

# **Entwicklung eines Informationssystems für die Neurowissenschaften**

Britta Schinzel

## **Formale Modellierung am Beispiel der Hirntopographie**

Modellierungen spielen in der Informatik eine tragende Rolle, da der Hauptgegenstand der Informatik, das Software-Programm ein Modell für etwas darstellt, und die Aktivitäten der InformatikerInnen, Software zu entwickeln, daher als Modellierungsleistungen zu betrachten sind. Der Begriff Modell wird demgemäss in der Informatik für viele Aufgaben verwendet: es gibt Simulations-, Analyse-, Entwicklungs- Daten-, Semantik-, BenutzerInnenmodelle und viele andere. Modelle modellieren so unterschiedliche Gegenstände und Aufgaben.

Bei der Entwicklung von erweiterbaren interaktiven Anwendungssystemen braucht man flexible Modellierungs- und Entwicklungsmethoden, die für den Wissensbereich der Anwendung adäquat sind und die die Wartung und Erweiterung der Daten- und Wissensbasis unterstützen. Weiter werden meist auch Softwarewerkzeuge für die Datenakquisition, Konsultation und Fortschreibung der Wissensbasis benötigt. Solches Methodenwissen und die Kenntnis bzw. Einübung in entsprechende Techniken werden in einem Informatik-Studium professionalisiert. Darüber hinaus werden im Bereich des Software Engineering Methoden zur Erschließung fremden Wissens und Kenntnisse über User-Führung und Benutzungsschnittstellen erlernt, die menschliches Handeln unterstützen. Mit der wachsenden Interaktivität anstelle von Vollautomatisierung, d.i. der Fähigkeit und Möglichkeit eines Programms, sich in kurzen Abständen an menschliches Handeln rückzukoppeln, wächst insgesamt die Rolle menschlichen Handelns im Design- und Entwicklungsprozess.

## **Modellierungsperspektiven: Abbildung und Konstruktion**

Bei Modellierungsprozessen handelt es sich immer um Erkenntnisprozesse, die eine Beziehung zwischen Modell und Realität herstellen sollen. Zu beachten ist aber, dass damit gleichzeitig für die Artefakte herstellende Informatik auch Herstellungs- und Konstruktionsprozesse in Gang gesetzt sind. Für die epistemologische Relation zwischen Modell und Original lassen sich drei Positionen unterscheiden: Die *Position der Abbild- oder Korrespondenztheorie* (Tarski 1983)

geht davon aus, dass unabhängig von Erkenntnismodellen die Realität, in der wir leben, existiert. Diese Wirklichkeit, die die Rolle des Originals spielt, ist objektiv gegeben. Falls es gelingt, sie in einem Modell abzubilden, so haben wir ein richtiges Modell, eine wahre Erkenntnis: das Modell ist ein Abbild eines Originals. Die *Konstruktivistische Position* (angelehnt an den "radikalen Konstruktivismus" und die Wissenssoziologie, z.B. Winograd & Flores, Maturana, Varela, oder Schmidt) lehnt die Annahme, dass es eine objektive Wirklichkeit für uns gibt, schon deshalb ab, weil unsere Erkenntniswerkzeuge, unsere Sinnesorgane (Sinnestäuschungen) und unser eingeschränktes Denken ebenso wie unsere technischen und wissenschaftlichen Hilfsmittel für eine „objektive“ Erfassung der Wirklichkeit nicht geeignet sind. Statt dessen bedingen/konstituieren Original und Modell sich gegenseitig, man geht davon aus, dass auch jedes Original (sobald benannt, ins Auge gefasst) Ergebnis einer (hermeneutisch bedingten) Modellierung ist. Modelle also bilden die Wirklichkeit nicht ab, sondern schaffen eine eigene Wirklichkeit. Von InformatikerInnen wird meist eine *pragmatische Position* eingenommen: Wirklichkeit ist nicht notwendig ein unbeeinflussbarer und objektiver Erkenntnisgegenstand, aber die Frage, ob unsere Modelle eine objektive Wirklichkeit abbilden oder ob sie diese erst konstruieren, ist falsch gestellt: Ein Modell ist gut, nicht weil es ein wahres Abbild der Wirklichkeit ist, sondern weil wir mit ihm erfolgreich in der Wirklichkeit operieren können. Die ontologische Frage also, die Frage nach dem Status der Wirklichkeit, in der sich die beiden anderen Positionen widersprechen, wird durch die Zweck-Mittel - Relation ersetzt.

Trotz einer pragmatischen Position lassen sich jedoch die oppositionellen erkenntnistheoretischen Annahmen in der Modellierung erkennen und sie können sogar je nach Zweck jeweils als geeignetere Mittel angesehen werden.<sup>1</sup> Grundannahmen der *Abbildungsperspektive* sind nach Grube (1995) einerseits die *Unabhängigkeitsannahme*, d.h. das Original existiert auch ohne Modell und vice versa und die *Vergleichbarkeitsannahme*: Original und Modell können miteinander verglichen werden. Das wichtigste Kriterium für die Relation zwischen Modell und Original ist, dass das Modell *korrekt und vollständig* und damit *verifizierbar* sein muss.

---

<sup>1</sup> Die unterschiedlichen Positionen innerhalb der Informatik zwischen der Forderung nach formaler Spezifikation und Verifikation auf der einen Seite und der Forderung nach Prozessorientierung und Einbeziehung der BenutzerInnen in den Entwicklungsprozess auf der anderen lassen sich auf die Einnahme jeweils einer dieser beiden Perspektiven zurückführen.

Bei der informatischen Modellierung ist aber oft unklar, was das Original und was das Modell ist. Bei der Entwicklung eines Datenmodells etwa, das einen Realitätsausschnitt abbildet, kann man das in Form der Spezifikation rational Rekonstruierte als Original betrachten. Dann sind die obigen Grundannahmen erfüllt, so dass die Kriterien für die Entwicklung und Bewertung des Datenmodells angewendet werden können. Das Original existiert unabhängig vom Modell, und sofern das Original tatsächlich unmissverständlich und eindeutig ist, ergeben sich auch beim Vergleich zwischen ihm und dem Modell für die Überprüfung von Korrektheit und Vollständigkeit keine prinzipiellen Schwierigkeiten.

Betrachtet man aber den zu modellierenden Realitätsausschnitt als Original und die Spezifikation als Modell, so wäre die Unabhängigkeitsannahme zwar erfüllt, nicht jedoch die Vergleichbarkeitsannahme, da unklar ist, in welchem Sinne der Realitätsausschnitt eindeutig und unmissverständlich ist, damit ein Modell mit ihm verglichen werden könnte. Dies zeigt, dass hier das Original, das als Bezug für eine Modellierung in der Abbildperspektive in Frage kommt, selbst ein Modell ist, dass hier also die *Konstruktionsperspektive* eingenommen werden muss. Dabei existieren Original und Modell nicht unabhängig voneinander, sondern konstituieren sich gegenseitig, und das Original kann nicht mit der „Wirklichkeit“ gleichgesetzt werden. Es wird davon ausgegangen, dass das Original, der Realitätsausschnitt, besser: die wie auch immer diffuse oder genaue Vorstellung des Modellierenden über den Realitätsausschnitt, schon das Ergebnis einer Modellierung ist. Zwecke sind unter dieser Perspektive für Original und Modell relevant, d.h. dass auch das Original nur für bestimmte Zwecke Original ist. Kriterium der Konstruktionsperspektive ist die *Angemessenheit*. Da das Original in der Konstruktionsperspektive nicht unabhängig gegeben ist, steht auch seine Angemessenheit zur Disposition.

Die *Modellierungskriterien* dirigieren den Prozess der Modellierung, die Wahl der Sprachmittel und Werkzeuge. Auf ihrer Grundlage wird entschieden, was ein Modell ist und von welcher Qualität es ist. Korrektheit und Vollständigkeit sind zwar Forderungen, die der Abbildperspektive zugeordnet werden, es ist aber möglich, dass die beiden Annahmen erfüllt sind und zugleich die Konstruktionsperspektive eingenommen wird. Das Angemessenheitskriterium ebenso wie die überdies zu fordernden Kriterien der Verständlichkeit, Kommunizierbarkeit und der Benutzungsfreundlichkeit lassen sich beiden Perspektiven zuordnen. Weitere

Modellierungskriterien sind etwa Effizienz, schnelle Erstellbarkeit, Voraussagbarkeit, Änderbarkeit, Erweiterbarkeit, Fehlertoleranz, Portabilität, Integrierbarkeit, Modularität, Sicherheit, Transparenz, Robustheit, Wiederverwendbarkeit, Wartbarkeit und Sozialverträglichkeit. Damit sind Qualitätskriterien genannt, die dann bei der Abbildungsperspektive in der Objektivität und der Funktionalität des Originals (hier der Spezifikation) verankert sind, bei der Konstruktionsperspektive nur indirekt vom Original (hier dem Aspekt der realen Welt) abhängen und in der Interessenlage der Modellierenden verankert sind.

Modelle werden in *Sprache*, in symbolische Repräsentationen, die für Menschen semantische und pragmatische Informationen enthalten und Handlungsanstöße geben, gefasst. Texte (*Spezifikationen*), die das Modell beschreiben, können natürlichsprachlich abgefasst sein (müssen dann aber in formale überführt werden) oder bereits in formaler Sprache. *Symbolische Repräsentationen* für die *semiotische Maschine Computer* sind - meist komplex strukturierte und in Beziehung gesetzte - Datenmengen. Menschen konstruieren und verstehen diese „technische Semiotik“ (Nake 1993) besser auf einer höheren sprachlichen Abstraktionsebene, etwa in logischer oder höherer Programmiersprache, als in Form von Daten und ihren Manipulationsvorschriften. Der Computer kann diese abstrakteren Symbolisierungen mittels Compilern oder Interpretern gemäß eindeutiger Transformationsvorschriften (hier ist die Abbildungsperspektive angebracht) auf die Datenebene bringen. Gesucht sind dann solche maschinellen Realisierungen des Modells, die die formale Spezifikation „erfüllen“. Die Erfüllungsrelation ist mathematischer Überprüfung zugänglich (Verifikation).

Wie aber kommt man zuvor von, wie auch immer „realweltlichen“ Sachverhalten, „Welt“-Modellen in unserer Vorstellung, zu Modellen auf der formalen Ebene, zu symbolischen Repräsentationen? Mit dem Prozess der „*rationalen (Re-)Konstruktion*“ (in Abwandlung von Wittgensteins *rationaler Rekonstruktion* (Stegmüller 1986)) für die *Anforderungsanalyse* ist zwar noch nicht gesagt, welche Methoden dabei zum Einsatz kommen oder ob gar gemäß ungetrübter Vorurteile ad hoc vorgegangen wird, aber dass dabei, am Ende jedenfalls, eine rationalistische, eindeutige, rigide Form des (Re-) Konstruierten erzeugt wird, werden muss, will man explizit, und am Ende formal symbolisieren. Dabei sollte eine konstruktivistische Perspektive eingenommen werden. Das Produkt dieses Prozesses ist die sogenannte *Spezifikation des Ist-Zustandes*, welche die symbolische Repräsentation von Aspekten der „Welt“ mit enthält. Der

Prozess der rationalen (Re-) Konstruktion ist also ein Modellierungs-, ein Erkenntnisprozess, der in Abhängigkeit von Voreinstellungen und Zielen der Beteiligten sehr unterschiedlich ablaufen kann, je nach Anwendungsgebiet unterschiedlich professionalisiert ist und entsprechend unterschiedliche Ergebnisse zeitigt (im Gegensatz zur Physik, wo die „Korrektheit“ der mathematischen Erklärung durch die Vorhersage von Phänomenen der realen Welt überprüfbar ist). Dabei gibt es, wie zu zeigen sein wird, verschiedene Grade der „Härte“ oder „Weichheit“ von Wissensdomänen, die sich in mehr oder weniger einengenden Möglichkeiten der rationalen (Re-)Konstruktion und deren Axiomatisierung widerspiegeln. Um zwei Extreme zu nennen, ist die Geometrie dreidimensionaler Körper sehr viel weniger kontingent axiomatisierbar<sup>2</sup> als etwa der Zusammenhang zwischen Neuroanatomie und Hirnfunktion des Menschen (ganz zu schweigen vom Zusammenhang beider mit Verhalten). Letztere Domäne hängt von vielen Vorannahmen über genetische Determinierung versus Plastizität des Gehirns und seiner Funktion ab, von subjektiven Validierungen der Wichtigkeit, Güte und Validität unterschiedlicher wissenschaftlicher Methoden und Ergebnisse aus verschiedenen Schulen, u.s.w.

Es wird klar, daß die Beziehung zwischen realweltlichen Aspekten und dem axiomatischen Modell einer Interpretation unterliegt, hermeneutischen Charakter hat, und daher einer mathematischen Verifikation nicht zugänglich ist.

Die Spezifikation beschreibt das Endprodukt zweier Leistungen in der *Anforderungsanalyse*, der Beschreibung des Ist-Zustandes und der des Soll-Zustandes in einem Text, sei es in formaler oder natürlichsprachlicher Form.<sup>3</sup> Die Forschungssituation der Anforderungsanalyse ist mit der von qualitativen empirischen Studien vergleichbar. Sie wird jedoch wegen der Zurichtung auf die formal-technische Gesamtaufgabe und die technische Problemlösung teilweise anders bewältigt als etwa in Psychologie oder Soziologie: das sogenannte *Requirements Engineering* befasst sich

---

<sup>2</sup> Diese aber befindet sich in Nähe der mathematischen Geometrie, welche ja Anfang des 20. Jahrhunderts Grundlage für formalistische Auffassung in der Mathematik wurde, weil eine unter verschiedenen Axiomatisierungen des Parallelenaxioms zur nichteuklidischen Geometrie führte, die mit der Anschauung nicht mehr vereinbar war. Sie ist daher auch nicht so „hart“ wie es scheinen mag.

<sup>3</sup> Die Forschungssituation ist dabei nicht mit der von NaturwissenschaftlerInnen vergleichbar, die die mathematische Sprache zur Formulierung von Naturgesetzen verwenden, die Einzelphänomene sowohl zu erklären wie auch vorherzusagen gestattet, also zur Beschreibung bestimmter Aspekte der Realität. Die Rolle von SoftwareentwicklerInnen bei der Spezifikation dagegen beinhaltet, ein textuelles (formales oder natürlichsprachliches) Modell zu konstruieren um es in Software zu übertragen, womit letztlich weitere Handlungen angestoßen werden. Die Rolle der Mathematik ist dabei eine völlig andere: sie ist nicht Mittel zur Beschreibung von Phänomenen der realen Welt (und daher kontinuierlich), sondern sie bezieht sich auf die Interpretationsstrukturen der verwendeten formalen Sprache, und ist daher diskret (Scheffe 1999).

mit der Spezifikation in Form von Checklisten und Werkzeugunterstützung, mittels derer Analyse, Wissenserwerb, Anforderungsmanagement, Konsensbildung, Entwurf und Darstellung etc. (oft auch schon Prozessdefinition, Qualitätssicherung, Evaluation, Validierung) unterstützt werden können. Im Requirements Engineering sind der (Re-)Konstruktions-, wie der die Aufgabe bewältigende Konstruktionsprozess wegen deren kontingenten Inhalte und Ziele natürlich nicht vordefiniert und abgebildet. Vielmehr helfen die Werkzeuge bei der Definition von Form und Prozess auf einer Metaebene. Im wissenschaftlichen Bereich werden diese Hilfsmittel jedoch kaum angewandt und die Spezifikation findet sich meist in (natürlichsprachlicher) Form von Projektanträgen an Drittmittelgeber. Die gängigen Verfahren sind hier informelle Gespräche, Interviews mit ExpertInnen anderer Disziplinen und eigene Literaturstudien der EntwicklerInnen. Der eigentliche Modellierungsprozess wird dann in der Umsetzung des so Erhobenen und auf die Software-Aufgabe Zugerichteten in eine formale Sprache gesehen. Doch lassen sich alle diese Bereiche kaum trennen.

## **Formale Sprachen und Wahrheitssemantik**

Ein Wort über formale Sprachen und deren Semantik ist in diesem Kontext angebracht: Die erste formale Sprache war die, die zur Klärung der Arithmetik gefunden wurde, Freges Begriffsschrift, die Vorläuferin der Prädiaktenlogik, welche in der Informatik gerne benutzt wird. Sie enthielt noch einige Antinomien, die sich durch Russels und später Zermelos Typentheorien jedoch entfernen ließen.

Da bei formalen Sprachen kein Bezug zu Gegenständen, auf die sie sich bezieht, gegeben ist, entsteht die Frage nach der Bedeutung von Worten und Sätzen. Sätze haben auch in natürlicher Sprache oft Wahrheitsgehalte, nämlich solche, die der Übereinstimmung des Denkens mit seinem Gegenstand entsprechen. Es liegt also nahe, auch in formalen Sprachen Bedeutungen mit Wahrheitswerten zu koppeln. Dies geschieht meist über den syntaktisch korrekten Aufbau von Sätzen und über die Ableitung aus Axiomen mittels zugelassener Ableitungsregeln, so etwa in der Prädikatenlogik wie in Tarskis Wahrheitssemantik formaler Sprachen (Tarski 1983). Auch die Semantik von Programmiersprachen wird auf solche Art definiert. Zu beachten ist dabei aber, daß die auf den Menschen gerichtete intentionale Bedeutung der sprachlichen Formulierung, die Sinnzuweisung: „Das System erfüllt die Anforderungen“ eine nicht mathematisch fassbare hermeneutische Interpretationsbeziehung darstellt, während die auf die Maschine gerichtete

mathematisch-logische Interpretation „Das System erfüllt die formale Spezifikation“ unabhängig von Realitätsbezug, aber verifizierbar ist durch: die Semantik der Äquivalenzrelation zwischen Spezifikation und interpretiertem Code liefert den Wahrheitswert 1“.

## **Modellierung in der Prädikatenlogik erster Stufe**

Hier soll am Beispiel eines Informationssystems für die Neurologie eine mögliche Formalisierungsmethode in der Sprache der Prädikatenlogik 1. Stufe (PL 1) vorgestellt werden, die in einem bestimmten Sinn verlässlich ist. Sie sichert nämlich die Erweiterbarkeit des Wissens dadurch, dass durch sogenanntes „Model-Checking“ die Konsistenz (Widerspruchsfreiheit) des Gesamtwissens im System überprüfbar ist. Dies ist notwendig, um komplexer verbundenes Systemwissen ableiten oder evtl. auch induktive Schlüsse ziehen zu können. Die Existenz von Widersprüchen, wie sie durch einfaches „drauflos“ Programmieren und mächtigere Sprachwerkzeuge als PL 1 leicht entstehen können, würde komplizierte „Truth Maintenance“ des Systems erfordern. PL 1 ist deshalb in der Informatik so beliebt, weil in ihrer Vollständigkeit und Widerspruchsfreiheit von Theorien noch beweisbar bleibt. Das ist in anderen für die Informatik von der sprachlichen Ausdruckskraft möglicherweise adäquateren Logiken, wie Modallogiken mit Sprachmitteln für weniger rigide Zusammenhänge (fast immer, möglicherweise, etc.) nicht mehr der Fall<sup>4</sup>.

Zunächst geht es um die Repräsentation (in PL 1) der Geometrie und Nachbarschaftsverhältnisse im Gehirn, die an konkrete 3D - Hirnbilder und -Atlanten gebunden werden können, um eine automatische Verknüpfung und Verarbeitung von neurologischem Wissen mit Lokalitäten im Gehirn zu ermöglichen. Die „Rationale (Re-)Konstruktion“ des zur Formalisierung in Frage stehenden Wissens geschieht unter der Konstruktionsperspektive durch Aufnahme relevanter Entitäten, ihres strukturellen Aufbaus in größere Entitäten, ihrer Rollen und der Beziehungen untereinander. Aus dem so vorbereiteten Wissen wird seligiert, was zur Bearbeitung der funktionellen Aufgaben des Systems notwendig ist. Das nun ausgewählte Wissen wird modularisiert (in sinnvoll abgeschlossene Einheiten unterteilt), um einen modularen Systemaufbau zu ermöglichen. Selektion und Modularisierung orientieren sich auch an der vom System letztendlich erwünschten Funktionalität, im Falle eines Informationssystems also an der

---

<sup>4</sup> Hier ergibt sich ein Konflikt zwischen den beiden auf Mensch und Maschine gerichteten Rollen der informatischen Formalisierung.

Darstellung und Abfrage von Wissen. Dabei kann nicht nur das explizit eingegebene (akquirierte) Wissen abgefragt werden, sondern auch Wissen, das das System selbst produziert: hierbei kann es sich um integriertes, logisch abgeleitetes oder induziertes verallgemeinerndes Wissen handeln, das gewissermaßen implizit im akquirierten Wissen steckt. Die Form oder Repräsentation der Ausgabe kann sich von den Eingaberepräsentationen unterscheiden, wie es etwa bei induktiv aus Daten erzeugten Visualisierungen immer der Fall ist.

Zurück zur Formalisierung: es soll ein Topographiemodul für das Gehirn entwickelt werden, an den sich weiteres formalisiertes und verbildlichtes Wissen anschließen kann. Eine symbolische Wissensbasis für die Hirntopographie kann etwa mit einem digitalen Hirn-Atlanten gekoppelt werden. Die rationale Rekonstruktion kann als wichtige Objektklassen der Hirntopographie etwa Struktur und (Blut- oder Nerven-) Bahn, dazu Nomenklatur, Literaturzitat und Autor identifizieren. Wichtige Beziehungen für diese Objektklassen sind dann „Strukturen sind durch Bahnen verbunden“, „Strukturen lassen sich hierarchisch gliedern“, „Strukturen sind in Nomenklaturen eingebettet“ oder „der Eintrag einer Verbindung ist an ein Literaturzitat gekoppelt“.

Typische Fragestellungen bei der Anfrage ans System könnten sein: Wann erlaubt man die Ableitung von mehrstufigen Projektionen? Was passiert, wenn eine Struktur in verschiedenen Nomenklaturen unterschiedlich eingebettet ist? Können bekannte Afferenzen einer Struktur auch auf ihre Substrukturen übertragen werden?

Die unten (und im Kapitel Hintergrundwissen) vorgestellte Modellierungsmethode (siehe auch Berndel 1992; Brendel et al 1993), die nach der Phase der *Problemanalyse* die Phasen der *Modellkonstruktion* (Modell hier im Sinne der Logik, s.u.), der *Axiomatisierung* und der *Implementierung* unterscheidet, erlaubt die Evaluation von Widerspruchsfreiheit (Konsistenz) und Vollständigkeit (d.h. dass innerhalb der axiomatischen Theorie alle wahren Sätze abgeleitet werden können) der Wissensbasis. Zudem erlaubt sie die hinsichtlich Konsistenz sichere Erweiterbarkeit des Systems. Auf diese Weise ist der Kern des Systems sozusagen a priori „automatisch“ verifiziert und damit von innen her sicher. Von der Außenseite her allerdings sind vor allem Adäquatheit der Wissensmodellierung mit Vollständigkeit der Spezifikation bezüglich der zu leistenden Aufgaben, Benutzbarkeit und Integrationsfähigkeit wichtige Anforderungen für Sicherheit, Wartbarkeit und Wiederverwendbarkeit des Systems. Während die Problemanalyse konstruktivistische Positionen verfolgt, können ausgehend von der Theorie für die

Hirntopographie in PL 1 alle weiteren Phasen unter der Abbildungsperspektive modelliert werden, da bereits die Sprache PL 1 Konsistenz und Vollständigkeit (letztere allerdings wird mit der unten beschriebenen Verwendung saturierter Modelle abgeschwächt) zu überprüfen erlaubt. Die Modellierungsmethode unterstützt eine strukturierte Überführung der theoretischen Konzeption in eine praktische Implementierung.

Die Verwendung einer logischen Repräsentation in der Sprache der Prädikatenlogik erster Stufe PL 1 erlaubt die Darstellung von Formeln, streng aufgebaut aus Funktionszeichen angewendet auf Variablen über einer Menge von Individuen und verknüpft mit logischen Symbolen, wobei zudem über den (Individuen-) Variablen quantifiziert werden kann. Für die Interpretation oder Semantik der PL 1 weist man den Individuen und Variablen ein konkretes Universum zu, den Relationen Relationen darin, und kann über den Aufbau der Formeln eine eindeutige (binäre Wahrheits-) Semantik angeben, die Formeln, für deren Variablen Werte des Universums eingesetzt wurden, die Wahrheitswerte 1 (wahr) oder 0 (falsch) zuweist. Bestimmte Formeln heißen Sätze, und zwar genau dann, wenn sie für alle PL 1-Interpretationen den Wahrheitswert 1 erhalten. Eine Theorie ist eine Menge von Formeln, die die Modellierung eines bestimmten Wissensbereichs darstellen soll. Die Prädikatenlogik erster Stufe ist die umfassendste logische Sprache, in der noch Konsistenz (Widerspruchsfreiheit) und Vollständigkeit (jeder wahre Satz ist ableitbar) von Theorien bewiesen werden können. Aus Komplexitätsgründen können jedoch in der informatischen Modellierung nur kleine Teilsprachen der PL 1 verwendet werden. So sind etwa die den Datenbankeinträgen entsprechenden Formeln einfach und atomar, d.h. sie bestehen nur aus einem Relationszeichen, dies angewendet auf Variablen, Konstanten und logische Symbole. Sie erlauben die Modellierung von sogenanntem Faktenwissen, Bsp.: „Blutbahn1 ist rot“, formal: **rot (b1)**, oder "Hirnareal 1 ist mit Hirnareal 2 durch Bahn 1 verbunden": **Verbindung(s1, b1, s2)**. Aber allgemeines Wissen hat meist kompliziertere logische Formen und erfordert oft auch die Verwendung von Quantoren: „für alle“ (  $\forall$  ) und „es gibt“ (  $\exists$  ), Bsp.: „für alle Hirnareale gilt, dass sie ein benachbartes Areal haben“:  $\forall s \exists t$  (**Nachbar (s,t)**).

Für medizinisches Hintergrundwissen und für die Hirntopographie genügen einfache atomare Formeln nicht. Doch kann durchaus ein großer Teil des Wissens in „datenbankfähige“ Form gebracht werden, wo Erweiterung, Konsistenz und Streichungen effektiv möglich sind und von

wo es - für praktische Zwecke jedenfalls - online<sup>5</sup> abgerufen werden kann. Wichtig ist also aus Effizienzgründen die Unterteilung des zu modellierenden Wissens in eine möglichst klein zu haltende Menge von Regelwissen und die Modellierung eines möglichst großen Teils in Form von Faktenwissen.

Für die *Axiomatisierung* der Hirntopographie muss eine Theorie (eine diese Hirntopographie repräsentierende Menge von Formeln in PL 1) aufgestellt werden. Dies beginnt etwa mit der Festlegung relevanter Objekte (Variablen und Relationen über dem Universum, das die Topographie repräsentieren soll) wie z.B. Hirnareale, Bahnen, Versorgungswege, histologische Einheiten etc.

Hierauf erfolgt eine Zusammenstellung der interessierenden Relationen zwischen den o.a. Objekten. Dazu gehört z.B. die o.a. Relation „Verbindung“. Eine weitere Relation wird durch die morphologische Hierarchie gegeben. Hier werden Fakten der Form "Hirnareal 1 ist morphologische Untereinheit von Hirnareal 2" ausgedrückt: **Sub\_Morph(s1, s2)**

Neben den neuroanatomischen Fakten muss auch das von Menschen implizit gewusste und vorausgesetzte Hintergrundwissen erfasst werden. Darunter ist in unserem Zusammenhang die Gesamtheit der logischen und geometrischen Sachverhalte, die Objekte und Relationen betreffen, zu verstehen. Zum Beispiel: "Wenn Hirnareal 1 morphologische Untereinheit von Hirnareal 2 ist und wenn Hirnareal 2 morphologische Untereinheit von Hirnareal 3 ist, dann ist Hirnareal 1 morphologische Untereinheit von Hirnareal 3"; formal:

$$" s1s2s3([\text{Sub\_Morph}(s1,s2) \dot{\cup} \text{Sub\_Morph}(s2,s3)] \textcircled{R} \text{Sub\_Morph}(s1,s3))$$

Ferner gehören der Theorie H für das Hintergrundwissen auch typische Schlussweisen und Denkmuster neuroanatomischer Expertise an. Selbstverständlich muss die Axiomatisierung unvollständig bleiben. Es geht darum, wesentliche Teile von Faktenwissen und Hintergrundwissen formal zu beschreiben. Endprodukt der Axiomatisierungsphase ist also eine logische Theorie:

$$\mathbf{T = F \dot{\cup} H.}$$

---

<sup>5</sup> Tatsächlich ist die SQL-Anfrage bereits PSPACE-vollständig, also nicht mehr praktisch ausführbar im Sinne der Komplexitätstheorie. Doch betrifft dies normal kurze Anfragen nicht.

bestehend aus Formeln für Fakten- und Hintergrundwissen. Die Bestandteile F und H haben jeweils die Gestalt:

$$\{ f \mid f \text{ atomare oder quantifizierte PL-1 Formel} \}$$

Der Faktenanteil ist der Datenbankbestand. Er ist für die Modellierung von H nicht wesentlich, muss allerdings einer fortwährenden Konsistenzprüfung unterzogen werden. F wird üblicherweise durch einen "Entity-Relationship-Ansatz" modelliert, auf einer anderen Abstraktionsstufe als die atomaren Formeln, leicht graphisch darzustellen und daher für die Entwerfenden besser zu überblicken.

Die Phase der *Modellkonstruktion* liefert das Bindeglied zwischen Axiomatisierung und der letztendlichen Implementierung auf einem Rechner. Es muss für die aufgestellte Hintergrundtheorie H ein Modell konstruiert werden. Dieses Modell besteht aus einer sogenannten Trägermenge M, dem Universum, in der die Objekte konkretisiert und zusammengefasst sind (interpretiert werden). In diesem Fall sind das die konkreten Hirnareale, Bahnen etc. Hinzu kommen Interpretationen für die Beziehungen zwischen den Objekten. Diese Relationen müssen dann entsprechend ihrer Stelligkeit als Teilmengen von M, M<sup>2</sup>, M<sup>3</sup> ... interpretiert werden. Etwas formaler:

$$\text{Modell} = ( M, \{ R^{ij} \mid R^{ij} \hat{=} M^j \} )$$

Jedoch muss die Interpretation der Objekte und ihrer Beziehungen so erfolgen, dass alle in H enthaltenen Formeln gemäß der Semantik von PL 1 in dem konstruierten **Modell** gültig sind, d.h. die Interpretation I (= "wahr") erhalten. Also:

$$(*) \quad \text{Für alle } f \hat{=} H: \text{Modell}_{\text{PL1}} f$$

In der Praxis konstruiert man nicht ein Modell, sondern eine parametrisierte Modellschar Modell(n). So möchte man ja z.B. die Bedingung der Skalierbarkeit erfüllen, d.h. Modelle in beliebig vorgegebener Größe n zur Verfügung haben.

Die Phase der Modellkonstruktion liefert wesentlich zwei Ergebnisse:

1. Den Konsistenznachweis für **H**. Wegen der Eigenschaft (\*) von **Modell** wissen wir, dass **H** keinen Widerspruch enthält, denn es ist nicht möglich, dass für einen gültigen PL1 - Satz **f** und seine Negation  $\neg f$  gilt:

$$\mathbf{Modell}_{PL1} f \quad \mathbf{und} \quad \mathbf{Modell}_{PL1} \neg f$$

2. Das mengentheoretische Modell kann direkt als Grundlage für eine Implementierung auf einem Rechner dienen.

Mittels eines solchen Ansatzes kann zwischen *explizitem und implizitem Wissen* folgendermaßen unterschieden werden: **H** ist, wie schon oben erwähnt, i.a. eine *unvollständige* Theorie. Dabei nennen wir eine PL1-Theorie **T** *vollständig*, genau dann wenn:

$$\mathbf{Für\ alle\ PL1-Sätze\ } f \mathbf{\ gilt: } f \hat{=} T \mathbf{\ oder } \neg f \hat{=} T.$$

Nach Konstruktion der parametrisierten Modellschar **Modell(n)** (n sei hier die Größe von **Modell(n)**) ist **H** in eine vollständige Theorie **H'** eingebettet, nämlich in:

$$\mathbf{H'} := Th(Modell(n)) := \{ f \mid f \text{ PL1-Satz und } Modell(n)_{PL1} f \}$$

Es ist  $H \hat{=} H'$  wegen (\*). Die von der EntwicklerIn aufgestellte Axiomatisierung **H** ist dann das explizite und die Differenz  $H' \setminus H$  das implizite Wissen der Modellierung.

Mit **Ded(T)** bezeichnen wir den *deduktiven Abschluss* (Abschluss unter dem Folgerungsbegriff) einer Theorie **T**, d.h.

$$\mathbf{Ded(T)} := \{ f \mid f \text{ PL1-Satz und } T_{PL1} f \},$$

ist die Menge all der Sätze, die logisch aus **T** ableitbar sind. Mithilfe der bereitgestellten Begriffe kann nun ein formales *Evaluationskriterium* für die Wissensmodellierung angegeben werden. Natürlich gilt:

$$\mathbf{H} \hat{=} \mathbf{Ded(H)} \hat{=} \mathbf{H'}$$

Eine Modellierung ist dann (auf der formalen Ebene, versteht sich) umso *adäquater*, je kleiner die Differenz  $H' \setminus \mathbf{Ded(H)}$  ausfällt. Eine kleine solche Differenz oder gar  $H' = \mathbf{Ded(H)}$  bedeutet nämlich, dass bei der Modellkonstruktion keine zusätzlichen Sachverhalte eingeflossen sind.

Ferner kann im Fall  $\mathbf{H}' = \mathbf{Ded}(\mathbf{H})$  das gesamte implizite Wissen durch logische Inferenz erschlossen werden. Es ist jedoch wichtig, darauf hinzuweisen, dass dieses Kriterium ein immanentes ist, d.h: es sagt nichts darüber aus, wie gut der Gegenstandsbereich - das neuroanatomische Wissen - axiomatisiert wurde. Dies wäre auch durch ein formales Kriterium nicht zu erfassen. Jedoch kann man daraus ermessen, wie adäquat das konstruierte Modell für die Axiomatisierung ist.

### *Saturierte Modelle*

Die Modelle **Modell(n)** erfüllen die Axiome der Hintergrundtheorie **H**. Der Faktenbestand **F** jedoch, der in einem ER-Modell bzw. einem relationalen Datenbankschema repräsentiert ist, muss ebenfalls modelliert sein, d.h auch für die atomaren Fakten  $\mathbf{f}_i$  muss gelten:

$$\mathbf{Modell}(n)_{PL1} \mathbf{f}_i$$

Dies ist die eigentliche Schnittstelle zwischen den unterschiedlichen Repräsentationsformen. Diese Verbindung kann realisiert werden, indem man Modelle **Modell(n)** konstruiert, die *saturiert* sind. Saturierte Modelle (Chang 1973) sind solche, in die man andere Modelle eins-zu-eins und strukturerhaltend (monomorph) einbetten kann. Genauer: Sei **Modell** saturiert und gelte  $\mathbf{Modell}_{PL1} \mathbf{H}$ . Sei ferner **Modell'** ein weiteres Modell von **H** mit kleinerer Elementanzahl. Dann gibt es eine eindeutige strukturerhaltende (monomorphe) Abbildung von **Modell'** in **Modell**, d.h. **Modell'** ist schon (bis auf Isomorphie) in **Modell** eingebettet, anders ausgedrückt: **Modell** beschreibt **Modell'** mit.

Der Vorteil bei der Verwendung saturierter Modelle gegenüber der einfachen Angabe eines konkreten "Hirnmodells" für die ganze Theorie  $\mathbf{F} \Rightarrow \mathbf{H}$  ohne nachträgliche Einbettung des Faktenbestandes in das saturierte Modell von **H** ergibt sich aus der hybriden Wissensbasis. Durch saturierte Modelle kann der "konstante" Teil des neuroanatomischen Wissens, einmal modelliert, immer wieder genutzt werden. Die sich fließend ändernde und erweiternde Faktenbasis lässt sich einfacher in einem Datenbankschema verwalten und dann in die saturierten Modelle der Hintergrundtheorie einbetten, so dass die Modelle nicht immer wieder neu konstruiert werden müssen. Die Methode ist trotz Abbildungsperspektive und formaler Rigidität offen gegen Erweiterungen.

Der Hauptbestandteil der abschließenden Phase der *Implementierung* ist die Umsetzung des obigen **Modells** in entsprechende Datenstrukturen bzw. die Prozeduralisierung der Semantik für die Relationen. Semantik<sup>6</sup> meint hier eine auf den Rechner übertragene korrespondenztheoretische „operationale“ oder „Transformations- Bedeutung“, wie nämlich die formalen Sprachmittel auf der Basisebene des Maschinencodes von ihm ausgeführt werden müssen („technische Semiotik“). So muss z.B. die o.a. **Sub\_Morph** - Relation durch die Suche in einer baumartigen Datenstruktur ermittelt werden. Für die meist  $\leq 3$  - stelligen Relationen, die in **T** auftreten, existieren effiziente Standard-Datenstrukturen und entsprechende Such- und Manipulationsalgorithmen in entsprechenden elektronischen Bibliotheken.

Durch die strukturierte 3-Phasen Modellierung kann ein gutes Monitoring der gewünschten Systemperformanz gewährleistet werden. Insbesondere die Modellkonstruktion liefert eine Brücke zwischen rein logisch-deskriptivem Material und der Realisierung auf dem Rechner und hat zudem den Vorteil beweisbarer Korrespondenz unter der Abbildungsperspektive.

### *Hintergrundwissen*

Axiomatisierung und Modellkonstruktion sind zwei Prozesse, die wechselseitig ineinander greifen und von Anfang an zusammen betrachtet werden müssen. In der Wissensmodellierung kann man nicht beliebige Formeln aufstellen, ihnen im Nachhinein Bedeutungen zuweisen und hoffen, damit ein Modell eines interessierenden Problemkreises gefunden zu haben. Vielmehr hat man diesen Problemkreis mitsamt seinen im (Re-)Konstruktionsprozess identifizierten Objekten, Relationen, Regeln von Anfang an im Auge und versucht, ihn durch eine Axiomatisierung formal zu beschreiben. Anders gesagt, eine Axiomatisierung ist an möglichst konkreten Vorgaben orientiert, und das maximal mögliche Maß an Adäquatheit für Domäne und Aufgabe, an „Re-“ im vollen Bewusstsein der „Konstruktion“, ist zu versuchen. Die im folgenden vorgestellten Axiome werden daher durch kleine schematische Zeichnungen motiviert. Die Relationssymbole erhalten mnemotechnische Abkürzungen.

---

<sup>6</sup> Informatische Semantik ist immer formal, und daher vom natürlichsprachlichen Standpunkt aus betrachtet Syntax. Es gibt verschiedene Arten von Programmiersprachensemantiken, operationale, prozedurale, denotationelle, etc., die Unterschiedliches formal fassen. Es geht dabei um die Transformationen, die der Computer aufgrund der formal-sprachlichen Anweisungen ausführen soll. Nur selten lassen sie sich in Form von geschlossenen mathematischen Formeln angeben. Meist müssen sie wie bei der Semantik von PL 1 stufenweise als Zustandstransformationen des Rechners aufgebaut werden.

Die Axiome sind in einer prädikatenlogischen Sprache erster Stufe (gemäß den üblichen Standards, siehe z.B [Eb]) formuliert, das Alphabet der Sprache enthält folgende Symbole:

- a) Logische Symbole: Junktoren, Quantoren, Klammersymbole, Gleichheitszeichen
- b) Variablen:
 

$s, s_0, s_1, s_2, \dots$	(Variablen für Struktursymbole)
$b, b_0, b_1, b_2, \dots$	(Variablen für Bahnsymbole)
- c) Funktionssymbole
  - 0-stellig (Konstanten):  $e_s, e_B$
- d) Relationssymbole
  - 2-stellig:
 

$Neuro\_weg,$	$Norm\_Weg,$	$Sub\_Weg,$
$Morph\_Teil\_Von,$	$Funk\_Teil\_Von,$	
$Sub\_Morph,$	$Sub\_Funk,$	
$Funk\_Morph\_Einbett$		
  - 3-stellig:
 

$Verbindung$
--------------

*a) Heterarchieaxiome*

Funktionelle Zentren und Morphologische Areale lassen sich in Hierarchien einordnen. Einzelne Strukturen sollen aber in verschiedene funktionelle bzw. morphologische (Ober-)Systeme aufgenommen werden können, wodurch die Baumstruktur von Hierarchien verletzt wird. Dafür müssen die Hierarchien zu Heterarchien (i.e. zu irreflexiven Ordnungen) erweitert werden, d.h die Eindeutigkeit der „Mutterbeziehung eines Knotens“ wird aufgelöst. Hier ist dies festlegbar mit:

Kein Funktionelles Zentrum (morphologisches Areal) ist eine Sub-Struktur von sich selbst (Irreflexivität der Sub-Beziehung):

- 1a)**  $\forall s \neg Sub\_Funk(s, s)$
- 1b)**  $\forall s \neg Sub\_Morph(s, s)$

Ist eine Struktur ein Teil einer anderen Struktur, so ist es auch eine Sub-Struktur davon:

- 2a)**  $\forall s_1 s_2 [Funk\_Teil\_Von(s_1, s_2) \Rightarrow Sub\_Funk(s_1, s_2)]$
- 2b)**  $\forall s_1 s_2 [Morph\_Teil\_Von(s_1, s_2) \Rightarrow Sub\_Morph(s_1, s_2)]$

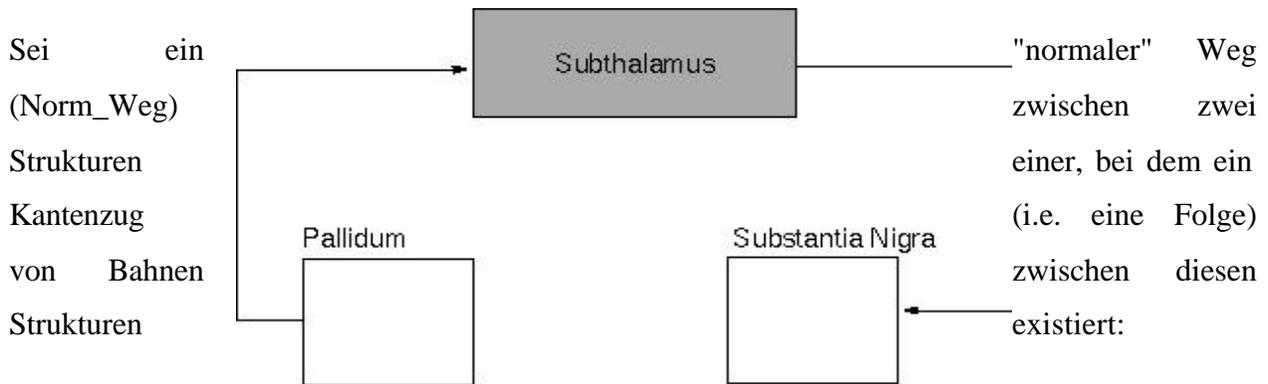
Die Sub-Beziehung ist transitiv, d.h. ist  $s_1$  Teil von  $s_2$  und  $s_2$  Teil von  $s_3$  so ist  $s_1$  Teil von  $s_3$

- 3a)**  $\forall s_1 s_2 s_3 [Funk\_Teil\_Von(s_1, s_2) \wedge Sub\_Funk(s_2, s_3) \Rightarrow Sub\_Funk(s_1, s_3)]$
- 3b)**  $\forall s_1 s_2 s_3 [Morph\_Teil\_Von(s_1, s_2) \wedge Sub\_Morph(s_2, s_3) \Rightarrow Sub\_Morph(s_1, s_3)]$

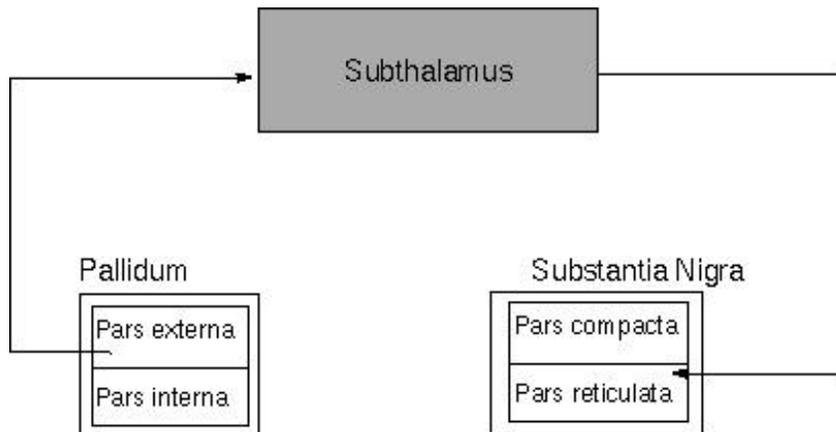
*b) Wegaxiome*

Die Relation Verbindung dient in diesem System als Eingaberelation und wird vom Wissensakquisiteur durch die Angabe von Verbindungsfakten festgelegt. Ein Verbindungsfaktum kann zwischen je zwei Funktionellen Zentren bzw. je zwei Morphologischen Arealen, aber auch

zwischen einem Funktionellen Zentrum und einem Morphologischen Areal existieren. Mit den Wegaxiomen lassen sich mehrstufige Projektionen (also Wege im Sinne der Graphentheorie) inferieren. Auf folgender Abbildung existiert ein Weg vom Pallidum über den Subthalamus zur Substantia nigra:



- 4) "  $s_1 s_2 b$  Verbindung( $s_1, b, s_2$ )  $\otimes$  Norm\_weg( $s_1, s_2$ )
- 5) "  $s_1 s_2 s_3 b$  [Verbindung( $s_1, b, s_2$ )  $\dot{\cup}$  Norm\_Weg( $s_2, s_3$ )]  $\otimes$  Norm\_Weg( $s_1, s_3$ )



Für die Sachverhalte in der folgenden Abbildung kann weiter axiomatisiert werden:

Es existiert ein Norm\_Weg von der Pars externa des Pallidums über den Subthalamus zur Substantia nigra, Pars reticulata. Mit der Pars externa hat aber erst recht deren Oberstruktur Pallidum eine Wegverbindung mit der Substantia nigra. So kann gefolgert werden, dass eine Struktur alle Norm\_Wege seiner Substrukturen besitzt. Solche umgekehrt vererbten Wege nennen wir "Sub\_Wege".

Die Symbolisierung ist dreigeteilt, je nachdem, ob die Substruktur-Beziehung vor (Axiom 6), nach (Axiom 7) oder vor *und* nach (Axiom 8) in einem Norm\_weg existiert:

- 6) "  $s_1 s_2 s_3 [ ( \text{Sub\_Funk}(s_1, s_2) \dot{\cup} \text{Sub\_Morph}(s_1, s_2) ) \dot{\cup} \text{Norm\_Weg}(s_2, s_3) ] \textcircled{R}$   
 $\text{Sub\_Weg}(s_1, s_3)$
- 7) "  $s_1 s_2 s_3 [ \text{Norm\_Weg}(s_1, s_2) ] \dot{\cup} ( \text{Sub\_Funk}(s_2, s_3) \dot{\cup} \text{Sub\_Morph}(s_2, s_3) ) \textcircled{R}$   
 $\text{Sub\_Weg}(s_1, s_3)$
- 8) "  $s_1 s_2 s_3 s_4 [ ( \text{Sub\_Funk}(s_1, s_2) \dot{\cup} \text{Sub\_Morph}(s_1, s_2) ) \dot{\cup} \text{Norm\_Weg}(s_2, s_3) \dot{\cup} ( \text{Sub\_Funk}(s_3, s_4) \dot{\cup} \text{Sub\_Morph}(s_3, s_4) ) ] \textcircled{R}$   
 $\text{Sub\_Weg}(s_1, s_4)$

Das letzte Axiom fasst Norm\_Wege und Sub\_Wege unter dem Oberbegriff Neuro\_Weg zusammen:

- 9) "  $s_1 s_2 [ \text{Norm\_Weg}(s_1, s_2) \dot{\cup} \text{Sub\_Weg}(s_1, s_2) ] \ll \text{Neuro\_Weg}(s_1, s_2)$

Das Axiomensystem besteht aus den Axiomen 1) - 9)<sup>7</sup>, die dazugehörige Theorie T ist der Abschluss dieses Axiomensystems unter dem Folgerungsbegriff Ded („1 – 9“).

Man sieht, dass die Axiomatisierung der Teil\_von - Beziehung wie der des transitiven Abschlusses von Verbindungen durch Wege bis auf Namensgebungen wenige Freiheiten bietet. Doch wäre bezüglich der Geometrie sehr viel mehr, und manches davon dann auch kontingent axiomatisierbar. Dennoch kann man mit der Hirngeometrie von einem ausgesprochen „harten“, d.h. verhältnismäßig eindeutigen und damit gern als „objektiv“ bezeichneten Wissensbereich sprechen. Dies ändert sich in zunehmendem Maße für die folgenden Beziehungen zwischen Objekten.

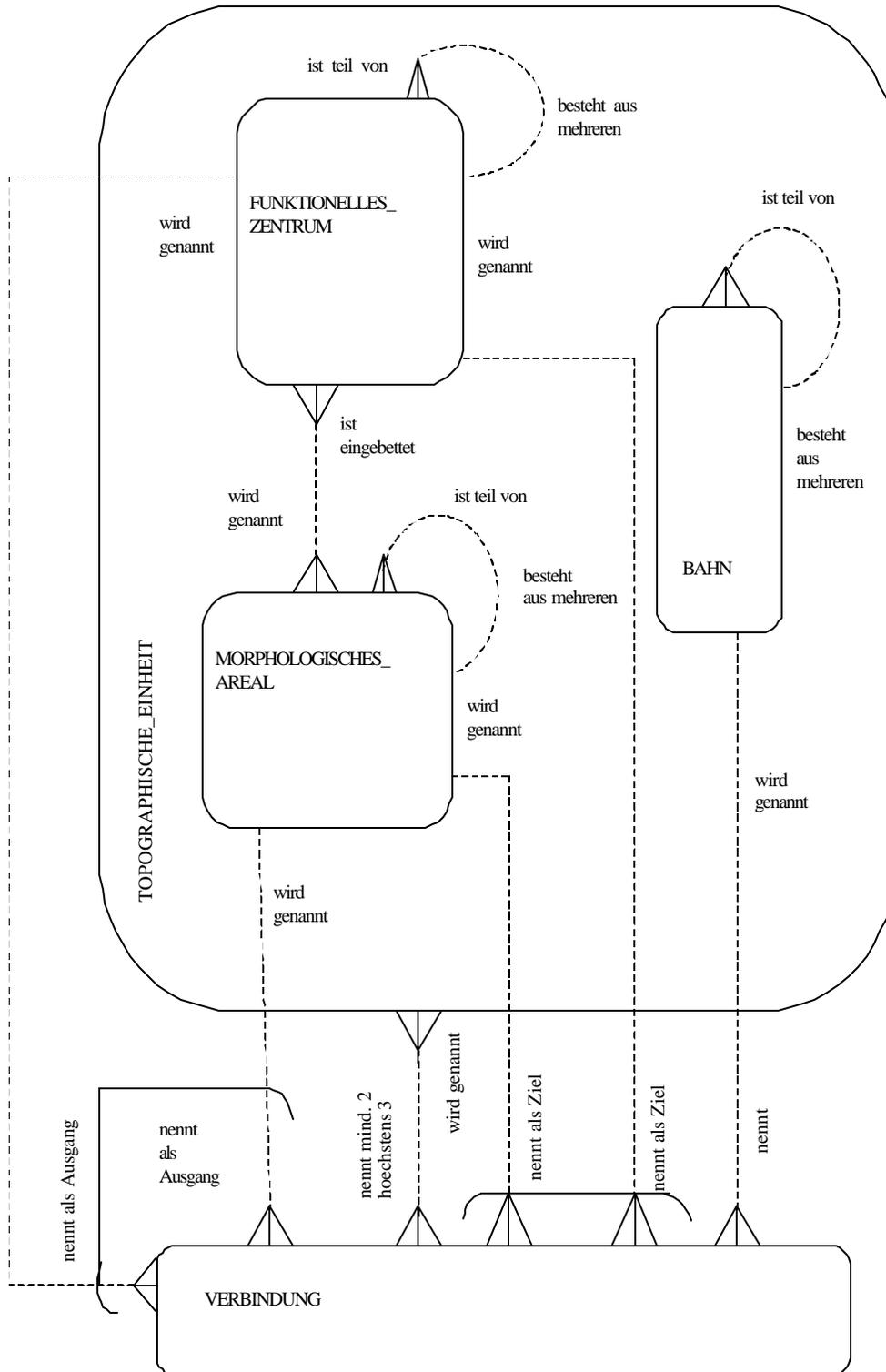
### **Wissensbasis**

---

<sup>7</sup> Weitere Axiome, etwa für die Unterscheidung von Nachbarschaften zwischen morphologischen Arealen in zusammenhängenden Strukturen und Nachbarschaften in Sulci, können mittels einer Nachbarschaftsrelation für zusammenhängende Areale eingeführt und für Sulci mit Hilfe des Faltungsoperators differenziert werden.

Das (atomare) *Faktenwissen* kann in einer relationalen Datenbank gespeichert werden. Dafür werden üblicherweise CASE-Werkzeuge (Computer Aided Software Engineering) verwendet, die den Entwicklungsprozess von der Strategie- bis hin zur Implementationsphase unterstützen.

Konzeptuelle Modelle werden meist in Form von Entity-Relationship-Diagrammen (ERD) angegeben. Hier eines, das die Struktur des Hirntopographiewissens beschreibt.



Mit den Entities werden Objekte zusammengefasst, um die wichtigsten Beziehungen zwischen ihnen zu definieren.

Zur Beschreibung von Wissen über Hirnregionen bzw. Areale und den Projektionen zwischen den verschiedenen Arealen können etwa die Entities TOPOGRAPHISCHE\_EINHEIT, MORPHOLOGISCHES\_AREAL, FUNKTIONELLES\_ZENTRUM, BAHN und VERBINDUNG eingeführt werden:

Unter dem Entitätstyp TOPOGRAPHISCHE\_EINHEIT sind morphologische Areale und funktionelle Zentren zusammengefasst. Morphologische Areale sind bestimmte räumliche Bereiche mit definierten Arealgrenzen. Sie können die Kopplung zu den Strukturen eines (3 D-)digitalen Hirnatlas bilden. Jede in einem Voxelatlas enthaltene Struktur hat dann sein symbolisches Pendant in der Menge der morphologischen Areale.

MORPHOLOGISCHES\_AREAL besitzt daher auch ein Koordinatenattribut, dessen Inhalt ein Zeiger auf die Menge der Voxelkoordinaten einer vermessenen Struktur ist. Zusätzlich kann die Wissensbasis auch morphologische Areale enthalten, die im digitalen Atlas nicht enthalten sind, d.h. noch nicht vermessen sind.

Funktionelle Zentren sind im Gegensatz zu morphologischen Arealen Objekttypen ohne geschlossene räumliche Abgrenzungen. Sie können einen Bezug zu einem oder mehreren morphologischen Arealen haben oder auf das gesamte Hirn verteilt sein. Ist eine topographische Unterscheidung zwischen morphologischen Arealen und funktionellen Zentren nicht möglich, so kann sie beispielsweise dadurch erreicht werden, dass die beiden Typen in unterschiedliche (morphologische oder funktionelle) Nomenklaturen eingeordnet werden. Dies ist ein Beispiel für Modellierung, die schon kontingent ist und die ModelliererIn herausfordert, die Diversifizierung von Morphologie und Funktion möglichst adäquat zu verknüpfen.

Auch die folgende Unterscheidung zwischen Bahn und Verbindung, die zunächst nicht unmittelbar einleuchten mag, hat kompliziertere pragmatische Gründe.

Eine BAHN verbindet zwei Strukturen (morphologische oder funktionelle) miteinander. Wie bei den MORPHOLOGISCHES\_AREALen besitzt BAHN ein "Koordinaten-Attribut" für die Kopplung von digitaler und symbolischer Komponente. Der Typ einer Bahn bezeichnet das "Medium", durch das der Zusammenhang realisiert wird. Er kann z.B. durch eine Faser, eine Blutbahn oder auch über den Liquor erfolgen.

Die rekursiven Beziehungen "Teil von" und „besteht aus mehreren“ der Entities MORPHOLOGISCHES\_AREAL, FUNKTIONELLES\_ZENTRUM und BAHN bilden eine hierarchische Anordnung innerhalb der topographischen Einheiten und Bahnen ab, die Relation „ist eingebettet“ und „wird genannt“ gestattet unterschiedlich strikte Relationen zwischen MORPHOLOGISCHES\_AREAL und FUNKTIONELLES\_ZENTRUM.

Die Entity VERBINDUNG hat zwei Beziehungen "nennt als Ausgang" und "nennt als Ziel" zu der Entity TOPOGRAPHISCHE\_EINHEIT sowie eine Beziehung "nennt" zu BAHN. Durch diese Beziehungsdefinitionen ist man in der Lage, die Projektionen mit Anfangs- und Zielzentrum sowie der verbindenden Bahn zu beschreiben. Es ist daher nicht notwendig, die Afferenzen und Efferenzen explizit anzugeben. Überdies öffnen sie Erweiterungsmöglichkeiten für diesbezüglich unterschiedliche Literaturangaben.

Weitere sinnvolle Entitäten (der Übersichtlichkeit halber in der Graphik nicht eingetragen) können WEG, DESKRIPTOR, NOMENKLATUR, LITERATUR und AUTOR sein: Das System soll nämlich in der Lage sein, nach bestimmten Regeln einzelne Verbindungen zu mehrstufigen Projektionen (Wegen) zusammenzufassen. Damit man die inferierten Wege von explizit eingetragenen Wegen unterscheiden kann, kann eine (hier nicht eingetragene) Entity WEG mit einer Assoziation zu VERBINDUNG eingeführt werden.

Die Zusammenhänge topographischer Einheiten mit Nomenklaturen können wie folgt strukturiert werden: Jeder Vertreter der Entities BAHN und TOPOGRAPHISCHE\_EINHEIT erhält mindestens eine Beschriftung in Form einer Entity DESKRIPTOR. Die Entity enthält Informationen über den Namen, evtl. Kurznamen und die verwandte Sprache bei der "Beschriftung" einer Bahn oder Struktur. Der kombinierte Primärschlüssel (Kombination von Attributnummer und der Assoziation "wird benannt" zur NOMENKLATUR) soll sicherstellen, dass eine Beschriftung innerhalb einer Nomenklatur eindeutig ist. Neben dem Namen einer Nomenklatur könnte die Entity NOMENKLATUR dann einen Attributtyp enthalten, dessen Wertebereich etwa "standard", "funktionell" und "künstlich" umfasst, wobei die ersten beiden Nomenklaturen Beschreibungen von morphologischen Arealen und funktionellen Zentren enthalten. Die künstliche Nomenklatur soll eine vor der BenutzerIn versteckte Nomenklatur sein, die dazu dient, die "überlappende Einbettung" von Strukturen verschiedener Nomenklaturen zu realisieren, um Zusammenhänge zwischen Nomenklaturen in verschiedenen Literaturstellen zu identifizieren. Dazu sind weitere Entitäten zur Anordnung von Strukturen einer oder verschiedener Nomenklaturen in eine transitive, asymmetrische, irreflexive Ordnung, in eine Heterarchie nötig. Damit ließen sich dann auch funktionelle Zentren verschiedenen morphologischen Arealen zuordnen (s.o.).

Für alle Einträge in der Faktenbasis kann die Quelle des Wissens eingetragen werden. Dafür sollen sämtliche Entities mit den Entities LITERATUR und/oder AUTOR verbunden werden. Die Entity LITERATUR enthält z.B. Angaben über Titel, Erscheinungsjahr und das Journal, in dem der Artikel veröffentlicht wurde. Das Attribut *hypothese* kann einen kurzen Text (als Textkonserve, ohne logische Integration ins System) aufnehmen, in dem man die Kernaussage des Artikels beschreiben kann.

Soviel soll genügen, um die Prinzipien, aber auch die Offenheit der Axiomatisierung gegenüber Zielen, Orientierungen, Vorlieben, Unkenntnis, aber auch Idiosynkrasien und biases zu verdeutlichen. Vor allem das, was nicht modelliert, was aus- und weggelassen wird, ist verdächtig, diese negative Seite der Freiheit zu transportieren. *Modellierung*, so sieht man, ist kein objektiver, distanzierter, neutraler Prozess, sie ist immer ein interpretativer Akt, bei dem die Bedeutungszuschreibung selektiven Vorverständnissen, Zielen und Wünschen folgt (Funken, Schinzel 1996). Einmal formalisiert, ist jedoch das Modell von dem gesamten Entstehungszusammenhang „dekontextualisiert“ und wird dann als Technofakt in andere Zusammenhänge „rekontextualisiert“. Die im sogenannten Softwarezement festgebackenen kontingenten Modellentscheidungen werden gleichzeitig darin auch versteckt und unsichtbar, wirken aber dennoch als Leitbilder, Denkmuster und Ordnungen, fehlende Möglichkeiten bei der Benutzung oder als Normierungen (Schinzel 1998). Und auch wenn die Adäquatheit bei der Modellierung mit bedacht war, hängt sie fortan vom neuen Verstehenskontext und den BenutzerInnen des Softwareproduktes ab und unterliegt somit hermeneutischen

Verstehensprozessen. Die Lücke zwischen der Modellierung, die die informellen Vorverständnisse der EntwicklerInnen in Software ein- und festgebacken hat, und dem Konstruktionsprozess von Sinnstrukturen der AnwenderInnen ist dann u.U. nicht zu schließen und/oder führt zu neuen Sinngebungen. Solche (Miss-)Verständnisse zeigen die Beiträge von Sigrid Schmitz und von Carmen Masanek in diesem Heft.

#### Literatur:

Brendel, Oliver (1992): Funktioneller Hirnatlas - Axiomatisierung und Repräsentation der Topographie, Bericht 7/92 des IIG Freiburg.

Brendel, Oliver, Widdig, Rolf; Piron, Frank; Schinzel, Britta (1993): Der Hybride Konsiliar für die Neurowissenschaften (HYBRIKON). Aus der Sicht der Informatik. 2. Deutschen Tagung Expertensysteme, 17. - 19. Februar 1993 in Hamburg. Workshop 6: Wissensbasen in der Medizin (MEDWIS) S. 19- 21.

Chang, C.C. (1973); Keisler, H.J.: Model Theory, North-Holland, Amsterdam

Chen, P. S.; Knöll, Heinz-Dieter (1991): Der Entity-Relationship-Ansatz zum logischen Systementwurf: Datenbank- und Programmentwurf. BI-Wiss.- Verlag; Mannheim, Wien, Zürich

Ebbinghaus, H.-D.; Flum, J. ; Thomas, W. (1986): Einführung in die mathematische Logik. Wissenschaftliche Buchgesellschaft, Darmstadt

Funken, Chr.; Schinzel, B. (1996): Die Aufgabenermittlung als Fehlerquelle. für W.H. Otto (Hrsg.): Faszination Computer – Die Informationstechnologie im Expertenspiegel. Festschrift für Konrad Zuse (unveröffentlicht), 18 S.

Grube, Gernot (1995): Modellierung in der Informatik. In: Fischer, M.; Grube, G.; Reisin, F.-M. (Hg.): Abbild oder Konstruktion – Modellierungsperspektiven in der Informatik KIT Report 125. TU Berlin

Krämer, Sybille (1988): Symbolische Maschinen: die Idee der Formalisierung in geschichtlichem Abriß. Wiss. Buchges. Darmstadt

Nake, Frieder (1993): Von der Interaktion. Über den instrumentalen und den medialen Charakter des Computers. In: ders. (Hg.): Die erträgliche Leichtigkeit der Zeichen. Ästhetik Semiotik Informatik. Agis Verlag Baden-Baden, S. 165 – 191. Schaub, Martin (1992): Künstliche und natürliche Sprache; OLMS Philosophische Texte und Studien; Hildesheim, Zürich, New York

Schinzel, B. (1998): Women's ways of Tackling the Specification Problem, AISB Quarterly (Journal of the Society of Artificial Intelligence and Simulation of Behaviour), No 100, Summer 1998, pp 18-23.

Stegmüller, Wolfgang (1986): Rationale Rekonstruktion von Wissenschaft und ihrem Wandel. Reclam

Schefe, Peter (1999): Softwaretechnik und Erkenntnistheorie; Informatik Spektrum 22: 122-135.

Tarski, Alfred (1983): Der Wahrheitsbegriff in den formalisierten Sprachen; in Berka, K., Kreiser  
L: Logik-Texte Berlin, S 443.

Winograd, T., Flores, F. (1987): Understanding Computers and Cognition. A New Foundation for  
Design; Addison Wesley, Reading, Mass.