Form, welche einfach lesbar ist. Ein Beispiel für die Anzeige einer Regel ist:

```
IF DrugTherapySuccess IS good THEN DBS-Qualification IS good.
```

```
WENN DrugTherapySuccess IST gut DANN DBS-Qualification IST gut.
```

## 10.2 Programmbeschreibung

An dieser Stelle wird ein Überblick über die verschiedenen Teile von FuzzyDBSexpert gegeben. Komplexe technische Aspekte, welche potentiell nur für Informatiker und Programmierer interessant sind, werden in den späteren Kapiteln detaillierter beschrieben. FuzzyDBSexpert gliedert sich aus der Benutzersicht in drei Bereiche (siehe Abbildung 10.12).



Abbildung 10.12: Hauptbereiche von FuzzyDBSexpert

## 10.2.1 Administrator-Teil

Der Administrator-Teil von FuzzyDBSexpert dient zur

- Benutzerverwaltung,
- UPDRS-Sets-Verwaltung und
- Variablenverwaltung.

### 10.2.1.1 Benutzerverwaltung

Die Benutzerverwaltung dient dazu, Konten für Benutzer des Systems, also Administratoren, Experten und Anwender, anzulegen, zu bearbeiten und zu löschen. Die Berechtigungsstufen sind in FuzzyDBSexpert hierarchisch aufgebaut, das bedeutet, dass höher eingestufte Benutzer auch die Rechte niedriger eingestufter Benutzer besitzen. Die Hierarchie der Benutzer ist folgendermaßen aufgebaut:

1. Administratoren (höchste Berechtigungsstufe)
2. Experten
3. Anwender (niedrigste Berechtigungsstufe)

Benutzer werden mit folgenden Attributen erfasst:

- Rolle\* (Administrator, Experte oder Anwender)
- Login\* (Benutzername zum Einloggen in FuzzyDBSexpert)
- Passwort\*

- Titel
- Name\*
- Vorname\*
- Geschlecht\*
- E-Mail-Adresse\*
- Institution
- Adresse
- Postleitzahl
- Stadt
- Bundesland
- Land

Attribute, welche mit einem \* gekennzeichnet sind, müssen zwingend eingegeben werden, alle anderen sind optional. Die E-Mail-Adresse wird unter anderem mittels eines regulären Ausdrucks auf Ihre Gültigkeit hin überprüft:

$$^{\wedge} [^{\wedge} \backslash \backslash @] + \backslash \backslash @ [^{\wedge} \backslash \backslash @] + \backslash \backslash . [^{\wedge} \backslash \backslash @] + \$$$

Benutzer können mit Hilfe des Benutzer-Managers und des enthaltenen DatabaseCRUDPanels, welches später noch beschrieben wird, hinzugefügt, editiert und gelöscht werden.

The screenshot shows a web interface titled 'Benutzer-Manager'. At the top, there is a message: 'Bitte benutzen Sie folgende Schaltflächen, um die Tabelle Benutzer zu bearbeiten.' Below this message are five buttons: 'Zeile hinzufügen', 'Markierte Zeilen löschen', 'Änderungen verwerfen', 'Änderungen speichern', and 'Zurück'. Below the buttons is a table with the following columns: 'Ausw.', 'Rolle', 'Login', 'Passwort', 'Titel', 'Name', 'Vorname', 'Geschlecht', and 'EMail'. The table contains three rows of user data:

Ausw.	Rolle	Login	Passwort	Titel	Name	Vorname	Geschlecht	EMail
<input type="checkbox"/>	a	vitek		Admin	Vitek	Marcus	männlich	mar
<input type="checkbox"/>	s	vitek		Expert	Vitek	Marcus	männlich	mar
<input type="checkbox"/>	u	vitek		User	Vitek	Marcus	männlich	mar

Abbildung 10.13: Benutzer-Manager

### 10.2.1.2 UPDRS-Set-Verwaltung

Die UPDRS, eine genormte medizinische Skala zur Beurteilung der Beeinträchtigung von Parkinson-Patienten durch deren Krankheit, wurde bereits in Kapitel 3 beschrieben. Sie wird auf

Deutsch und Englisch geliefert. Da sie jedoch auch in anderen Sprachen implementiert werden können soll und weiters die UPDRS zum jetzigen Zeitpunkt eine Revision durchläuft und nach der angelaufenen Testphase als sogenannte MDS-UPDRS veröffentlicht wird [Goetz, 2007], ist es notwendig, auch andere Versionen der UPDRS in das System eingeben zu können. Dies ermöglicht FuzzyDBSexpert durch entsprechende Bildschirme zur Verwaltung der UPDRS, welche im Anschluss beschrieben werden.

### 10.2.1.2.1 UPDRS-Sets-Manager

Im UPDRS-Sets-Manager (siehe Abbildung 10.14) können UPDRS-Sets hinzugefügt, bearbeitet und gelöscht werden, letzteres nur, sofern diese nicht von einem Regel-Set benutzt werden.



Abbildung 10.14: UPDRS-Sets-Manager

UPDRS-Sets haben folgende Attribute:

- UPDRS-Set-Name

Der Name für das UPDRS-Set; dieser sollte einen Hinweis auf die Sprache beinhalten, da der Name für die Auswahl zur Erstellung von Regel-Sets angezeigt wird und somit aussagekräftig beschreiben sollte, um welche UPDRS es sich handelt.

- UPDRS-Set-Referenz

An dieser Stelle soll ein Literaturverweis eingetragen werden, welcher auf die Quelle dieser UPDRS verweist, beziehungsweise ein Hinweis, wie eine eventuelle Übersetzung in eine andere Sprache zustande gekommen ist.

Zu den UPDRS-Sets gehören einerseits Optionen-Sets - auf eine Frage der derzeit aktuellen UPDRS können entweder zwei (Werte 0 und 1) oder fünf (Werte 0–4) verschiedene Antworten gegeben werden - und andererseits die Fragen selbst.

### 10.2.1.2.2 UPDRS-Optionen-Sets-Manager

UPDRS-Optionen-Sets dienen dazu, verschiedene mögliche Antworten, welche für eine Frage der UPDRS anwendbar sind, zusammenzufassen. Im UPDRS-Optionen-Sets-Manager können Optionen-Sets hinzugefügt, bearbeitet und gelöscht werden, letzteres aber nur, wenn diese nicht von einer der UPDRS-Fragen benutzt werden.

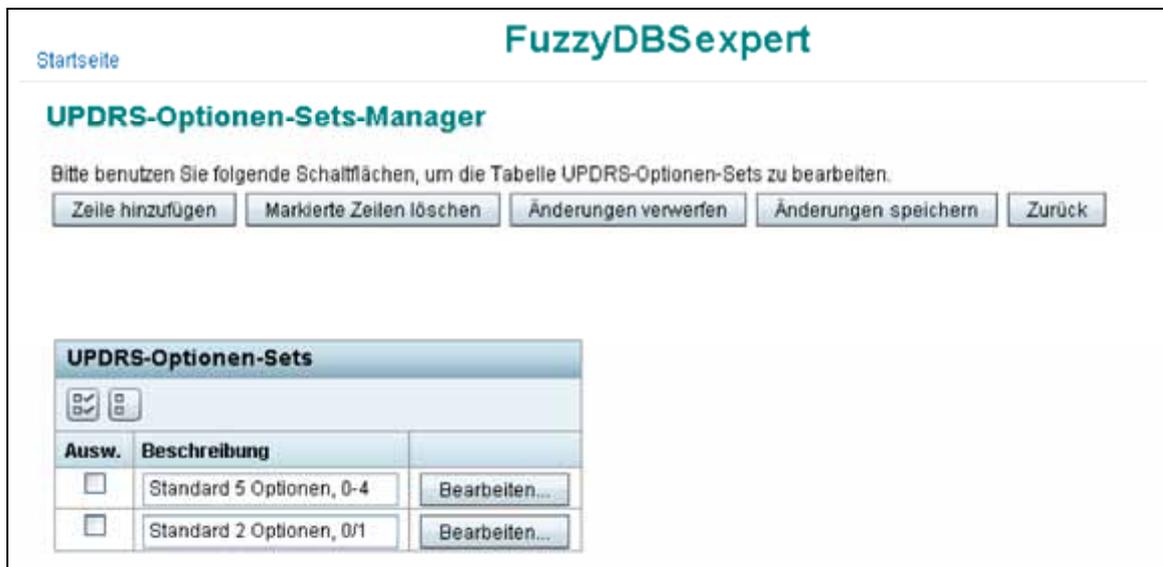


Abbildung 10.15: UPDRS-Optionen-Sets-Manager

UPDRS-Optionen-Sets besitzen eine Beschreibung, welche einen Hinweis auf die enthaltenen Antwort-Optionen geben soll und können mit Hilfe des DatabaseCRUDPanel verwaltet werden.

### 10.2.1.2.3 UPDRS-Optionen-Set-Optionen-Manager

Jedes Optionen-Set fasst eine gewisse Anzahl von möglichen Antworten auf eine UPDRS-Frage zusammen. Diese Optionen können mit Hilfe des UPDRS-Optionen-Set-Optionen-Managers (siehe Abbildung 10.16) hinzugefügt, bearbeitet und gelöscht werden.

Ausw.	Visualisierung	Wert
<input type="checkbox"/>	0	0
<input type="checkbox"/>	1	1
<input type="checkbox"/>	2	2
<input type="checkbox"/>	3	3
<input type="checkbox"/>	4	4

Abbildung 10.16: UPDRS-Optionen-Set-Optionen-Manager

Eine Option besitzt folgende Attribute:

- Visualisierung

Eine Option wird mit dem Zeichen, welches im Feld „Visualisierung“ eingetragen wird, visualisiert. Diese Zeichen werden einem Benutzer zur Auswahl angezeigt, welcher die Antworten für einen UPDRS-Datensatz ausfüllt. Dies macht die Visualisierung unabhängig von den Werten der Antworten und flexibler für zukünftige Änderungen oder beim Wunsch, andere Symbole beim Ausfüllen der UPDRS-Tabelle anzuzeigen, als für die Berechnung verwendet werden.

- Wert

Der Wert des Attributes „Wert“ wird für die Berechnung der Teilsummen verwendet und stellt eine ganze Zahl dar.

#### 10.2.1.2.4 UPDRS-Sektionen-Manager

Bei FuzzyDBSexpert werden die einzelnen Sektionen der UPDRS, wie in Kapitel 10.1.1.1.2 erläutert, getrennt betrachtet. Die UPDRS besteht aus den vier Sektionen UPDRS-I bis -IV. Der UPDRS-Sektionen-Manager (siehe Abbildung 10.17) ermöglicht es, Sektionen einer UPDRS hinzuzufügen, zu bearbeiten und zu löschen.



Abbildung 10.17: UPDRS-Sektionen-Manager

Eine UPDRS-Sektion besitzt folgende Attribute:

- Sektions-Bezeichnung

Dieses Attribut enthält die lange Bezeichnung der UPDRS-Sektion, welcher diese möglichst eindeutig identifizieren soll. Es ist üblich, die UPDRS-Sektionen mit UPDRS-I bis UPDRS-IV abzukürzen.

- Variable

Jeder gewichtete Summenwert einer Sektion soll als Variable zur Verfügung stehen, deswegen muss hier eine Verbindung einer UPDRS-Sektion zu einer bestimmten Variablen geschaffen werden. Wenn in der Folge ein Patienten-Datensatz ausgewertet wird, steht nach Berechnung des gewichteten Summenwertes dieser als berechnete Eingangs-Variable zur Verfügung und kann für Regeln verwendet werden. Details hierzu werden in Kapitel 10.2.1.3 erklärt.

#### 10.2.1.2.5 UPDRS-Sektion-Fragen-Manager

Der Hauptbestandteil eines UPDRS-Sets sind die Fragen. Diese können mit dem UPDRS-Sektion-Fragen-Manager (siehe Abbildung 10.18) hinzugefügt, bearbeitet und gelöscht werden.

**UPDRS-Sektion-Fragen-Manager**

Bitte benutzen Sie folgende Schaltflächen, um die Tabelle UPDRS-Sektion-Fragen zu bearbeiten.

UPDRS-Sektion-Fragen				
Ausw.	Frage	Fragentext	Fragenspezialisierung	Optionen-Set
<input type="checkbox"/>	18	Sprache		Standard 5 Optionen, 0-4
<input type="checkbox"/>	19	Gesichtsausdruck		Standard 5 Optionen, 0-4
<input type="checkbox"/>	20.1	Ruhetremor	Gesicht	Standard 5 Optionen, 0-4
<input type="checkbox"/>	20.2	Ruhetremor	rechte Hand	Standard 5 Optionen, 0-4
<input type="checkbox"/>	20.3	Ruhetremor	linke Hand	Standard 5 Optionen, 0-4
<input type="checkbox"/>	20.4	Ruhetremor	rechter Fuß	Standard 5 Optionen, 0-4
<input type="checkbox"/>	20.5	Ruhetremor	linker Fuß	Standard 5 Optionen, 0-4
<input type="checkbox"/>	21.1	Aktions- oder Haltungstremor der Hände	rechts	Standard 5 Optionen, 0-4
<input type="checkbox"/>	21.2	Aktions- oder Haltungstremor der Hände	links	Standard 5 Optionen, 0-4
<input type="checkbox"/>	22.1	Rigidität	Nacken	Standard 5 Optionen, 0-4
<input type="checkbox"/>	22.2	Rigidität	rechte obere Extremität	Standard 5 Optionen, 0-4

Abbildung 10.18: UPDRS-Sektion-Fragen-Manager

Eine UPDRS-Frage besteht aus folgenden Attributen:

– Frage

Das Attribut Frage gibt die Nummer der Frage in der UPDRS an, ergänzt um einen inkrementellen Wert, falls eine Frage mehrmals – meist für mehrere Extremitäten – gestellt werden muss.

– Fragentext

Das Attribut UPDRS-Frage soll die ursprüngliche Fragestellung aus der UPDRS enthalten beziehungsweise eine Übersetzung davon. Die Quelle der UPDRS oder deren Übersetzung kann im UPDRS-Set (siehe Kapitel 10.2.1.2.1) angegeben werden.

– Fragenspezialisierung

Die Fragenspezialisierung soll, falls in der UPDRS eine Frage mehrmals gestellt werden soll – meist für mehrere Extremitäten –, eine nähere Spezifizierung der Extremität ermöglichen (siehe Tabelle 10-4).

Tabelle 10-4: Beispiel von UPDRS-Fragen mit mehreren Extremitäten

UPDRS-Nummer	Frage	Fragenspezialisierung
22.1	Rigidität	Nacken
22.2	Rigidität	rechte obere Extremität
22.3	Rigidität	linke obere Extremität
22.4	Rigidität	rechte untere Extremität
22.5	Rigidität	linke untere Extremität

– Optionen-Set

In diesem Feld wird jenes Optionen-Set ausgewählt, welches die passenden Optionen für die Beantwortung der UPDRS-Frage enthält.

### 10.2.1.3 Variablenverwaltung

Variablen stellen in FuzzyDBSexpert alle Werte dar, die dem Patienten zugewiesen werden können (siehe Kapitel 10.1.1.1.1). Diese können mit Hilfe des Variablen-Managers (siehe Abbildung 10.19) hinzugefügt, bearbeitet und gelöscht werden.

Ausw.	Variablen-Bezeichnung	U.Gr. UOD	O.Gr. UOD	Einheit	Typ	
<input type="checkbox"/>	UPDRS I	0,0	16,0		UPDRS automatische E	Beschreibungen bearbeiten...
<input type="checkbox"/>	UPDRS II	0,0	52,0		UPDRS automatische E	Beschreibungen bearbeiten...
<input type="checkbox"/>	UPDRS III	0,0	108,0		UPDRS automatische E	Beschreibungen bearbeiten...
<input type="checkbox"/>	UPDRS IV	0,0	23,0		UPDRS automatische E	Beschreibungen bearbeiten...
<input type="checkbox"/>	Hoehn & Yahr	0,0	5,0		Eingangsgröße	Beschreibungen bearbeiten...
<input type="checkbox"/>	Schwab & England	0,0	100,0	%	Eingangsgröße	Beschreibungen bearbeiten...
<input type="checkbox"/>	DBS-Qualifikation	0,0	1,0		Ausgangsgröße	Beschreibungen bearbeiten...
<input type="checkbox"/>	UPDRS Sum	0,0	199,0		UPDRS Summenwert	Beschreibungen bearbeiten...
<input type="checkbox"/>	DrugTherapy/Success	0,0	1,0		Eingangsgröße	Beschreibungen bearbeiten...
<input type="checkbox"/>	TherapeuticFluctuation	0,0	1,0		Eingangsgröße	Beschreibungen bearbeiten...
<input type="checkbox"/>	Age	0,0	120,0	years	Eingangsgröße	Beschreibungen bearbeiten...
<input type="checkbox"/>	DurationOfDisease	0,0	100,0		Eingangsgröße	Beschreibungen bearbeiten...

Abbildung 10.19: Variablen-Manager

Es ist meiner Meinung nach sinnvoll, Variablen für alle Benutzer zentral zu verwalten, damit einerseits nicht jeder Experte Variablen definieren muss, welche häufig zum Einsatz kommen, wie zum Beispiel das Patientenalter und die Dauer der Krankheit, und andererseits die Typisierung der Variablen nur von einem Administrator vorgenommen werden kann, da dies eine potentielle Fehlerquelle darstellt und mit Bedacht verwendet werden muss. Eine Regel-Set-Variable hat immer einen Bezug zu einer Variable, welche von einem Administrator definiert wurde (siehe Kapitel 10.1.1.2.1). Dort werden die Eckdaten definiert, welche eine Fuzzy-Variable beschreiben und für

jeden Benutzer gleich sein sollen. An dieser Stelle jedoch sollen Experten Ihre Meinungen in Form von Fuzzy-Sets zu diesen Variablen einbringen können, welche diese erst verwendbar machen.

Fuzzy-Sets sind in FuzzyDBSexpert durch einen Namen und die Form charakterisiert. Die Form bedingt die anzugebende Anzahl an Parametern, welche diese dann im Kontext der Variablen definieren. Folgende Formen sind in FuzzyDBSexpert realisiert:

Tabelle 10-5: Formen für Fuzzy-Variablen in FuzzyDBSexpert

Bezeichnung	Anz. Param.	Beschreibung
TriangleFuzzySet 2P	2	Dreieckiges Fuzzy-Set definiert durch Mitte und Breite
TriangleFuzzySet 3P	3	Dreieckiges Fuzzy-Set definiert durch linken, mittleren und rechten Punkt
RectangleFuzzySet	2	Rechteckiges Fuzzy-Set
SingletonFuzzySet	1	Fuzzy-Set bestehend aus einem einzigen Punkt, der 1 ist
TrapezoidFuzzySet	4	Trapezförmiges Fuzzy-Set
GaussianFuzzySet	2	Gauß'sche Glockenkurve
GaussianFuzzySet wide	4	Gauß'sche Glockenkurve mit einem Plateau
PIFuzzySet	2	Glockenförmige Kurve
SFuzzySet	2	S-förmiges Fuzzy-Set, welches links unten beginnt
ZFuzzySet	2	Z-förmiges Fuzzy-Set, welches links oben beginnt
LeftLinearFuzzySet	2	Lineares Fuzzy-Set, welches links unten beginnt
RightLinearFuzzySet	2	Lineares Fuzzy-Set, welches links oben beginnt
LeftGaussianFuzzySet	2	Halbe Glockenkurve, welche links unten beginnt
RightGaussianFuzzySet	2	Halbe Glockenkurve, welche links oben beginnt

Die Beschreibungen der Formen für Fuzzy-Sets werden auch im Programm in der jeweiligen Sprache angezeigt. Genaue Beschreibungen der Fuzzy-Sets findet man in [NRCFuzzy], 2007].

Damit eine Beschreibung der genauen Bedeutung einer Variablen in jeder unterstützten Sprache ermöglicht wird, kann man für jede Variable für jede im System definierte Sprache eine Beschreibung verfassen. Falls eine Sprache neu hinzukommt, wird die Tabelle automatisch um einen entsprechenden Eintrag erweitert und ein Administrator kann diesen entsprechend ausfüllen.



Abbildung 10.20: Variablen-Beschreibungen-Manager

Variablen können, sobald sie definiert wurden, in Regel-Sets verwendet werden. Einfache Eingangsgrößen müssen bei der Anwendung in den Patientendatensätzen vom Benutzer mit Werten befüllt werden, die automatischen Eingangsgrößen werden anhand der ausgefüllten UPDRS ermittelt und die Ausgangsgrößen berechnen sich mit Hilfe der Regeln, Inferenz- und Defuzzifizierungsmethoden aus potentiell jeder dieser Eingangsgrößen.

## 10.2.2 Experten-Teil

Der Experten-Teil von FuzzyDBSexpert ermöglicht es den Experten unter Verwendung der im Administrator-Teil definierten Variablen und der UPDRS, Regel-Sets zusammenzustellen, auf deren Grundlage eine Fuzzy-Inferenz durchgeführt werden kann und welche Anwender verwenden können, um ihre Patientendatensätze auszuwerten.

### 10.2.2.1 Regel-Sets

Regel-Sets decken folgende Aufgaben ab:

- Auswahl der vom Experten gewünschten Variablen
- Definition der Fuzzy-Sets für alle Eingangs- und Ausgangsvariablen
- Definition der Regeln
- Definition der unterschiedlichen Gewichte zur Anwendung auf die UPDRS-Fragen
- Bestimmung je eines Gewichtes für jede UPDRS-Frage

### 10.2.2.2 Regel-Sets-Manager

Regel-Sets können mit dem Regel-Sets-Manager (siehe Abbildung 10.21) verwaltet werden. Regel-Sets können hinzugefügt, bearbeitet und gelöscht werden.



Abbildung 10.21: Regel-Sets-Manager

Regel-Sets besitzen folgende Attribute:

- Regel-Set-Bezeichnung

An dieser Stelle sollte ein das Regel-Set beschreibender Name eingetragen werden.

- UPDRS-Set

Hier kann das zu verwendende UPDRS-Set ausgewählt werden. Dies hat einerseits Einfluss auf die Sprache der Fragen und kann andererseits später dazu verwendet werden, um ein neues UPDRS-Set auszuwählen, welches nach der voraussichtlich dieses Jahr erscheinenden MDS-UPDRS [Goetz, 2007] erstellt wurde.

- Standard Regel-Exekutor

An dieser Stelle kann der standardmäßige Regel-Exekutor (siehe Kapitel 10.1.2.2) angegeben werden. Falls für die einzelnen Regeln kein anderer Regel-Exekutor ausgewählt wurde, wird dieser defaultmäßig angewendet.

- Defuzzifizierungsmethode

Hier kann die gewünschte Defuzzifizierungsmethode (siehe Kapitel 10.1.2.4) angegeben werden, mit der die Werte der Ausgangsvariablen in konventionelle Zahlenwerte umgewandelt werden.

- Erstelldatum

In diesem Feld wird ein automatischer Zeitstempel der Erstellung eingetragen, damit die Benutzer wissen, wie aktuell dieses Set ist.

- Aktiviert

Mit diesem Bitschalter kann angegeben werden, ob dieses Set zur Verwendung stehen soll oder nicht. Als Anwendungsbeispiel können Sets, welche gerade entwickelt werden, hiermit von der Benutzung ausgeschlossen werden.

- Öffentlich

Mit diesem Bitschalter kann angegeben werden, ob dieses Set von jedem Benutzer verwendet werden darf oder nicht. Falls dieses Regel-Set nicht öffentlich ist, kann nur der Ersteller dieses Set verwenden.

Im Regel-Set-Manager finden sich weiters noch Schaltflächen, um die Variablen, Regeln und Gewichte des jeweiligen Sets bearbeiten zu können.

### 10.2.2.2.1 Regel-Set-Variablen-Manager

Mit Hilfe des Regel-Set-Variablen-Managers (siehe Abbildung 10.22) können Variablen zu einem Regel-Set hinzugefügt, geändert und gelöscht werden.



Abbildung 10.22: Regel-Set-Variablen-Manager

Regel-Set-Variablen besitzen in FuzzyDBSexpert keine Attribute, es wird nur definiert, welche Variablen an diesem Regel-Set teilnehmen sollen. Die Fuzzy-Sets für die einzelnen Variablen können durch Klicken auf die entsprechende Schaltfläche „Variablen-Sets bearbeiten...“ mit Hilfe des Regel-Set-Variablen-Sets-Managers verwaltet werden.

### 10.2.2.2.2 Regel-Set-Variablen-Sets-Manager

Mit Hilfe des Regel-Set-Variablen-Sets-Managers (siehe Abbildung 10.23: Regel-Set-Variablen-Sets-Manager) können die Fuzzy-Sets für eine Variable hinzugefügt, bearbeitet und gelöscht werden.



Abbildung 10.23: Regel-Set-Variablen-Sets-Manager

Diese Sets besitzen folgende Attribute:

- Form

An dieser Stelle wird die Form des Fuzzy-Sets definiert, von der die Anzahl der anzugebenden Parameter, welche die Form genau definieren, abhängt. Die verschiedenen möglichen Formen sind in Tabelle 10-5 aufgelistet. Daneben wird, grau hinterlegt, die jeweilige Beschreibung der Form in der gewählten Sprache eingeblendet, um sich unter den nicht so leicht verständlichen Namen der Form etwas vorstellen zu können.

- Fuzzy-Set-Name

Dies sollte ein das Set beschreibender Name sein, da dieser zur Beschreibung für andere Benutzer dient, wie zum Beispiel „jung“ im Kontext der Variable „Patientenalter“.

Durch Klicken auf die Schaltfläche „Parameter bearbeiten...“ gelangt man zum Regel-Set-Variablen-Set-Parameter-Manager, mit dem man die Parameter für das jeweilige Fuzzy-Set definieren kann.

### 10.2.2.2.3 Regel-Set-Variablen-Set-Parameter-Manager

Der Regel-Set-Variablen-Set-Parameter-Manager (siehe Abbildung 10.24) dient zur Eingabe der für das Fuzzy-Set der Variable benötigten Parameter, um dessen Form eindeutig zu definieren. Je nach Form des Fuzzy-Sets (siehe Tabelle 10-5) werden dazu ein bis mehrere Parameter benötigt. Das System erstellt automatisch die passende Anzahl von Parametern für ein neues Fuzzy-Set, welche dann eingegeben werden können. Sollte sich die Form des Fuzzy-Sets nachträglich ändern, so fügt

das System entweder Parameter hinzu oder löscht von hinten überzählig gewordene Parameter wieder weg, wobei schon eingegebene Werte erhalten bleiben. Weiters überprüft das System, ob sich die Parameter im Wertebereich der Variablen, wie er im Administrator-Teil (siehe Kapitel 10.2.1.3) definiert wurde, befinden.

Regel-Set-Variablen-Set-Parameter-Manager

Bitte benutzen Sie folgende Schaltflächen, um die Tabelle Regel-Set-Variablen-Set-Parameter zu bearbeiten.

Zeile hinzufügen    Markierte Zeilen löschen    Änderungen verwerfen    Änderungen speichern    Zurück

Regel-Set-Variablen-Set-Parameter			
Parameter-Beschreibung	U.Gr. UOD	O.Gr. UOD	Wert
linke untere Ecke	0.0	16.0	4.0
mittlere Spitze	0.0	16.0	8.0
rechte untere Ecke	0.0	16.0	12.0

Abbildung 10.24: Regel-Set-Variablen-Set-Parameter-Manager

Im Regel-Set-Variablen-Set-Parameter-Manager können die Parameter für das jeweilige Fuzzy-Set eingegeben werden, wobei die Anzahl der Einträge automatisch passend für das Fuzzy-Set erstellt wird und eine Überprüfung stattfindet, ob sich die angegebenen Parameter innerhalb des Wertebereiches (UOD) der jeweiligen Variable befinden.

### 10.2.2.3 Regel-Set-Regeln

Der Aufbau der Regeln in FuzzyDBSexpert wurde schon in Kapitel 10.1.1.2.2 ausführlich erläutert, dieses Wissen kann man direkt in der Praxis anwenden. Regeln können mit den in den folgenden Kapiteln beschriebenen Werkzeugen eingegeben und modifiziert werden.

#### 10.2.2.3.1 Regel-Set-Regel-Manager

Mit Hilfe des Regel-Set-Regel-Managers (siehe Abbildung 10.25) können Regeln einem Regel-Set hinzugefügt, bearbeitet und gelöscht werden.

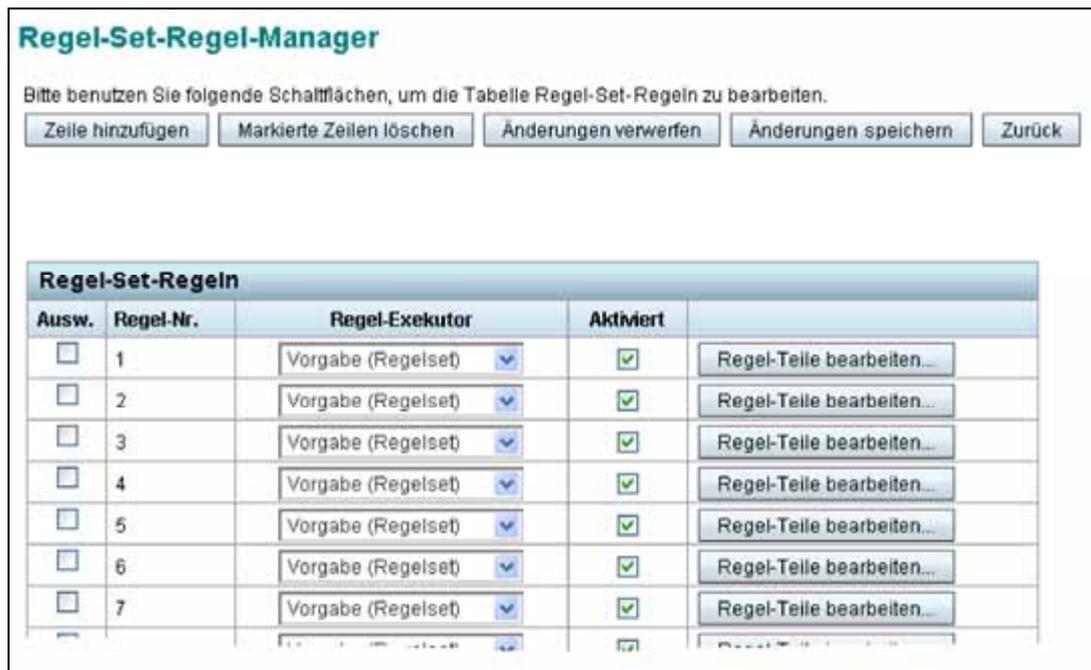


Abbildung 10.25: Regel-Set-Regel-Manager

Regeln besitzen in FuzzyDBSexpert folgende Attribute:

- Regel-Exekutor

An dieser Stelle kann ein vom standardmäßigen Regel-Exekutor abweichender Regel-Exekutor (siehe Kapitel 10.1.2.2 und 10.2.2.2) angegeben werden.

- Aktiviert

Mit diesem Bitschalter wird angegeben, ob die jeweilige Regel zur Auswertung von Patientendatensätzen verwendet werden soll oder nicht.

Mit der Schaltfläche „Regel-Teile bearbeiten...“ gelangt man in den Regel-Set-Regel-Teil-Manager, um die Regel-Teile der entsprechenden Regel bearbeiten zu können.

### 10.2.2.3.2 Regel-Set-Regel-Teil-Manager

Mit dem Regel-Set-Regel-Teil-Manager (siehe Abbildung 10.26) kann man den Regeln Regel-Teile hinzufügen, diese bearbeiten und löschen.



Abbildung 10.26: Regel-Set-Regel-Teil-Manager

Regel-Teile sind durch folgende Attribute definiert:

- Regel-Teil-Typ

Mit diesem Attribut wird bestimmt, ob es sich um einen Bedingungsteil oder um einen Schlussfolgerungsteil handelt. Die Reihenfolge der Bedingungs- und Schlussfolgerungsteile muss hierbei nicht eingehalten werden, diese werden später automatisch richtig zugeordnet. Wird ein Regel-Teil-Typ ausgewählt, so wird die Liste der zur Verfügung stehenden Variablen entsprechend aufgebaut.

- Variable

Wie in Kapitel 10.1.1.2.2 beschrieben, muss ein Regel-Teil immer einen Bezug zu genau einer Variable haben. Von dieser Variable können die Fuzzy-Sets zur weiteren Definition der Regel ausgewählt werden. Abhängig vom ausgewählten Regel-Typ werden nur Eingangs- oder nur Ausgangsvariablen zur Auswahl angeboten.

### 10.2.2.3.3 Regel-Set-Regel-Teil-Atom-Manager

Der Regel-Set-Regel-Teil-Atom-Manager (siehe Abbildung 10.27) dient dazu, die Regeln, in Abhängigkeit der vorhin gewählten Variablen, vollständig spezifizieren zu können. Es ist möglich, Atome hinzuzufügen, zu bearbeiten und zu löschen.

**Regel-Set-Regel-Teil-Atom-Manager**

Bitte benutzen Sie folgende Schaltflächen, um die Tabelle Regel-Set-Regel-Teil-Atome zu bearbeiten.

Regel-Set-Regel-Teil-Atome		
Ausw.	Atom-Nr.	Linguistischer Ausdruck
<input type="checkbox"/>	1	poor

Abbildung 10.27: Regel-Set-Regel-Teil-Atom-Manager

Regel-Teil-Atome haben nur ein Attribut, nämlich einen Ausdruck. Nachdem es in FuzzyDBSexpert möglich sein soll, auch andere Fuzzy-Bibliotheken für Java, als das NRC FuzzyJ-Toolkit, zu verwenden, ist dies ein generisches Feld, in dem einfach ein Text eingetragen wird. Welche Bedeutung dieser für das System hat und ob daraus vielleicht sogar Programmcode generiert wird, bleibt der jeweiligen Implementierung überlassen.

In der vorliegenden Implementierung wird der gesamte Text als linguistischer Ausdruck vom NRC FuzzyJ-Toolkit geparkt und verarbeitet. Um den Benutzer bei der Eingabe des linguistischen Ausdrucks zu unterstützen, wurde der LinguisticExpressionConstructor geschaffen. Dieser bietet der Variable entsprechende Fuzzy-Sets, die Modifikatoren (siehe Tabelle 10-2) und die Operatoren (siehe Tabelle 10-3) zur Auswahl an und überprüft nach der Eingabe, ob es sich um einen gültigen linguistischen Ausdruck handelt. Der LinguisticExpressionConstructor wird in Kapitel 10.2.4.5 näher besprochen.

#### 10.2.2.4 Regel-Set-Gewichte

Jede einzelne Frage der UPDRS kann durch einen Experten, wie schon in Kapitel 10.1.1.2.3 beschrieben, gewichtet werden. Hierzu müssen zuerst die gewünschten Gewichte in Visualisierung und Wert vom Experten definiert werden. Dabei wird defaultmäßig für alle UPDRS-Fragen ein Gewicht mit dem Wert 1 angelegt. Gewichte können mit dem Regel-Set-Gewichte-Manager verwaltet werden, die Zuweisung der Gewichte geschieht mit dem Regel-Set-UPDRS-Gewichte-Manager.

##### 10.2.2.4.1 Regel-Set-Gewichte-Manager

Mit Hilfe des Regel-Set-Gewichte-Managers (siehe Abbildung 10.28) können Gewichte hinzugefügt, bearbeitet und gelöscht werden.

**Regel-Set-Gewichte-Manager**

Bitte benutzen Sie folgende Schaltflächen, um die Tabelle Regel-Set-Gewichte zu bearbeiten.

Regel-Set-Gewichte		
Ausw.	Visualisierung	Wert
<input type="checkbox"/>	0	1.0
<input type="checkbox"/>	--	0.25
<input type="checkbox"/>	-	0.5
<input type="checkbox"/>	+	2.0
<input type="checkbox"/>	++	4.0

Abbildung 10.28: Regel-Set-Gewichte-Manager

In obigem Beispiel ist ein Set mit fünf verschiedenen Gewichten abgebildet, wobei jener Eintrag mit dem Gewicht 1 immer defaultmäßig erzeugt wird. Das Symbol kann selbstverständlich nachträglich geändert werden. Die Gewichte besitzen folgende zwei Attribute:

- Visualisierung

Durch Einführung einer anderen – symbolischen – Visualisierung für Gewichte kann man die konkreten Werte der Gewichte verschleiern und Benutzer können sich so mehr auf die eigene Intuition verlassen, ohne auf Zahlenwerte achten zu müssen.

- Wert

Hier wird eine positive Zahl angegeben, welche das Gewicht repräsentieren soll. Der bei der UPDRS gewählte Wert der Antwort wird später mit diesem Wert multipliziert. Standardmäßig haben alle Fragen der UPDRS eine Gewichtung von 1, dies kann im Regel-Set-UPDRS-Gewichte-Manager geändert werden.

#### 10.2.2.4.2 Regel-Set-UPDRS-Gewichte-Manager

Der Regel-Set-UPDRS-Gewichte-Manager (siehe Abbildung 10.29) dient dazu, den einzelnen Fragen der UPDRS jeweils eines der im Regel-Set-Gewichte-Manager definierten Gewichte zuzuweisen. Standardmäßig besitzen alle Fragen der UPDRS das Gewicht 1.

**Regel-Set-UPDRS-Gewichte-Manager**

Bitte benutzen Sie folgende Schaltflächen, um die Tabelle Regel-Set-UPDRS-Gewichte zu bearbeiten.

Regel-Set-UPDRS-Gewichte			
UPDRS-Nr.	Frage	Spezialisierung	Gewicht
1	Intellektuelle Einschränkung		0
2	Denkstörungen		0
3	Depression		0
4	Motivation/Initiative		0
5	Sprache		0
6	Speichelsekretion		0
7	Schlucken		0

Abbildung 10.29: Regel-Set-UPDRS-Gewichte-Manager

Wie in obiger Abbildung zu sehen ist, werden die vorhin definierten Gewichte zur Auswahl für die Gewichtung jeder einzelnen Frage angeboten.

## 10.2.3 Anwender-Teil

Im Anwender-Teil von FuzzyDBSexpert können einerseits Patienten und deren Datensätze administriert und andererseits Patientendatensätze anhand von Regel-Sets, welche von Experten zur Verfügung gestellt werden, ausgewertet werden.

### 10.2.3.1 Patienten-Manager

In FuzzyDBSexpert können Anwender mit Hilfe des Patienten-Managers (siehe Abbildung 10.30) Patienten hinzufügen, bearbeiten und löschen.

**Patienten-Manager**

Bitte benutzen Sie folgende Schaltflächen, um die Tabelle Patientendaten zu bearbeiten.

Patientendaten					
Ausw.	Patienten-Code	Geburtsdatum	Geschlecht	Öffentlich	
<input type="checkbox"/>	12345	25.09.1975	männlich	<input checked="" type="checkbox"/>	<input type="button" value="Datensätze bearbeiten..."/>
			männlich		
			weiblich		

Abbildung 10.30: Patienten-Manager

Patienten haben in FuzzyDBSexpert folgende Attribute:

- Patienten-Code

Der Patienten-Code soll den jeweiligen Patienten eindeutig identifizieren und zwar am Besten indirekt personenbezogen. Es wird davon abgeraten, hier die Sozialversicherungsnummer, Kombinationen von Vornamen, Nachnamen und Geburtsdatum oder Ähnliches einzutragen, da die Daten über das Internet übermittelt werden und bei direkt personenbezogenen Daten das Datenschutzgesetz anzuwenden und das Projekt viel aufwändiger bezüglich Security wäre, was den Rahmen dieser Arbeit sprengen würde.

- Geburtsdatum
- Geschlecht
- Öffentlich

Mit diesem Bitschalter kann angegeben werden, ob dieser Patienteneintrag mit all seinen Datensätzen öffentlich zugänglich sein soll, das bedeutet, auch von anderen Benutzern für die Evaluierungszwecke verwendet werden darf.

### 10.2.3.1.1 Patienten-Datensatz-Manager

Mit Hilfe des Patienten-Datensatz-Managers (siehe Abbildung 10.31) können für einen Patienten Datensätze erhoben, geändert oder gelöscht werden. Ein Patienten-Datensatz stellt eine Bestandsaufnahme der UPDRS und aller Variablen zu einem bestimmten Zeitpunkt dar.



Abbildung 10.31: Patienten-Datensatz-Manager

Ein Patienten-Datensatz besitzt folgende Attribute:

- Datensatz-Bezeichnung

Mit dieser Bezeichnung soll der Patienten-Datensatz eindeutig identifizierbar sein.

- Regel-Set-Basis

An dieser Stelle stehen alle öffentlichen und zusätzlich die eigenen Regel-Sets zur Auswahl. Mit Hilfe dieser Angabe wird später die Liste zur Erhebung der Werte für die vom Regel-Set benötigten Variablen erstellt.

- UPDRS-Set

Hier werden alle UPDRS-Sets zur Auswahl angeboten, derzeit sind dies die UPDRS auf Deutsch und Englisch. Anhand dieser Angabe wird später die Liste zur Darstellung der UPDRS-Fragen zusammengestellt.

Ein weiteres, aber nicht veränderliches Attribut, ist das Datum der Erhebung, welches in Form eines Zeitstempels automatisch generiert wird. Mit Hilfe der entsprechenden Schaltflächen kann man die Patienten-Datensatz-Variablen oder die Patienten-Datensatz-UPDRS eingeben.

### 10.2.3.1.2 Patienten-Datensatz-Variablen-Manager

Im Patienten-Datensatz-Variablen-Manager (siehe Abbildung 10.32) können für die durch das vorhin angegebene Regel-Set benötigten, nicht automatisch berechneten Eingangs-Variablen, Werte angegeben werden.

**Patienten-Datensatz-Variablen-Manager**

Bitte benutzen Sie folgende Schaltflächen, um die Tabelle Patienten-Datensatz-Variablen zu bearbeiten.

Patienten-Datensatz-Variablen						
Ausw.	Linguistischer Ausdruck		Numerischer Wert	U.Gr. UOD	O.Gr. UOD	Variablen-Beschreibung
Hoehn & Yahr		<- Bearbeiten	4.0	0.0	5.0	Das Stadium nach Hoehn und Yahr im ON-2
Schwab & England		<- Bearbeiten	70.0	0.0	100.0	Der Prozentwert der modifizierten Schwab- & England
DrugTherapySuccess	not very good	<- Bearbeiten		0.0	1.0	Der Erfolg der medikamentösen Therapie a bedeutet.
TherapeuticFluctuations	moderate	<- Bearbeiten		0.0	1.0	Die Therapeutischen Fluktuationen aufgrund 1, wobei "0" absolut keine Fluktuationen be
Age		<- Bearbeiten	66.0	0.0	120.0	Das Patientenalter in Jahren.
DurationOfDisease		<- Bearbeiten	8.5	0.0	100.0	Die Dauer der Parkinson'schen Krankheit d

Abbildung 10.32: Patienten-Datensatz-Variablen-Manager

Für die einzelnen Variablen kann einer der beiden Einträge ausgefüllt werden:

- Linguistischer Ausdruck

An dieser Stelle kann ein linguistischer Ausdruck für die Variable angegeben werden, welcher aus Fuzzy-Sets für diese Variable, Modifikatoren und Operatoren bestehen kann. Um den Benutzer bei der Eingabe eines für die jeweilige Variable gültigen linguistischen Ausdruckes zu unterstützen, kann dieser die Eingabe mit Hilfe des LinguisticExpressionConstructors zusammenstellen, welcher in Kapitel 10.2.4.5 näher beschrieben wird.

- Numerischer Wert

In vielen Fällen ist die Angabe eines Wertes in Form einer Zahl (zum Beispiel für Zeitintervalle) einfacher und adäquater. Aus diesem Grund kann an dieser Stelle statt eines linguistischen Ausdruckes auch eine Zahl eingegeben werden.

Das System überprüft bei der Eingabe automatisch, ob nur einer oder keiner der beiden Werte angegeben wurde und gibt andernfalls eine Fehlermeldung aus.

### 10.2.3.1.3 Patienten-Datensatz-UPDRS-Manager

Dieser Teil wird den meisten Anwendern, welche mit der Parkinson'schen Krankheit vertraut sind, am bekanntesten vorkommen. Mit Hilfe des Patienten-Datensatz-UPDRS-Managers (siehe Abbildung 10.33) kann nämlich die UPDRS ausgefüllt werden.

**Patienten-Datensatz-UPDRS-Manager**

Bitte benutzen Sie folgende Schaltflächen, um die Tabelle Patienten-Datensatz-UPDRS zu bearbeiten.

Patienten-Datensatz-UPDRS			
UPDRS Nr.	Frage	Spezialisierung	Punkte
1	Intellektuelle Einschränkung		<input checked="" type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4
22.1	Rigidität	Nacken	<input checked="" type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4
22.2	Rigidität	rechte obere Extremität	<input checked="" type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4
22.3	Rigidität	linke obere Extremität	<input checked="" type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4
22.4	Rigidität	rechte untere Extremität	<input checked="" type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4
22.5	Rigidität	linke untere Extremität	<input checked="" type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4
38	Treten "Off"-Perioden plötzlich auf, z. B. innerhalb von wenigen Sekunden?		<input checked="" type="radio"/> 0 <input type="radio"/> 1
39	Für welche Dauer befindet sich der Patient tagsüber durchschnittlich im "Off"-Stadium?		<input checked="" type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4
40	Leidet der Patient an Appetitlosigkeit, Übelkeit oder Erbrechen?		<input checked="" type="radio"/> 0 <input type="radio"/> 1
41	Leidet der Patient an Schlafstörungen, z. B. Schlaflosigkeit oder Schläfrigkeit?		<input checked="" type="radio"/> 0 <input type="radio"/> 1
42	Hat der Patient orthostatische Symptome?		<input checked="" type="radio"/> 0 <input type="radio"/> 1

Abbildung 10.33: Patienten-Datensatz-UPDRS-Manager

Um die Eingabe zu beschleunigen, wird auf die Darbietung der Antwortmöglichkeiten in Form einer Drop-down-Komponente verzichtet und stattdessen eine technisch kompliziertere Radio-Button-Auswahl angeboten, was eine Auswahl der Antwort mit einem einzigen Mausklick ermöglicht. Dies wird den meisten Anwendern sehr entgegenkommen, da es sich meist um Personen mit Zeitdruck handeln wird. Es ist zu beachten, dass für manche Fragen unterschiedlich viele Elemente zur Auswahl stehen. Alle diese Aspekte werden automatisch in FuzzyDBSexpert modelliert, konstruiert und zur Anzeige gebracht.

### 10.2.3.2 Evaluierungs-Teil

Mit Hilfe des Evaluierungs-Teiles können Benutzer ein Regel-Set auf einen Patienten-Datensatz anwenden, Ergebnisse berechnen und diese anzeigen lassen.

#### 10.2.3.2.1 Patienten-Datensatz-Evaluierung

Im entsprechenden Bildschirm (siehe Abbildung 10.34) werden zuerst die für eine Berechnung notwendigen Daten ausgewählt und danach wird die Berechnung durch Klicken auf die entsprechende Schaltfläche gestartet.

**Patienten-Datensatz-Evaluierung** zurück

**Wählen Sie einen Patienten:**  
12345

**Wählen Sie einen Datensatz:**  
TestPatientSet1

**Wählen Sie ein Regel-Set aus:**  
FirstTestRuleSet

**Berechnen**

Patienten-Code: 12345  
Geburtsdatum: 25.09.75  
Geschlecht: männlich

Datensatz-Bezeichnung: TestPatientSet1  
Erhebungsdatum: 12.01.07 00:50

Regel-Set-Bezeichnung: FirstTestRuleSet  
Standard-Regel-Exekutor: MamdaniMinMaxMin  
Defuzzifizierungsmethode: Center of Area

Abbildung 10.34: Patienten-Datensatz-Evaluierung

Zuerst muss ein Patient ausgewählt werden; die Auswahl eines Datensatzes ohne vorherige Auswahl eines Patienten ist nicht möglich. Falls für den gewählten Patienten noch gar keine Datensätze angelegt sind, wird ein Hinweis angezeigt und ein anderer Patient muss gewählt werden. Im Fenster rechts neben der Patientenauswahl wird eine kurze Information über den gewählten Patienten angezeigt, damit sich der Benutzer besser orientieren kann. Wenn der Patient Datensätze besitzt, kann einer dieser Datensätze mit der zweiten Drop-down-Komponente ausgewählt werden. Im Fenster rechts neben der Datensatzauswahl wird eine kurze Information über den gewählten Patienten-Datensatz angezeigt. Unabhängig vom Patienten kann ein Regel-Set ausgewählt werden, welches auf die Patientendaten angewendet werden soll. Falls dieses keine Regeln enthält, wird ein entsprechender Hinweis angezeigt und ein anderes Regel-Set muss ausgewählt werden. Sollten alle drei ausgewählten Elemente gültig sein, so kann durch Klicken auf die entsprechende Schaltfläche die Berechnung gestartet werden. Nach erfolgreicher Berechnung wird die Ergebnis-Übersicht angezeigt.

### 10.2.3.2 Ergebnis-Übersicht

In der Ergebnis-Übersicht wird eine kurze Information über den Patienten, den Patienten-Datensatz und das Regel-Set, alles selbstverständlich lokalisiert, das bedeutet in der jeweils gewählten Sprache, angezeigt.

**Ergebnis-Übersicht** [zurück](#)

### Ergebnisse

Die Ergebnisse für den/die Patienten/Patientin 12345, geboren am 25.09.75, Geschlecht männlich, basieren auf dem Datensatz TestPatientSet1, erhoben am 12.01.07 00:50 und auf dem Regel-Set FirstTestRuleSet erstellt von Admin Marcus Vitek.

Der Score-Wert für die Eignung des/der Patienten/Patientin für eine DBS-OP beträgt: 0,5

Abbildung 10.35: Ergebnis-Übersicht

Die Anzeige in der Ergebnis-Übersicht deckt aber nur einen kleinen Teil der Anzeigemöglichkeiten ab. Für andere Anzeigemöglichkeiten existieren entsprechende Schaltflächen.

### **Regel-Anzeige**

In der Regel-Anzeige werden alle Regeln angezeigt, die das Regel-Set enthält und zwar in einer Form, welche für Benutzer lesbar ist und nicht hierarchisch, wie sie in der Datenbank und im Programm gehalten werden.

[zurück](#)

### Anzeige der Regeln

Folgende Regeln kamen zur Anwendung:

WENN DrugTherapySuccess IST bad DANN DBS-Qualification IST poor  
WENN DrugTherapySuccess IST moderate DANN DBS-Qualification IST arguable  
WENN DrugTherapySuccess IST good DANN DBS-Qualification IST well  
WENN TherapeuticFluctuations IST light DANN DBS-Qualification IST poor  
WENN TherapeuticFluctuations IST moderate DANN DBS-Qualification IST arguable  
WENN TherapeuticFluctuations IST strong DANN DBS-Qualification IST well  
WENN Age IST young DANN DBS-Qualification IST poor  
WENN Age IST medium DANN DBS-Qualification IST arguable  
WENN Age IST old DANN DBS-Qualification IST well  
WENN DurationOfDisease IST short DANN DBS-Qualification IST poor  
WENN DurationOfDisease IST medium DANN DBS-Qualification IST arguable  
WENN DurationOfDisease IST long DANN DBS-Qualification IST well

Einige Regeln des Regel-Sets könnten nicht zur Anwendung gekommen sein, wenn die notwendigen Patientendaten nicht angegeben wurden.

Abbildung 10.36: Beispiel einer Regel-Anzeige

### Variablen-Anzeige

In der Variablen-Anzeige werden alle Variablen, also Eingangs-, automatisch berechnete Eingangs-, die UPDRS-Summen- und alle Ausgangsvariablen angezeigt. Für jede dieser Variablen kann im Administrator-Teil (siehe Kapitel 10.2.1.3) eine Beschreibung in jeder im System definierten Sprache (zur Zeit Deutsch und Englisch) eingegeben werden, um die genaue Bedeutung jeder Variablen erklären zu können, damit kein Raum für Fehlinterpretationen bleibt. Diese Beschreibung kann durch Klicken auf die entsprechende Schaltfläche ein- und ausgeschaltet werden.

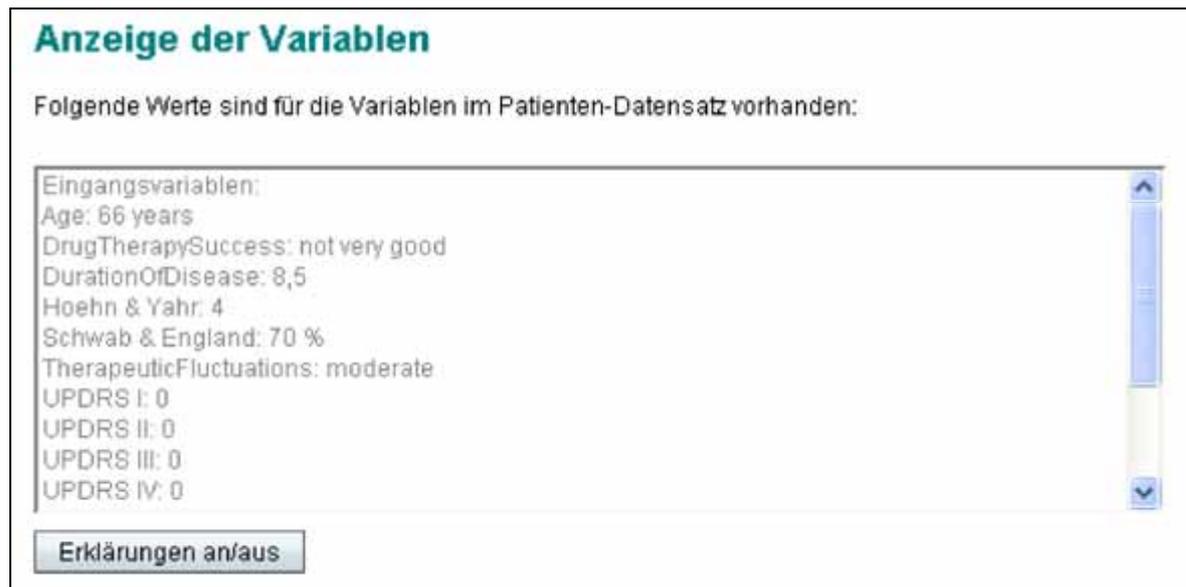


Abbildung 10.37: Beispiel einer Variablen-Anzeige

### UPDRS-Ergebnis-Anzeige

In der UPDRS-Ergebnis-Anzeige wird der UPDRS-Fragebogen mit den entsprechenden Antworten als Zahlenwerte dargestellt. Dieses Formular kann hier kopiert und in anderen Programmen weiter verwendet werden, was einen Mehrwert für Anwender darstellt, welche die UPDRS ausgefüllt haben, da dies nicht noch einmal getan werden muss, um die eigene, private Dokumentation zu komplettieren. Nicht nur der komplette Fragebogen, sondern auch die Teilsummen- und Summenwerte der UPDRS-I-IV, und zwar wahlweise gewichtet oder ungewichtet, werden angezeigt. Für herkömmliche Anwendungen und Vergleiche ist natürlich der ungewichtete UPDRS-Summenwert interessant.

## Anzeige der UPDRS-Ergebnisse zurück

Die Ergebnisse für den/die Patienten/Patientin 12345, geboren am 25.09.75 , Geschlecht männlich basieren auf dem Datensatz TestPatientSet1 , erhoben am 12.01.07 00:50 und auf dem Regel-Set FirstTestRuleSet erstellt von Admin Marcus Vitek.

UPDRS-Teil- und Gesamtsummen:

Ungewichtete Resultate
Gewichtete Resultate

UPDRS I: 0 (UPDRS-I - Kognitive Funktionen, Verhalten und Stimmung.)  
UPDRS II: 0 (UPDRS-II - Aktivitäten des täglichen Lebens.)  
UPDRS III: 0 (UPDRS-III - Motorische Untersuchung.)  
UPDRS IV: 0 (UPDRS-IV - Komplikationen der Behandlung (in der letzten Woche).)  
UPDRS Sum: 0 (UPDRS-Summenwert über alle UPDRS-Sektionen.)

UPDRS-Tabelle			
Frage	Fragentext	Fragenspezialisierung	Punkte
1	Intellektuelle Einschränkung		0
2	Denkstörungen		0
3	Depression		0
4	Motivation/Initiative		0

Abbildung 10.38: Beispiel einer Anzeige der UPDRS-Ergebnisse

### Grafische Darstellung der Ausgangsvariablen (Plot)

Als zusätzliches Feature ist es in FuzzyDBSexpert möglich, unter Verwendung der Funktion zur grafischen Darstellung von Fuzzy-Sets des NRC FuzzyJ-Toolkits, die akkumulierten Fuzzy-Sets der Ausgangsvariablen auf einfache Weise zu visualisieren.

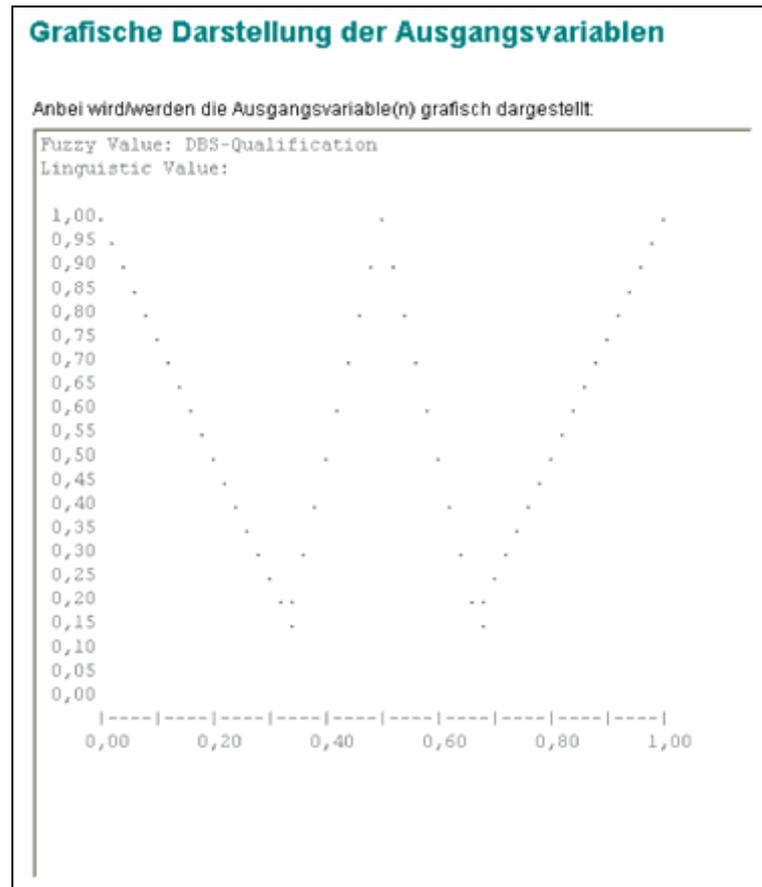


Abbildung 10.39: Beispiel der grafischen Darstellung einer Ausgangsvariablen

## 10.2.4 Bereichsübergreifende Features

Folgende Merkmale von FuzzyDBSexpert sind entweder im gesamten Programm oder zumindest in mehreren Teilen davon vorhanden.

### 10.2.4.1 Lokalisierung

FuzzyDBSexpert ist durchgehend lokalisiert, was bedeutet, dass in den folgenden Unterkapiteln beschriebene Elemente je nach Auswahl (derzeit Deutsch und Englisch verfügbar) adaptiert auf die jeweilige Sprache und Kultur angezeigt werden. Dies inkludiert nicht nur Textelemente, sondern auch Datumsangaben und Zahlenformate. Die Sprachauswahl wird am ersten Bildschirm von FuzzyDBSexpert getroffen, wobei die Liste der verfügbaren Sprachen dynamisch generiert wird.



Abbildung 10.40: Sprachauswahl in FuzzyDBExpert

Allgemein gibt es viele verschiedene Ansätze zur Lokalisierung, in FuzzyDBExpert werden folgende vier angewendet:

- Java-unterstützte Lokalisierung
- Lokalisierung über ein Ressource-Bundle
- dynamische datenbankunterstützte Lokalisierung
- statische datenbankunterstützte Lokalisierung

#### 10.2.4.1.1 Java-unterstützte Lokalisierung

Das Java-Framework unterstützt die Lokalisierung mit Hilfe von sogenannten *Locale*-Objekten, welche ausführlich in der Java-API<sup>1</sup> beschrieben sind. Ein *Locale*-Objekt repräsentiert eine bestimmte geografische, politische oder kulturelle Region. In FuzzyDBExpert werden *Locale*-Objekte verwendet, um Nummernformate und Zeitangaben automatisch zu lokalisieren.

#### 10.2.4.1.2 Lokalisierung über ein Ressource-Bundle

Java unterstützt die Verwendung von Ressource-Bundles, welche unter anderem zur Lokalisierung verwendet werden können. Ressource-Bundles sind so implementiert, dass sie das Suchen der benötigten lokalisierten Objekte – zum Beispiel Texte – weitgehend abstrahieren, sodass man sich damit nicht im Detail beschäftigen muss. In mehreren Property-Dateien – je Sprache eine – werden

---

<sup>1</sup> Application Programming Interface, die Programmierschnittstelle, welche von einem Softwaresystem anderen Programmen zur Verfügung gestellt wird, um dieses zu benutzen.

Attribut-Wert-Tupel gespeichert. Bei Bedarf wird ein Attribut, also zum Beispiel ein Seitentitel, in der Property-Datei gesucht und der entsprechende Wert davon zurückgeliefert. Das bringt - gegenüber einem fest programmierten Wert - den Vorteil mit sich, dass ein Programmierer die Attribute nicht im Code suchen und durch einen anderen Wert, zum Beispiel bei Übersetzung des Programmes in eine andere Sprache, ersetzen muss, sondern dies an einer zentralen Stelle tun kann, meist ohne das Programm neu zu kompilieren. Außerdem ist auf diese Weise die Unterstützung von verschiedenen Sprachen möglich. Folgende Texte sind über eine Property-Datei lokalisiert:

- Seitentitel, welche in der Titelleiste des Browser-Fensters angezeigt werden
- auf der Seite sichtbare Seitentitel
- Texte auf den Seiten (bis auf wenige Ausnahmen)
- Tabellentitel
- Tabellenspaltentitel

#### **10.2.4.1.3 Dynamische datenbankunterstützte Lokalisierung**

Variablen werden durch Administratoren angelegt. Administratoren müssen nicht unbedingt Programmierer sein, deswegen kommt hier eine Lokalisierung der Beschreibung einer neuen Variablen mit Hilfe der Property-Datei nicht in Frage. Aus diesem Grund wurde der Bildschirm zum Bearbeiten von Variablen um eine Schaltfläche pro Variable erweitert, welche zu einem weiteren Bildschirm führt, wo die Beschreibung der Variablen in allen im System definierten Sprachen möglich ist (siehe Kapitel 10.2.1.3). Kommt eine Sprache hinzu, so wird diese Tabelle automatisch um diesen Eintrag erweitert. Auf diese Weise können alle Variablen um die Beschreibung in der neu hinzugekommenen Sprache ergänzt werden.

#### **10.2.4.1.4 Statische datenbankunterstützte Lokalisierung**

Einige Elemente, welche in der Datenbank gespeichert sind, besitzen auch lokalisierte Beschreibungen. Es handelt sich hierbei um Elemente, welche in starkem Zusammenhang mit der Programmlogik stehen und daher nur von einem Programmierer geändert werden dürfen, welcher auch den Programmcode entsprechend erweitern kann und sich über die Konsequenzen seiner

Handlungen im Klaren sein muss. Diese Elemente können über keine Benutzeroberfläche von FuzzyDBSexpert bearbeitet werden, sondern nur durch direkte Manipulation der Datenbank, sei es auch über eine für nicht im Datenbankmanagement versierte Personen leicht verständliche Benutzeroberfläche, wie zum Beispiel den MySQL Query Browser<sup>1</sup>. Folgende Elemente sind statisch lokalisiert und müssen bei Erweiterung um eine Sprache oder zwecks Änderung direkt in der Datenbank geändert werden:

- Texte zur Beschreibung der Form der Fuzzy-Sets
  - Beispiel: „Trapezförmiges Fuzzy-Set“
- Texte zur Beschreibung der Parameter der Form der Fuzzy-Sets
  - Beispiel: „rechte untere Ecke“
- Regel-Teil-Typen-Texte
  - zwei Einträge: „Bedingung“, „Schlussfolgerung“
- Variablentypen-Texte
  - Beispiel: „Ausgangsgröße“, „Eingangsgröße“
- Texte für die Bezeichnung von Geschlechtern
  - zwei Einträge: „männlich“, „weiblich“

#### **10.2.4.2 Flexibilität**

Die Flexibilität eines Programmes kann vielerlei Aspekte betreffen. Einer davon wäre die in Kapitel 10.2.4.1 beschriebene sprachenabhängige Gestaltung der Texte. Im Folgenden sollen weitere Aspekte, in denen FuzzyDBSexpert flexibel ist, genannt werden.

---

<sup>1</sup> MySQL Query Browser: <http://www.mysql.com/products/tools/query-browser/>

#### 10.2.4.2.1 Flexibilität durch und für die Benutzer

FuzzyDBSexpert ist ein durch Benutzer umfassend konfigurierbares Programm. Über die Benutzeroberfläche können folgende Dinge erweitert werden:

- UPDRSs können über die Definition von Sektionen und deren Fragen durch Administratoren frei definiert werden (wobei es ratsam ist, sich an die vorgegebenen Standards zu halten, damit die Skalen vergleichbar bleiben).
- Die möglichen Antwortoptionen für die UPDRS können von Administratoren in Wert und Visualisierung frei definiert werden.
- Variablen können von Administratoren beliebig hinzugefügt werden, inklusive deren Beschreibung in allen verfügbaren Sprachen.
- Regeln und Fuzzy-Sets für die Variablen, welche von Administratoren definiert wurden, können für jeden Experten extra beliebig in Sets geführt werden.
- Alle Fragen der UPDRS können von jedem Experten für jedes Regel-Set extra mit frei definierbaren Gewichten versehen werden.
- Patienten können von jedem Benutzer angelegt und bei Bedarf veröffentlicht werden, das bedeutet, dass jeder Benutzer auf die Datensätze dieses Patienten zugreifen darf, um diese zu evaluieren.
- Patienten-Datensätze können von allen Benutzern extra anhand von gewählten Regel-Sets und der UPDRS flexibel gestaltet und eingegeben werden, da abhängig von der gewählten UPDRS die Fragenliste und abhängig vom gewählten Regel-Set die anzugebenden Variablen flexibel gestaltet werden.

#### 10.2.4.2.2 Austauschbarkeit der Fuzzy-Bibliothek

FuzzyDBSexpert wurde so gestaltet, dass versucht wurde, den nicht zu vermeidenden Aufwand zur Anpassung an eine andere (Java-) Fuzzy-Bibliothek möglichst gering zu halten. Dies geschieht unter anderem durch möglichst wenige Schnittstellen zu den Objekten und durch möglichst punktuelle Verwendung der Objekte in dieser Bibliothek. Dazu werden FuzzyDBSexpert-eigene Objekte im Speicher zur Repräsentierung von Fuzzy-Regeln, Fuzzy-Variablen und der Fuzzy-Inferenzmaschine gehalten (siehe Kapitel 10.3.3.1) und erst in letzter Instanz die Übersetzung auf die Objekte der verwendeten Bibliothek durchgeführt. Wie diese Trennung genau geschieht, wird im technischen Teil dieser Arbeit genauer beschrieben.

### 10.2.4.2.3 Austauschbarkeit der Datenbank

FuzzyDBSexpert verwendet keine MySQL-spezifischen Ausdrücke, sondern nur Standard-SQL-92, welches von jeder nennenswerten Datenbank unterstützt wird. Die Datenbank kann über das Ausführen einer .SQL-Datei datenbankunabhängig über die Verwendung von `CREATE TABLE` und `INSERT` Statements komplett inklusive Demo-Daten aufgebaut werden.

### 10.2.4.2.4 Austauschbarkeit des Web-Containers

Für FuzzyDBSexpert kann jeder Webserver mit Servlet/JSP-Engine verwendet werden, welcher JSF (JavaServer Faces) unterstützt. Das Programm wird in Form einer .WAR-Datei geliefert, welche jede Servlets/JSP-Engine, die für JavaServer Faces konfiguriert ist, verarbeiten kann.

### 10.2.4.2.5 Flexibilität der Lokalisierung

FuzzyDBSexpert kann durch einen einzigen zusätzlichen Eintrag in eine Tabelle um die Unterstützung einer Sprache erweitert werden, welche auch automatisch im Sprachen-Auswahlschirm erscheint (siehe Abbildung 10.40). Es darf aber nicht vergessen werden, dass alle lokalisierten Texte (siehe Kapitel 10.2.4.1) für die neue Sprache eingetragen werden müssen und das nicht nur in der Datenbank.

### 10.2.4.2.6 Design

Das Design der Seiten ist durch vorrangige Verwendung von Cascading Stylesheets<sup>1</sup> und JSPF (JavaServer Page-Fragments) sehr einheitlich, aber auch sehr flexibel. So befindet sich zum Beispiel auf jeder Seite ein Header, welcher immer das gleiche Element darstellt, die unterschiedlichen Überschriften werden dynamisch erzeugt.

---

<sup>1</sup> Cascading Stylesheets (CSS) – eine HTML-Ergänzungssprache, mit der man HTML-Elemente exakt formatieren und positionieren kann (siehe <http://de.selfhtml.org/css/index.htm>)

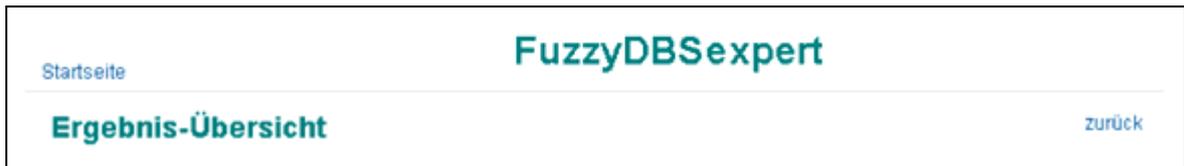


Abbildung 10.41: Beispiel eines Headers in FuzzyDBSexpert

Auch die anderen Überschriften und die Platzierung der Inhalte und des DatabaseCRUDPanels auf der Seite sind beispielsweise über Stylesheet-Elemente realisiert und brauchen nur an einer Stelle geändert werden, um die Änderung auf alle Seiten zu übertragen. Es kann auch ein komplett anderes Design – Thema genannt – auf das Projekt angewendet werden, dies wird direkt durch die Entwicklungsumgebung unterstützt.

### 10.2.4.3 DatabaseCRUDPanel

Das DatabaseCRUDPanel (siehe Abbildung 10.42) ist ein zentrales Element in FuzzyDBSexpert. FuzzyDBSexpert arbeitet mit sehr vielen Tabellen und Benutzer müssen indirekt auf vielen von diesen Tabellen über eine Benutzeroberfläche operieren können. Weiters gibt es nur ein bestimmtes Set an immer wiederkehrenden Datenbankoperationen, aus denen sich auch die Namensgebung herleitet, da CRUD ein verbreitetes Akronym für

- Create
- Read
- Update und
- Delete

ist. Eine Benutzeroberfläche, welche auch die meisten Fehler abfängt und dem Benutzer in einer komfortablen Weise anzeigt, kann sehr aufwändig werden. Aus diesen Gründen wurde ein zentrales Element geschaffen, welches nur in eine Seite eingebaut und etwas konfiguriert werden muss, mit dem alle notwendigen Datenbankoperationen und anderen genannten Anforderungen abgedeckt werden können:

- Datenbankoperationen
  - Einträge hinzufügen (auch multiple, wenn möglich)
  - Einträge löschen (auch multiple, wenn angebracht)
  - Änderungen von Einträgen speichern oder verwerfen

- Ansprechende Meldungen bei
  - erfolgreichem oder fehlerhaftem Speichern von neuen Einträgen oder Änderungen
  - erfolgreichem oder fehlerhaftem Löschen von Einträgen
  - Versuch der Bearbeitung von Untereinträgen oder Klicken der Zurück-Schaltfläche bei Existenz eines neu hinzugefügten Eintrages, der noch nicht gespeichert wurde
- Wiederverwendung
  - durch die Wiederverwendung dieses Widgets<sup>1</sup> in vielen Seiten müssen Änderungen, Erweiterungen und Verbesserungen nur an einer einzigen Stelle durchgeführt werden und deshalb kann dieses Bedienelement auch sehr aufwändig gestaltet werden



Abbildung 10.42: DatabaseCRUDPanel

Das multiple Hinzufügen von Einträgen, also das Hinzufügen von mehreren Einträgen hintereinander, ohne diese sofort ausfüllen zu müssen, ist dann möglich, wenn es keine Untereinträge gibt, welche unbedingt ausgefüllt sein müssen, damit der hinzugefügte Eintrag valide wird.

#### 10.2.4.4 Kaskadiertes Löschen

Die Operation des Löschens ist in einer Datenbank immer ein wenig problematisch. Häufig hängen mit einem Eintrag auch andere Einträge zusammen, welche ohne diese nicht existieren können. In Datenbanken werden diese Zusammenhänge mit Hilfe von Fremdschlüsseln abgebildet. Deswegen ist ohne die Unterstützung durch kaskadiertes Löschen nur das Löschen von Einträgen möglich, auf die keine Fremdschlüssel zeigen, was unter anderem bei Einträgen der untersten Hierarchie, wie zum Beispiel den Regel-Teil-Atomen in FuzzyDBSexpert, der Fall ist. Da es jedoch für einen Benutzer beispielsweise unzumutbar ist, wenn er eine Regel löschen möchte, zuerst die zugehörigen Regel-Teil-Atome, dann die Regel-Teile und erst dann die Regel zu löschen, musste die Datenbank

---

<sup>1</sup> Als Widgets (Fensterkontrollelemente) bezeichnet man die Interaktionselemente oder Steuerelemente in grafischen Benutzeroberflächen. Der Begriff entstand durch das Zusammensetzen und Verkürzen der englischen Worte "window" und "gadget".

mit gut durchdachtem kaskadiertem Löschen konfiguriert werden. Folgende Abbildung zeigt, für welche Einträge die zusammenhängenden Untereinträge mitgelöscht werden, wobei hier die Namen der Tabellen in der Datenbank verwendet werden (siehe Kapitel 10.3.2.3):

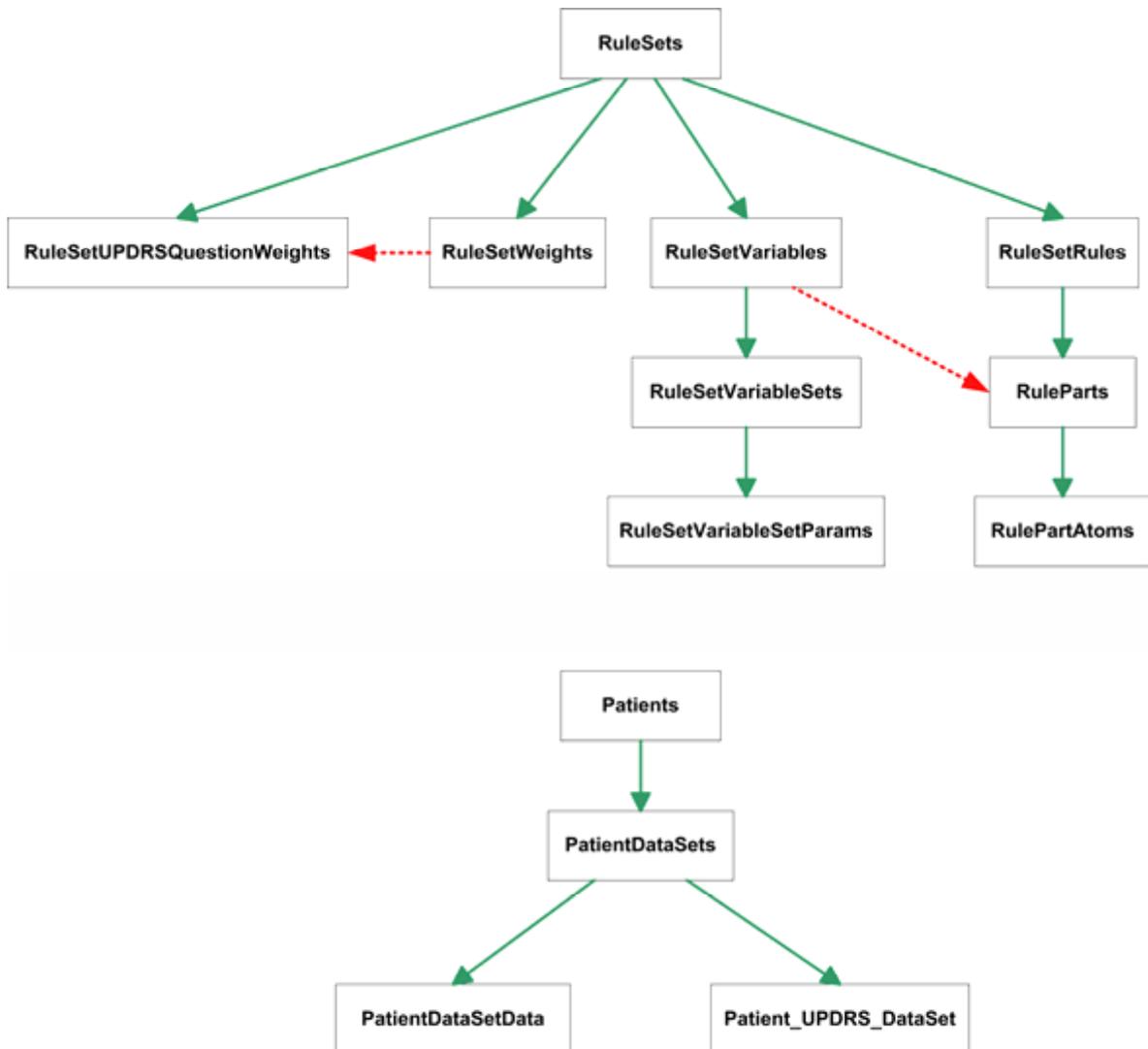


Abbildung 10.43: Tabellen mit Lösch-Kaskaden

Diese Grafik ist so zu interpretieren, dass beim Löschen von hierarchisch betrachtet höher gestellten Elementen alle niedriger liegenden Elemente, also jene, auf die die Pfeilspitzen zeigen, ebenfalls gelöscht werden. In der Grafik sind diese Lösch-Kaskaden mit grünen (durchgehenden) Pfeilen angezeigt. Man kann davon ausgehen, dass alle anderen – nicht dargestellten – Elemente nicht ohne Weiteres löschtbar sind, also das Löschen restriktiert wird, wenn es auf die Elemente zeigende Referenzen gibt, was durchaus Sinn macht. So ist es zum Beispiel nicht sinnvoll, ein Regel-Set oder UPDRS-Set zu löschen, wenn damit schon mindestens ein Patienten-Datensatz angelegt wurde. Es ist aber ohne weiteres möglich, ein Regel-Set zu löschen, welches nicht in einem Patienten-Datensatz verwendet wird. In der Grafik sind zwei interessante Fälle mit roten (punktierten) Pfeilen angezeigt. Wenn es für die Ausgangselemente Einträge gibt, welche von diesen abhängig sind, also jene an der Pfeilspitze, wird das Löschen restriktiert. Die Datenbank

muss deswegen so intelligent implementiert sein, zuerst die abhängigen Elemente zu löschen und dann erst jene, von denen die abhängigen Elemente abhängig waren. Beispielsweise müssen die Einträge der Tabelle RuleParts vor jenen der Tabelle RuleSetVariables für ein Regel-Set gelöscht werden, damit ein Eintrag in der RuleSet-Tabelle entfernt werden kann.

### 10.2.4.5 LinguisticExpressionConstructor

In FuzzyDBSexpert ist es an einigen Stellen möglich beziehungsweise notwendig, gültige linguistische Ausdrücke für eine Fuzzy-Variable anzugeben. Folgende Tabelle soll eine Übersicht darüber geben:

Tabelle 10-6: Vorkommen linguistischer Ausdrücke in FuzzyDBSexpert

Vorkommen	Notwendigkeit der Angabe
Regel-Teil-Atom	notwendig
Patienten-Datensatz-Variable	möglich

Mit etwas Erfahrung ist es natürlich möglich, linguistische Ausdrücke ohne Hilfsmittel zu formulieren, dazu muss man jedoch mindestens die Bezeichnungen für die jeweiligen Fuzzy-Sets der betreffenden Variablen wissen. Die in FuzzyDBSexpert möglichen Modifikatoren (siehe Tabelle 10-2) und Operatoren (siehe Tabelle 10-3) kann man nach einiger Zeit auswendig wissen. Sollte ein eingegebener linguistischer Ausdruck im Kontext einer bestimmten Variable nicht gültig sein, so wird der Inferenzprozess beim Konstruieren der Fuzzy-Variablen oder der Fuzzy-Regeln stoppen und den Dienst versagen. Aus diesen Gründen wurde ein weiteres Widget geschaffen, der LinguisticExpressionConstructor.

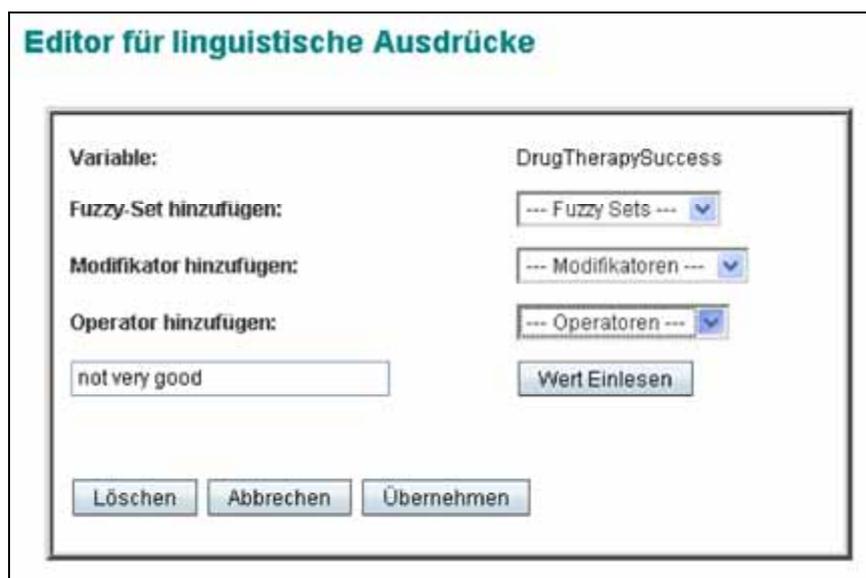


Abbildung 10.44: Der LinguisticExpressionConstructor

Er bietet mit Hilfe von Drop-down-Komponenten das Einfügen von gültigen Fuzzy-Sets für die jeweilige Fuzzy-Variable, Modifikatoren und Operatoren an. Sollte es dennoch notwendig oder gewünscht sein, selbst einen linguistischen Ausdruck zu schreiben oder den vorhandenen zu modifizieren, so kann man den selbst geschriebenen Text mit Hilfe der „Wert Einlesen“-Schaltfläche einlesen. Wenn man mit dem linguistischen Ausdruck zufrieden ist und diesen übernehmen möchte, klickt man auf die „Übernehmen“-Schaltfläche. An dieser Stelle findet eine Überprüfung statt, ob der linguistische Ausdruck für die Variable gültig ist. Falls nicht, erfolgt die Anzeige einer entsprechenden Fehlermeldung. Man sollte in diesem Fall überprüfen, ob man nur für die Variable definierte Fuzzy-Sets verwenden wollte (dazu kann man in der Drop-down-Komponente für die Fuzzy-Sets nachsehen, welche Einträge angeboten werden) oder, ob man sich bei einem der Modifikatoren oder Operatoren verschrieben hat.

## 10.3 Technische Implementierung

In diesem Kapitel werden die Datenbankstruktur, die Klassenstruktur, spezielle technische Aspekte und ausgewählte Entwurfsmuster (Design Patterns) detailliert besprochen. Dieses Kapitel richtet sich daher an Programmierer, welche die Funktionsweise von FuzzyDBSexpert im Detail verstehen möchten oder auch Tipps und Beispiele für die Lösung von bestimmten Aufgaben für eigene Programme suchen.

### 10.3.1 Überblick über die eingesetzten Technologien

Im Folgenden wird ein kurzer Überblick über die in FuzzyDBSexpert eingesetzten Technologien, jeweils mit Referenzen, gegeben. Alle diese Technologien sind kostenlos verfügbar, jedoch werden diese unter unterschiedlichsten Lizenzen herausgegeben.

#### 10.3.1.1 Datenbank

In FuzzyDBSexpert kommt eine MySQL-Datenbank [MySQL, 2007] in der Version 4.1 zum Einsatz. FuzzyDBSexpert stellt keine hohen Ansprüche an die Datenbank, was deren Funktionsumfang und Leistungsfähigkeit betrifft und verwendet auch keine proprietären MySQL-spezifischen Ausdrücke, sondern nur den SQL-92 Standard. Die gesamte Datenbank kann inklusive Demo-Daten durch Ausführen eines .SQL-Skripts, welches konform mit dem SQL-92 Standard ist, unter Verwendung von CREATE TABLE- und INSERT-Statements aufgebaut werden. So ist es zum Beispiel möglich, FuzzyDBSexpert zusammen mit einer Postgres-Datenbank [Postgres, 2007] zu betreiben, wenn diese dem Datenbankadministrator geläufiger ist.

### 10.3.1.2 Softwareentwicklung

Als Programmiersprache kommt in FuzzyDBSexpert Java von Sun zum Einsatz. Genauer sind dies hauptsächlich folgende Technologien:

- Java EE (Java 2 Platform, Enterprise Edition), Java EE Web Application Technologies [J2EE, 2007]

Java EE ist die Spezifikation einer Softwarearchitektur für die transaktionsbasierte Ausführung von Anwendungen, welche in der Programmiersprache Java geschrieben wurden. In dieser Spezifikation werden Softwarekomponenten und Dienste definiert, welche dazu dienen, einen allgemein akzeptierten Rahmen zur Verfügung zu stellen, um auf dessen Basis aus modularen Komponenten verteilte, mehrschichtige Anwendungen entwickeln zu können. Genau definierte Schnittstellen zwischen Komponenten und Schichten sollen dafür sorgen, dass Softwaremodule verschiedener Hersteller interoperabel sind, wenn diese gemäß der Spezifikation implementiert sind und dass verteilte Anwendungen gut skalierbar sind. Für FuzzyDBSexpert wird aber nur der Teil der Technologien aus dem Bereich der Web-Anwendungstechnologien, einem kleinen Bereich von Java EE, verwendet und soll an dieser Stelle genauer erläutert werden:

- JSF (JavaServer Faces 1.2)<sup>1</sup>

JavaServer Faces ermöglichen es, GUIs<sup>2</sup> für Web-Anwendungen zu programmieren, welche einfach zu erzeugen und flexibel in der Wahl der Komponenten für die Darstellung sind. JavaServer Faces realisieren implizit das MVC-Entwurfsmuster (Model-View-Controller Design Pattern, siehe Kapitel 10.3.3.1 und [Fowler, 2002, S. 114–117]). Seiten, welche mit JSF implementiert wurden, haben einerseits einen JSP-Teil, welcher die Visualisierung darstellt und andererseits einen Java-Teil, welcher die Business-Logik realisiert.

- JSP (JavaServer Pages 2.1)<sup>3</sup>

JavaServer Pages, welche zur Ausführung in Servlets übersetzt werden, bieten die Möglichkeit, einfach und schnell Webseiten mit dynamischem Inhalt zu generieren.

---

<sup>1</sup> <http://java.sun.com/javaee/javaxserverfaces/>

<sup>2</sup> Graphical User Interface (Benutzerschnittstelle)

<sup>3</sup> <http://java.sun.com/products/jsp/>

JSPs können bei Entwicklung von Webapplikationen mit dem Java Studio Creator 2 (siehe Kapitel 10.3.1.4) unter Verwendung von Drag-and-drop-Technik erzeugt werden, jedoch müssen zum Beispiel zur Lokalisierung oder bei komplizierteren Problemen die JSPs händisch modifiziert oder Ausdrücke in der Expression Language (ein Teil der JavaServer Pages Standard Tag Library (JSTL)) formuliert werden. Der Java Studio Creator 2 bietet zur direkten Bearbeitung der JSPs einen eingebauten XML-Editor.

- Servlets (Java Servlet 2.5)<sup>1</sup>

Servlets, in welche JSPs übersetzt werden, bieten einen einfachen und konsistenten Weg, Webserver in ihrer Funktionalität zu erweitern. Ein Servlet kann man beschreiben als Applet (ein kleines Programm, welches in Java geschrieben ist und in einem Webbrowser läuft), welches serverseitig läuft. Servlets müssen nicht direkt programmiert werden, jedoch sind Grundkenntnisse von Vorteil, um die Verzeichnisstruktur, den Deployment-Descriptor und die Funktionsweise der Servlets zu kennen und zu wissen, wie man beispielsweise Servlet Context Parameter und Filter verwendet.

- XML 1.0<sup>2</sup>

XML (Extensible Markup Language) ist zwar kein Teil der J2EE, wird hier jedoch sehr oft verwendet, wie zum Beispiel für diverse Konfigurationsdateien (Deployment-Descriptor `web.xml`, JSF-spezifische Dateien, wie `faces-config.xml`, `managed-beans.xml`, `navigation.xml`) oder für den Aufbau der .JSP-Dateien, welche auch XML-Dateien sind. Aus diesem Grund ist es wichtig, Grundkenntnisse in XML zu haben, was für Programmierer oder Informatiker obligatorisch sein sollte.

- J2SE 5.0 (Java Platform, Standard Edition) [J2SE, 2007]

Natürlich kommt auch herkömmliches Java zum Einsatz, einerseits zur Implementierung der verschiedensten notwendigen Klassen und andererseits als Basis für den Java-Teil der JavaServer Faces.

---

<sup>1</sup> <http://java.sun.com/products/servlet/>

<sup>2</sup> <http://www.w3.org/XML/>

– Fuzzy-Bibliothek

Als in Java implementierte Bibliothek, welche alle Fuzzy-Aspekte abdeckt, wurde das NRC FuzzyJ-Toolkit [NRCFuzzy], 2007] in der Version 1.10a ausgewählt. Das NRC FuzzyJ-Toolkit ist für Lehr- und Forschungszwecke kostenlos und wird in Form einer .JAR-Datei zur Verfügung gestellt.

### 10.3.1.3 Abgrenzung zu anderen Technologien

In diesem Kapitel soll erläutert werden, warum eine Entscheidung zugunsten der JSF-Spezifikation mit Java als Programmiersprache gegenüber anderen Web-Anwendungstechnologien und Programmiersprachen gefallen ist. Technologieentscheidungen sind in vielen Fällen nicht einfach zu treffen, vor allem nicht, wenn man prinzipiell offen gegenüber allen Technologien und Programmiersprachen ist und nicht davor zurückschreckt, sich bei Bedarf jede dieser Technologien in kurzer Zeit aneignen zu müssen.

Relativ bald war die Entscheidung gegen das ASP.NET Framework, bei dem viele verschiedene Programmiersprachen wie C# (CSharp), Visual Basic .NET oder J# (JSharp) zum Einsatz kommen können, gefallen, da diese Technologie proprietäre Software ist und es eine faktische Plattformabhängigkeit von dem Betriebssystem Microsoft Windows Server und den IIS (Microsoft Internet Information Server) und damit vergleichsweise hohe Kosten gibt, auch wenn das Projekt Mono<sup>1</sup> sehr weit fortgeschritten ist.

Als ernstzunehmende Programmiersprache für das Web wurde PHP<sup>2</sup> in Betracht gezogen, für das es auch einige Web-Anwendungs-Frameworks gibt. Diese ist, wie Java, Open Source und daher nicht von einer einzigen Firma abhängig. Prinzipiell wäre gegen den Einsatz von PHP nichts einzuwenden, wenn es sich um ein kleineres Projekt gehandelt hätte. Eine Internet- und Interviewrecherche hat ergeben, dass bei PHP eine gute Datenbankunterstützung und Internet-Protokolleinbindung gegeben ist, außerdem sind sehr viele Programmbibliotheken und vorgefertigte Musterlösungen für PHP vorhanden, da es sehr verbreitet ist. Der Schwachpunkt liegt jedoch in der Herkunft von PHP, es wurde als Skriptsprache, welche in HTML-Seiten eingebunden wird, entwickelt und erst später mit Objektorientierung erweitert. In Java ist die Objektorientierung ein zentraler Teil des Kerndesigns. Java besitzt echtes Exception-Handling und eine sehr leicht

---

<sup>1</sup> Mono ist eine Open-Source-Portierung des Microsoft-.NET-Frameworks (<http://www.mono-project.com>)

<sup>2</sup> PHP ist ein rekursives Backronym für „PHP: Hypertext Preprocessor“, eine serverseitig interpretierte (Script)-Sprache

verwendbare Lokalisierung, beides ist in PHP etwas schwierig zu handhaben. Die Skalier- und Wartbarkeit ist bei PHP auch sehr viel schlechter als bei der Java-Technologie. Bei Verwendung von PHP ist die Trennung zwischen dem Benutzer-Interface und der Business-Logik sehr unscharf, wo hingegen bei der J2EE mit Struts<sup>1</sup> oder JSF im Fall von FuzzyDBSexpert das MVC-Muster (siehe Kapitel 10.3.3.3.1) sehr gut umgesetzt werden kann und daher diese Dinge unabhängig voneinander geändert beziehungsweise skaliert werden können. Das wichtigste Argument für J2EE ist aber, dass alle Java-APIs verwendet werden können, also mächtige und komplexe Web-Anwendungen unter Verwendung von ebenfalls mächtigen und komplexen Bibliotheken realisiert werden können. So ist zum Beispiel mit dem NRC FuzzyJ-Toolkit eine Java-Bibliothek verfügbar, die einen Funktionsumfang hinsichtlich Fuzzy bietet, welcher mindestens eine eigene Masterarbeit darstellen könnte und im Rahmen dieser Arbeit nicht machbar gewesen wäre.

Nachdem die Entscheidung für Java als Programmiersprache gefallen war, galt es noch, sich zwischen den verschiedensten Webanwendungstechnologien zu entscheiden. Zur Auswahl standen unter anderem Struts, Spring<sup>2</sup>, WebWork<sup>3</sup>, Tapestry<sup>4</sup> und JSF. Auch zwischen diesen Technologien ist es nicht einfach, sich zu entscheiden, zumal alle diese Technologien ernst zu nehmen sind, aber gleichsam vor Entwicklung des Projektes dem Autor mehr oder weniger unbekannt waren. Nach längerem Studium von Vergleichen dieser Technologien im Internet und Einholen von Ratschlägen von Webanwendungsprogrammierern fiel die Entscheidung schließlich zugunsten JSF, hauptsächlich, weil es eine vielversprechende, neue und mächtige Technologie, welche das MVC-Konzept realisiert, ist und mit dem Sun Java Studio Creator 2 eine sehr gute kostenlose Entwicklungsumgebung verfügbar ist, welche direkt für die Entwicklung mit JSF geschaffen wurde. Außerdem wurde durch die Entwicklung von Sun ein Quasi-Standard für Web-Anwendungen geschaffen, welcher sich nahtlos in das Java-Umfeld eingliedert.

#### 10.3.1.4 Entwicklungsumgebung

Als Entwicklungsumgebung kommt der Sun Java Studio Creator 2 Update 1 [Creator, 2007] zum Einsatz. Diese Entwicklungsumgebung wurde von Sun Microsystems für die Entwicklung von Webprogrammen, welche JavaServer Faces einsetzen, entwickelt und ist daher relativ neu und stetiger Weiterentwicklung unterworfen. Begonnen wurde die Entwicklung von FuzzyDBSexpert mit dem Java Studio Creator 2 Beta, dazwischen wurde das Final Release dieser Version

---

<sup>1</sup> <http://struts.apache.org/>

<sup>2</sup> <http://www.springframework.org/>

<sup>3</sup> <http://www.opensymphony.com/webwork/>

<sup>4</sup> <http://tapestry.apache.org/>

herausgegeben und schließlich wurde ein Branch namens Update 1, wobei der originale Branch wohl früher oder später auslaufen wird, verwendet. Leider gibt es aufgrund des frühen Stadiums des Creators noch einige Fehler in dem Programm, auf die man bei der Entwicklung Rücksicht nehmen muss, um sich unnötigen Zeitaufwand zu sparen.

### 10.3.1.5 Web-Container

Als serverseitige Ablaufumgebung kommt während der Entwicklungszeit der Sun Application Server 8.2<sup>1</sup> zum Einsatz, welcher beim Java Studio Creator 2 mitgeliefert wird und eng mit dieser IDE im Zuge der Entwicklung zusammenarbeitet. Dieser ist ein Anwendungsserver, eine Software, die als Ablaufumgebung für Anwendungsprogramme diesen spezielle Dienste zur Verfügung stellt, wie beispielsweise Transaktionen, Authentifizierung oder Datenbanken über definierte Schnittstellen. Teil dieses Anwendungsservers ist eine Servlet/JSP-Engine und Teil davon ist der Webserver, also jene Basis, welche die Informationen über das HTTP-Protokoll zur Verfügung stellt.

In der Praxis ist der Sun Application Server meiner Erfahrung nach für diesen Anwendungszweck zu ressourcenhungrig, was den Arbeitsspeicherbedarf betrifft, langsam und instabil und deswegen sollte eine andere serverseitige Ablaufumgebung zum Einsatz kommen, wie zum Beispiel der Apache Tomcat<sup>2</sup>, welcher eine Servlet/JSP-Engine darstellt, mit oder ohne Apache HTTP-Server<sup>3</sup>, einem sehr bekannten und verbreiteten Webserver, als Basis. Im Prinzip kann jeder Webserver (zusammen) mit einer JSP-Engine eingesetzt werden, welcher für JSF konfiguriert werden kann. In der Praxis kommt auf fast 60 % der Webserver weltweit der Apache HTTP-Server zum Einsatz [Netcraft, 2007].

## 10.3.2 Datenbank

### 10.3.2.1 Entwurf

Der konzeptionelle Entwurf der Datenbank wurde mit Hilfe des Entity-Relationship-Modells (ERM) durchgeführt. Aus Erfahrung und unter Verwendung des ERM befand sich das abgeleitete Relationenschema bereits in der vierten Normalform (4NF), Details zu den Normalformen finden sich in [Heuer, 2000, S. 238–260] und [Kemper, 2006, S. 181–193].

---

<sup>1</sup> <http://developers.sun.com/prodtech/appserver/>

<sup>2</sup> <http://tomcat.apache.org/>

<sup>3</sup> <http://httpd.apache.org/>

Aufgrund des vorwiegend streng hierarchischen Aufbaus, welcher sich als Konsequenz aus dem Aufbau der Variablen (siehe Kapitel 10.1.1.2.1), der Regeln (siehe Kapitel 10.1.1.2.2) und der UPDRS (siehe Kapitel 10.1.1.1.2), sowie deren Zusammenfassung in Sets ergibt, treten hauptsächlich 1:n-Beziehungen auf.

Einige Details des Datenbankschemas wurden im Laufe der Entwicklung der Software geändert, nachdem sie sich als unpraktikabel oder zusätzlich notwendig erwiesen haben, da dies beim Entwurf nicht absehbar war, deswegen wurde das Datenbankschema in mehreren Iterationen verändert.

Als Beispiel einer notwendigen Erweiterung soll die Modellierung des Geschlechts genannt werden. So war beispielsweise das Geschlecht zuerst nicht als eigene Datenbanktabelle vorgesehen, die Lokalisierung sollte mit Hilfe der Property-Datei geschehen (siehe Kapitel 10.2.4.1.2). Dafür war nur ein bezeichnender Schlüssel für das Geschlecht notwendig. Es erwies sich in der Praxis jedoch als einfacher, die Lokalisierung über eine statische, nicht von einem Benutzer änderbare, Datenbanktabelle durchzuführen (siehe Kapitel 10.2.4.1.4). Dadurch wurde es notwendig, eine zusätzliche Tabelle für die Texte zur Bezeichnung der Geschlechter in allen unterstützten Sprachen hinzuzufügen. Eine Tabelle für die Sprachen war bis zuletzt auch nicht vorgesehen. Nur die universelle Erweiterbarkeit machte es erforderlich, eine Tabelle für die Sprachen anzulegen, aus der zum Beispiel die Optionen-Liste zur Sprachauswahl auf der ersten Seite von FuzzyDBSexpert generiert wird (siehe Kapitel 10.2.4.1).

Als Beispiel einer sich als überflüssig erwiesenen Tabelle wird hier eine im Anfangsstadium vorgesehene Tabelle für Variablen-Typen genannt. In dieser sollten die verschiedenen Variablentypen (siehe Tabelle 10-1) mit dem entsprechenden Namen und der Eingabemethode verzeichnet sein. So war diese Tabelle unter anderem dafür vorgesehen, bei der Auswahl der Variablen für die Regel-Teile zu helfen, also für die Auswahl der Bedingungsteil- und Schlussfolgerungsteil-Variablen. Es geht bei der Konstruktion der Regel-Teile durch den Benutzer aber nicht direkt um die Variablentypen, also Eingangs- und Ausgangsvariablen, sondern darum, die Variablen eingeschränkt durch die Definition des Regel-Teils als Bedingungs- oder Schlussfolgerungsteil zur Auswahl anzubieten. In der Software wird weiters an manchen Stellen die Auswahl von einfachen Eingangsgrößen, an anderen Stellen der automatisierten UPDRS-Eingangsgrößen (aber nicht der UPDRS-Summe) und an wieder anderen Stellen von allen Eingangsgrößen benötigt. Natürlich hätte dies alles über eine zusätzliche Identifikation für alle Eingangsgrößen in der Tabelle für die Variablentypen modelliert werden können. Dies hätte aber die zweite Normalform (2NF), welche keine partiellen Abhängigkeiten von Schlüsseln und Attributen erlaubt, verletzt, und es wäre für diese Abbildung eine eigene Tabelle notwendig gewesen. Nachdem dies insgesamt keine Vereinfachung und keinen zusätzlichen Nutzen ergibt, wurde die Tabelle für die Speicherung der Variablentypen wieder verworfen.

Das grundlegende Konzept des hierarchischen Aufbaus der Variablen, Regeln und der UPDRS und der Zusammenfassung in Sets wurde jedoch nie verändert.

### 10.3.2.2 Primärschlüssel

Die Daten- und Datenbankstruktur von FuzzyDBSexpert ist größtenteils streng hierarchisch aufgebaut. Dies ergibt sich als Konsequenz aus dem Aufbau der Variablen (siehe Kapitel 10.1.1.2.1), der Regeln (siehe Kapitel 10.1.1.2.2) und der UPDRS (siehe Kapitel 10.1.1.1.2), sowie deren Zusammenfassung in Sets. Aus diesem Grund hat sich die Verwendung von zusammengesetzten Datenbankschlüsseln (compound oder composite keys) angeboten. Zusammengesetzte Datenbankschlüssel setzen sich aus mindestens zwei Attributen zusammen, deren Kombination eindeutig sein muss. Jedes dieser Attribute kann ein Fremdschlüssel sein und diese Fremdschlüssel können selbst wieder zusammengesetzte Schlüssel sein. Dies soll anhand der Hierarchie der Regeln bis zur Tabelle der Regel-Teil-Atome veranschaulicht werden:

Tabelle 10-7: Beispiel für zusammengesetzte Datenbankschlüssel

Tabelle	Primärschlüssel
RuleSets	RuleSetID
RuleSetRules	RuleSetID + RuleID
RuleParts	RuleSetID + RuleID + RulePartID
RulePartAtoms	RuleSetID + RuleID + RulePartID + RulePartAtomID

In obigem Beispiel kann man erkennen, dass Primärschlüssel ab der dritten Hierarchiestufe sowohl selbst zusammengesetzte Schlüssel sind, als auch zusammengesetzte Schlüssel enthalten.

Diese zusammengesetzten Schlüssel haben einen weiteren Vorteil für die Programmierung. In FuzzyDBSexpert sind Regeln, Variablen-Sets und UPDRS-Sets intuitiv hierarchisch über die Benutzerschnittstelle aufgebaut. Wird zum Beispiel eine neue Regel für ein Regel-Set erstellt, dann erhält diese Regel mehrere Regel-Teile und diese Regel-Teile erhalten schließlich die entsprechenden Atome. Es wird so gesehen immer in diesen Ebenen navigiert und die Tabellen weiter unten in der Hierarchie erhalten jeweils ein zusätzliches Schlüsselattribut. Dies lässt sich nun sehr gut mit Hilfe eines Pools an Schlüsseln modellieren, wobei ein Attribut oder mehrere Attribute fix und nur eines davon variabel ist. So sind beispielsweise für Regel-Teile einer bestimmten Regel die RuleSetID und RuleID immer gleich, jedoch hat jeder Regel-Teil ein in Kombination mit diesen ein eindeutiges Schlüsselattribut RulePartID. Dies wird in FuzzyDBSexpert nun so modelliert, dass

die fixen Attribute in einer *Map*<sup>1</sup> gespeichert und bei Bedarf auf die SQL-Abfragen angewendet werden. Das jeweils aktuelle variable Schlüsselattribut wird extra geführt, da es für verschiedenste Zwecke, beispielsweise beim Hinzufügen von neuen Einträgen, verwendet wird. Beim Hinuntersteigen in der Hierarchie wird das aktuelle variable Schlüsselattribut in diesen Schlüssel-Pool aufgenommen und bei Hinaufsteigen in der Hierarchie aus diesem wieder entfernt (siehe Kapitel 10.3.3.3).

### 10.3.2.3 Konzeptuelle Modellierung mit UML<sup>2</sup>

Die folgenden Diagramme wurden mit Hilfe von UML2 [UML, 2007] modelliert. Die Modellierung erfolgt in Anlehnung an die konzeptionelle Modellierung mit UML in [Kemper, 2006, S. 57–62]. In den folgenden Diagrammen werden die Angaben der Sichtbarkeit und Datentypen weggelassen, da die Übersichtlichkeit sehr darunter leiden würde. Weiters werden die Bezeichnungen für jene Assoziationen weggelassen, bei welchen es sich um 1:n-Beziehungen handelt, wo es um die exklusive Zuordnung von existenzabhängigen Teil-Objekten zu genau einem übergeordneten Objekt, also um eine Komposition, handelt. Diese exklusive Zuordnung existenzabhängiger Unterobjekte wird in UML mit der ausgefüllten Raute (◆) auf der Seite der übergeordneten Klasse angegeben und – speziell in den folgenden Diagrammen – wird die logisch zu folgernde Multiplizität 1 (die Multiplizität 1...1 wird vereinfacht mit 1 dargestellt) auf dieser Seite der Beziehung wegen der Übersichtlichkeit weggelassen. Die Multiplizität auf der anderen Seite kann, sofern nicht anders angegeben, mit \* (Multiplizität  $n$ ,  $n \in \mathbb{N}$ ) angenommen werden.

Zusätzlich werden in den folgenden Diagrammen die Primärschlüssel (Primary Keys, PK) und Fremdschlüssel (Foreign Keys, FK) benannt, was in UML nicht vorgesehen ist, damit man leichter auf das darunter liegende Datenbankmodell schließen kann. Primärschlüsselattribute werden zusätzlich unterstrichen.

Fett gedruckte Attribute sind zwingend erforderlich, etwa, weil sie Bestandteil von Primär- oder Fremdschlüsseln sind oder einfach zweckmäßig angegeben werden müssen.

---

<sup>1</sup> Eine *Map* ist eine Datenstruktur, welche Schlüssel auf Werte abbildet. Eine *Map* kann keine doppelten Schlüssel enthalten und jeder Schlüssel darf maximal einen Wert abbilden.

<sup>2</sup> Unified Modeling Language

In den folgenden Diagrammen kann es sein, dass eine Entität in mehreren Diagrammen auftritt. Dies dient zur besseren Visualisierung der Zusammenhänge und bedeutet natürlich nicht, dass diese Tabelle doppelt vorhanden ist.

#### **10.3.2.3.1 Datenbankmodell des Wissensbasis-Teiles**

Im Diagramm des Datenbankmodells des Wissensbasis-Teiles (siehe Abbildung 10.45) wird jener Teil der Datenbank dargestellt, welcher zur Speicherung der Wissensbasis, also hauptsächlich der Regeln und Fuzzy-Sets zu den Variablen, beiträgt.



### 10.3.2.3.2 Datenbankmodell des UPDRS-Teiles

Im Diagramm des Datenbankmodells des UPDRS-Teiles der Datenbank (siehe Abbildung 10.46) werden alle Tabellen dargestellt, welche zur Speicherung der UPDRS, deren Optionen, Gewichte und Werte der Patientendatensätze in Bezug auf die UPDRS dienen.

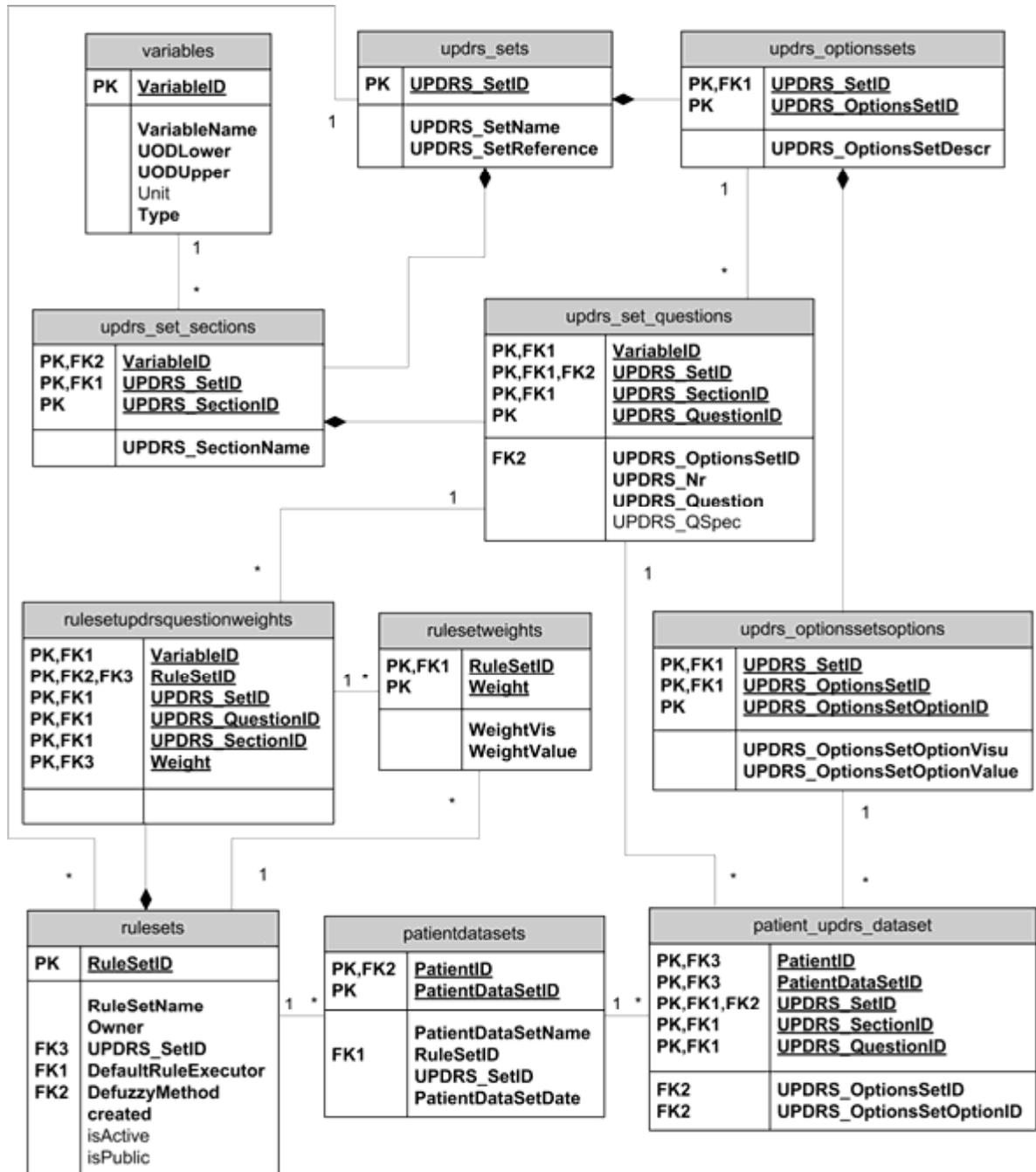


Abbildung 10.46: Datenbankmodell des UPDRS-Teiles

### 10.3.2.3.3 Datenbankmodell des Benutzer- und Patienten-Teiles

Im folgenden Diagramm (siehe Abbildung 10.47) werden die Tabellen, welche die Benutzer- und Patientendaten inklusive den Patienten-Datensätzen aufnehmen, gezeigt.

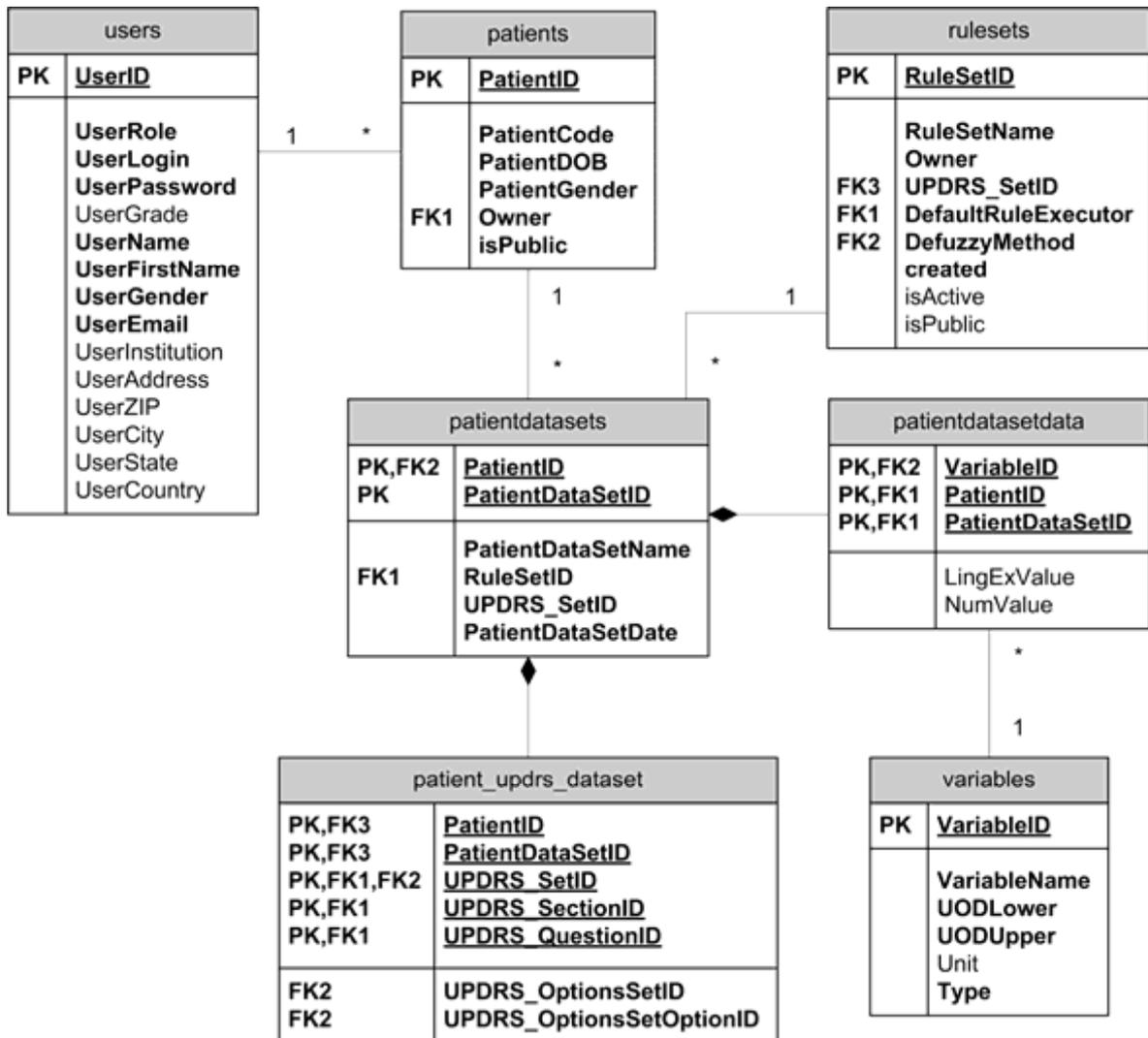


Abbildung 10.47: Datenbankmodell des Benutzer- und Patienten-Teiles

### 10.3.2.3.4 Datenbankmodell der Tabellen zur Lokalisierung

In diesem Kapitel werden die Tabellen für die Verwendung zur statischen datenbankunterstützten Lokalisierung (siehe Kapitel 10.2.4.1.4) beschrieben (siehe Abbildung 10.48). An dieser Stelle ist eine kleine Besonderheit zu nennen. Normalerweise werden, soweit möglich, Attribute, welche nur ganz bestimmte Werte annehmen dürfen, über Fremdschlüsselbeziehungen modelliert. Bei den Sprachen ist es nun so, dass für mehrere Sprachen, beispielsweise österreichisches und deutsches Deutsch oder amerikanisches und britisches Englisch, das gleiche *Locale*-Objekt (zum Beispiel instanziiert mit dem Konstruktor (de) oder (en), siehe Kapitel 10.2.4.1.1) verwendet werden kann,

wenn man kleine Sprachdifferenzen ignoriert, was sehr viel redundante Schreibarbeit vermeidet. Deswegen kann dieser Sprachcode nicht als Schlüssel in der Datenbank verwendet werden, da amerikanisches und britisches Englisch, falls notwendig, unterschieden werden müssen. Aus diesem Grund wurde ein künstlicher, aufsteigender, ganzzahliger Schlüssel für die Tabelle der Sprachen gewählt, zur Lokalisierung aber mit den Sprachkürzeln gearbeitet. Sollte es einmal zur Diskriminierung zwischen amerikanischem und britischem Englisch kommen, müsste man die gesamte Lokalisierung nachziehen. Derzeit wird das Sprachkürzel „en“ für beide Sprachen verwendet und deswegen kann das Sprachkürzel auch nicht als Fremdschlüssel dienen, da es kein Primärschlüssel und damit auch nicht eindeutig ist, es kann ja für britisches oder amerikanisches Englisch stehen. Dieses Problem wurde dadurch umgangen, dass die Attribute für die Sprachkürzel in den Lokalisierungstexttabellen nicht mit einem Fremdschlüssel beschränkt werden, aber zumindest einen Teil des Primärschlüssels darstellen. Somit ist sichergestellt, dass pro zu lokalisierendem Text nur ein einziger Eintrag für jede Sprache existieren kann. Diese „lose Bindung“ der Texte zu den zu lokalisierenden Einträgen, da zusätzlich noch das Sprachkürzel im Primärschlüssel verwendet aber nicht als Fremdschlüssel modelliert wird, ist in dem folgenden Diagramm mit Hilfe von strichlierten Linien dargestellt.

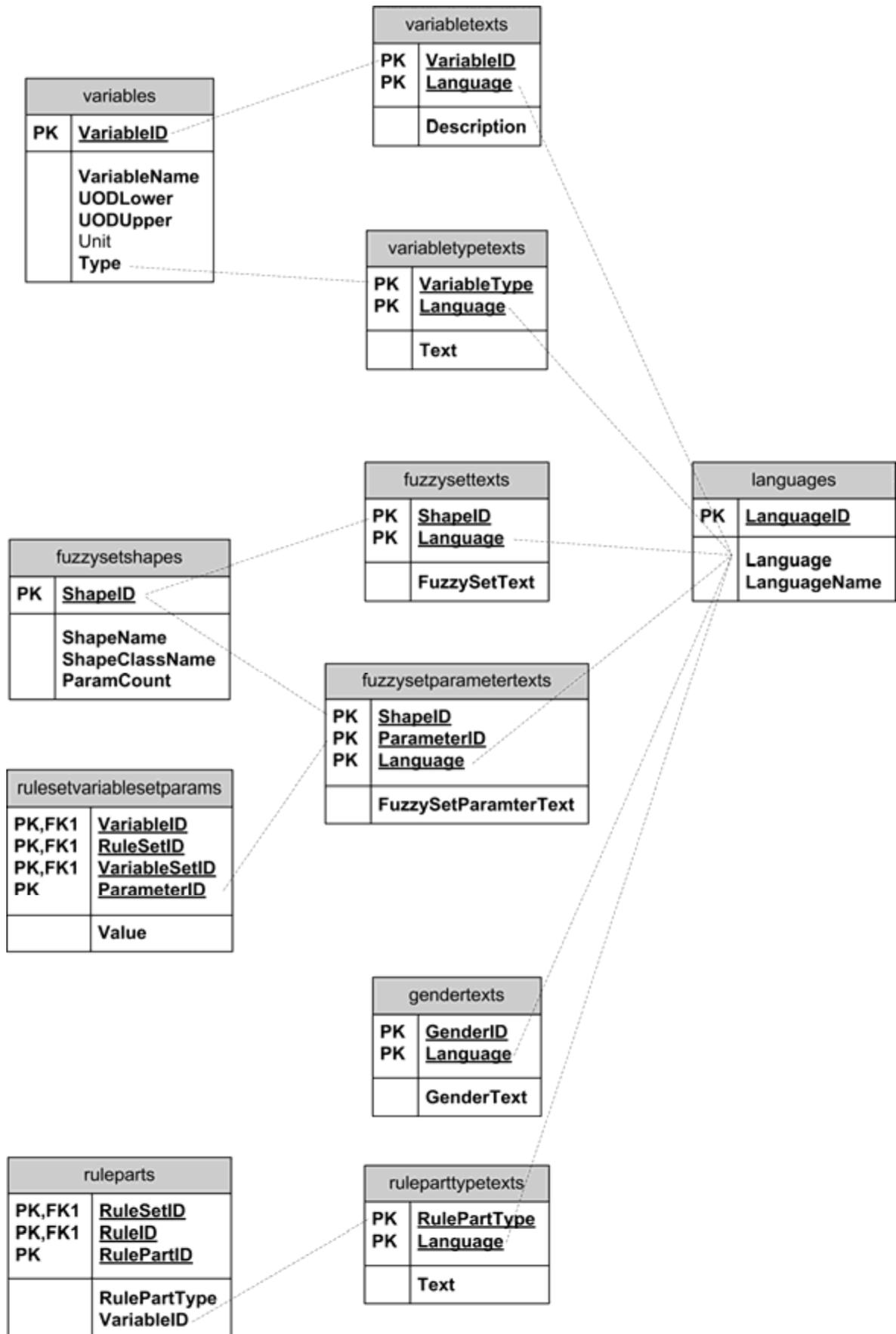


Abbildung 10.48: Datenbankmodell der Tabellen zur Lokalisierung

### 10.3.2.4 SQL-Script

Die SQL-Script-Datei dient dazu, die Datenbank inklusive der Demo-Daten mit Hilfe von Standard SQL-92 Befehlen automatisiert aufbauen zu können. SQL-Scripts sind Textdateien, welche SQL-Anweisungen enthalten, wie `CREATE TABLE` zum Erstellen von Tabellen oder `INSERT INTO <Tabelle> VALUES`, um diese Tabellen mit Inhalten zu füllen. Üblicherweise gibt es in den Datenbankprogrammen auf Kommandozeilenebene oder in einem grafischen Verwaltungs-Tool Funktionen, welche es ermöglichen, .SQL-Dateien zu laden und auszuführen. In MySQL kann man SQL-Scripts mit Hilfe des folgenden Befehls von der MySQL-Kommandozeile aus aufrufen:

```
mysql> source <Dateiname>
```

Die SQL-Script-Datei ist in gedruckter Form im Anhang zu finden, aber ohne Demo-Daten, da dies zu lange wäre – über 1000 Zeilen – und für den Entwickler nicht so interessant ist im Gegensatz zu den Primär- und Fremdschlüsseln, Datentypen und sonstigen Beschränkungen.

### 10.3.3 Programm

Die verwendeten Technologien und die IDE wurden schon in Kapitel 10.3.1.2 beschrieben. Im Rahmen dieser Diplomarbeit kann und soll auch keine weitere Beschreibung dieser Technologien stattfinden. Die mir zur Einarbeitung beziehungsweise als dauernde Referenz dienende Literatur, wird im Kapitel 10.3.4 diskutiert. In diesem Teil der vorliegenden Arbeit werden nur prinzipielle Modelle der Daten und der Software, meist in Form von UML-Klassendiagrammen, vorgestellt und es wird auf einige interessante Teilaspekte, also spezielle interessante Lösungen oder Probleme, eingegangen.

#### 10.3.3.1 Grundlegende Konzepte

FuzzyDBSexpert besitzt Datenstrukturen, welche Variablen, Regeln und die UPDRS auf verschiedensten Ebenen repräsentieren:

- Datenbank (siehe voriges Kapitel)
- Templates für Fuzzy-Datenstrukturen

Diese Templates dienen dazu, jene Informationen aufzunehmen, welche für das jeweilige Element, zum Beispiel eine Variable, benötigt werden und sollen die Funktionalität von jener der Fuzzy-Bibliothek abstrahieren, damit weniger Aufwand notwendig ist, eine andere Fuzzy-Bibliothek verwenden zu können. Dieses Muster wird Fassade (Facade) genannt und ist in [GoF<sup>1</sup>, 1995, S. 174–183] beschrieben. Die *FuzzyEngine* stellt in diesem Fall eine Fassade für die *FuzzyEngineBean* bereit, welche das FuzzyJ-Toolkit vor der *FuzzyEngineBean* versteckt. Die Informationen in diesen Datenstrukturen sind ähnlich jenen, die in der Datenbank stehen.

#### – Konkrete Fuzzy-Strukturen

Die Templates für die Fuzzy-Datenstrukturen müssen im Laufe der Inferenz in konkrete Fuzzy-Datenstrukturen der jeweiligen verwendeten Bibliothek (in diesem Fall das NRC FuzzyJ-Toolkit) umgewandelt werden, damit diese eine Funktionalität erhalten, mit der zum Beispiel eine Fuzzy-Inferenz oder eine Defuzzifizierung durchgeführt werden kann.

Diese Trennung und Interaktion der Datenstrukturen soll anhand folgender Grafik veranschaulicht werden:

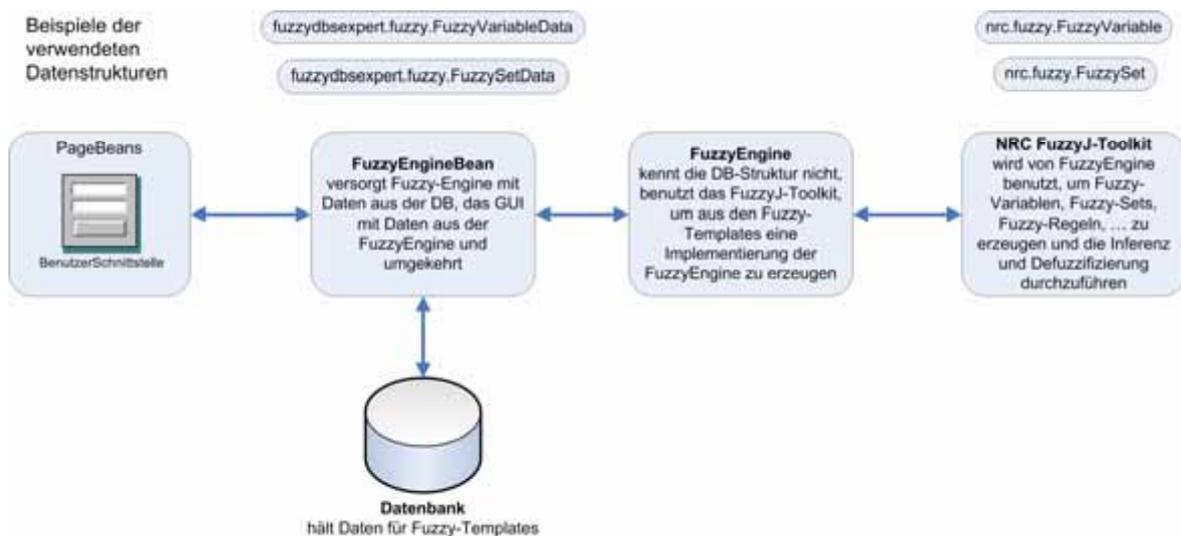


Abbildung 10.49: MVC-Konzept des Fuzzy-Teils von FuzzyDBSexpert

Diese Struktur realisiert insgesamt das MVC-Konzept (Model-View-Controller), welches in [Fowler, 2002, S. 114–117] beschrieben ist, wobei die einzelnen Bestandteile möglichst strikt

<sup>1</sup> GoF ist ein bekanntes Akronym für „Gang of Four“, wie diese Gruppe der vier Autoren gerne bezeichnet wird.

voneinander getrennt sein sollten (siehe Abbildung 10.50). Details zu deren Funktionalität sind den entsprechenden Klassendiagrammen in den folgenden Kapiteln zu entnehmen.

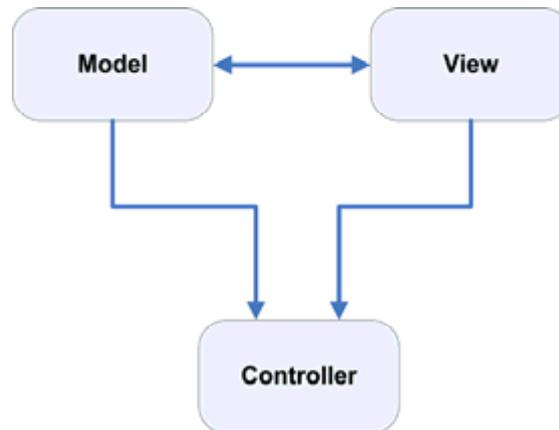


Abbildung 10.50: MVC Muster [vgl. Fowler, 2002, S. 114]

– Model

Das Model ist ein Objekt, welches domänenspezifische Informationen repräsentiert und alle Daten und Verhalten enthält, welche nicht für die Benutzerschnittstelle notwendig sind [vgl. Fowler, 2002, S. 114]. Im Fall von *FuzzyDBSexpert* ist das Model die Datenbank beziehungsweise auch die daraus resultierenden Strukturen der Fuzzy-Templates, sowie die *FuzzyEngine*, die unter Benutzung des NRC FuzzyJ-Toolkits, den Großteil der Business-Logik des Fuzzy-Teils darstellt.

– View

Die View repräsentiert die Anzeige des Models in der Benutzerschnittstelle und dient nur zur Visualisierung. Alle Änderungen der Daten werden vom dritten Teil des MVC-Musters verwaltet – dem Controller [vgl. Fowler, 2002, S. 115]. Die View sollte vollkommen unabhängig von der Business-Logik und dem zugrunde liegenden Model sein. Die Views sind im Fall von *FuzzyDBSexpert* die JSPs, welche Page-Beans im Hintergrund haben. JavaBeans sind Java-Klassen, welche als Container zur Datenübertragung verwendet werden. Die Page-Beans, welche, wie der Name schon sagt, zu entsprechenden Seiten, also JSPs gehören, sind selbst Bestandteil eines MVC-Musters, nämlich dem der JavaServer Faces, welche implizit das MVC-Muster realisieren.

– Controller

Der Controller nimmt die Benutzereingaben entgegen und aktualisiert das Model entsprechend [vgl. Fowler, 2002, S. 115]. Der Controller ist im Fall von *FuzzyDBSexpert* die *FuzzyEngineBean*, welche mit der *FuzzyEngine* interagiert.

### 10.3.3.2 Pakete

In FuzzyDBSexpert werden, wie in jedem anderen nicht allzu kleinem Java-Projekt, die Klassen in verschiedene Pakete aufgeteilt, welche in diesem Fall alle im Paket *fuzzydbsexpert* liegen:

- *adminweb, expertweb* und *userweb*

Diese drei Pakete enthalten die Java-Klassen, welche für die einzelnen Seiten verwendet werden, sind also hauptsächlich Page-Beans. Beim Java Studio Creator 2 ist in den meisten Tutorials und Beispielen vorgesehen, dass neue Seiten direkt im Hauptverzeichnis angelegt werden, doch das wird bei einer großen Anzahl von Seiten unübersichtlich.

- *data*

Im Paket *data* befinden sich Klassen, welche Daten repräsentieren, zum Beispiel Werte für die Patienten-Eingangs- und Ausgangsvariablen.

- *fuzzy*

Im Paket *fuzzy* befinden sich prinzipiell alle Klassen, welche mit dem Begriff „fuzzy“ zu tun haben, zum Beispiel Templates für die Fuzzy-Strukturen oder die *FuzzyEngineBean* und die *FuzzyEngine*.

- *locale*

Im Paket *locale* befinden sich die Properties-Dateien (siehe Kapitel 10.2.4.1.2), je eine pro Sprache, und eine Bean, welche das Festlegen der Sprache und das Erhalten eines lokalisierten Textes aus den Properties-Dateien unterstützt.

- *tools*

Das Paket *tools* enthält Klassen, welche das Handling von Session-Timeouts vereinfachen. Das Problem der Session-Timeouts wird in Kapitel 10.3.3.5 näher erläutert.

- *user*

Im Paket *user* befinden sich Klassen für die Benutzerverwaltung und deren Authentifizierung.

- *widget*

Im Paket *widget* befinden sich die Klassen für das DatabaseCRUDPanel (siehe Kapitel 10.2.4.3 und 10.3.3.3) und den LinguisticExpressionConstructor (siehe Kapitel 10.2.4.5 und 10.3.3.4).

Abbildung 10.51 zeigt die Pakete und Klassen von FuzzyDBSexpert in einem UML-Diagramm:

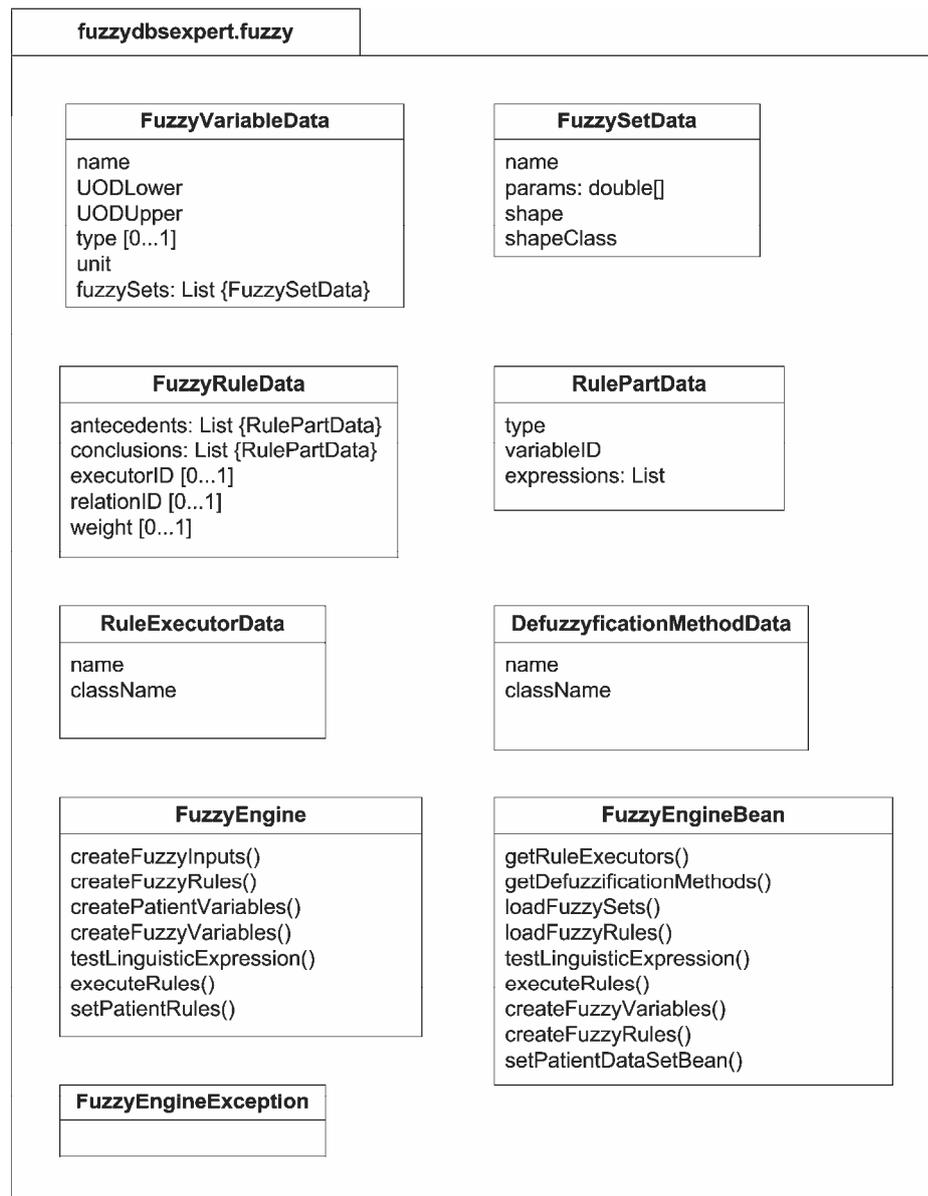


Abbildung 10.51: Pakete und Klassen in FuzzyDBSexpert

Einige der Pakete, welche speziell für FuzzyDBSexpert interessant sind, werden in den folgenden Kapiteln detaillierter beschrieben. Das obige Klassendiagramm und die folgenden Klassendiagramme sind in der UML2 Notation [UML, 2007] modelliert, aber nur besonders interessante Attribute, deren Datentypen, Wertebereiche und Methoden eingezeichnet.

#### **10.3.3.2.1 Paket *fuzzydbsexpert.fuzzy***

Das Paket *fuzzydbsexpert.fuzzy* enthält alle Klassen (siehe Abbildung 10.52), welche mit dem Begriff „fuzzy“ zu tun haben.

Abbildung 10.52: Paket *fuzzydbsexpert.fuzzy*

Dies sind zum Großteil einerseits Klassen, welche Fuzzy-Templates darstellen, und andererseits Klassen, welche die Business-Logik, die mit diesen Templates und dem NRC FuzzyJ-Toolkit arbeiten, implementieren.

- Fuzzy-Templates

- *FuzzyVariableData*

Diese Klasse enthält alle Informationen, welche eine Fuzzy-Variable charakterisieren. Sie enthält am Anfang, wenn noch keine Fuzzy-Sets geladen wurden, keine Liste mit den Fuzzy-Sets, sondern nur die Informationen, welche für alle Benutzer gleich sind. Erst, wenn die Fuzzy-Sets für diese Variable aus einem bestimmten Regel-Set geladen

werden, enthält die Klasse *FuzzyVariableData* eine Liste mit den Elementen vom Typ *FuzzySetData*.

- *FuzzySetData*

Diese Klasse besitzt alle Informationen, welche zur Charakterisierung eines Fuzzy-Sets notwendig sind. Besonders hervorzuheben ist hier das Attribut *shapeClass*, welches den Klassennamen der Form angibt. Diese Klasse muss, bevor die Inferenz stattfinden kann, im Zuge der Umwandlung in die konkreten Fuzzy-Strukturen des NRC FuzzyJ-Toolkits mit Hilfe von *Reflection* dynamisch erzeugt werden (siehe Kapitel 10.3.3.6).

- *FuzzyRuleData*

Intuitiv enthält diese Klasse alle Informationen, welche zusammen eine Fuzzy-Regel ausmachen. Sie besitzt unter anderem Listen von Bedingungs- und Schlussfolgerungsteilen von der Klasse *RulePartData* und einen Verweis zu einem Regel-Exekutor, falls diese Regel einen vom standardmäßig definierten Regel-Exekutor abweichenden definiert hat.

- *RulePartData*

Diese Klasse stellt die einzelnen Bedingungs- und Schlussfolgerungsteile dar und besitzt deswegen einen Verweis auf eine bestimmte Variable, auf die sich dann die einzelnen Ausdrücke beziehen. Die Ausdrücke sind in Form einer Liste (*List*-Klasse) gespeichert, deren Einträge in der richtigen Reihenfolge eingefügt werden, da es hier auf die richtige Abfolge der Ausdrücke ankommt.

- *RuleExecutorData*

Die Regel-Exekutoren müssen, bevor die Inferenz stattfinden kann, in eine konkrete Klasse des NRC FuzzyJ-Toolkits umgewandelt werden, was dynamisch über *Reflection* geschieht (siehe Kapitel 10.3.3.6).

- *DefuzzificationMethodData*

Die Defuzzifizierungsmethode im NRC FuzzyJ-Toolkit ist eine Methode, welche auf Fuzzy-Variablen, also in diesem Fall die Fuzzy-Ausgangsvariablen, angewendet wird. Diese Methode wird auch dynamisch über *Reflection* aufgerufen (siehe Kapitel 10.3.3.6).

– Business-Logik

○ *FuzzyEngineBean*

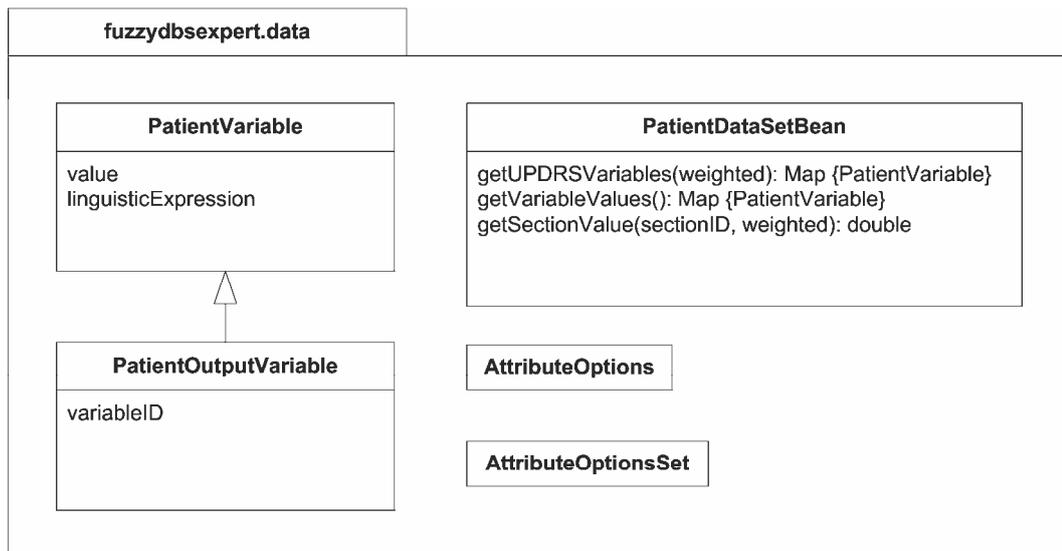
Die *FuzzyEngineBean* ist – zusammen mit der benutzten *FuzzyEngine* – eine der wichtigsten Klassen, welche, wie schon in Kapitel 10.3.3.1 dargestellt, die Business-Logik für den Fuzzy-Teil von *FuzzyDBSexpert* implementiert. Sie hat Zugriff auf die Datenbank und kann daher alle Fuzzy-Templates laden und aus den Daten die entsprechenden Strukturen erstellen. Weiters werden von ihr die Patientendaten verwaltet und die Inferenz gestartet, welche die Ausgangsvariablen mit Werten versieht, falls nicht das Auftreten einer *FuzzyEngineException* dies verhindert. Sie wird auch zum Testen auf die Gültigkeit eines linguistischen Ausdrucks verwendet, da hierfür die komplette Fuzzy-Variable mit allen Fuzzy-Sets des NRC FuzzyJ-Toolkits notwendig ist. Diese Funktion wird im *LinguisticExpressionConstructor* (siehe Kapitel 10.3.3.4) verwendet.

○ *FuzzyEngine*

Die *FuzzyEngine* wird nur von der *FuzzyEngineBean* benutzt und nie von außerhalb, also zum Beispiel direkt von den dargestellten Seiten aus (siehe auch Abbildung 10.49). Die *FuzzyEngine* stellt die eigentliche Inferenzmaschine, welche alle für die Inferenz notwendigen Daten enthält, dar. Sie benutzt direkt das NRC FuzzyJ-Toolkit, um aus den Fuzzy-Templates konkrete Klassen mit dem Toolkit zu erstellen und deren Methoden zu verwenden. Daher benötigt sie auch keinen Zugriff auf die Datenbank, weil sie alle notwendigen Informationen von der *FuzzyEngineBean* erhält. Auf diese Weise versteckt die *FuzzyEngine* die Implementierungsdetails und das NRC FuzzyJ-Toolkit vor der *FuzzyEngineBean*. Dieses Entwurfsmuster wird Fassade genannt (siehe Kapitel 10.3.3.1) und ist in [GoF, 1995, S. 174–183] näher beschrieben.

### 10.3.3.2 Paket *fuzzydbsexpert.data*

Das Paket *fuzzydbsexpert.data* (siehe Abbildung 10.53) enthält Klassen, welche Patientendaten repräsentieren, also hauptsächlich Variablen und jene Klassen, die diese benutzen.

Abbildung 10.53: Paket *fuzzydbsexpert.data*

– *Patientvariable*

Die Klasse *Patientvariable* repräsentiert den Wert einer Variablen, soll also hauptsächlich den Umstand modellieren, dass eine Variable entweder einen numerischen Wert oder einen linguistischen Ausdruck als Wert haben kann, aber nie beides gleichzeitig.

– *PatientOutputVariable*

Eine Patienten-Ausgangvariable ist eine Patientenvariable, welche zusätzlich einen Verweis auf die übergeordnete Variable besitzt. Diese Struktur ist notwendig, damit bei der Ausgabe der Ergebnisse die Variable und somit deren Eckdaten gefunden und dem Benutzer präsentiert werden können.

– *PatientDataSetBean*

Die *PatientDataSetBean* bietet alle Funktionen, welche im Zusammenhang mit den Patientendaten notwendig sind. Diese Klasse hat Zugriff auf die Datenbank und bietet deshalb Methoden, um die Werte für einen Patienten-Datensatz für die Eingangsvariablen aus der Datenbank zu laden und als *Map* mit Elementen vom Typ *PatientVariable* zur Verfügung zu stellen. Die Schlüssel für die einzelnen Elemente sind die ID-Nummern der jeweiligen Variablen. Weiters können die UPDRS-Werte des jeweiligen Patienten-Datensatzes geladen und die daraus berechenbaren gewichteten und ungewichteten Sektionswerte als numerische Werte und als *Map* mit Elementen vom Typ *PatientVariable* zur Verfügung gestellt werden.

### 10.3.3.3 DatabaseCRUDPanel

Das DatabaseCRUDPanel ist ein zentrales Element in FuzzyDBSexpert (siehe Kapitel 10.2.4.3), da zum Eingeben des generischen und fallspezifischen Wissens sehr viele Datenbankoperationen

notwendig sind. Aus diesem Grund wurde dieses zentrale Element geschaffen, um einerseits Redundanzen im Code zu vermeiden und viele Code-Zeilen zu sparen und andererseits dieses Element sehr aufwändig gestalten zu können, da dies nur einmal für alle Seiten mit diesem Element getan werden muss. Die grundlegenden Funktionen des DatabaseCRUDPanels wurden schon in Kapitel 10.2.4.3 genannt. Mit Hilfe dieses Widgets kann man Einträge – wenn möglich multiple, das heißt mehrere auf einmal – einer Tabelle hinzufügen und löschen und durchgeführte Änderungen speichern oder verwerfen. Das DatabaseCRUDPanel besteht aus mehreren Komponenten (siehe Abbildung 10.54), welche die Seite und die Business-Logik realisieren. Der sichtbare Teil ist ein JSPF (JavaServer Page Fragment), also ein Fragment einer Seite, welches nicht die volle Funktionalität einer JSP besitzt, vor allem, was den Lifecycle<sup>1</sup> betrifft. Daher besteht das DatabaseCRUDPanel nicht nur aus einer dem JSPF zugehörigen Page-Bean (*DatabaseCRUDPanel*-Klasse), sondern auch aus einer eng verbundenen *SessionBean*<sup>2</sup>, namens *DatabaseCRUDPanelBean*, welche hauptsächlich die Business-Logik implementiert.

---

<sup>1</sup> Der Lifecycle beschreibt den Lebenszyklus einer Seite, welcher verschiedene Stufen von der Initialisierung bis hin zur Zerstörung durchläuft und muss vom Programmierer unbedingt beherrscht werden.

<sup>2</sup> Eine Session-Bean ist eine Klasse, welche spezielle Anforderungen hinsichtlich Namensgebung, Aufbau und Verhalten erfüllt und eine Lebensdauer eines Client-Server-Zyklus hat, also so lange ein Benutzer in FuzzyDBSexpert eingeloggt ist, der Browser nicht geschlossen wird und kein Session-Timeout auftritt.

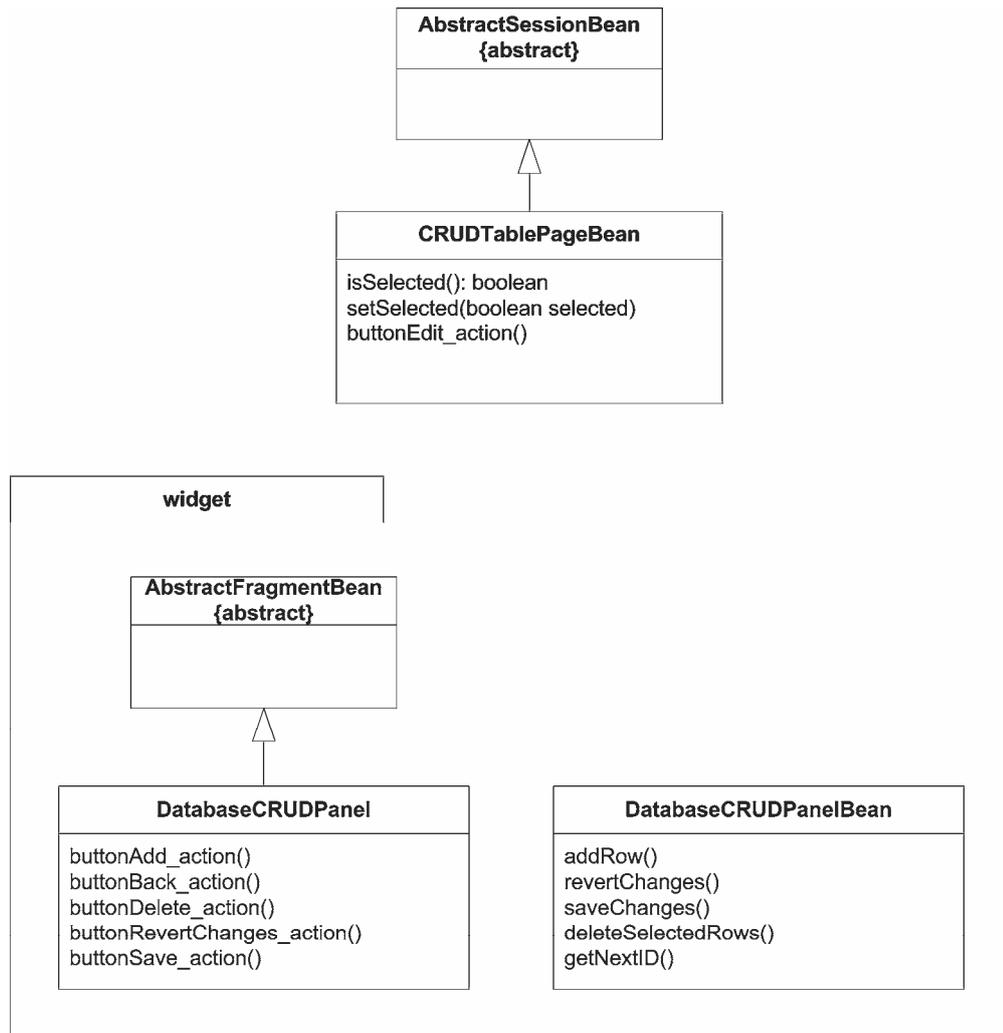


Abbildung 10.54: DatabaseCRUDPanel-Klassendiagramm

Weiters sind für alle Seiten, auf welchen mehrere Zeilen selektierbar und/oder Untereinträge editierbar sein sollen, bestimmte Methoden notwendig, wie in der Beispiel-Klasse *CRUDTablePageBean* angegeben. Die Funktionalitäten dieser Klassen sollen hier kurz beschrieben werden.

### 10.3.3.3.1 Grundlegendes

Damit das *DatabaseCRUDPanel* verwendet werden kann, muss einerseits das JSPF auf der jeweiligen Seite eingebunden werden und andererseits die *DatabaseCRUDPanelBean* in der

jeweiligen *init()*-Methode<sup>1</sup> der Seite konstruiert und in den Session-Scope<sup>2</sup> gebracht werden. Das ist alles, was notwendig ist, um die Seite mit den grundlegenden Datenbankfunktionen auszustatten. Sollen mehrere Zeilen einer Datenbanktabelle selektierbar oder Untereinträge zu bearbeiten sein, müssen noch zusätzliche Methoden auf der jeweiligen Seite implementiert werden (siehe Abbildung 10.54: DatabaseCRUDPanel-Klassendiagramm).

### 10.3.3.3.2 DatabaseCRUDPanel Klasse

Diese Klasse, welche im Zuge der Implementierung der JavaServer Faces die zugehörige Page-Bean zu dem JSPF darstellt, liefert gemäß dem MVC-Entwurfsmuster nur wenig Beitrag zur Business-Logik. Sie ist hauptsächlich dazu da, um Methoden in der *DatabaseCRUDPanelBean* aufzurufen, Meldungen über den Erfolg oder Misserfolg einer Operation zu empfangen und entsprechende lokalisierte Darstellungen dieser zu präsentieren (siehe Abbildung 10.55).

---

<sup>1</sup> Die *init()*-Methode ist eine der Methoden, welche im Zuge des LifeCycles einer Page-Bean automatisch aufgerufen werden.

<sup>2</sup> Der Session-Scope ist der Bereich, auf den man innerhalb einer Client-Server-Session Zugriff hat.

**Patienten-Manager**

Bitte benutzen Sie folgende Schaltflächen, um die Tabelle Patientendaten zu bearbeiten.

**⚠ Änderungen nicht gespeichert.**  
Bitte speichern oder verwerfen Sie die Änderungen, bevor Sie fortfahren.

Patientendaten					
Ausw.	Patienten-Code	Geburtsdatum	Geschlecht	Öffentlich	
<input type="checkbox"/>	12345	25.09.1975	männlich ▾	<input checked="" type="checkbox"/>	Datensätze bearbeiten...
<input type="checkbox"/>		12.01.2007	männlich ▾	<input type="checkbox"/>	Datensätze bearbeiten...

**✔ Änderungen erfolgreich gespeichert.**

**✔ Änderungen erfolgreich verworfen.**

**✔ Einträge erfolgreich gelöscht.**

**❌ Fehler beim Löschen.**

Abbildung 10.55: Beispiele für Meldungen des DatabaseCRUDPanels

### 10.3.3.3 DatabaseCRUDPanelBean

Die *DatabaseCRUDPanelBean* ist eine Java-Bean im Session-Scope, welche die gesamte Business-Logik des DatabaseCRUDPanels implementiert. Wie in Kapitel 10.3.2.2 beschrieben, besteht der Schlüssel einer Datenbanktabelle in FuzzyDBSexpert meist aus mehreren Attributen, jene, die fix für die jeweiligen zu bearbeitenden Einträge sind und ein einziges, welches variabel ist. Die Werte der fixen Schlüsselattribute sind in einer *Map* im Session-Scope gespeichert und können bei Bedarf dort entnommen werden. Der inkrementelle Schlüsselwert, also jener, welcher bei Hinzufügen eines Eintrages inkrementiert werden muss, wird gesondert geführt. Wenn man einen Untereintrag einer Tabelle editieren möchte, so wird das jeweilige inkrementelle Schlüsselattribut den fixen Schlüsseln hinzugefügt und der Attributname gespeichert, damit beim Zurückgehen in der Hierarchie dieser wieder aus dem fixen Schlüssel-Pool entfernt werden kann. An die Stelle des alten variablen Schlüsselattributes tritt das entsprechende neue Schlüsselattribut. Beim Aufsteigen in der Hierarchie tritt an die Stelle des neuen variablen Schlüsselattributes wieder das vorherige variable Schlüsselattribut und wird auch gleichzeitig aus dem Schlüssel-Pool der fixen Schlüssel entfernt. Um diese Funktionalität zur Verfügung stellen zu können, sind der *DatabaseCRUDPanelBean* folgende Dinge – hier nur kurz umschrieben, die Details sind der Dokumentation im Source-Code zu entnehmen – zu übergeben:

- ein Schlüsselwert, um den Namen der zu bearbeitenden Tabelle lokalisiert anzeigen zu können
- den Paketnamen (*admin*, *expert*, *user* oder *null*), um die Check-Boxen ansprechen zu können, welche für die Selektierung von einzelnen Einträgen verwendet werden

- Objekte, die Zugriff auf die Datenbanktabelle ermöglichen (*CachedRowSetDataProvider* und *RowSet*)<sup>1</sup>
- Objekte mit Datenbankzugriff, die den letzten Schlüsselwert des inkrementellen Schlüssels zur Verfügung stellen können (*CachedRowSetDataProvider* und *RowSet*)
- eine Tabelle, welche die Werte der aktuellen fixen Datenbankschlüssel enthält
- ein *String*-Array, welches die für die jeweilige Datenbanktabelle aktuellen Attributnamen für die fixen Datenbankschlüssel enthält
- einen *String* mit dem Namen des Attributes des zu inkrementierenden Datenbankschlüssels
- einen Standard-Wert für den inkrementellen Datenbankschlüssel
- einen Bitschalter, welcher anzeigt, ob die Daten aus der Datenbank aktualisiert werden müssen oder ob diese noch aktuell sind

Mit Hilfe dieser Informationen können folgende Operationen, deren Funktionsweise beschrieben wird, durchgeführt werden:

- Einträge hinzufügen

Die *DatabaseCRUDPanelBean* besitzt eine Methode, um Einträge, auch mehrere auf einmal, hinzufügen zu können. Dazu wird mit Hilfe des *CachedRowSetDataProvider* eine Zeile hinzugefügt und alle fixen Schlüsselattribute mit den Werten aus der *Map* besetzt. Das variable Attribut bekommt den vorher festgelegten Standard-Wert. Die korrekten Werte für das inkrementelle Attribut werden erst beim Speichern gesetzt.

- Einträge löschen

Mit Hilfe des *DatabaseCRUDPanels* kann man mehrere Einträge auf einmal löschen. Dabei können verschiedene Fälle auftreten:

- das Löschen ist für diese Tabelle nicht erlaubt

---

<sup>1</sup> <http://developers.sun.com/prodtech/javatools/jscreator/reference/docs/apis/>

- das Löschen schlägt fehl (zum Beispiel, weil noch ein Fremdschlüssel auf ein Attribut dieses Eintrages zeigt und das Löschen restriktiert wird), dann können folgende daraus resultierende Fälle auftreten:
  - die Transaktion<sup>1</sup> kann nicht zurückgenommen werden, da die darunter liegende Struktur kein Rollback<sup>2</sup> unterstützt
  - die Transaktion kann nach einem fehlgeschlagenen Löschen aus anderen Gründen nicht zurückgenommen werden
  - die Transaktion kann nach einem fehlgeschlagenen Löschen erfolgreich zurückgenommen werden
- das Löschen ist erfolgreich

Je nach Fall wird dem DatabaseCRUDPanel ein entsprechender Fehlercode zurückgeliefert, der es diesem ermöglicht, eine genaue, lokalisierte Meldung anzuzeigen. Das Löschen von Einträgen kann nicht rückgängig gemacht werden, da die Änderungen sofort in die Datenbank geschrieben werden.

#### – Änderungen speichern

Für jeden Eintrag wird zuerst überprüft, ob das inkrementelle Schlüsselattribut *null* ist. Wenn dies der Fall ist, handelt es sich um einen neuen Eintrag und für diesen wird das inkrementelle Schlüsselattribut auf den nächsthöheren, freien, ganzzahligen Wert gesetzt. Zum Schluss werden die Änderungen mit der Methode *commit()* bestätigt.

#### – Änderungen verwerfen

Durch diese Funktion wird die Methode *revertChanges()* aufgerufen, welche alle hinzugefügten Einträge und Änderungen an bestehenden Einträgen verwirft.

Das DatabaseCRUDPanel besitzt drei verschiedene Konstruktoren, einen für die komplette Ausstattung an Funktionalität und zwei, wo die Funktionen für das Hinzufügen beziehungsweise das Hinzufügen und Editieren von Untereinträgen nicht unterstützt werden, die Angabe einiger sonst für diese Funktionen notwendiger Objekte aber nicht erforderlich ist.

---

<sup>1</sup> Eine Transaktion ist eine feste Folge von Operationen, welche als Einheit betrachtet werden.

<sup>2</sup> Als Rollback bezeichnet man das Zurücknehmen der einzelnen Operationen einer Transaktion.

#### 10.3.3.3.4 CRUDTablePageBean

Wenn für eine Seite mit DatabaseCRUDPanel das Markieren von Einträgen beziehungsweise das Bearbeiten von Untereinträgen möglich sein soll oder die „Zurück“-Schaltfläche betätigt wird, so wird die *CRUDTablePageBean* benötigt. Im Normalfall sollte gemeinsame Funktionalität einfach vererbt werden, damit sie den spezielleren Klassen zur Verfügung steht. In diesem Fall sollte also jede Seite, welche diese Funktionalität benötigt, von einer Seite erben, welche diese implementiert hat. Leider ist es in diesem Fall so, dass die Page-Beans direkt von der *AbstractPageBean* ableiten müssen. Nachdem es in Java auch keine multiple Vererbung gibt, musste eine Bean geschaffen werden, die diese Funktionen zur Verfügung stellt. Diese ist eine *SessionBean* und steht daher automatisch zur Verfügung. Folgende Funktionalität wird von der *CRUDTablePageBean* zur Verfügung gestellt:

- Zeilen selektieren
- Zeilen auf Selektierung überprüfen
- Verarbeitung nach Betätigen der Schaltfläche zum Editieren von Untereinträgen

Falls Zeilen hinzugefügt wurden, wird diese Aktion abgebrochen und eine entsprechende lokalisierte Warnmeldung angezeigt. Andernfalls wird das aktuelle variable Schlüsselattribut zur *Map* mit den fixen Schlüsselattributen hinzugefügt.

- Verarbeitung nach Betätigen der „Zurück“-Schaltfläche

Nach dem Betätigen dieser Schaltfläche gelangt man eine Ebene in der Hierarchie hinauf. Wenn Änderungen noch nicht in die Datenbank geschrieben wurden, wird diese Aktion abgebrochen und eine entsprechende lokalisierte Warnmeldung angezeigt. Andernfalls wird das vorhergehende variable Schlüsselattribut, welches zur *Map* der fixen Schlüsselattribute hinzugefügt wurde, wieder entfernt und das alte variable Schlüsselattribut reaktiviert.

#### 10.3.3.4 LinguisticExpressionConstructor

Der *LinguisticExpressionConstructor* ist ein weiteres wichtiges Element in *FuzzyDBSexpert*. Er soll es dem Benutzer ermöglichen, ohne die Fuzzy-Sets von bestimmten Variablen, Modifikatoren und Operatoren auswendig wissen zu müssen, linguistische Ausdrücke für jeweils eine Variable konstruieren zu können (siehe Kapitel 10.2.4.5).

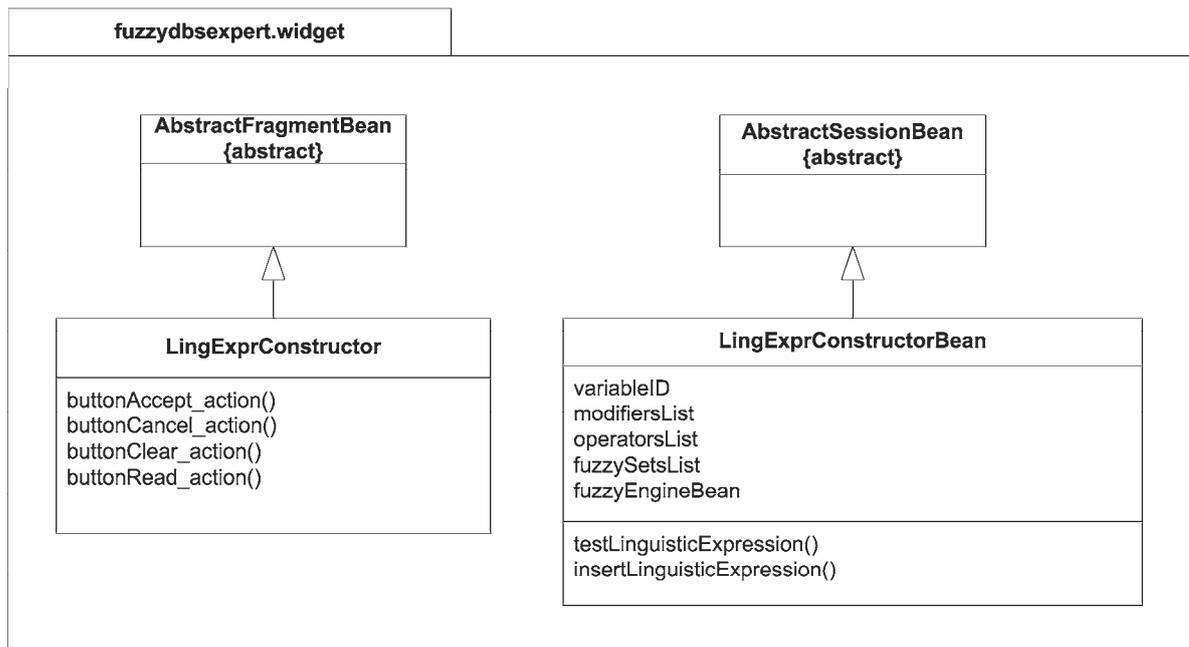


Abbildung 10.56: Klassendiagramm des LinguisticExpressionConstructors

Der LinguisticExpressionConstructor ist, wie das vorhin beschriebene DatabaseCRUDPanel, auch ein JSP-Fragment mit einer Bean (*LingExprConstructorBean*), welche den Großteil der Business-Logik implementiert. Die *LingExprConstructorBean* hat, wie dem Klassendiagramm (siehe Abbildung 10.56) zu entnehmen ist, folgende Hauptaufgaben:

- Entgegennehmen des Schlüssels der betreffenden Variable und entsprechende Anpassung der Datenbankabfragen
- Zurverfügungstellen der Inhalte der Drop-down-Komponenten für
  - Fuzzy-Sets einer bestimmten Variable mit einem lokalisierten Standardeintrag
  - Modifikatoren mit einem lokalisierten Standardeintrag
  - Operatoren mit einem lokalisierten Standardeintrag
- Einfügen des linguistischen Ausdrucks an die richtige Stelle in der Datenbank
- Testen des linguistischen Ausdrucks

Wie schon in Kapitel 10.3.3.2.1 beschrieben, besitzt die FuzzyEngineBean eine Methode, linguistische Ausdrücke für Variablen auf ihre Gültigkeit hin zu testen. Dazu sind folgende Schritte notwendig:

- Laden der Daten der Fuzzy-Variablen aus der Datenbank

- Erstellen der konkreten Fuzzy-Variable
  - Laden der Daten der Fuzzy-Sets aus dem entsprechenden Regel-Set für diese Variable aus der Datenbank
  - Erstellung der konkreten Fuzzy-Sets und Zuweisung der Fuzzy-Variablen
  - Setzen des linguistischen Ausdrucks als Wert für diese Variable zur Überprüfung, ob dieser für diese Variable gültig ist oder nicht

Das Einfügen je eines Standardeintrages für die Drop-down-Komponenten (siehe Abbildung 10.44) hat einerseits den Zweck, den Benutzer bei der Bedienung zu unterstützen, indem angezeigt wird, mit welcher Komponente welche Dinge ausgewählt werden können. Andererseits ist es für die Funktion wichtig, dass die Ereignisse (Events) der Komponenten korrekt verarbeitet werden können. Es ist hier zweckmäßig, das *onChange*-Event als Trigger zu benutzen. Falls aber schon der erste Eintrag ein auszuwählender Eintrag ist und der Benutzer genau diesen auswählen möchte, so wird das *onChange*-Event nicht gefeuert, da sich bei der Auswahl nichts ändert. Das *onClick*-Event hingegen feuert leider schon, wenn die Drop-down-Komponenten-Liste geöffnet wird, also zu früh. Deswegen muss bei solchen Komponenten meistens zuerst ein Standardeintrag angefügt werden, damit eine Auswahl des Benutzers korrekt verarbeitet werden kann.

In der dem *LinguisticExpressionConstructor* zugehörigen Page-Bean, also der Klasse *LingExprConstructorBean*, werden hauptsächlich die verschiedenen Aktionen der Drop-down-Komponenten und der Schaltflächen verarbeitet und entsprechende Methoden in der *LingExprConstructorBean* aufgerufen. Sie dient auch dazu, dem Benutzer, abhängig vom Ausgang der Gültigkeitsprüfung oder von anderen Fehlern, entsprechende lokalisierte Warnhinweise oder Fehlermeldungen anzuzeigen.

### 10.3.3.5 Handhabung des Session-Timeouts

Ein Problem bei Web-Anwendungen stellen Session-Timeouts dar, also das Ablaufende der maximalen Inaktivitätszeit innerhalb einer Session. Nach einer gewissen inaktiven Zeit, wo keine Anfrage in der konkreten Client-Server-Session an den Server geschickt, also etwa eine Schaltfläche geklickt wird, wird diese Session geschlossen und der Benutzer muss sich nochmals im System anmelden. Alle Eintragungen, welche etwa in der Zwischenzeit in einem Formular durchgeführt wurden, gehen dadurch verloren. Der Wert für die Zeitspanne von der letzten Anfrage an den Server bis zum Auftreten des Session-Timeouts kann im Deployment-Descriptor (*web.xml*) eingestellt werden:

```
<session-config>
  <session-timeout>30<!-- minutes --></session-timeout>
</session-config>
```

Nun soll der Benutzer im Falle eines Session-Timeouts automatisch auf die Login-Seite weitergeleitet werden und ein Hinweis über den Ablauf der Session angezeigt bekommen. Für die Weiterleitung auf die Login-Seite ist der *SecurityCheckFilter* im Paket *fuzzydbsexpert.tools* zuständig. Dieser überprüft, falls die *SessionBean* noch vorhanden ist, ob der Benutzer existiert. Falls der Benutzer nicht existiert, wird der Benutzer sofort auf die Login-Seite weitergeleitet. Der Filter und seine Zuordnung müssen im Deployment-Descriptor eingetragen werden:

```
<filter>
  <filter-name>Security Page Filter</filter-name>
  <filter-class>
    fuzzydbsexpert.tools.SecurityCheckFilter
  </filter-class>
</filter>

<filter-mapping>
  <filter-name>Security Page Filter</filter-name>
  <url-pattern>/faces/*</url-pattern>
</filter-mapping>
```

Leider hilft der Filter nicht, den Hinweis für den Benutzer über das erfolgte Session-Timeout anzuzeigen, da vom Filter aus nicht auf die *ApplicationBean*<sup>1</sup> zugegriffen werden kann. Es muss in diesem Fall die *ApplicationBean* als Mittel der Verständigung verwendet werden, da ja die *SessionBean* wegen des Session-Timeouts schon zerstört sein kann. Daher muss noch ein *HttpSessionListener* verwendet werden – der *TimeoutSessionListener* – welcher, sobald die Session zerstört wird, in der *ApplicationBean* einen Bitschalter setzt, damit der Benutzer auf dem Login-Bildschirm eine entsprechende Fehlermeldung angezeigt bekommt. Auch dieser *HttpSessionListener* muss im Deployment-Descriptor eingetragen werden:

```
<listener>
  <listener-class>
    fuzzydbsexpert.tools.TimeoutSessionListener
  </listener-class>
</listener>
```

### 10.3.3.6 Verwendung von Reflection

Ein wahrscheinlich nicht so bekannter Aspekt von Java ist, dass man von einem beliebigen Objekt eine Methode aufrufen kann, auch wenn weder die zugehörige Klasse, noch die aufzurufende Methode zum Zeitpunkt der Entwicklung der Anwendung bekannt ist. Diese Funktionalität wird

---

<sup>1</sup> Die *ApplicationBean* ist eine Java-Bean, welche sich im Application-Scope befindet, also so lange existiert, wie die gesamte Applikation läuft.

durch das Java Reflection API<sup>1</sup> ermöglicht. Reflection wird an drei Stellen in FuzzyDBSexpert, alle befinden sich in der FuzzyEngine, eingesetzt:

- Erstellung von Fuzzy-Sets

Sowohl die Anzahl der Parameter, deren Werte, als auch die Form wird jeweils dynamisch aus der Datenbank geladen, um mit diesen Daten die Fuzzy-Sets für die Fuzzy-Variablen zu erstellen, indem ein jeweils passender Konstruktor und damit die Klasse erstellt wird.

- Erstellung von Regel-Exekutoren

Der Regel-Exekutor für jede Regel wird anhand des Klassennamens, welcher aus der Datenbank geladen wird, dynamisch erstellt und der Regel zugewiesen.

- Aufrufen der Defuzzifizierungsmethode

Die Defuzzifizierungsmethode wird anhand des Methodennamens dynamisch aus der Datenbank erstellt und an den Fuzzy-Ausgangswerten aufgerufen.

Im folgenden Abschnitt werden zwei Beispiele für den Einsatz von Reflection im Zusammenhang mit FuzzyDBSexpert näher erläutert.

```
// add sets to the fuzzyVariables
for (int j = 0; j < fuzzyVariableData.getFuzzySets().size(); j++) {
    FuzzySetData fuzzySetData =
        (FuzzySetData) fuzzyVariableData.getFuzzySets().get(j);

    // create fuzzy set with reflection because it is known only from the
    // database which class and constructor parameter count is needed for
    // this particular fuzzy variable
    Class clazz = Class.forName(fuzzySetData.getShapeClass()); // FuzzySet's
                                                                // class

    // create constructor (types) array
    Class consTypes[] = new Class[fuzzySetData.getParamCount()];
    for (int foo = 0; foo < fuzzySetData.getParamCount(); foo++) {
        consTypes[foo] = Double.TYPE; // class for primitive type double
    }

    // create constructor data array
    Object[] consData = new Object[fuzzySetData.getParamCount()];
    for (int foo = 0; foo < fuzzySetData.getParamCount(); foo++) {
        consData[foo] = new Double(fuzzySetData.getParams()[foo]);
        // Double is automatically converted to double
    }

    Constructor constructor = clazz.getConstructor(consTypes);
```

---

<sup>1</sup> Application Programming Interface: beschreibt Klassen und Methoden, die zur Programmierung notwendig sind

```

        Object fuzzySet = constructor.newInstance(consData);
        fuzzyVariable.addTerm(fuzzySetData.getName(), (FuzzySet) fuzzySet);
    }
    fuzzyVariables.put(variableID, fuzzyVariable);

```

In obigem Beispiel werden für eine Fuzzy-Variable die zugehörigen Sets erstellt und der Variable zugewiesen. Zuerst wird für jedes Fuzzy-Set der Variablen die zugehörige Klasse *FuzzySetData* aus einer *Map* geholt, welches als Template für das konkrete Fuzzy-Set dient. Danach wird eine Klasse des Typs der Klasse der Fuzzy-Form durch Reflection erzeugt. Die Klassennamen für die Fuzzy-Set-Formen sind in der Datenbank verzeichnet. Die zugehörige Tabelle ähnelt jener der Fuzzy-Set-Formen (siehe Tabelle 10-5), abgesehen davon, dass zusätzlich der Klassenname für die Form angegeben ist, beispielsweise *nrc.fuzzy.TriangleFuzzySet*. Danach wird, da für den Konstruktor eine unbestimmte Anzahl an Parametern erforderlich ist, jeweils ein *Array* für die Parameter-Typen und eines für die Parameter-Werte erstellt. Nachdem die Parameter vom Typ *double* sind und dies aber keine Klasse darstellt, sondern nur einen primitiven Datentypen, aber Klassen für die Konstruktor-Typen benötigt werden, kann hier nur *Double.TYPE* verwendet werden, welche eine Klasse für den primitiven Typ *double* darstellt. Die Werte für die entsprechenden Parameter werden anhand der Daten aus dem *FuzzySetData*-Objekt in der richtigen Reihenfolge in das Werte-*Array* übertragen. Danach wird ein *Constructor*-Objekt mit Hilfe der Parameter-Typen erstellt und schließlich ein *FuzzySet*-Objekt durch Instanzierung mit Hilfe des vorhin erzeugten Konstruktors und den Werten für die Parameter erzeugt, welches zum Schluss noch gecastet<sup>1</sup> werden muss, damit es der Fuzzy-Variable zugewiesen werden kann.

Als zweites Beispiel soll der Aufruf einer unbekanntnen Methode einer bekannten Klasse gezeigt werden, nämlich der Defuzzifizierungsmethode für Fuzzy-Variablen.

```

// create method with reflection
Method method = globalFuzzyOutput[0].getClass().getMethod(
    ((DefuzzificationMethodData) defuzzificationMethods.get(
        defuzzificationMethodID)).getClassName(), null);
// defuzzify all results
for (int i = 0; i < patientOutputVariables.size(); i++) {
    double result = ((Double)method.invoke(globalFuzzyOutput[i],
        null)).doubleValue();
    ((PatientOutputVariable)patientOutputVariables.get(i)).setValue(
        result);
}

```

---

<sup>1</sup> Beim Casten handelt es sich um eine Typenumwandlung von einem Datentypen in einen anderen. In diesem Fall wird ein Objekt vom Typ *Objekt* (also dem allgemeinsten Typen, den es für ein Objekt geben kann) explizit in ein Objekt vom Typ *FuzzySet* (genauer *nrc.fuzzy.FuzzySet*) umgewandelt. Dies ist notwendig, da die Methode *addTerm* nur Objekte vom Typ *FuzzySet* aber – verständlicherweise – nicht vom Typ *Objekt* als zweiten Parameter akzeptiert.

Zuerst wird der Name der Methode mit Hilfe des Objektes *DefuzzificationMethodData*, welches die Methodennamen für Defuzzifizierungsmethoden bereithält, ermittelt und damit ein *Method*-Objekt erstellt. Die Methodennamen werden aus der Datenbank geladen, wo sie sich in einer Tabelle befinden, welche folgendermaßen aufgebaut ist:

Tabelle 10-8: Defuzzifizierungsmethoden-Methodennamen in FuzzyDBSexpert

Defuzzifizierungsmethode	Methodenname
Center of Area	<i>centerOfAreaDefuzzify</i>
Mean of Maxima	<i>maximumDefuzzify</i>
Moment	<i>momentDefuzzify</i>
Weighted Average	<i>weightedAverageDefuzzify</i>

Danach wird diese Methode für jede der Fuzzy-Ausgangsvariablen aufgerufen und das Ergebnis der entsprechenden Patienten-Ausgangsvariable zugewiesen.

### 10.3.3.7 Ablauf der Inferenz

In diesem Kapitel soll der Ablauf der Inferenz und alles, was dazu notwendig ist, sequentiell beschrieben werden, damit es nicht nur möglich ist, diese zu verstehen und Anregungen für eigene Implementierungen zu gewinnen, sondern auch einen leichteren Einstieg in das Verständnis des betreffenden Programmcodes zu finden. Sobald der Benutzer auf die Schaltfläche zum Berechnen der Ausgangsvariablen am Schirm für die Patienten-Datensatz-Evaluierung (siehe Kapitel 10.2.3.2.1) klickt, laufen folgende Vorgänge ab:

- Überprüfung, ob der Patient, ein Datensatz und das für die Berechnung zugrunde liegende Regel-Set ordnungsgemäß ausgewählt wurden
- Laden des Patienten-Datensatzes aus der Datenbank
  - Laden der Variablen-Werte aus der Datenbank
    - Überprüfung, ob ein linguistischer Ausdruck oder ein Zahlenwert für die jeweilige Variable angegeben wurde
    - Erstellung einer Patienten-Variablen mit dem jeweiligen Wert beziehungsweise ignorieren dieser Variable, falls weder ein linguistischer Ausdruck, noch ein Zahlenwert für diese angegeben wurde (weil dieser Wert zum Beispiel unbekannt oder nicht ermittelbar ist)
    - Hinzufügen aller Patientenvariablen einer *Map*
  - Hinzufügen der gewichteten UPDRS-Sektionswerte zu den Variablen-Werten

- Laden der UPDRS-Werte für jede UPDRS-Sektion aus der Datenbank
  - Berechnung des gewichteten Summenwertes für jede Sektion
  - Erstellung von Patienten-Variablen mit diesen Summenwerten und Hinzufügen zur *Map* mit den Patienten-Variablen
  - Hinzufügen des UPDRS-Gesamtsummenwertes zur *Map* mit den Patienten-Variablen
- Erstellung der *FuzzyEngineBean*
- Laden globaler Parameter für das Regel-Set aus der Datenbank (Standard-Regel-Exekutor, Defuzzifizierungsmethode)
  - Erstellung der *FuzzyEngine* mit Hilfe obiger und folgender Parameter
    - Variablendaten aus der Datenbank
    - *Map* der Regel-Exekutoren (Methodennamen für die *Reflection*) aus der Datenbank
    - *Map* der Defuzzifizierungsmethoden (Klassennamen für die *Reflection*) aus der Datenbank
- Vorbereiten der *FuzzyEngine* über die *FuzzyEngineBean* auf die Inferenz
- Übergeben des Patienten-Datensatzes an die *FuzzyEngine*
  - Zuweisung einer erstellten Liste (*List*) von Fuzzy-Regeln mit entsprechenden *FuzzyRuleData*-Objekten (eine Liste deswegen, da die Regeln sequentiell abgearbeitet werden müssen und eine Liste eher dieses Verhalten unterstützt als eine *Map* (obwohl es auch bei einer *Map* möglich ist, die Elemente sequentiell durchzugehen)
    - Erstellung je eines *FuzzyRuleData*-Objektes pro Regel
      - Laden des Regel-Exekutors für diese Regel, falls dieser angegeben wurde
      - Erstellung der Listen für die Bedingungs- und Schlussfolgerungsteile der Regel mit *RulePartData*-Objekten
        - Laden der Liste der Ausdrücke der Regeln aus der Datenbank (Tabelle *RulePartAtoms* in der Datenbank)

- Zuweisung der Fuzzy-Sets zu den Variablen
  - Zuweisung aller Sets zu den Patienten-Variablen (falls keine Sets zu einer Variablen existieren – zum Beispiel wenn zwar im Patienten-Datensatz ein Wert zu einer Variable vorhanden ist, aber diese Variable in dem bestimmten Regel-Set nicht verwendet wird – wird diese Variable ausgelassen)
    - Laden der allgemeinen Daten jedes Sets, wie Name, Form und Klassenname der Form für die *Reflection*
    - Erstellen einer Liste (*List*) mit den Parameter-Werten für die Fuzzy-Form aus der Datenbank (eine Liste deswegen, weil die Parameter unbedingt in der richtigen Reihenfolge stehen müssen, sonst wäre die Form der Fuzzy-Sets falsch)
    - Erstellen eines *FuzzyVariableData*-Objektes (jetzt inklusive der Sets) für jede Fuzzy-Variable
- Überprüfung, ob mindestens eine Regel und eine Patienten-Variable vorhanden sind; falls nicht, erfolgt die Anzeige einer entsprechenden Fehlermeldung (z. B. beim Versuch der Verwendung eines Regel-Sets, wo keine Regel definiert ist oder alle Regeln auf inaktiv gestellt wurden)
- Erstellung einer *Map* von konkreten *FuzzyVariable*-Objekten aus den Daten der *Map* von *FuzzyVariableData*-Objekten (Konkretisierung der Fuzzy-Variablen aus den Templates für die Fuzzy-Variablen)
  - Erzeugen der jeweiligen *FuzzyVariable*-Objekte
  - Erzeugung und Hinzufügen der *FuzzySet*-Objekte aus den *FuzzySetData*-Objekten über *Reflection* (siehe Kapitel 10.3.3.6) zu den jeweiligen *FuzzyVariable*-Objekten
- Erstellung einer *Map* von konkreten *FuzzyRule*-Objekten aus den Daten der Liste von *FuzzyRuleData*-Objekten (Konkretisierung der Fuzzy-Regeln aus den Templates für die Fuzzy-Regeln)
  - Falls der Regel-Exekutor für diese Regel angegeben wurde, wird dieser, sonst der Standard-Regel-Exekutor, als konkretes *FuzzyRuleExecutor*-Objekt über *Reflection* erzeugt
  - Erzeugung der konkreten *FuzzyRule*

- Anhängen der Bedingungs- und Schlussfolgerungsteile in Form von konkreten *FuzzyValue*-Objekten mit jeweils verschiedenen Methoden (entweder *addAntecedent()* oder *addConclusion()*)
    - Erzeugung der *FuzzyValue*-Objekte aus den *FuzzyRuleData*-Objekten, aus denen *FuzzyVariable*-Objekte mit zugewiesenen linguistischen Ausdrücken erzeugt werden
  - Sollte sich eine der für diese Regel benötigten Variablen nicht in der Patienten-Variablen-*Map* befinden, also keinen konkreten Wert besitzen, da sie entweder überhaupt nicht erhoben wurde oder unbekannt ist, so wird diese Regel wieder verworfen und künftig somit ignoriert
- Überprüfung, ob mindestens eine Regel zur Ausführung übrig geblieben ist (wenn die Patientendaten und das zu verwendende Regel-Set inkompatibel sind, das heißt, die Menge der Variablen, welche sich im Patienten-Datensatz befinden und jene, welche vom Regel-Set benötigt werden, disjunkt sind oder es keine einzige Regel gibt, wo alle Variablen im Patienten-Datensatz Werte zugewiesen haben, kann es dazu kommen, dass keine einzige Regel zur Ausführung übrig bleibt)
- Ausführung der Regeln und Berechnung der Ausgangswerte
- Erstellung der konkreten Fuzzy-Eingangswerte als *Map* mit Objekten vom Typ *FuzzyVariable*, welche einen Wert zugewiesen bekommen haben
    - Falls der Wert ein linguistischer Ausdruck ist, wird dieser direkt dem *FuzzyVariable*-Objekt zugewiesen, das NRC FuzzyJ-Toolkit kann diesen linguistischen Ausdruck dann für diese Variable interpretieren
    - Falls der Wert ein Zahlenwert ist, wird aus diesem Zahlenwert ein Singleton-Fuzzy-Set erzeugt und dieses dem *FuzzyVariable*-Objekt zugewiesen
  - Ausführen jeder Fuzzy-Regel
    - Zuweisen aller für die Regel jeweils benötigten Fuzzy-Eingangswerte
    - Überprüfen, ob die Regel einen bestimmten Mindestbeitrag zum Gesamtergebnis liefern würde oder nicht, letztenfalls wird die Regel nicht zur Ausführung gebracht und spart damit Rechenzeit
    - Ausführen der jeweiligen Fuzzy-Regel auf Basis des Regel-Exekutors
    - Für jede Ausgangs-Variable wird der Beitrag dieser Regel hinzugefügt

- Falls die Ausgangs-Variable noch keinen Wert besitzt, wird der Ausgangswert der Regel der neue Wert der Ausgangs-Variable
  - Andernfalls wird der Beitrag des Ergebnisses der Regel zur Ausgangs-Variable mit Hilfe der Fuzzy-Union hinzugefügt
- Erstellung der jeweiligen Defuzzifizierungsmethode mittels Reflection (siehe Kapitel 10.3.3.6)
  - Aufrufen der Defuzzifizierungsmethode für jede Ausgangs-Variable und Speicherung des Wertes in das jeweilige *PatientOutputVariable*-Objekt

Diese sequentielle Auflistung mag an dieser Stelle recht kompliziert aussehen, dieser Ablauf ist aber einerseits notwendig, um das Fassade-Muster zu realisieren (siehe Kapitel 10.3.3.1) und dadurch weniger abhängig von der Fuzzy-Java-Bibliothek zu sein und wird andererseits bei gleichzeitiger Durchsicht des Programmcodes leichter verständlich. Jedenfalls sollte dieser Ablauf verstanden werden, wenn das Programm geändert werden oder als Beispiel für eigene Lösungen dienen sollte.

## 10.3.4 Literaturhinweise

In diesem Kapitel werden Literaturhinweise von Quellen angeführt, welche mir bei der Durchführung des Projektes, einerseits zum Einlernen in die Technologien und andererseits auch als Nachschlagewerke, gedient haben. Es geht mir hier nicht darum, eine möglichst lange Liste zu den Themen verfügbarer Literatur aufzuzählen, sondern meiner Meinung nach Hinweise für qualitativ hochwertige und dem Lernenden absolut zu empfehlende Literatur anzubieten. Auch wenn ein Vielfaches an verfügbarer spezieller Literatur durchgesehen wurde, sollen hier nur jene Quellen genannt werden, welche über längere Zeit hinweg als Wissensquelle und nicht nur zum Nachschlagen kleiner Spezialgebiete gedient haben, diese finden sich als Quellenangaben an der jeweils adäquaten Stelle.

Quellen aus dem Web werden hier, soweit vorhanden, bevorzugt genannt, da diese sofort und (fast) immer verfügbar und zudem kostenlos sind. Die technischen Themen haben aufgrund ihres Gebietes natürlich potentiell gute Quellen im Internet.

### 10.3.4.1 Java

Auf der Homepage des Herstellers der Java-Technologie (<http://java.sun.com>) findet man alle Informationen und die API-Dokumentation zu den verschiedensten Java-Versionen, welche wegen des ausgezeichneten Umfangs und Aufbaus eigentlich jedes Nachschlagewerk in Buchform obsolet macht. Um Java zu lernen, kann man einen Blick auf die hervorragenden Java-Tutorials werfen und

diese bei Gefallen durcharbeiten (<http://java.sun.com/docs/books/tutorial/index.html>). Das Java-Tutorial ist auch in Form eines Buches erhältlich (siehe angegebener Link).

Wenn man zu Java gedruckte Literatur bevorzugt, kann man sich, abgesehen von obigem Buch, das Buch „Head First Java“ [HFJava, 2004] genauer ansehen, wenn man die Programmiersprache Java lernen möchte. Die Bücher der Serie „Head First“ aus dem O'REILLY-Verlag versuchen, mit modernen Lernprinzipien, Wissen nachhaltig zu vermitteln. Als gedrucktes Einstiegs- und Nachschlagewerk kann ich „Java in a Nutshell“ [JavaNS, 2005] in der jeweils aktuellen Version aus dem gleichen Verlag empfehlen, wobei aber ein großer Teil des Buches aus der API besteht und dieser Teil aus oben genanntem Grund eigentlich unnötig ist. Viele Codebeispiele zu allen möglichen Gebieten von Java bietet „Das Java Codebook“ (neu: „Das Java 6 Codebook“) aus dem Addison-Wesley-Verlag.

### **10.3.4.2 Servlets & JSP**

Servlets und JSP sind Teile der Java EE Web Application Technologies, welche Teil der Java Platform, Enterprise Edition (Java EE) und deswegen auf den entsprechenden Unterseiten der Sun-Homepage für Java zu finden sind (<http://java.sun.com/javaee/technologies/webapps/>). An dieser Stelle findet sich auch die API-Dokumentation. Es ist auch ein Java EE-Tutorial auf dieser Seite verfügbar, welches dem Einsteiger nützlich sein kann (<http://java.sun.com/javaee/5/docs/tutorial/doc/index.html>).

Um Servlets und JSP zu lernen, kann das Buch „Head First Servlets & JSP“ [HFServlets]JSP, 2005] empfohlen werden. Abgesehen von einigen anderen Büchern, welche manchmal als Nachschlagewerke gedient haben, kann das „Das J2EE Codebook“ (neu: „Das Java EE 5.0 Codebook“) aus dem Addison-Wesley-Verlag empfohlen werden, welches kurz und prägnant zu vielen Themen der J2EE bzw. Java EE 5.0 einen Überblick und für viele Fragen Problemlösungen mit Code-Beispielen bietet.

### **10.3.4.3 JSF**

JSF ist zwar auch Teil der Java EE Web Application Technologies, wird aber wegen der herausragenden Rolle in FuzzyDBSexpert extra besprochen. Da JSF relativ neu ist, gibt es noch nicht so viel Literatur, wie etwa zu Servlets und JSP. Grundsätzlich können die unter Servlets & JSP genannten Online-Quellen verwendet werden, da JSF ein Teil der Java EE Web Application Technologies ist. Weiters ist der „Java Studio Creator Field Guide, Second Edition“ mit JSF-Teilen durchzogen. Wer unbedingt gedruckte Literatur bevorzugt, dem können prinzipiell – nicht nur JSF-bezogen – alle Bücher der „In Action“-Serie aus dem Manning-Verlag und der „Core“-Serie aus dem Prentice Hall International-Verlag empfohlen werden, in beiden Verlagen sind entsprechende Bücher erschienen.

#### 10.3.4.4 Creator

Der Java Studio Creator ist die relativ neue IDE von Sun und deswegen existiert noch sehr wenig Literatur dazu. Der Großteil der Werke kommt direkt von Sun in Form von sehr empfehlenswerten Tutorials (<http://developers.sun.com/prodtech/javatools/jscreator/learning/tutorials/index.jsp>) oder dem, wenn man mit dem Java Studio Creator in einem mäßig fortgeschrittenen Stadium – also bei Projekten ab einigen wenigen Seiten mit der Notwendigkeit nicht-trivialer Business-Logik zu implementieren – arbeiten möchte, absolut essentiellen „Java Studio Creator Field Guide, Second Edition“ (<http://developers.sun.com/prodtech/javatools/jscreator/learning/bookshelf/index.jsp>), welcher kostenlos im PDF-Format heruntergeladen werden kann. Nicht immer hilft dieses Field-Guide weiter, in diesem Fall kann man sich an das sehr aktive Sun Java Studio Creator Community Forum wenden (<http://forum.java.sun.com/forum.jspa?forumID=881>).

#### 10.3.4.5 Datenbank

Nachdem eine MySQL-Datenbank zum Einsatz kommt, welche Open Source ist, ist natürlich auch ein umfangreiches, empfehlenswertes kostenloses Handbuch online verfügbar (<http://dev.mysql.com/doc/refman/5.0/en/index.html>).

Gedruckte Literatur wurde hauptsächlich in allgemeinen SQL-Fragen oder beim Datenbank-Entwurf verwendet, zum Beispiel bei Fragen zu den Normalformen. In diesem Fall können die beiden in meinem Studium verwendeten Bücher „Datenbank-Systeme“ [Kemper, 2006] und „Datenbanken: Konzepte und Sprachen“ [Heuer, 2000] empfohlen werden.

#### 10.3.4.6 Web-Container

Neben dem mit dem Creator mitgelieferten Sun Application Server zum Testen wird der Apache Tomcat als zuverlässiger und ressourcensparender Servlet/JSP-Container zum Betrieb eingesetzt. Die zugehörige Dokumentation ist bei der jeweiligen Version des Servers zu finden (<http://tomcat.apache.org/>). Dieser kann zwar auch als Webserver fungieren, bei größeren Projekten ist es jedoch ratsam, die Servlet/JSP-Engine vom Webserver zu trennen und zum Beispiel einen Apache HTTP-Server als Webserver zu verwenden (<http://httpd.apache.org/>).

# 11 Schlussdiskussion

## 11.1 Zusammenfassung

„Fuzzy inference systems (FIS) have been applied for regulating depth of anaesthesia, and for recommending regimen changes in diabetes. It appears that very little has yet been done in applying FIS in neurological DSS“ [Mobyen, 2006].

Mit FuzzyDBSexpert ist es gelungen, ein webbasiertes, entscheidungsunterstützendes System zu realisieren, in welchem durchgängig Fuzzy-Logik Eingang gefunden hat. Es gibt zwar eine Arbeit ([Eitel, 2003]), welche das gleiche Ziel in Hinsicht einer Entscheidungsunterstützung für eine Tiefe Hirnstimulation verfolgt hat, jedoch wurde dort keine Fuzzy-Logik eingesetzt und die verfügbaren Technologien haben sich in der Zwischenzeit auch weiterentwickelt. Experten können in FuzzyDBSexpert generisches Wissen in Form von Regeln, Fuzzy-Sets zu Fuzzy-Variablen, und nicht zuletzt Gewichten für jede einzelne Frage der UPDRS, in das System einbringen. Patientendaten können entweder als linguistische Ausdrücke oder als numerische Werte angegeben werden, ähnlich wie in CADIAG-IV [Boegl, 2004, S. 25], je nachdem, welche Form sich besser für die Angabe der jeweiligen Merkmalsausprägung eignet. Manche Werte sind gut als numerische Werte anzugeben, wie zum Beispiel die Krankheitsdauer. Andere werden aufgrund von Unwissen, Unsicherheit oder gewollter beziehungsweise ungewollter Unschärfe besser als linguistischer Ausdruck anzugeben sein, wie etwa die Wirkung des L-Dopas. Außerdem lassen sich mit Hilfe der Fuzzy-Logik fließende Übergänge zwischen den einzelnen Fuzzy-Sets, also den Merkmalsausprägungen, realisieren.

Die Verwendung von Fuzzy-Logik stellt einen sehr intuitiven Ansatz für die Benutzer dar, jedoch ist es schwierig, die Ergebnisse im Detail nachzuvollziehen. Dies würde eine prototypische Realisierung in einem Fuzzy-Framework erfordern, wo man zum Beispiel die Eingangsgrößen einfach mit der Maus über den gesamten Wertebereich regeln und die Veränderung in den Ausgangsvariablen nachvollziehen kann.

In FuzzyDBSexpert lässt sich auch widersprüchliches Wissen einbringen, was ein weiterer Vorteil der Fuzzy-Logik ist, ohne die Fuzzy-Inferenz dadurch zu beeinträchtigen. Nicht nur Expertenwissen in sich kann widersprüchlich sein, es gibt natürlicherweise auch verschiedene Meinungen unter den Experten. Dies stellt für FuzzyDBSexpert kein Problem dar, jeder Experte kann seine eigenen Gewichtungen der UPDRS, Fuzzy-Sets und Fuzzy-Regeln in das System einbringen und das in Form eines oder mehrerer Regel-Sets. Für die Berechnung der Ergebnisse können die Benutzer, also zum Beispiel Mediziner, welche keine Experten auf dem Gebiet der Tiefen Hirnstimulation sind, oder Patienten und deren Angehörige, eines dieser der Öffentlichkeit zugänglich gemachten Regel-Sets verwenden, um die Patientendaten in Form der UPDRS und

anderer Variablen auswerten zu lassen. FuzzyDBSexpert kommt auch mit unvollständig vorliegenden Daten zurecht. Wenn FuzzyDBSexpert auf eine Regel stößt, welche unbekannte Daten benötigt, so wird diese einfach ignoriert und das System fährt mit der nächsten Regel fort, bis alle Regeln abgearbeitet sind. So ist es zum Beispiel möglich, Regel-Sets zu erstellen, die nur den dritten Teil der UPDRS (UPDRS-III: Motorische Untersuchung) benötigen, damit nicht so viele Daten eingegeben werden müssen und somit eine verkürzte Eingabe der Patientendaten und damit eine höhere Akzeptanz bei den Benutzern möglich wird. Dazu ist jedoch zu bemerken, dass immer möglichst viele der benötigten Daten angegeben werden sollten, da das Ergebnis umso zuverlässiger wird, je mehr der geforderten Daten eingegeben wurden. Außerdem kann durch die Eingabe der UPDRS der Status des Patienten und bei mehrmaliger Eingabe dessen Verlauf über die Zeit in normierter, vergleichbarer wissenschaftlicher Weise dokumentiert werden. Als Mehrwert werden die Teilsummen und die Gesamtsumme der UPDRS berechnet, dies muss also nicht mehr manuell geschehen, mit der Möglichkeit, sich bei der Berechnung zu irren. Außerdem kann die UDRS in die eigene Dokumentation kopiert werden, was für einige Experten für die eigene Dokumentation als zusätzlicher Nutzen interessant ist.

Wenn die Wissensbasis zufriedenstellend aufgebaut ist – „perfekt“ wird sie wohl nie sein, aber es besteht zumindest die Möglichkeit, sich möglichst nahe am aktuellen Stand von Wissenschaft und Forschung zu bewegen – sind im Grunde relativ wenige Angaben, abgesehen von der UPDRS, zu machen, um zu einem Ergebnis in Form der Eignung zur Tiefen Hirnstimulation oder anderen Ergebnissen – es können beliebig viele weitere Ausgangsvariablen berechnet werden – zu gelangen.

Abgesehen von der UPDRS ist FuzzyDBSexpert so generisch und flexibel gestaltet, dass es für viele andere Problemstellungen als webbasiertes, wissensbasiertes System dienen kann, da das gesamte Wissen über Benutzerschnittstellen ins System eingegeben werden kann. Mit einigen kleinen Umbenennungen kann sogar die UPDRS in irgendeinen anderen Fragebogen mit Punktwertungen, welcher aus verschiedenen Sektionen und Fragen pro Sektion mit verschiedensten Antwortmöglichkeiten für die Fragen besteht, umgestaltet werden und das alles über die bestehende Benutzerschnittstelle. Lediglich einige Bezeichnungen müssten auf generische Namen geändert werden. So betrachtet kann FuzzyDBSexpert als Prototyp für weitere webbasierte, wissensbasierte Systeme dienen, nicht nur für medizinische Anwendungen.

## 11.2 Ausblick

Fuzzy-Inferenzsysteme werden zunehmend erfolgreich in der Medizin eingesetzt. Beispiele hierfür sind CADIAG-II, MedFrame/CADIAG-IV, FuzzyARDS und FuzzyKBWear [vgl. Adlassnig, 2001].

FuzzyDBSexpert ist derzeit noch nicht im Einsatz. Es ist notwendig, eine fundierte Wissensbasis aufgrund statistischer Analysen und Expertenmeinungen aufzubauen, zu testen und iterativ zu

verfeinern, sodass verlässliche und mit den statistischen Analysen zu vereinbarende Ergebnisse auftreten. Das Problem dabei ist, dass spezialisierte Experten oftmals wenig Zeit für diese Analysen und die Erstellung von Regeln anhand von diesen Daten und ihrem Expertenwissen haben und deswegen meist die Unterstützung durch einen Wissensingenieur benötigen. Andererseits wird ohne ein wenig Engagement wenig Fortschritt erzielbar sein. Die notwendigen Analysen und iterativen Anpassungen der Wissensbasis konnten aber im Zuge dieser Arbeit nicht durchgeführt werden, da ein Großteil der Zeit für den Entwurf und die Realisierung des Systems verwendet wurde und dies sonst den Rahmen dieser Arbeit gesprengt hätte. Es ist vorstellbar, im Rahmen weiterer Masterarbeiten oder anderer wissenschaftlicher Arbeiten den Datenbestand zu analysieren, zusammen mit dem Expertenwissen eine Wissensbasis aufzubauen, zu evaluieren und zu verfeinern.

Aufgrund des im vorigen Kapitel genannten generischen und flexiblen Aufbaus von FuzzyDBSexpert kann dieses System jedenfalls als ein Beispiel für weitere Entwicklungen von webbasierten, wissensbasierten Systemen analysiert und in Betracht gezogen werden, nicht nur auf dem medizinischen Sektor.

## Literaturverzeichnis

- [Adlassnig] Adlassnig, Klaus-Peter: The Section on Medical Expert and Knowledge-Based Systems at the Department of Medical Computer Sciences of the University of Vienna Medical School. In: *Artificial Intelligence in Medicine*, Vol. 21, 2001, S. 139–146.
- [ÄrzteWoche] Ärzte Woche: Parkinson: Neue Behandlungsoption (20. Jahrgang, Nr. 1–2, 2006) <http://www.aerztewoche.at/viewArticleDetails.do?articleId=3989> (21.02.2007)
- [Beierle] Beierle, Christoph; Kern-Isberner Gabriele: *Methoden wissensbasierter Systeme. Grundlagen, Algorithmen, Anwendungen*, 2., überarbeitete und erweiterte Auflage. – Wiesbaden: Vieweg Verlag, 2003.
- [Benabid1] Benabid, Alim-Louis; Pollak, Pierre; Hoffmann, Dominique; LeBas, Jean-Francois: Chronic High-Frequency Thalamic Stimulation in Parkinson's Disease. In: Koller, William C., Paulson, George (Hrsg.): *Therapy of Parkinson's Disease*, Second Edition, Revised and Expanded, S. 381–401. - New York/Basel/Hong Kong: Marcel Dekker, Inc., 1995.
- [Benabid2] Benabid, Alim-Louis; Deuschl, Günther; Lang, Anthony E.; Lyons, Kelly E.; Rezai, Ali R.: Deep Brain Stimulation for Parkinson's Disease. In: *Movement Disorders*, Vol. 21 Suppl. 14, 2006, S. 168–170.
- [Benazzouz] Benazzouz, Abdelhamid; Gao, Dongming M.; Ni, Zhongge G.; Piallat, Brigitte; Bouali-Benazzouz, Rabia; Benabid, Alim-Louis: Effect of High-Frequency Stimulation of the Subthalamic Nucleus on the Neuronal Activities of the Substantia Nigra Pars Reticulata and Ventrolateral Nucleus of the Thalamus in the Rat. In: *Neuroscience*, Vol. 99 No. 2, 2000, S. 289–295.
- [Bothe] Bothe, Hans-Heinrich: *Fuzzy Logic. Einführung in die Theorie und Anwendungen*, 2. Auflage. – Heidelberg: Springer-Verlag, 1995.
- [Boegl] Boegl, Karl; Adlassnig, Klaus-Peter; Hayashi, Yoichi; Rothenfluh, Thomas E., Leitich, Harald: Knowledge acquisition in the fuzzy knowledge representation framework of a medical consultation system. In: *Artificial Intelligence in Medicine*, Vol. 30, 2004, S. 1–26
- [Dostrovsky] Dostrovsky, J. O.; Levy, R.; Wu, J. P.; Hutchison, W. D.; Tasker, R. R.; Lozano, A.M.: Microstimulation-Induced Inhibition of Neuronal Firing in Human Globus Pallidus. In: *Journal of Neurophysiology*, Vol. 84 No. 1, 2000, S. 570–574.
- [Egly] Egly, Uwe; Eiter Thomas; Tompits, Hans: *Skriptum zur Lehrveranstaltung Einführung in Wissensbasierte Systeme (VU 2.0)* – Wien: Institut für Informationssysteme, Abt. Wissensbasierte Systeme 184/3, Technische Universität Wien, 2002.

- [Eitel] Eitel, Roland: POTexpert – Ein Konzept für computergestützte Entscheidungshilfe zur operativen Behandlung von Parkinson-Patienten, Diplomarbeit. – Wien: Technische Universität Graz, Universität Wien, 2003.
- [Fazzini] Fazzini, Enrico; Dogali, Michael; Beric, Alexander; Eidelberg, David; Sterio, Djorfje; Gianutsos, John; Newman, Bonnie; Kluger, Alan; Perrine, Kenneth; Loftus, Sheree; Chin, Linda; Samelson, Dale; Kolodny, Edwin: The Effects of Unilateral Ventral Posterior Medial Paddidotomy in Patients with Parkinson's Disease and Parkinson's Plus Syndromes. In: Koller, William C., Paulson, George (Hrsg.): Therapy of Parkinson's Disease, Second Edition, Revised and Expanded, S. 353–379. - New York/Basel/Hong Kong: Marcel Dekker, Inc., 1995.
- [Fahn] Fahn Stanley, Elton Richard. L.: Unified Parkinson's Disease Rating Scale. In: Fahn, S.; Marsden, C. D.; Calne, D.B.; Goldstein, M. (Hrsg.): Recent Developments in Parkinson's Disease, Vol. 2., S. 153–163. – Florham Park NJ: Macmillan Healthcare Information, 1987.
- [Fowler] Fowler, Martin: Patterns of Enterprise Application Architecture. – Amsterdam: Addison-Wesley, 2002.
- [Fuchs] Fuchs, Gerd A.: Die Parkinsonsche Krankheit, Ursachen und Behandlungsformen. – München: Beck, 2002.
- [Gerlach] Gerlach, Manfred; Reichmann, Heinz; Riederer, Peter: Die Parkinson-Krankheit. Grundlagen – Klinik – Therapie, Dritte, überarb. und erw. Auflage. – Wien: Springer-Verlag, 2003.
- [Gottlob] Gottlob, Georg; Frühwirth, Thomas; Horn, Werner (Hrsg.): Expertensysteme. – Wien: Springer-Verlag, 1990.
- [Goetz] Goetz, Christopher G.; Fahn, Stanley; Martinez-Martin, Pablo; Poewe, Werner; Sampaio, Cristina; Stebbins, Glenn T.; Stern, Matthew B.; Tilley, Barbara C.; Dodel, Richard; Dubois, Bruno; Holloway, Robert; Jankovic, Joseph; Kulisevsky, Jaime; Lang, Anthony E.; Lees, Andrew; Leurgans, Sue; LeWitt, Peter A.; Nyenhuis, David; Olanow, C. Warren; Rascol, Olivier; Schrag, Anette; Teresi, Jeanne A.; Van Hilten, Jacobus J.; LaPelle, Nancy: Movement Disorder Society-Sponsored Revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): Process, Format, and Clinimetric Testing Plan. In: Movement Disorders, Vol. 22 No. 1, 2007, S. 41–47.
- [GoF] Gamma, Erich; Helm, Richard; Johnson, Ralph E.; Vlissides, John: Design Patterns. Elements of Reusable Object-Oriented Software. – Amsterdam: Addison-Wesley, 1995.
- [Gross] Gross, Robert E.; Krack, Paul; Rodriguez-Oroz, Maria; Rezai, Ali R.; Benabid, Alim-Louis: Electrophysiological Mapping for the Implantation of Deep Brain Stimulators for Parkinson's Disease and Tremor. In: Movement Disorders, Vol. 21 Suppl. 14, 2006, S. 259–283.

- [Hauser] Hauser, Robert; Lyons, Kelly E.; Pahwa, Rajesh; Zesiewicz, Theresa A.; Golbe, Lawrence I.: Parkinson's Disease. Questions and Answers, Fourth Edition. – Hampshire, West Palm Beach: Merit Publishing International, 2003.
- [Hausotter] Hausotter, Wolfgang: Parkinson in der Praxis, 1. Auflage – Bern/Göttingen/Toronto/Seattle: Verlag Hans Huber, 2003.
- [Heuer] Heuer, Andreas; Saake, Gunter: Datenbanken: Konzepte und Sprachen, 2., aktualisierte und erweiterte Auflage. – Bonn: mitp-Verlag, 2000.
- [HFJava] Bashym, Bryan; Sierra, Kathy; Bates, Bert: Head First Servlets & JSP. – Sebastopol, CA, USA: O'Reilly Media Inc., 2004.
- [HFServletsJSP] Sierra, Kathy; Bates, Bert: Head First Java™, Second Edition. – Sebastopol, CA, USA: O'Reilly Media Inc., 2005.
- [JavaNS] Flanagan, David: Java in a Nutshell. A Desktop Quick Reference, 5th Edition. – Sebastopol, CA, USA: O'Reilly Media Inc., 2005.
- [J2EE] Sun: Java Platform, Enterprise Edition (Java EE): <http://java.sun.com/javaee/> (21.02.2007)
- [Kelly] Kelly, Patrick J.: Stereotactic Thalamotomies. In: Koller, William C.; Paulson, George (Hrsg.): Therapy of Parkinson's Disease, Second Edition, Revised and Expanded, S. 331–351. – New York/Basel/Hong Kong: Marcel Dekker, Inc., 1995.
- [Kemper] Kemper, Alfons; Eickler, André: Datenbanksysteme. Eine Einführung, 6. Auflage. – München: Oldenbourg Wissenschaftsverlag, 2004.
- [Lang1] Lang, Anthony E.: Clinical Rating Scales and Videotape Analysis. In: Koller, William C., Paulson, George (Hrsg.): Therapy of Parkinson's Disease, Second Edition, Revised and Expanded, S. 21–46. – New York/Basel/Hong Kong: Marcel Dekker, Inc., 1995.
- [Lang2] Lang, Anthony E.; Houeto, Jean-Luc; Krack, Paul; Kubu, Cynthia; Lyons, Kelly E.; Moro, Elena; Ondo, William; Pahwa, Rajesh; Poewe, Werner; Tröster, Alexander I.; Utti, Ryan; Voon, Valerie: Deep Brain Stimulation: Preoperative Issues. In: Movement Disorders, Vol. 21 Suppl. 14, 2006, S. 171–196.
- [LeWitt] LeWitt, Peter; Oertel, Wolfgang: Parkinson's Disease. The Treatment Options. – London: Martin Dunitz Ltd, 1999.
- [Machado] Machado, Andre; Rezai, Ali R.; Kopell, Brian H.; Gross, Robert E.; Sharan, Ashwini D.; Benabid, Alim-Louis: Deep Brain Stimulation for Parkinson's Disease: Surgical Technique and Perioperative Management. In: Movement Disorders, Vol. 21 Suppl. 14, 2006, S. 247–258.

- [MathWorks] MathWorks: MATLAB – Fuzzy Logic Toolbox (2006): [http://www.mathworks.com/access/helpdesk/help/pdf\\_doc/fuzzy/fuzzy.pdf](http://www.mathworks.com/access/helpdesk/help/pdf_doc/fuzzy/fuzzy.pdf) (21.02.2007)
- [MDVU] WE MOVE: MD Virtual University: <http://www.mdvu.org/library/ratingscales/pd/> (21.02.2007)
- [Medtronic] Medtronic: Activa Parkinson's Control Therapy: <http://www.medtronic.com/neuro/parkinsons/product.html> (21.02.2007)
- [Mobyen] Mobyen, Uddin Ahmed; Westin, Jerker; Nyholm, Dag; Dougherty, Mark; Groth, Torgny: A Fuzzy Rule-Based Decision Support System for Duodopa Treatment in Parkinson. In: Minock, Michael; Eklund, Patrik; Lindgren, Helena (Editors): Proceedings of SAIS 2006. The 23rd Annual Workshop of the Swedish Artificial Intelligence Society. Umeå, May 10–12, 2006, S. 45–50. <http://sais2006.cs.umu.se/proceedings/> (21.02.2007)
- [MünchenIsar] München rechts der Isar, Neurologische Klinik und Polyklinik der technischen Universität: [http://www.neuro.med.tu-muenchen.de/de/\\_neurologen/\\_fachinformationen/d\\_updrs.shtml](http://www.neuro.med.tu-muenchen.de/de/_neurologen/_fachinformationen/d_updrs.shtml) (21.02.2007)
- [Netcraft] Netcraft: December 2006 Web Server Survey: [http://news.netcraft.com/archives/2006/12/05/december\\_2006\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2006/12/05/december_2006_web_server_survey.html) (21.02.2007)
- [NRCFuzzyJ] National Research Council of Canada's Institute for Information Technology: NRC FuzzyJ Toolkit for the Java™ Platform: [http://www.iit.nrc.ca/IR\\_public/fuzzy/fuzzyJToolkit2.html](http://www.iit.nrc.ca/IR_public/fuzzy/fuzzyJToolkit2.html) (21.02.2007)
- [ÖPG] Österreichische Parkinson Gesellschaft: Konsensusrichtlinien der Österreichischen Parkinson Gesellschaft zur tiefen Gehirnstimulation bei Morbus Parkinson (2003). [http://www.parkinson.at/fileadmin/p-aktuell/2003/P-aktuell-Sonderausgabe2\\_2003.pdf](http://www.parkinson.at/fileadmin/p-aktuell/2003/P-aktuell-Sonderausgabe2_2003.pdf) (21.02.2007)
- [Parkinson] Parkinson, James: An Essay on the shaking palsy. – Sherwood, Neely & Jones, 1817.
- [Pena] Reyes, Peña; Andrés, Carlos: Coevolutionary Fuzzy Modeling. – Heidelberg: Springer-Verlag, 2004.
- [Pinter1] Pinter, M. M.; Alesch, F.; Murg, M.; Seiwald, M.; Helscher, R. J.; Binder, H.: Deep brain stimulation of the subthalamic nucleus for control of extrapyramidal features in advanced idiopathic Parkinson's disease: one year follow-up. In: Journal of Neural Transmission, Vol. 106, 1999, S. 693–709.
- [Pinter2] Pinter, Michaela. M.; Murg, Monika; Alesch, Francois; Freudl, B.; Helscher, R. J.; Binder, H.: Does Deep Brain Stimulation of the Nucleus Ventralis Intermedius Affect Postural Control and Locomotion in Parkinson's Disease? In: Movement Disorders, Vol. 14 No. 6, 1999, S. 958–693.

- [Pollak] Pollak, Pierre; Fraix, Valérie; Krack, Paul; Moro, Elena; Mendes, Alexandre; Chabardes, Stephan; Koudsie, Adnan; Benabid, Alim-Louis: Treatment Results: Parkinson's Disease. In: In: Movement Disorders, Vol. 17 Suppl. 3, 2002, S. 75–83.
- [Przuntek] Przuntek, Horst; Müller, Thomas: Morbus Parkinson. Interdisziplinäre Reflektionen. – Darmstadt: Steinkopff Verlag, 2003.
- [Rezai] Rezai, Ali R.; Kopell, Brian H.; Gross, Robert E.; Vitek, Jerrold L.; Sharan, Ashwini D.; Limousin, Patricia; Benabid, Alim-Louis: Deep Brain Stimulation for Parkinson's Disease: Surgical Issues. In: Movement Disorders, Vol. 21 Suppl. 14, 2006, S. 197–218.
- [Sadegh-Zadeh] Sadegh-Zadeh, Kazem: The Fuzzy Revolution: Goodbye to the Aristotelian Weltanschauung. In: Artificial Intelligence in Medicine, Vol 21, 2001, S. 1–25.
- [Saß] Saß, Henning (Bearb.): American Psychiatric Association: Diagnostisches und statistisches Manual psychischer Störungen DSM-IV: übersetzt nach der vierten Auflage des Diagnostic and Statistical Manual of Mental Disorders der American Psychiatric Association / dt. Bearb. u. Einf. von Henning Saß. 3., unveränd. Aufl. – Göttingen (u.a.): Hogrefe, Verlag für Psychologie, 2001.
- [Schnabel] Schnabel, Manfred: Expertensysteme in der Medizin. Eine Einführung mit Beispielen. – Stuttgart: Fischer Verlag, 1996.
- [UML] Object Management Group: Unified Modeling Language: <http://www.uml.org/> (21.02.2007)
- [Zadeh] Zadeh, Lotfi A.: Fuzzy Sets. In: Information and Control 8, 1965, S. 338–353.
- [Zirh] Zirh, T.A.; Lenz, F.A.; Reich, S.G.; Dougherty, P. M.: Patterns of Bursting Occurring in Thalamic Cells during Parkinsonian Tremor. In: Neuroscience, Vol. 83 No. 1, 1998, S. 107–121.

# Abbildungsverzeichnis

ABBILDUNG 2.1: DIE WICHTIGSTEN DOPAMINERGENEN NEURONENSYSTEME IM MENSCHLICHEN GEHIRN [GERLACH, 2003, S. 47] .....	4
ABBILDUNG 2.2: FORTSCHREITEN DER KLINISCHEN AUSWIRKUNGEN [HAUSER, 2003, S. 62] .....	7
ABBILDUNG 2.3: SCHEMA DER WIRKUNG VON L-DOPA IN PHASE II [GERLACH, 2003, S. 235] .....	8
ABBILDUNG 2.4: SCHEMA DER WIRKUNG VON L-DOPA IN PHASE III [GERLACH, 2003, S. 235] .....	8
ABBILDUNG 2.5: SCHMÄLERUNG DES THERAPEUTISCHEN FENSTERS ÜBER DIE ZEIT [HAUSER, 2003, S. 66] .....	9
ABBILDUNG 2.6: DBS-ELEKTRODE [MEDTRONIC, 2007] .....	22
ABBILDUNG 2.7: IMPULSGENERATOR MIT ELEKTRODE UND KABEL [MEDTRONIC, 2007] .....	22
ABBILDUNG 2.8: RÖNTGENBILD MIT EINER DBS-ELEKTRODE; DAS KABEL WURDE NOCH NICHT VERLEGT [BENABID1, 1995, S. 389] .....	23
ABBILDUNG 2.9: MRT-BILD MIT EINER DBS-ELEKTRODE [BENABID1, 1995, S. 391] .....	23
ABBILDUNG 5.1: KOMPONENTEN EINES WISSENSBASIERTEN SYSTEMS .....	34
ABBILDUNG 6.1: BEISPIEL EINER VORWÄRTSVERKETTUNG [VGL. BEIERLE, 2006, S. 82] .....	47
ABBILDUNG 8.1: SCHEMA EINER FUZZY-INFERENZ [MATHWORKS, 2004, S. 2-27] .....	56
ABBILDUNG 8.2: DREI TERME DER LINGUISTISCHEN VARIABLE „TEMPERATUR“ WERDEN DURCH ZUGEHÖRIGKEITS- FUNKTIONEN ÜBER DEN NUMERISCHEN TEMPERATURWERTEN BESCHRIEBEN (50°C SIND HIER ZU 0,8 „NIEDRIG“, ZU 0,2 „MITTEL“ UND ZU 0,0 „HOCH“) [BOTHE, 1995, S. 11] .....	56
ABBILDUNG 8.3: BEKANNTE IMPLIKATIONSOPERATOREN BEI GEGEBENER REGEL „IF U IS A THEN V IS C“: (A) FUZZY- EINGANGSVARIABLE A MIT EINEM GEGEBENEN ELEMENT X (B) FUZZY-AUSGANGSGRÖÖE C DER REGEL (C) MINIMUM- IMPLIKATION VON $C' = \mu_{A \rightarrow C}(x)$ (D) PRODUKT-IMPLIKATION VON $C' = \mu_{A \rightarrow C}(x)$ [PENA, 2004, S. 11] .....	58
ABBILDUNG 8.4: AUSGEHEND VON EINER GESCHÄTZTEN SCHARFEN MENGE A' KANN EINE REALISTISCHE ZUGEHÖRIGKEITSFUNKTION $\mu_A(x)$ DADURCH KONSTRUIERT WERDEN, DASS NACHBARSCHAFTSBEZIEHUNGEN ZWISCHEN DEN EINZELNEN ELEMENTEN X DEN SCHARFEN KURVENVERLAUF DEFORMIEREN [BOTHE, 1995, S. 28] .....	60
ABBILDUNG 8.5: KONSTRUKTION EINER ZUGEHÖRIGKEITSFUNKTION DURCH EINE REPRÄSENTATIVE AUSWAHL DISKRETER ELEMENTE X, WELCHE BEKANNT SIND UND DIREKT BEWERTET WERDEN KÖNNEN [BOTHE, 1995, S. 28] .....	60
ABBILDUNG 8.6: DREIECK- UND TRAPEZOIDALFUNKTION AUS MATLAB [MATHWORKS, 2004, S. 2-10] .....	61
ABBILDUNG 8.7: GAUß'SCHE GLOCKENKURVE, ZWEI ÜBERLAGERTE GAUß'SCHE GLOCKENKURVEN UND VERALLGEMEINERTE GLOCKENKURVE AUS MATLAB [MATHWORKS, 2004, S. 2-11] .....	61
ABBILDUNG 8.8: SIGMOIDALE KURVE, DIFFERENZ UND PRODUKT ZWEIER SIGMOIDALER KURVEN AUS MATLAB [MATHWORKS, 2004, S. 2-11] .....	61
ABBILDUNG 8.9: Z-, PI- UND S-KURVE ALS BEISPIELE POLYNOMIELLER KURVEN AUS MATLAB [MATHWORKS, 2004, S. 2-13] .....	61
ABBILDUNG 8.10: LOGISCHE OPERATOREN IN ZWEI- UND MEHRWERTIGER LOGIK [MATHWORKS, 2004, S. 2-14] .....	62
ABBILDUNG 8.11: GEBRÄUCHLICHE T-NORM OPERATOREN: (A) MINIMUM, (B) PRODUKT UND (C) BEGRENZTES PRODUKT [PENA, 2004, S. 8] .....	63
ABBILDUNG 8.12: GEBRÄUCHLICHE S-NORM OPERATOREN: (A) MAXIMUM, (B) PROBABILISTISCHE SUMME UND (C) BESCHRÄNKTE SUMME [PENA, 2004, S. 8] .....	64
ABBILDUNG 9.1: STRUKTUR VON FUZZYDBSEXPERT .....	69
ABBILDUNG 9.2: SCHEMA DER PROGRAMMSTRUKTUR VON FUZZYDBSEXPERT .....	74
ABBILDUNG 10.1: PATIENTENDATEN-SET .....	78
ABBILDUNG 10.2: BEISPIEL EINER VARIABLEN IN FUZZYDBSEXPERT .....	79
ABBILDUNG 10.3: PATIENTEN-UPDRS-DATENSATZ .....	80
ABBILDUNG 10.4: GEWICHTUNG UND SUMMENBILDUNG DER UPDRS .....	82
ABBILDUNG 10.5: AUFBAU EINES REGEL-SETS .....	83
ABBILDUNG 10.6: AUFBAU EINER VARIABLEN IN FUZZYDBSEXPERT .....	83
ABBILDUNG 10.7: BEISPIEL EINER REGEL-SET-VARIABLE .....	84
ABBILDUNG 10.8: AUFBAU EINER REGEL-SET-REGEL .....	85
ABBILDUNG 10.9: BEISPIEL EINER REGEL IN FUZZYDBSEXPERT .....	85
ABBILDUNG 10.10: BEISPIEL DER ANWENDUNG DES MAMDANI-MIN-MAX-MIN-OPERATORS [NRCFUZZYJ, 2007] .....	88
ABBILDUNG 10.11: BEISPIEL DER ANWENDUNG DES LARSEN-PRODUCT-MAX-MIN-OPERATORS [NRCFUZZYJ, 2007] .....	89
ABBILDUNG 10.12: HAUPTBEREICHE VON FUZZYDBSEXPERT .....	91
ABBILDUNG 10.13: BENUTZER-MANAGER .....	92
ABBILDUNG 10.14: UPDRS-SETS-MANAGER .....	93
ABBILDUNG 10.15: UPDRS-OPTIONEN-SETS-MANAGER .....	94

ABBILDUNG 10.16: UPDRS-OPTIONEN-SET-OPTIONEN-MANAGER.....	95
ABBILDUNG 10.17: UPDRS-SEKTIONEN-MANAGER.....	96
ABBILDUNG 10.18: UPDRS-SEKTION-FRAGEN-MANAGER.....	97
ABBILDUNG 10.19: VARIABLEN-MANAGER.....	98
ABBILDUNG 10.20: VARIABLEN-BESCHREIBUNGEN-MANAGER.....	100
ABBILDUNG 10.21: REGEL-SETS-MANAGER.....	101
ABBILDUNG 10.22: REGEL-SET-VARIABLEN-MANAGER.....	102
ABBILDUNG 10.23: REGEL-SET-VARIABLEN-SETS-MANAGER.....	103
ABBILDUNG 10.24: REGEL-SET-VARIABLEN-SET-PARAMETER-MANAGER.....	104
ABBILDUNG 10.25: REGEL-SET-REGEL-MANAGER.....	105
ABBILDUNG 10.26: REGEL-SET-REGEL-TEIL-MANAGER.....	106
ABBILDUNG 10.27: REGEL-SET-REGEL-TEIL-ATOM-MANAGER.....	107
ABBILDUNG 10.28: REGEL-SET-GEWICHTE-MANAGER.....	108
ABBILDUNG 10.29: REGEL-SET-UPDRS-GEWICHTE-MANAGER.....	109
ABBILDUNG 10.30: PATIENTEN-MANAGER.....	110
ABBILDUNG 10.31: PATIENTEN-DATENSATZ-MANAGER.....	111
ABBILDUNG 10.32: PATIENTEN-DATENSATZ-VARIABLEN-MANAGER.....	112
ABBILDUNG 10.33: PATIENTEN-DATENSATZ-UPDRS-MANAGER.....	113
ABBILDUNG 10.34: PATIENTEN-DATENSATZ-EVALUIERUNG.....	114
ABBILDUNG 10.35: ERGEBNIS-ÜBERSICHT.....	115
ABBILDUNG 10.36: BEISPIEL EINER REGEL-ANZEIGE.....	116
ABBILDUNG 10.37: BEISPIEL EINER VARIABLEN-ANZEIGE.....	117
ABBILDUNG 10.38: BEISPIEL EINER ANZEIGE DER UPDRS-ERGEBNISSE.....	118
ABBILDUNG 10.39: BEISPIEL DER GRAFISCHEN DARSTELLUNG EINER AUSGANGSVARIABLEN.....	119
ABBILDUNG 10.40: SPRACHAUSWAHL IN FUZZYDBSEXPRT.....	120
ABBILDUNG 10.41: BEISPIEL EINES HEADERS IN FUZZYDBSEXPRT.....	125
ABBILDUNG 10.42: DATABASECRUDPANEL.....	126
ABBILDUNG 10.43: TABELLEN MIT LÖSCH-KASKADEN.....	127
ABBILDUNG 10.44: DER LINGUISTICEXPRESSIONCONSTRUCTOR.....	128
ABBILDUNG 10.45: DATENBANKMODELL DES WISSENSBASIS-TEILES.....	139
ABBILDUNG 10.46: DATENBANKMODELL DES UPDRS-TEILES.....	140
ABBILDUNG 10.47: DATENBANKMODELL DES BENUTZER- UND PATIENTEN-TEILES.....	141
ABBILDUNG 10.48: DATENBANKMODELL DER TABELLEN ZUR LOKALISIERUNG.....	143
ABBILDUNG 10.49: MVC-KONZEPT DES FUZZY-TEILS VON FUZZYDBSEXPRT.....	145
ABBILDUNG 10.50: MVC MUSTER [VGL. FOWLER, 2002, S. 114].....	146
ABBILDUNG 10.51: PAKETE UND KLASSEN IN FUZZYDBSEXPRT.....	148
ABBILDUNG 10.52: PAKET <i>FUZZYDBSEXPRT.FUZZY</i> .....	150
ABBILDUNG 10.53: PAKET <i>FUZZYDBSEXPRT.DATA</i> .....	153
ABBILDUNG 10.54: DATABASECRUDPANEL-KLASSENDIAGRAMM.....	155
ABBILDUNG 10.55: BEISPIELE FÜR MELDUNGEN DES DATABASECRUDPANELS.....	157
ABBILDUNG 10.56: KLASSENDIAGRAMM DES LINGUISTICEXPRESSIONCONSTRUCTORS.....	161

# Tabellenverzeichnis

TABELLE 2-1: WIRKUNGEN EINER TIEFEN HIRNSTIMULATION BEI MORBUS PARKINSON [PRZUNTEK, 2003, S. 53] .....	12
TABELLE 5-1: LOGISCHE VERKNÜPFUNGEN .....	42
TABELLE 10-1: VARIABLENTYPEN IN FUZZYDBSEXPert .....	79
TABELLE 10-2: FUZZY-MODIFIKATOREN .....	86
TABELLE 10-3: OPERATOREN IN FUZZYDBSEXPert .....	86
TABELLE 10-4: BEISPIEL VON UPDRS-FRAGEN MIT MEHREREN EXTREMITÄTEN.....	98
TABELLE 10-5: FORMEN FÜR FUZZY-VARIABLEN IN FUZZYDBSEXPert .....	99
TABELLE 10-6: VORKOMMEN LINGUISTISCHER AUSDRÜCKE IN FUZZYDBSEXPert .....	128
TABELLE 10-7: BEISPIEL FÜR ZUSAMMENGESetzte DATENBANKSCHLÜSSEL .....	136
TABELLE 10-8: DEFUZZIFIZIERUNGSMETHODEN-METHODENNAMEN IN FUZZYDBSEXPert .....	166

# Anhang

## UPDRS-Skala

Die deutsche Übersetzung der UPDRS-Skala wurde von der Homepage München rechts der Isar, Neurologische Klinik und Poliklinik der technischen Universität [MünchenIsar, 2007], entnommen und zwar so, wie sie dort angeführt ist. Das bedeutet, dass die modifizierte Skala nach Hoehn und Yahr und die modifizierte Skala nach Schwab und England in Punkt 4 hineingemischt wurden und eigentlich nicht direkt zur UPDRS-Skala gehören.

(aus: Fahn S, Elton RL, Members of the UPDRS Development Committee (1987) Unified Parkinson's Disease Rating Scale. In: S Fahn, CD Marsden, DB Calne, M Goldstein (eds) Recent Developments in Parkinson's Disease. Vol II. Macmillan Healthcare Information, Florham Park (NJ), pp 153–163, 293–304)

### I. Kognitive Funktionen, Verhalten und Stimmung

#### 1. Intellektuelle Einschränkung:

0 - Keine.

1 - Leicht. Vergesslichkeit mit teilweiser Erinnerung an Ereignisse und keine anderweitigen Schwierigkeiten.

2 - Mäßiger Gedächtnisverlust mit Desorientierung und mäßigen Schwierigkeiten beim Meistern komplexer Probleme. Leichte, aber definitive Einschränkung zu Hause mit der Notwendigkeit einer gelegentlichen Hilfe.

3 - Schwerer Gedächtnisverlust mit zeitlicher und häufig örtlicher Desorientierung. Schwere Einschränkung bei der Bewältigung von Problemen.

4 - Schwerer Gedächtnisverlust, Orientierung nur zur Person erhalten. Kann keine Urteile fällen und keine Probleme lösen. Benötigt bei der persönlichen Pflege viel Hilfe. Kann nicht mehr alleine gelassen werden.

#### 2. Denkstörungen: (als Folge von Demenz oder Medikamenten-Intoxikationen)

0 - Keine.

1 - Lebhaftige Träume.

2 - "Gutartige" Halluzinationen mit erhaltener Einsicht.

3 - Gelegentliche bis häufige Halluzinationen und Wahnvorstellungen; keine Einsicht; könnte sich störend auf die täglichen Aktivitäten auswirken.

4 - Persistierende Halluzinationen, Wahnvorstellungen oder floride Psychose. Kann sich nicht selbst versorgen.

#### 3. Depression:

0 - Nicht vorhanden.

1 - Zeitweise Traurigkeit oder Schuldgefühl stärker als normal, niemals Tage oder Wochen anhaltend.

2 - Anhaltende Depression (1 Woche oder länger).

3 - Anhaltende Depression mit vegetativen Symptomen (Schlaflosigkeit, Appetitlosigkeit, Gewichtsabnahme, Verlust des Interesses).

4 - Anhaltende Depression mit vegetativen Symptomen und Selbstmordgedanken oder -absichten.

#### 4. Motivation/Initiative:

0 - Normal.

1 - Weniger energisch als sonst; stärker passiv.

- 2 - Fehlende Initiative oder Desinteresse an nicht routinemäßigen Aktivitäten.
- 3 - Fehlende Initiative oder Desinteresse an täglichen (routinemäßigen) Aktivitäten.
- 4 - In sich gekehrt, völliges Fehlen von Motivation.

## II. Aktivitäten des täglichen Lebens (jeweils getrennt in on/off-Perioden ermitteln)

### 5. Sprache:

on off

0 - Normal.

- 1 - Leicht beeinträchtigt. Keine Verständigungsschwierigkeiten.
- 2 - Mäßig beeinträchtigt. Wird bisweilen gebeten, etwas zu wiederholen.
- 3 - Stark beeinträchtigt. Wird häufig gebeten, etwas zu wiederholen.
- 4 - Meistens unverständlich.

### 6. Speichelsekretion:

on off

0 - Normal.

- 1 - Gering, aber eindeutig vermehrter Speichel im Mund; nachts gelegentlich Speichelaustritt.
- 2 - Mäßig vermehrte Speichelsekretion; eventuell minimaler Speichelaustritt.
- 3 - Deutlich vermehrte Speichelsekretion mit leichtem Speichelaustritt.
- 4 - Ausgeprägter Speichelaustritt, muß ständig Papiertuch oder Taschentuch benutzen.

### 7. Schlucken:

on off

0 - Normal.

- 1 - Selten Schwierigkeiten beim Schlucken.
- 2 - Gelegentliche Schwierigkeiten beim Schlucken.
- 3 - Braucht weiche Nahrung.
- 4 - Muß über transnasale Magensonde oder Gastrostomie ernährt werden.

### 8. Handschrift:

on off

0 - Normal.

- 1 - Etwas langsam oder klein.
- 2 - Mäßig langsam oder klein; sämtliche Wörter leserlich.
- 3 - Stark beeinträchtigt; nicht alle Wörter leserlich.
- 4 - Die Mehrzahl der Wörter ist unleserlich.

### 9. Speisen schneiden und mit Utensilien umgehen:

on off

0 - Normal.

- 1 - Etwas langsam und unbeholfen, aber keine Hilfe erforderlich.
- 2 - Kann die meisten Speisen schneiden, jedoch unbeholfen und langsam; etwas Hilfe erforderlich.
- 3 - Speisen müssen von jemandem geschnitten werden, kann aber noch langsam essen.
- 4 - Muß gefüttert werden.

**10. Anziehen:**

on off

0 - Normal.

- 1 - Etwas langsam, aber keine Hilfe erforderlich.
- 2 - Gelegentliche Hilfe beim Knöpfen, beim Schlüpfen in die Ärmel.
- 3 - Beträchtliche Hilfe erforderlich, kann aber manches alleine schaffen.
- 4 - Hilfslos.

**11. Hygiene:**

on off

0 - Normal.

- 1 - Etwas langsam, aber keine Hilfe erforderlich.
- 2 - Braucht beim Duschen und Baden Hilfe; oder bei Körperpflege sehr langsam.
- 3 - Braucht beim Waschen, Zähnebürsten, Haarekämmen und beim Gang auf die Toilette Hilfe.
- 4 - Foley-Katheter oder andere mechanische Hilfsmittel.

**12. Umdrehen im Bett und Bettwäsche zurechtziehen:**

on off

0 - Normal.

- 1 - Etwas langsam und unbeholfen, benötigt aber keine Hilfe.
- 2 - Kann sich alleine, jedoch unter großen Schwierigkeiten, herumdrehen und die Bettwäsche zurechtziehen.
- 3 - Beginnt, kann sich aber nicht alleine im Bett umdrehen oder die Bettwäsche zurechtziehen.
- 4 - Hilfslos.

**13. Fallen (unabhängig von Starre):**

on off

0 - Kein.

- 1 - Seltenes Fallen.
- 2 - Gelegentliches Fallen, weniger als einmal pro Tag.
- 3 - Fällt durchschnittlich einmal pro Tag.
- 4 - Fällt häufiger als einmal pro Tag.

**14. Erstarren beim Gehen:**

on off

0 - Kein.

- 1 - Seltenes Erstarren beim Gehen; eventuell verzögerter Start.
- 2 - Gelegentliches Erstarren beim Gehen.
- 3 - Regelmäßiges Erstarren. Gelegentliches Fallen nach Erstarren.
- 4 - Häufiges Fallen nach Erstarren.

**15. Laufen:**

on off

0 - Normal.

- 1 - Leichte Schwierigkeiten. Eventuell fehlendes Mitschwingen der Arme, eventuell Neigung das Bein nachzuziehen.
- 2 - Mäßige Schwierigkeiten, benötigt jedoch wenig oder keine Hilfe.
- 3 - Schwere Gehstörung, benötigt Hilfe.
- 4 - Kann selbst mit Hilfe nicht mehr gehen.

**16. Tremor:**

on off

0 - Keiner.

- 1 - Leicht und selten auftretend.
- 2 - Mäßig; für den Patienten lästig.
- 3 - Stark, bei zahlreichen Aktivitäten hinderlich.
- 4 - Ausgeprägt; bei den meisten Aktivitäten hinderlich.

**17. Sensorische Beschwerden infolge von Parkinsonismus:**

on off

0 - Keine.

- 1 - Gelegentliches Taubheitsgefühl, Kribbeln oder leichte Schmerzen.
- 2 - Häufiges Taubheitsgefühl, Kribbeln oder Schmerzen, nicht störend.
- 3 - Häufig schmerzhaft empfindungen.
- 4 - Unerträgliche Schmerzen.

**III. Motorische Untersuchung****18. Sprache:**

0 - Normal.

- 1 - Leichte Abnahme von Ausdruck, Diktion und/oder Volumen.
- 2 - Monoton, verwaschen, aber verständlich; mäßig behindert.
- 3 - Deutliche Beeinträchtigung, schwer zu verstehen.
- 4 - Unverständlich.

**19. Gesichtsausdruck:**

0 - Normal.

- 1 - Minimal veränderte Mimik, könnte ein normales "Pokergesicht" sein.

2 - Leichte, aber eindeutig abnorme Verminderung des Gesichtsausdruckes.

3 - Mäßig verminderte Mimik; Lippen zeitweise geöffnet.

4 - Maskenhaftes oder erstarrtes Gesicht mit stark oder völlig fehlendem Ausdruck; Lippen stehen um 7 mm auseinander.

**20. Ruhetremor: (G = Gesicht, RH = rechte Hand, LH = linke Hand, RF = rechter Fuß, LF = linker Fuß)**

G RH LH RF LF

0 - Keine.

1 - Leicht und selten vorhanden.

2 - Geringe Amplitude persistierend; oder mäßige Amplitude, aber nur intermittierend auftretend.

3 - Mäßige Amplitude, die meiste Zeit vorhanden.

4 - Ausgeprägte Amplitude, die meiste Zeit vorhanden.

**21. Aktions- oder Haltungstremor der Hände: (R = rechts, L = links)**

R L

0 - Fehlt.

1 - Leicht; bei Bewegung vorhanden.

2 - Mäßige Amplitude, bei Bewegung vorhanden.

3 - Mäßige Amplitude, bei Beibehalten der Haltung und bei Bewegung vorhanden.

4 - Ausgeprägte Amplitude; beim Essen störend.

**22. Rigidity: (Geprüft bei passiver Bewegung der großen Gelenke am sitzenden Patienten. Zahnradphänomen kann ignoriert werden). (N = Nacken, ROE = rechte obere Extremität, LOE = linke obere Extremität, RUE = rechte untere Extremität, LUE = linke untere Extremität)**

N ROE LOE RUE LUE

0 - Fehlt.

1 - Leicht oder nur erkennbar bei Aktivierung durch spiegelbildliche oder andere Bewegungen.

2 - Leicht bis mäßig.

3 - Ausgeprägt, jedoch voller Bewegungsumfang bleibt erreicht.

4 - Stark; Schwierigkeit beim Ausführen aller Bewegungen.

**23. Fingerklopfen: (Patient berührt in rascher Reihenfolge und bei größtmöglicher Amplitude und mit jeder Hand gesondert den Daumen mit dem Zeigefinger). (R = rechts, L = links)**

R L

0 - Normal.

1 - Leichte Verlangsamung und/oder Verringerung der Amplitude.

2 - Mäßig eingeschränkt. Eindeutige und frühzeitige Ermüdung. Bewegung kann gelegentlich unterbrochen werden.

3 - Stark eingeschränkt. Verzögerter Start der Bewegungen oder Unterbrechung fortlaufender Bewegungen.

4 - Kann die Aufgabe kaum ausführen.

**24. Handbewegungen: (Patient öffnet und schließt die Hände in rascher Reihenfolge bei größtmöglicher Amplitude und mit jeder Hand gesondert). (R = rechts, L = links)**

R L

0 - Normal.

1 - Leichte Verlangsamung und/oder Verringerung der Amplitude.

2 - Mäßig eingeschränkt. Eindeutige und frühzeitige Ermüdung. Bewegung kann gelegentlich unterbrochen werden.

3 - Stark eingeschränkt. Verzögerter Start der Bewegungen oder Unterbrechung fortlaufender Bewegungen.

4 - Kann die Aufgabe kaum ausführen.

**25. Rasch wechselnde Bewegungen der Hände: (Pronation-Supinationsbewegungen der Hände, vertikal oder horizontal, mit größtmöglicher Amplitude, beide Hände gleichzeitig).**

R L

0 - Normal.

1 - Leichte Verlangsamung und/oder Verringerung der Amplitude.

2 - Mäßig eingeschränkt. Eindeutige und frühzeitige Ermüdung. Bewegung kann gelegentlich unterbrochen werden.

3 - Stark eingeschränkt. Verzögerter Start der Bewegungen oder Unterbrechung fortlaufender Bewegungen.

4 - Kann die Aufgabe kaum ausführen.

**26. Agilität der Beine: (Der Patient klopft in rascher Reihenfolge mit der Ferse auf den Boden und hebt dabei das ganze Bein an. Die Amplitude soll mindestens 7,5 cm betragen.)**

R L

0 - Normal.

1 - Leichte Verlangsamung und/oder Verringerung der Amplitude.

2 - Mäßig eingeschränkt. Eindeutige und frühzeitige Ermüdung. Bewegung kann gelegentlich unterbrochen werden.

3 - Stark eingeschränkt. Verzögerter Start der Bewegungen oder Unterbrechung fortlaufender Bewegungen.

4 - Kann die Aufgabe kaum ausführen.

**27. Aufstehen vom Stuhl: (Patient versucht mit vor der Brust verschränkten Armen von einem geradelehnigen Holz- oder Metallstuhl aufzustehen).**

0 - Normal.

1 - Langsam; kann mehr als einen Versuch benötigen.

2 - Stößt sich an den Armlehnen hoch.

3 - Neigt zum Zurückfallen und muß es eventuell mehrmals versuchen, kann jedoch ohne Hilfe aufstehen.

4 - Kann ohne Hilfe nicht aufstehen.

**28. Haltung:**

0 - Normal aufrecht.

1 - Nicht ganz aufrecht, leicht vorgebeugte Haltung; könnte bei einem älteren Menschen normal sein.

2 - Mäßig vorgebeugte Haltung; eindeutig abnorm, kann leicht zu einer Seite geneigt sein.

3 - Stark vorgebeugte Haltung mit Kyphose; kann mäßig zu einer Seite geneigt sein.

4 - Ausgeprägte Beugung mit extrem abnormer Haltung.

**29. Gang:**

0 - Normal.

1 - Geht langsam, kann einige kurze Schritte schlurfen, jedoch keine Festination oder Propulsion.

2 - Gehen schwierig, benötigt aber wenig oder keine Hilfe; eventuell leichtes Trippeln, kurze Schritte oder Propulsion.

3 - Starke Gehstörung, benötigt Hilfe.

4 - Kann überhaupt nicht gehen, auch nicht mit Hilfe.

**30. Haltungsstabilität: (Reaktion auf plötzliches Verlagern nach hinten durch Ziehen an den Schultern des Patienten, der mit geöffneten Augen und leicht auseinanderstehenden Füßen geradesteht. Der Patient ist darauf vorbereitet)**

0 - Normal.

1 - Retropulsion, gleicht aber ohne Hilfe aus.

2 - Fehlen einer Haltungsreaktion; würde fallen, wenn er nicht vom Untersucher aufgefangen würde.

3 - Sehr instabil; neigt dazu, spontan das Gleichgewicht zu verlieren.

4 - Kann nicht ohne Unterstützung stehen.

**31. Bradykinesie und Hypokinesie des Körpers: (Kombination aus Langsamkeit, Zögern, verminderten Mitbewegungen der Arme, geringe Bewegungsamplitude und allgemeine Bewegungsarmut)**

0 - Keine.

1 - Minimale Verlangsamung, Bewegung wirkt beabsichtigt; könnte bei manchen Menschen normal sein. Möglicherweise herabgesetzte Amplitude.

2 - Leichte Verlangsamung und Bewegungsarmut, die eindeutig abnorm sind. Alternativ auch herabgesetzte Amplitude.

3 - Mäßige Verlangsamung und Bewegungsarmut oder Herabsetzung der Amplitude.

4 - Ausgeprägte Verlangsamung, Bewegungsarmut oder Herabsetzung der Amplitude.

**IV. Komplikationen der Behandlung (in der vergangenen Woche)****A DYSKINESIEN****32. Dauer: Zu welcher Tageszeit treten die Dyskinesien auf?  
(Anamnestische Angaben)**

0 - Keine.

1 - 1–25% des Tages.

2 - 26–50% des Tages.

3 - 51–75% des Tages.

4 - 76–100% des Tages.

**33. Behinderung: Wie hinderlich sind die Dyskinesien?  
(Anamnestische Angaben; können durch Untersuchung in der Sprechstunde modifiziert werden)**

0 - Keine Behinderung.

1 - Leichte Behinderung.

2 - Mäßige Behinderung.

3 - Starke Behinderung.

4 - Vollständige Behinderung.

**34. Schmerzhaftes Dyskinesien: Wie schmerzhaft sind die Dyskinesien?**

0 - Kleine schmerzhaftes Dyskinesien.

1 - Leicht.

2 - Mäßig.

3 - Stark.

4 - Ausgeprägt.

**35. Auftreten von Dystonie am frühen Morgen: (Anamnestic Angaben)**

0 - Nein.

1 - Ja.

**V. Modifizierte Stadienbestimmung nach Hoehn und Yahr**

(aus: Hoehn MM (1992) The natural history of Parkinson's disease in the pre-levodopa and post-levodopa eras. Neurologic Clinics 10:331-340)

Stadium 0 = Keine Anzeichen der Erkrankung

Stadium 1 = Einseitige Erkrankung.

Stadium 1.5 = Einseitige und axiale Beteiligung.

Stadium 2 = Beidseitige Erkrankung ohne Gleichgewichtsstörung.

Stadium 2.5 = Leichte beidseitige Erkrankung mit Ausgleich beim Zugtest.

Stadium 3 = Leichte bis mäßige beidseitige Erkrankung; leichte Haltungsinstabilität; körperlich unabhängig.

Stadium 4 = Starke Behinderung; kann noch ohne Hilfe laufen oder stehen.

Stadium 5 = Ohne Hilfe an den Rollstuhl gefesselt oder bettlägrig.

**VI. Modifizierte Schwab- und England-Skala der Aktivitäten des täglichen Lebens**

on off

100% - Völlig unabhängig. Kann sämtliche Verrichtungen ohne Verlangsamung, Schwierigkeiten oder Behinderung ausführen. Völlig gesund. Keine Schwierigkeiten wahrgenommen.

90% - Völlig unabhängig. Kann sämtliche Verrichtungen mit geringer Verlangsamung, Schwierigkeiten und Behinderung ausführen. Kann doppelt so lange dazu brauchen. Schwierigkeiten werden bewußt.

80% - Bei den meisten Verrichtungen völlig unabhängig. Braucht dafür doppelt so viel Zeit. Ist sich der Schwierigkeiten und Verlangsamung bewußt.

70% - Nicht völlig unabhängig. Bei manchen Verrichtungen größere Schwierigkeiten. Braucht für einige drei- bis viermal so lange. Muß einen großen Teil des Tages auf die Verrichtungen verwenden.

60% - Leichte Abhängigkeit. Kann die meisten Verrichtungen ausführen, jedoch äußerst langsam und unter viel Anstrengung.

50% - Stärker abhängig. Hilfe bei der Hälfte der Verrichtungen, langsamer usw. Schwierigkeiten bei allem.

40% - Sehr abhängig. Kann bei sämtlichen Verrichtungen mithelfen, nur einige alleine sehr langsam.

30% - Kann bei Anstrengungen hie und da einige Verrichtungen alleine ausführen oder beginnen. Benötigt viel Hilfe.

20% - Kann nichts alleine tun. Kann bei manchen Verrichtungen etwas mithelfen. Stark behindert.

10% - Völlig abhängig, hilflos. Völlig behindert.

0% - Vegetative Funktionen wie Schlucken, Blasen- und Stuhlentleerung sind ausgefallen. Bettlägrig.

**B KLINISCHE FLUKTUATIONEN**

36. Lassen sich "off"-Perioden, z.B. zeitlich nach einer Medikamenteneinnahme voraussagen?

0 - Nein.

1 - Ja.

37. Sind "off"-Perioden zeitlich nach einer Medikamenteneinnahme vorhersagbar?

0 - Nein.

1 - Ja.

38. Treten "off"-Perioden plötzlich auf, z.B. innerhalb von wenigen Sekunden?

0 - Nein.

1 - Ja.

39. Für welche Dauer befindet sich der Patient tagsüber durchschnittlich im "off"-Stadium?

0 - Überhaupt nicht.

1 - 1-25% des Tages.

2 - 26-50% des Tages.

3 - 51-75% des Tages.

4 - 76-100% des Tages.

**C ANDERWEITIGE KOMPLIKATIONEN**

40. Leidet der Patient an Appetitlosigkeit, Übelkeit oder Erbrechen?

0 - Nein.

1 - Ja.

41. Leidet der Patient an Schlafstörungen, z.B. Schlaflosigkeit oder Schläfrigkeit?

0 - Nein.

1 - Ja.

42. Hat der Patient orthostatische Symptome?

0 - Nein.

1 - Ja.

## SQL-Script

Wie in Kapitel 10.3.2.4 beschrieben, erfolgt hier die Auflistung des SQL-Scripts, welches es ermöglicht, die Datenbank inklusive Demo-Daten komplett aufzubauen. An dieser Stelle werden aber die Demo-Daten weggelassen, da dieses Listing sonst zu lange – über 1000 Zeilen – geworden wäre.

```
-- FuzzyDBSexpert
-- Marcus Vitek
-- e9902392
-- FuzzyDBSexpert@vitek.at
--
-- Create FuzzyDBSexpert Tables
--
-- WARNING: this could delete all data!!!
-- use: SOURCE <filename>.sql;
-- or:  Open Script...

-- common constraints:
-- very short text (for Identifiers): VARCHAR(10)
-- short text: VARCHAR(20)
-- middle length text: VARCHAR(50)
-- long text (for forms, fully qualified class names): VARCHAR(256)
-- very long text: VARCHAR(1024)

-- select right character set for Umlauts for example
set names "utf8";

-- drop tables in a sequence considering foreign key constraints

-- drop UPDRS tables in an order that no foreign key constraint fires
DROP TABLE IF EXISTS FuzzyOperators;
DROP TABLE IF EXISTS FuzzyModifiers;
DROP TABLE IF EXISTS GenderTexts;
DROP TABLE IF EXISTS VariableTypeTexts;
DROP TABLE IF EXISTS FuzzySetParameterTexts;
DROP TABLE IF EXISTS FuzzySetTexts;
DROP TABLE IF EXISTS RuleSetVariableSetParams;
DROP TABLE IF EXISTS RuleSetVariableSets;
DROP TABLE IF EXISTS PatientDataSetData;

DROP TABLE IF EXISTS Patient_UPDRS_DataSet;
DROP TABLE IF EXISTS RuleSetUPDRSQuestionWeights;
DROP TABLE IF EXISTS UPDRS_Set_Questions;
DROP TABLE IF EXISTS UPDRS_OptionsSetsOptions;
DROP TABLE IF EXISTS UPDRS_OptionsSets;
DROP TABLE IF EXISTS UPDRS_Set_Sections;

DROP TABLE IF EXISTS RulePartAtoms;
DROP TABLE IF EXISTS RuleParts;
DROP TABLE IF EXISTS RuleSetRules;
DROP TABLE IF EXISTS RuleSetVariables;
DROP TABLE IF EXISTS RuleSetWeights;
DROP TABLE IF EXISTS PatientDataSets;
DROP TABLE IF EXISTS RuleSets;
DROP TABLE IF EXISTS UPDRS_Sets;
DROP TABLE IF EXISTS Patients;

-- 'independent' tables (no foreign key into another table)

DROP TABLE IF EXISTS RulePartTypeTexts;
DROP TABLE IF EXISTS RuleExecutors;
DROP TABLE IF EXISTS DefuzzyMethods;
DROP TABLE IF EXISTS FuzzySetShapes;
DROP TABLE IF EXISTS VariableTexts;
DROP TABLE IF EXISTS Variables;
DROP TABLE IF EXISTS Users;
DROP TABLE IF EXISTS Languages;
```

```
DROP TABLE IF EXISTS test;

CREATE TABLE test(
    testResult INT);

CREATE TABLE Languages(
    LanguageID INT NOT NULL, -- remember: only for languages which's code is
    the same
    Language VARCHAR(10) NOT NULL,
    LanguageName VARCHAR(50) NOT NULL,
    PRIMARY KEY (LanguageID));

CREATE TABLE FuzzySetShapes(
    ShapeID INT NOT NULL,
    ShapeName VARCHAR(50) NOT NULL,
    ShapeClassName VARCHAR(256) NOT NULL,
    ParamCount INT NOT NULL,
    PRIMARY KEY (ShapeID));

CREATE TABLE RuleExecutors(
    ExecutorID INT NOT NULL,
    Name VARCHAR(50) NOT NULL,
    ExecutorClassName VARCHAR(256) NOT NULL,
    PRIMARY KEY (ExecutorID));

CREATE TABLE DefuzzyMethods(
    DefuzzyID INT NOT NULL,
    Name VARCHAR(50) NOT NULL,
    DefuzzyMethodName VARCHAR(100) NOT NULL,
    PRIMARY KEY (DefuzzyID));

CREATE TABLE Users(
    UserID INT NOT NULL,
    UserRole CHAR(1) NOT NULL,
    UserLogin VARCHAR(20) UNIQUE NOT NULL,
    UserPassword VARCHAR(20) NOT NULL,
    UserGrade VARCHAR(20),
    UserName VARCHAR(40) NOT NULL,
    UserFirstName VARCHAR(40) NOT NULL,
    UserGender INTEGER NOT NULL,
    UserEmail VARCHAR(50) NOT NULL,
    UserInstitution VARCHAR(256),
    UserAddress VARCHAR(256),
    UserZIP VARCHAR(20),
    UserCity VARCHAR(20),
    UserState VARCHAR(20),
    UserCountry VARCHAR(20),
    PRIMARY KEY (UserID));

CREATE TABLE Patients(
    PatientID INT NOT NULL,
    PatientCode VARCHAR(50) NOT NULL,
    PatientDOB DATE NOT NULL,
    PatientGender INTEGER NOT NULL,
    Owner INT NOT NULL,
    isPublic BOOLEAN NOT NULL,
    PRIMARY KEY (PatientID),
    FOREIGN KEY (Owner) REFERENCES Users(UserID));

CREATE TABLE Variables(
    VariableID INT NOT NULL,
    VariableName VARCHAR(50) NOT NULL,
    UODLower DOUBLE NOT NULL,
    UODUpper DOUBLE NOT NULL,
    Unit VARCHAR(50),
    Type INT NOT NULL,
    PRIMARY KEY (VariableID));
-- Type: 1=output variable, 2=input variable, 3= UPDRS variable (auto
input)
--      4=UPDRS accumulated score

CREATE TABLE UPDRS_Sets(
    UPDRS_SetID INT NOT NULL,
    UPDRS_SetName VARCHAR(256) NOT NULL,
```

```

UPDRS_SetReference VARCHAR(256),
PRIMARY KEY (UPDRS_SetID));

CREATE TABLE UPDRS_OptionsSets(
UPDRS_SetID INT NOT NULL,
UPDRS_OptionsSetID INT NOT NULL,
UPDRS_OptionsSetDescr VARCHAR(50) NOT NULL,
PRIMARY KEY (UPDRS_SetID, UPDRS_OptionsSetID),
FOREIGN KEY (UPDRS_SetID) REFERENCES UPDRS_Sets(UPDRS_SetID));

CREATE TABLE UPDRS_OptionsSetsOptions(
UPDRS_SetID INT NOT NULL,
UPDRS_OptionsSetID INT NOT NULL,
UPDRS_OptionsSetOptionID INT NOT NULL,
UPDRS_OptionsSetOptionVisu VARCHAR (5) NOT NULL,
UPDRS_OptionsSetOptionValue INT NOT NULL,
PRIMARY KEY (UPDRS_SetID, UPDRS_OptionsSetID, UPDRS_OptionsSetOptionID),
FOREIGN KEY (UPDRS_SetID, UPDRS_OptionsSetID) REFERENCES
UPDRS_OptionsSets(UPDRS_SetID, UPDRS_OptionsSetID));

CREATE TABLE UPDRS_Set_Sections(
UPDRS_SetID INT NOT NULL,
UPDRS_SectionID INT NOT NULL,
UPDRS_SectionName VARCHAR(50) NOT NULL,
VariableID INT NOT NULL,
PRIMARY KEY (UPDRS_SetID, UPDRS_SectionID),
FOREIGN KEY (UPDRS_SetID) REFERENCES UPDRS_Sets(UPDRS_SetID),
FOREIGN KEY (VariableID) REFERENCES Variables(VariableID));

CREATE TABLE UPDRS_Set_Questions(
UPDRS_SetID INT NOT NULL,
UPDRS_SectionID INT NOT NULL,
UPDRS_QuestionID INT NOT NULL,
UPDRS_Nr CHAR(4) NOT NULL,
UPDRS_Question VARCHAR(256) NOT NULL,
UPDRS_QSpec VARCHAR(50),
UPDRS_OptionsSetID INT NOT NULL,
PRIMARY KEY (UPDRS_SetID, UPDRS_SectionID, UPDRS_QuestionID),
FOREIGN KEY (UPDRS_SetID, UPDRS_SectionID) REFERENCES
UPDRS_Set_Sections(UPDRS_SetID, UPDRS_SectionID),
FOREIGN KEY (UPDRS_SetID, UPDRS_OptionsSetID) REFERENCES
UPDRS_OptionsSets(UPDRS_SetID, UPDRS_OptionsSetID));

CREATE TABLE RuleSets(
RuleSetID INT NOT NULL,
RuleSetName VARCHAR(50) NOT NULL,
Owner INT NOT NULL,
UPDRS_SetID INT NOT NULL,
DefaultRuleExecutor INT NOT NULL,
DefuzzyMethod INT NOT NULL,
created TIMESTAMP,
isActive BOOL,
isPublic BOOL,
PRIMARY KEY (RuleSetID),
FOREIGN KEY (Owner) REFERENCES Users(UserID),
FOREIGN KEY (UPDRS_SetID) REFERENCES UPDRS_Sets(UPDRS_SetID),
FOREIGN KEY (DefaultRuleExecutor) REFERENCES RuleExecutors(ExecutorID),
FOREIGN KEY (DefuzzyMethod) REFERENCES DefuzzyMethods(DefuzzyID));

CREATE TABLE PatientDataSets(
PatientID INT NOT NULL,
PatientDataSetID INT NOT NULL,
PatientDataSetName VARCHAR(50) NOT NULL,
RuleSetID INT NOT NULL,
UPDRS_SetID INT NOT NULL,
PatientDataSetDate TIMESTAMP,
PRIMARY KEY (PatientID, PatientDataSetID),
FOREIGN KEY (PatientID) REFERENCES Patients(PatientID) ON DELETE CASCADE,
FOREIGN KEY (RuleSetID) REFERENCES RuleSets(RuleSetID),
FOREIGN KEY (UPDRS_SetID) REFERENCES UPDRS_Sets(UPDRS_SetID));

CREATE TABLE Patient_UPDRS_DataSet(
PatientID INT NOT NULL,
PatientDataSetID INT NOT NULL,

```

```

UPDRS_SetID INT NOT NULL,
UPDRS_SectionID INT NOT NULL,
UPDRS_QuestionID INT NOT NULL,
UPDRS_OptionsSetID INT NOT NULL,
UPDRS_OptionsSetOptionID INT NOT NULL,
PRIMARY KEY (PatientID, PatientDataSetID, UPDRS_SetID, UPDRS_SectionID,
UPDRS_QuestionID),
FOREIGN KEY (PatientID, PatientDataSetID) REFERENCES
PatientDataSets(PatientID, PatientDataSetID) ON DELETE CASCADE,
FOREIGN KEY (UPDRS_SetID, UPDRS_SectionID, UPDRS_QuestionID) REFERENCES
UPDRS_Set_Questions(UPDRS_SetID, UPDRS_SectionID, UPDRS_QuestionID),
FOREIGN KEY (UPDRS_SetID, UPDRS_OptionsSetID, UPDRS_OptionsSetOptionID)
REFERENCES UPDRS_OptionsSetsOptions(UPDRS_SetID, UPDRS_OptionsSetID,
UPDRS_OptionsSetOptionID),
FOREIGN KEY (UPDRS_SetID) REFERENCES UPDRS_Sets(UPDRS_SetID));

CREATE TABLE RuleSetWeights(
RuleSetID INT NOT NULL,
Weight INT NOT NULL,
WeightVis VARCHAR(20) NOT NULL,
WeightValue DOUBLE NOT NULL,
PRIMARY KEY (RuleSetID, Weight),
FOREIGN KEY (RuleSetID) REFERENCES RuleSets(RuleSetID) ON DELETE CASCADE);

CREATE TABLE RuleSetUPDRSQuestionWeights(
RuleSetID INT NOT NULL,
UPDRS_SetID INT NOT NULL,
UPDRS_SectionID INT NOT NULL,
UPDRS_QuestionID INT NOT NULL,
Weight INT NOT NULL,
PRIMARY KEY (RuleSetID, UPDRS_SetID, UPDRS_SectionID, UPDRS_QuestionID),
FOREIGN KEY (RuleSetID) REFERENCES RuleSets(RuleSetID) ON DELETE CASCADE,
FOREIGN KEY (UPDRS_SetID, UPDRS_SectionID, UPDRS_QuestionID) REFERENCES
UPDRS_Set_Questions(UPDRS_SetID, UPDRS_SectionID, UPDRS_QuestionID),
FOREIGN KEY (RuleSetID, Weight) REFERENCES RuleSetWeights (RuleSetID,
Weight));

CREATE TABLE PatientDataSetData(
PatientID INT NOT NULL,
PatientDataSetID INT NOT NULL,
VariableID INT NOT NULL,
LingExValue VARCHAR(50),
NumValue DOUBLE,
PRIMARY KEY (PatientID, PatientDataSetID, VariableID),
FOREIGN KEY (PatientID, PatientDataSetID) REFERENCES
PatientDataSets(PatientID, PatientDataSetID) ON DELETE CASCADE,
FOREIGN KEY (VariableID) REFERENCES Variables(VariableID));

CREATE TABLE RuleSetVariables(
RuleSetID INT NOT NULL,
VariableID INT NOT NULL,
PRIMARY KEY (RuleSetID, VariableID),
FOREIGN KEY (RuleSetID) REFERENCES RuleSets(RuleSetID) ON DELETE CASCADE,
FOREIGN KEY (VariableID) REFERENCES Variables(VariableID));

CREATE TABLE RuleSetVariableSets(
RuleSetID INT NOT NULL,
VariableID INT NOT NULL,
VariableSetID INT NOT NULL,
VariableSetShape INT NOT NULL,
VariableSetName VARCHAR(20),
PRIMARY KEY (RuleSetID, VariableID, VariableSetID),
FOREIGN KEY (RuleSetID, VariableID) REFERENCES RuleSetVariables(RuleSetID,
VariableID) ON DELETE CASCADE,
FOREIGN KEY (VariableSetShape) REFERENCES FuzzySetShapes(ShapeID));

CREATE TABLE RuleSetVariableSetParams(
RuleSetID INT NOT NULL,
VariableID INT NOT NULL,
VariableSetID INT NOT NULL,
ParameterID INT NOT NULL,
Value DOUBLE NOT NULL,
PRIMARY KEY (RuleSetID, VariableID, VariableSetID, ParameterID),

```

```

FOREIGN KEY (RuleSetID, VariableID, VariableSetID) REFERENCES
RuleSetVariableSets(RuleSetID, VariableID, VariableSetID) ON DELETE
CASCADE);

CREATE TABLE RuleSetRules (
  RuleSetID INT NOT NULL,
  RuleID INT NOT NULL,
  RuleExecutor INT,
  isActive BOOLEAN,
  RuleRelation INT,
  RuleWeight DOUBLE,
  PRIMARY KEY (RuleSetID, RuleID),
  FOREIGN KEY (RuleSetID) REFERENCES RuleSets(RuleSetID) ON DELETE CASCADE,
  FOREIGN KEY (RuleExecutor) REFERENCES RuleExecutors(ExecutorID));
-- RuleRelation is for future use when Java Fuzzy Toolkits
-- support other combine operators than AND
-- RuleWeight can be applied when the Java Fuzzy Toolkit
-- supports it

CREATE TABLE RuleParts (
  RuleSetID INT NOT NULL,
  RuleID INT NOT NULL,
  RulePartID INT NOT NULL,
  RulePartType INT NOT NULL,
  VariableID INT NOT NULL,
  PRIMARY KEY (RuleSetID, RuleID, RulePartID),
  FOREIGN KEY (RuleSetID, RuleID) REFERENCES RuleSetRules(RuleSetID, RuleID)
ON DELETE CASCADE,
  FOREIGN KEY (RuleSetID, VariableID) REFERENCES RuleSetVariables(RuleSetID,
VariableID));
-- RulePartType: 1 = Conclusion, 2 = Antecedent

CREATE TABLE RulePartAtoms (
  RuleSetID INT NOT NULL,
  RuleID INT NOT NULL,
  RulePartID INT NOT NULL,
  RulePartAtomID INT NOT NULL,
  RulePartExpression VARCHAR(100) NOT NULL,
  PRIMARY KEY (RuleSetID, RuleID, RulePartID, RulePartAtomID),
  FOREIGN KEY (RuleSetID, RuleID, RulePartID) REFERENCES
RuleParts(RuleSetID, RuleID, RulePartID) ON DELETE CASCADE);
-- This is a very universal way to store the single parts of rule parts.
In the expression one can code
-- IDs of a certain table or linguistic expressions for example. As other
Java Fuzzy Toolkits might store
-- rules very different this universal approach ought to be used.

-- Texts to describe the items of some tables in multiple languages --
-- variable texts (which can be edited)

CREATE TABLE VariableTexts(
  VariableID INT NOT NULL,
  Language VARCHAR(10) NOT NULL,
  Description VARCHAR(1024) NOT NULL,
  PRIMARY KEY (VariableID, Language),
  FOREIGN KEY (VariableID) REFERENCES Variables(VariableID));

-- constant texts (not editable)

CREATE TABLE FuzzySetTexts(
  ShapeID INT NOT NULL,
  Language VARCHAR(10) NOT NULL,
  FuzzySetText VARCHAR(100) NOT NULL,
  PRIMARY KEY (ShapeID, Language),
  FOREIGN KEY (ShapeID) REFERENCES FuzzySetShapes(ShapeID));

CREATE TABLE FuzzySetParameterTexts(
  ShapeID INT NOT NULL,
  ParameterID INT NOT NULL,
  Language VARCHAR(10) NOT NULL,
  FuzzySetParameterText VARCHAR(100) NOT NULL,
  PRIMARY KEY (ShapeID, ParameterID, Language),

```

```
        FOREIGN KEY (ShapeID) REFERENCES FuzzySetTexts(ShapeID));

CREATE TABLE RulePartTypeTexts (
    RulePartType INT NOT NULL,
    Language VARCHAR(10) NOT NULL,
    Text VARCHAR(20) NOT NULL,
    PRIMARY KEY (RulePartType, Language));
-- an own table for RulePartTypes would be too much...

CREATE TABLE VariableTypeTexts (
    VariableType INT NOT NULL,
    Language VARCHAR(10) NOT NULL,
    Text VARCHAR(20) NOT NULL,
    PRIMARY KEY (VariableType, Language));

CREATE TABLE GenderTexts (
    GenderID INTEGER NOT NULL,
    Language VARCHAR(10) NOT NULL,
    GenderText VARCHAR(20) NOT NULL,
    PRIMARY KEY (GenderID, Language));

-- not localized but flexible texts

CREATE TABLE FuzzyModifiers (
    ModifierID INT NOT NULL,
    Name VARCHAR(20) NOT NULL,
    LingEx VARCHAR(20) NOT NULL,
    PRIMARY KEY (ModifierID));

CREATE TABLE FuzzyOperators (
    OperatorID INT NOT NULL,
    Name VARCHAR(10) NOT NULL,
    LingEx VARCHAR(10) NOT NULL,
    PRIMARY KEY (OperatorID));
```