

Aus dem
Max-Delbrück-Zentrum für molekulare Medizin (MDC)
Berlin-Buch

DISSERTATION

**Ein Repräsentationsformat zur
standardisierten Beschreibung und
wissensbasierten Modellierung genomischer
Expressionsdaten**

Zur Erlangung des akademischen Grades
Doctor rerum medicarum (Dr. rer. medic.)

vorgelegt der Medizinischen Fakultät der Charité -
Universitätsmedizin Berlin

von
Daniel Schober
aus Ötjendorf, Stormarn

Gutachter: 1. Prof. em. Dr. med. J. G. Reich
2. Prof. Dr. St. Schuster
3. Prof. Dr.-Ing. Dr. med. habil. S. Pöpl

Datum der Promotion: 29.05.2006



In Völuspá 2,7 nennt die Seherin den Weltenbaum miötvid maeran. Was das zu bedeuten habe, ist eine viel erörterte Streitfrage. Detter-Heinzel II, 9 nennen die Bedeutung des Ganzen unklar, und ebenso noch Neckel in seinem Glossar S. 118. Weil man mit dem Maßbaume nichts anzufangen wußte, machte man aus ihm einen Honigbaum miövidr und erinnerte dabei an die Beziehungen des Baumes zum Honigfalle. Dagegen erklärte Gering schon in seinem Glossar S. 126 (ähnlich nun auch der Kommentar, Gering-Sijmons I, 5) den Ausdruck als den "nach wohlbedachtem Plane erschaffenen Baum", "das Symbol des planmäßig eingerichteten Weltganzen". Damit hat er den Weg zur Erklärung gewiesen, aber doch nicht beachtet, daß der Weltenbaum eine bestimmte Gliederung aufweist, die der von Zeit- und Weltordnung entspricht. Denn der Weltenbaum ist der Zahlenbaum, und seine Beziehungen zur Zeitrechnung hat K. v. Spieß in seiner Arbeit "Monatsbaum, Jahresbaum, Weltenbaum" (Wr. Zs. f. Volksk. 1923, H. 2-5) an reichem Stoffe gezeigt. Miövidr ist demnach der Baum, der dieselbe Gliederung wie die Welt selbst aufweist, an dem also die rechten Maße sichtbar werden, und daher stammt sein Name.

Edmund Mudrak

- Im Herzschlag der Dinge -

Inhaltsverzeichnis

1	Einleitung	12
1.1	Struktur der Arbeit	12
1.2	Sprachliche Konventionen	12
1.3	Problembeschreibung: Genannotation und Expressionsanalyse	13
1.3.1	Heterogenität der Genannotationen.....	14
1.3.2	Mangel an Kontext-Daten	15
1.3.3	Primitive Repräsentationsformalismen	15
1.3.4	Laborspezifische Annotationen.....	16
1.4	Ziel der Arbeit	16
2	Grundlagen	19
2.1	Ontologie als Schema zur Wissensrepräsentation.....	19
2.1.1	Ontologiedefinitionen und Anwendungsgebiete.....	19
2.1.2	Ontologie als Kommunikationsstandard und Modell.....	21
2.1.3	Bestandteile ontologischer Formalisierung (KR-Ideome)	21
2.1.3.1	Konzepte.....	21
2.1.3.2	Instanzen.....	22
2.1.3.3	Slots, <i>facets</i> und <i>constraints</i>	22
2.1.3.4	Frames und Forms	24
2.1.4	Objektorientierung und Vererbung	24
2.1.5	Datentypen und ihre Repräsentation durch Slot- <i>widgets</i>	25
2.1.6	Semantik der Ontologie (OKBC-CLIPS).....	26
2.1.7	Beschreibung des Wissensbank-Editors Protégé-2000	28
2.1.8	Erstellung von Ontologien und <i>ontology engineering</i> -Standards	30
2.2	Ontologiebasiertes Wissensmanagement in den Biowissenschaften	31
2.3	Anwendungsdomäne: <i>Toll-like Receptors</i> und dendritische Zellen.....	31
2.4	Expressionsanalyse mit dem Affymetrix® Human Genome U95Av2 GeneChip®	33

3	Ergebnisse	34
3.1	Erstellung der Gandr-Ontologie	34
3.1.1	Anforderungsspezifikation und Kompetenzfragen	34
3.1.2	Wissensakquisition	35
3.1.3	Manuelle Erstellung einer prototypischen Ontologie	36
3.1.4	Erweiterung der Ontologie um domänenspezifisches Vokabular	36
3.1.4.1	Erstellung des Ausgangs-Textkorpus zur KR-Ideom-Extraktion	36
3.1.4.2	POS-tagging und Extraktion potentieller KR-Ideome	37
3.1.5	Taxonomische Integration der Konzepte unter die prototypische Ontologie	39
3.1.5.1	Benennungs-Konventionen für Konzepte und Slots	40
3.1.5.2	Nutzung einer Text-Konkordanz zur Konzeptpositionierung	41
3.1.5.3	Gliederung in <i>top-level</i> -Module	42
3.1.5.4	Partonomien und Prozess-Taxonomien	43
3.1.5.5	Formalisierung als Konzept, Slot oder Instanz	43
3.1.5.6	Nutzung von Mehrfachvererbung	44
3.1.6	Erweiterung der Semantik	44
3.1.6.1	Hinzufügen von Slots und Relationen	44
3.1.7	Integration vorhandener Ontologien	45
3.1.8	Entkopplung einer molekularbiologischen <i>upper-level-Ontologie</i>	46
3.2	Beschreibung der Gandr-Ontologie	47
3.2.1	Grundlegende <i>top-level</i> -Module	47
3.2.2	Taxonomische Organisation und Kontext-Einbettung über relationale Slots	48
3.2.3	Abbildung ontologischer Ideome auf Protégé-GUI und CLIPS-Format	50
3.2.4	Größe und Metrik der Ontologie	54
3.2.5	Zugänglichkeit und Veröffentlichungsstatus der Ontologie	54
3.3	Erstellung und Beschreibung der Gandr-Wissensbank	55
3.3.1	Beschreibung der integrierten Daten (Instanzen)	55
3.3.1.1	Verknüpfungen zu externen Daten über Hyperlinks	56

3.3.2	Datenimport mit dem Datagenie-Plugin	57
3.3.3	Genannotation durch Verschieben (<i>drag and drop</i>) von probe set IDs	58
3.3.4	Größe der Wissensbank, Systemanforderungen und Performanz	59
3.4	Anwendungen der Gandr-Wissensbank	60
3.4.1	Formale Annotation von probe set IDs unter Nutzung von Mehrfachvererbung ..	61
3.4.2	Integration des Speicher- und Modellierungs-Formates	62
3.4.3	Assoziative und kontextsensitive Navigation	62
3.4.4	Ontologische Informationsextraktion mit dem Queries & Export-Tab	63
3.4.4.1	Export von Anfrageergebnissen	66
3.4.5	Visualisierungen der Wissensbank	67
3.4.5.1	DAG-basierte Visualisierung mit GraphViz und dem OntoViz-Plugin	67
3.4.5.2	Spring-Layout-Visualisierung mit Touch Graph und dem TGViz-Plugin	68
3.4.6	Wissensakquisition und Konsistenzprüfung über <i>constraints</i>	69
3.4.7	Wissensaustausch und Ontologieexport	70
3.4.8	Programmgesteuerte Manipulationen der Wissensbank (JessTab)	71
3.4.9	PROMPT <i>ontology-versioning</i> und <i>-merging</i>	72
3.5	Anpassungen der Benutzeroberfläche (GUI)	73
3.5.1	Darstellungsoptimierung über <i>Slot-widgets</i> und <i>browser keys</i> im Forms-Tab	73
3.6	Modifizieren und Erweitern der Ontologie	74
3.6.1	Einführung von Slot-Hierarchien	74
3.6.2	Erweiterung der KR-Semantik (Slot-Hierarchien, Metakonzepte und -slots)	75
3.7	Die Wissensbank als Internet-Anwendung: WebGandr	76
3.8	Aufbau der Gandr-Internetseite, Schulungsfilm und Dokumentation	77
4	Diskussion	78
4.1	Lexikalische Eigenschaften molekularbiologischer Terminologien	78
4.2	Neurokognitive Grundlagen der Wissensakquisition	78

4.2.1	Strukturtreue und Kontext erhöhen Interpretationsgeschwindigkeit	79
4.2.2	Festigung und Erweiterung des Wissensmodells.....	80
4.3	Anlehnung an <i>ontology engineering</i> -Methodologien (ONIONS)	81
4.4	Probleme bei der Erstellung von Ontologien.....	82
4.4.1	Kommunikation mit den Experten und Wissensakquisition.....	82
4.4.2	Taxonomisierungs-Probleme	82
4.4.2.1	Konzept oder Instanz, Subkonzept oder Slot	83
4.4.2.2	Konzept oder Instanz als Slotwert	84
4.4.2.3	Repräsentation von Transformationen und graduellen Zustandsübergängen ...	84
4.4.2.4	Kontextwandel, Synonyme, Redundanz und taxonomische Inkonsistenzen....	85
4.4.2.5	Gleiche Detailliertheit bei Geschwisterkonzepten	86
4.4.3	Fehler in zu integrierenden Ontologien	86
4.5	Beurteilung der Gandr-Ontologie	87
4.5.1	Ontologie-Typ.....	87
4.5.2	Beurteilung der Kodierungssprache und Expressivität.....	89
4.5.2.1	RDB vs. CLIPS vs. OWL	90
4.6	Beurteilung der Gandr-Wissensbank	90
4.6.1	Formale Klassifizierung der Anwendung nach dem System Uscholds	91
4.6.2	Beurteilung der Anwendung anhand der Anforderungsspezifikation.....	94
4.6.3	Beurteilung der IR-Kapazität	94
4.6.4	Dokumentation und Schulung.....	95
4.6.5	Akzeptanz beim Nutzer.....	96
4.7	Beurteilung der Visualisierungsansätze	97
4.7.1	Datengetriebene und konfigurierbare GUI	97
4.7.2	Visualisierungen der Wissensbank-Inhalte.....	98
4.7.2.1	Vorteile der Frames gegenüber tabellarischen Darstellungen	98
4.7.2.2	Vergleich mit Kohns <i>molecular interaction maps</i>	99

4.8	Vergleich mit anderen Annotations-Systemen und Ontologien.....	100
4.8.1	Affymetrix®-eigene Annotationsmöglichkeiten.....	101
4.8.2	Gene Ontology, GONG und GO-Mining-Tool	101
4.8.3	UMLS.....	103
4.8.4	MGED, MIAME und MAGE.....	104
4.8.5	TAMBIS.....	105
4.9	Ausblick	106
4.9.1	Ontologie-induzierte Konzept-Ikonographien	106
4.9.2	Internetseiten-Annotation im <i>semantic web</i> -Ansatz	107
4.9.3	Diskriminanzanalysen, Gruppierungsverfahren und maschinelles Lernen.....	108
4.10	Zusammenfassung.....	110
	Literaturverzeichnis.....	112
	Abkürzungen.....	122
	Abbildungsverzeichnis.....	124
	Danksagungen.....	125
	Curriculum Vitae.....	126
	Publikationsliste.....	128
	Erklärung.....	129
	Anhang.....	130

1 Einleitung

1.1 Struktur der Arbeit

Die **Einleitung** enthält eine Einführung in das Forschungsproblem der Repräsentation, Speicherung und Anwendung biologisch funktionaler Genbeschreibungen im Rahmen des Wissensmanagements. Aus der Problembeschreibung wird die Zielstellung der Arbeit hergeleitet. Die **Grundlagen** enthalten eine Themenhinführung, in der grundlegende Philosophien der Objektorientierung erklärt werden, die im Rahmen dieser Arbeit Lösungen der genannten Probleme ermöglichen. Er führt in das Thema Ontologie als Wissensrepräsentations-Schema ein und stellt die gegenwärtigen Methoden der Erstellung, Konzeptionierung und Anwendung von Ontologien dar. Weiter wird die biologische Anwendungsdomäne der Wissensrepräsentation vorgestellt. Die **Ergebnisse** beschreiben die hier angewandte Methodik der Erstellung der Ontologie, des Gen-Annotationssystems und der darauf aufbauenden Wissensbank. Diverse Anwendungen des Systems (formale Genannotation, kontextsensitives Navigieren, graphisches Visualisieren und ontologische Abfragen der Gendaten) werden an Beispielen erläutert. In der **Diskussion** wird das Gesamtsystem evaluiert. Die Ergebnisse werden auf abstrakterer Ebene beschrieben und im Vergleich zu etablierten Ansätzen anderer Gruppen diskutiert. Dabei wird der Mehrwert des hier vorgestellten Systems gegenüber den bisherigen Systemen herausgestellt und im **Ausblick** weitere Nutzungsmöglichkeiten und potentielle Erweiterungen des Systems vorgestellt. Am Schluß folgt eine kurze Zusammenfassung der Ergebnisse. Der **Anhang** enthält Zusatzinformationen, weitere Anwendungen und eine Bedienungsanleitung für das System.

1.2 Sprachliche Konventionen

Dem interdisziplinären Charakter der Arbeit entsprechend ließ es sich nicht vermeiden, Fachtermini aus verschiedenen Bereichen, insbesondere den Gebieten Linguistik, Symbolische KI, Wissensmanagement und Molekularbiologie, zu verwenden. Oft sind in den unterschiedlichen Fachdomänen verschiedene Begriffe für dieselben Dinge in Gebrauch, was dazu führt, daß Fachausdrücke den Domänenspezialisten nicht adäquat verwendet erscheinen, obwohl sie im Sinne der Fachterminologie einer anderen Domäne tatsächlich korrekt angewendet wurden. Solche Synonyme sind insbesondere Class-Concept, Property-Attribut und Objekt-Instanz. Es wurde versucht, wenn möglich, auf Anglizismen zu verzichten. Bei feststehenden Begriffen wurden diese dennoch aus dem Englischen übernommen und sind dann *kursiv* gesetzt. Für neuere noch nicht feststehende Begriffe wurden, wenn irgend möglich,

plausible deutsche Ausdrücke verwandt. Fachwörter, die als solche mit Metadaten für Suchmaschinen versehen sind, werden in **Fettdruck** dargestellt. Programmcode und KR-Ideome sind in *verbatim* gesetzt. Nutzer der Wissensbank können sowohl Menschen als auch Computer bzw. Programme sein; beide werden im folgenden kollektiv als **Agenten** bezeichnet. Der Begriff "Ontologie" wird hier im Sinne eines semantisch definierten Formalismus zur Wissensrepräsentation verstanden. Damit entfernen wir uns von einigen gängigen Ontologie-Definitionen, die den Standardisierungsaspekt von Ontologien besonders betonen. Die Begriffe Gen, Probe Set ID, Cellular Compound- und Probe Set Container-Instanz werden synonym verwendet, da auf den genutzten Genchips keine *splice*-Varianten untersucht wurden und Gene meist nur über ihre Genprodukte behandelt werden. Das im Rahmen dieser Dissertation erstellte Anwendungspaket wurde im Hinblick auf seine vielfältigen Nutzungsmöglichkeiten **GandrKB** genannt. Gandr ist ein Akronym für "*Gene annotation data representation*" und bedeutet im altnordischen soviel wie "Zauberstab". Das KB steht für *knowledge base*.

1.3 Problembeschreibung: Genannotation und Expressionsanalyse

Aufgrund der rasanten Entwicklungen neuer Techniken der Hochdurchsatz-Datenerfassung hält die Transformation der gewonnenen biologischen Daten in anwendbares Wissen (das sog. **knowledge-mining**) nicht annähernd mit dem Tempo der Datenerfassung Schritt. Wissenschaftler sehen sich einem anwachsenden Berg von Daten gegenüber, der kaum noch manuell ausgewertet werden kann [1, 2]. Wir befinden uns in einem Informationsdilemma, das J. Naisbitt [3] mit den Worten "Wir ertrinken in Information und hungern nach Wissen" beschreibt. Für eine schnelle und effiziente Datenverarbeitung sind computergestützte Ansätze gefordert, die den Wissenschaftler bei Abbau und Weiterverarbeitung dieses Datenberges unterstützen.

Nach der nahezu vollständigen Entschlüsselung des Humangenoms wendet man sich zunehmend dem Transkriptom und Proteom zu und untersucht, wie stark die Gene in verschiedenen Geweben oder Stoffwechselkonditionen abgelesen, in mRNA transkribiert und zu Proteinen translatiert werden. Auf dem Gebiet der Expressionsanalyse hat sich in den letzten Jahren das Verfahren der DNA-Microarrays und hierunter insbesondere der Affymetrix[®]-Ansatz als Standardverfahren etabliert [4, 5, 6]. Die Microarray-Technik basiert auf der Hybridisierung von Nukleinsäuren. Komplementäre Nukleinsäure-Einzelstränge lagern sich dabei spezifisch über Wasserstoffbrückenbindungen aneinander. Auf einer Silizium-Immobilisierungsmatrix, dem Microarray, liegen an definierten Positionen Nukleinsäuren zu untersuchender Gene, die sogenannten Proben-Nukleinsäuren. Diese hybridisieren dann mit unterschiedlich fluoreszenzmarkierten *target*-Nukleinsäuren aus verschiedenen zu untersuchenden Geweben

oder Gewebekonditionen. Die im Gewebe vorhandenen *target*-Nukleinsäuren werden, nach Abwaschen unspezifisch gebundener Nukleinsäuren, über einen Fluoreszenz-Scanner durch den Ort der Hybridisierung auf dem Chip identifiziert und über das Verhältnis der Fluoreszenzintensitäten, bei den für die Markierungen der zu vergleichenden *target-samples* charakteristischen Wellenlängen, quantifiziert.

Im Gegensatz zum hochtechnisierten Herstellungsverfahren dieser Microarrays muten die in den vorhandenen **Laborator Informations Management Systemen (LIMS)** standardmäßig mitgelieferten Ansätze zur Repräsentation, Abfrage und Verarbeitung der generierten Gendaten primitiv an [4]. Im folgenden werden einige der sich hierdurch ergebenden Mißstände erläutert.

1.3.1 Heterogenität der Genannotationen

Zur Verarbeitung großer Ausgangsdatenmengen, wie sie durch Microarray-Experimente generiert werden, ist zunächst eine Reduktion auf wenige im Hinblick auf eine laborspezifische Fragestellung relevante Datenobjekte erforderlich. Dieses **Information-Retrieval (IR)** wird über Anfragen an das LIMS nach bestimmten Suchkriterien durchgeführt. Eine Anfrageschnittstelle liefert dann als Ergebnis eine überschaubare im Idealfalle besonders interessante Teilmenge der Ausgangsdaten. Die bei der Auswertung von Microarray-Experimenten am häufigsten genutzten Suchkriterien sind, neben den Transkriptionswerten selbst, die Genannotationen, welche die auf dem Chip repräsentierten Gene funktionell beschreiben. Bei der Verwendung heterogener Annotationen als **IR-Suchattribute**, stößt man jedoch auf Schwierigkeiten, da sie nicht einheitlich verwendet bzw. verstanden werden. So können orthographische Varianten, Synonyme als Ausdruck verschiedener Fachterminologien oder Begriffe mit breitem Interpretationsspielraum die Quote an richtig positiven Anfragetreffern minimieren und dadurch die IR-Funktionalität des LIMS stark einschränken. Affymetrix[®] liefert in seinem *Data Mining Tool*[®] (*DMT*) genannten LIMS zu den funktionell nicht aussagekräftigen Genidentifiern (probe set IDs und *accession*-Nummern) an funktionellen Annotationen lediglich die von der Genbank übernommenen *feature-table*-Genannotationen. Diese Annotationen sind sehr **heterogen** aufgebaut, da sie von verschiedenen Wissenschaftlern aus unterschiedlichsten Fachrichtungen durchgeführt wurden. Dabei reflektieren diverse antonyme Fachtermini domänenspezifische Perspektiven. Man kann so unterschiedliche Beschreibungen wie "Cystatin B", "schwannoma-associated protein (SAM9) complete cds" und "mSUG1" finden. Weiter werden oft mehrere unterschiedliche Synonyme und Akronyme benutzt. Zum Beispiel können G-Protein gekoppelte Rezeptoren mit Beschreibungen wie "*G-Protein coupled receptor*", "*GPCR-protein*", "*seven transmembrane domain protein*" oder "*7 TM protein*" annotiert sein, die alle ungefähr dasselbe

bedeuten. Sucht man nun in einer Tabellenspalte "G-Protein coupled Receptor", so erhält man nur die Zeilen, in denen diese Zeichenkette (*string*) explizit auftaucht. Man erhielte nur dann alle richtig positiven Treffer, wenn man nach all den verschiedenen Begriffen gleichzeitig suchen würde, was in der Praxis zu zeitaufwendig wäre. Andernfalls werden nur die explizit in der Anfrage und Annotation auftauchenden Begriffe gefunden, inhaltlich relevante, aber über Synonyme beschriebene Einträge also vernachlässigt. Die heterogene nicht formale Annotation macht es dem Auswerter also unmöglich, seine Daten vollständig und inhaltlich über diese Annotation abzufragen und erschwert weiter den Einsatz von automatischen Auswertungsstrategien (z.B. *clustering*-Verfahren).

1.3.2 Mangel an Kontext-Daten

Generell ist die **Annotierung** der auf den Microarrays repräsentierten Gene nicht besonders ausführlich und wenig umfangreich. Affymetrix[®] liefert zu jedem Gen im Schnitt nur 117 Zeichen an beschreibendem Text. Zusätzliche Informationen sind zwar auf externen Internetseiten abrufbar, können dadurch jedoch nicht in Anfragen an den eigenen Datenbestand einbezogen werden. Gerade für Mediziner, d.h. an systemischen Zusammenhängen interessierte Wissenschaftler, ist die Bereitstellung zusätzlicher Kontextinformationen für eine korrekte Interpretation von Bedeutung.

1.3.3 Primitive Repräsentationsformalismen

Zur Repräsentation, Visualisierung und Auswertung ihrer experimentellen Daten nutzen Wissenschaftler meist einfache Tabellenkalkulationsprogramme. Die Ergebnisse eines Affymetrix[®] Microarrays z.B. werden in Form einer riesigen Microsoft Excel[®]-Tabelle mit zehntausenden Genen, ihren Expressionswerten und Annotationen repräsentiert. Der Nachteil hierbei ist die für das menschliche Wahrnehmungsverhalten nachteilige tabellarische Repräsentation, welche die besonders ausgeprägten Parallel- und Muster-Verarbeitungsfähigkeiten des Gehirns kaum nutzt. Weiter können die Tabelleneinträge nicht verknüpft bzw. Verknüpfungen nicht visualisiert oder abgefragt werden. Erst die Einordnung generierter Daten in ein Modell des holistischen Ganzen bedeutet Wissenschaft im engeren Wortsinn, also die Schaffung von Wissen. Dabei ist zunächst zweitrangig, ob es sich um ein internes sprachliches, also gedankliches Modell, das sich der Wissenschaftler von seiner Forschungsdomäne bildet, oder ein gegenüber dem geistigen Modell extern repräsentiertes formales Computermodell handelt. Wichtig ist, daß Daten nicht wie in Tabellen zusammenhanglos als Listen präsentiert werden, sondern daß diese eingebettet in ihrem semantischen Kontext und somit als unmittelbar zugängliches Wissen repräsentiert werden.

Semantisch definierte komplexere Anfragen der Form "Wie werden Gene exprimiert, die Kinasen sind und im Toll-like Receptor Pathway eine Rolle spielen ?" oder "Welche im Zellkern exprimierten Gene werden stark exprimiert ?" sind in tabellenorientierten Programmen aufgrund der Primitivität der zugrundeliegende Repräsentationssemantik schwer zu stellen.

1.3.4 Laborspezifische Annotationen

Wissenschaftler sollten die Möglichkeit haben, Gene mit eigenen, in der Forschergruppe etablierten Annotationen zu versehen. Diese anwendungsspezifische Annotation sollte auf einer standardisierten Terminologie aufbauen. Die in den gegenwärtig eingesetzten LIMS für die Genannotation zur Verfügung gestellten Annotationswerkzeuge sind sehr einfach und erschweren eine schnelle und vor allem konsistente Genannotation mit einem für das Arbeitsgebiet spezifischen Vokabular. Meist wird lediglich in einer Excel[®]-Tabelle eine zusätzliche Spalte angelegt und mit Beschreibungstext versehen. Die Mehrheit der LIMS erlaubt hier lediglich primitive Datentypen wie *string* und *number* und keine selbstdefinierten, geschweige denn komplexe zusammengesetzte Datentypen.

1.4 Ziel der Arbeit

Den erwähnten Mißständen wird dadurch begegnet, daß Microarray-Daten und Genannotationen über eine formale Semantik modernen Wissensmanagement Werkzeugen zugänglich gemacht werden. Ziel der Arbeit ist es, zu zeigen, daß eine formale objektorientierte Datenrepräsentation, eine sog. Ontologie, durch Laborbiologen zur Genannotation verwandt werden kann und hier gegenüber den bisher genutzten tabellarischen Annotationen einige Vorteile bietet. Es wird eine an den Bedürfnissen der Domänenspezialisten, d.h. Medizinern und Molekularbiologen orientierte Methode der Annotation mit eindeutigen und dennoch einleuchtenden Beschreibungselementen vorgestellt. Am Beispiel von Affymetrix[®] Microarray-Daten wird gezeigt, daß ontologisch strukturierte Repräsentationen genutzt werden können, um laborspezifische, also eigene, an einen Forschungsgegenstand angepaßte Genannotationen schnell und dabei konsistent zu erstellen. Es wird gezeigt, wie die annotierten Daten als "Wissensmodelle" graphisch visualisiert sowie intelligent und vollständig abgefragt werden können. "Ein Gen annotieren" bedeutet hier, einem Microarray Genidentifier einen ontologisch definierten Begriff der bereitgestellten Terminologie zuzuweisen.

Weiter werden neue Techniken und Methodiken zur Erstellung derartiger Ontologien untersucht, wobei geprüft wird, ob und wie Techniken des *Information Retrieval*, der Informations-Extraktion und der natürlichen Sprachverarbeitung bei der Erstellung domänenspezifischer

ontologischer Annotationsvokabulare genutzt werden können. Am konkreten Beispiel des Toll-like Receptor-Signaltransduktion (TLR-ST) wird gezeigt, wie dem Laborwissenschaftlern u.a. folgende Vorteile des ontologiebasierten Wissensmanagements nutzbar gemacht werden können.

- **Parallele Genannotation beschleunigt Annotationsprozess:** Über den Einsatz moderner Wissensmanagement- und Modellierungs-Werkzeuge soll eine schnelle Annotierung sehr vieler Microarray-Gene gleichzeitig über ihnen gemeinsame Eigenschaften ermöglicht werden. Dabei sollen Eigenschaften von allgemeineren Annotations-Klassen auf speziellere "vererbt" werden (siehe Abschnitt 2.1.4).
- **Akquisition formalen Wissens:** Unformal, d.h. als freier Text gespeicherte Annotationsdaten und über externe Internetseiten zugängliche Annotationen sollen mit Hilfe einer Ontologie und hieraus abgeleiteten formularartigen Dateneingabemasken formalisiert werden. Die Formalisierung des Wissens soll über die Ontologie überprüfbar sein und so die Konsistenz der Genannotation erhöhen.
- **Gemeinsames Vokabular ermöglicht gemeinsame Interpretation:** Durch Nutzung eines definierten standardisierten Repräsentationsformates soll eine gemeinsame konsistente Interpretation der Annotationen und Daten unter Menschen und Arbeitsgruppen einerseits, andererseits aber auch zwischen Menschen und Computern sowie zwischen verschiedenen Computerprogrammen ermöglicht werden. Die Formalisierung der Annotation über eine definierte Semantik soll Mißverständnisse verhindern, die aus unterschiedlichen Auffassungen der Annotations-Begriffe resultieren bzw. deren Interpretationsspielraum einschränken.
- **Wiederbenutzung der Annotation:** Ontologisch formalisierte Wissensrepräsentationen können in verschiedenste Repräsentationsstandards transformiert werden, wodurch dem Nutzer eine breite Palette weiterer Bearbeitungswerkzeuge zur Verfügung gestellt wird. Austausch, Wiederverwendung und Verbreitung der Annotation (Ontologie) und Wissensbank werden erleichtert, da die Ontologie getrennt von den annotierten Daten gespeichert und verschickt werden kann (*ontology sharing* und *reuse*). Weiter kann sie mit anderen Ontologien, d.h. in ähnlichem Format vorliegenden Annotationen verglichen oder fusioniert werden (*ontological merging*).
- **Repräsentation des funktionalen Kontext als Modell:** Bisher wurden Datenspeicherung und Modellbildung in strikt getrennten Ansätzen verfolgt. Formale Wissensrepräsentationen jedoch ermöglichen es, die Vorteile beider Ansätze in einem holistischen Ansatz zu vereinen, was Zeit und Ressourcen einsparen kann. In letzter Konsequenz wird hier ein Wechsel der Methodik postuliert. Neuere Entwicklungen aus dem Bereich des Wissensmanagements erlauben eine zunehmende Abkehr von reinen Datenspeichern, hin zu semantischen, d.h. integrativeren, Modellspeichern. Mit der Gandr-Ontologie soll ein Speichermodell für einen derartigen Modellspeicher vorgelegt und gezeigt werden, wie Anwender, die selbst keine "Wissensingenieure" sind, von derartigen Modellspeichern profitieren können. Der an systemischen Zusammenhängen interessierte Mediziner benötigt unmittelbaren Zugang zu in Wechselbeziehung zu einander stehenden Daten im jeweiligen Interessenfokus. Solche aufeinander Bezug nehmenden Annotationen sollten den systemischen Kontext eines Gens in einem Stoffwechsel- oder Signaltransduktionsweg repräsentieren können. Hierüber soll Agenten ein schnelles und unmittelbares Verifizieren daten- bzw. kontextinduzierter

Hypothesen ermöglicht werden. Neurokognitiven und wahrnehmungspsychologischen Erkenntnissen Rechnung tragend, wird hier eine Repräsentation von Wissen als assoziatives semantisches Netzwerk der herkömmlichen tabellarischen Repräsentation vorgezogen, da semantische Netze aufgrund ihrer strukturtreuen Abbildung auf ihr neurokognitives Korrelat ein intuitiveres Arbeiten ermöglichen (siehe 4.2.1). In einen Kontext eingebundenes Wissen wird ferner vom Menschen schneller erfaßt und gelernt als tabellarisch präsentierte Daten.

- **Visualisierung von Interaktionsnetzen und Signalkaskaden:** Der Repräsentationsformalismus soll eine datengetriebene, d.h. automatische, graphische Visualisierung des Annotationsmodells als interaktives Netzwerk erlauben. So sollen annotierte Gene automatisch als Interaktionsnetze bzw. Signaltransduktionskaskaden visualisiert und analysiert werden können. Über derartige Graphiken soll ein schneller, assoziativer und intuitiver Zugang des Annotationskontexts der auf dem Microarray repräsentierten Gene, ermöglicht werden.
- **Inhaltsbasierte Abfragen auch nach implizitem Wissen:** Über die ontologisch strukturierte Genannotation und eine entsprechende Anfrageschnittstelle sollen semantisch komplexe, inhaltsbasierte Anfragen an den Datenbestand in einer intuitiven Weise gestellt werden können. Die Nutzung einer entsprechenden Schnittstelle soll qualitativ bessere Suchergebnisse liefern. Die verwendete Repräsentation soll Agenten ein Erschließen der Bedeutung ermöglichen und neben dem expliziten auch in den Daten implizit vorhandenes Wissen zur Verfügung stellen. Bei der Generierung der Anfragen soll die ontologische Struktur inhaltliche, d.h. sachbezogene Hilfestellung geben. Diese Anfrage-Möglichkeiten sollen eine komfortablere und vollständigere Versuchsauswertung ermöglichen.

2 Grundlagen

2.1 Ontologie als Schema zur Wissensrepräsentation

Die gebräuchlichste Form der Wissensrepräsentation ist einfacher Text in natürlicher Sprache. Die semantische Struktur ist durch die Grammatik beschrieben und besitzt für Menschen relativ starke Ausdruckskraft. Sie stellt zwar das am weitesten verbreitete Repräsentationsformat für Wissen dar, kann jedoch aufgrund der syntaktischen und semantischen Komplexität in voller Ausdrucksstärke, also umfassend und eindeutig, nur durch Menschen interpretiert werden. Erst durch eine weitergehende Formalisierung in einer expliziten Wissensrepräsentationssprache kann Wissen auch von Computern interpretiert und automatisch weiterverarbeitet werden. Eine derartige Wissensrepräsentation enthält syntaktisch definierte Beschreibungskomponenten, die der Beschreibung von Daten in einer formalen Struktur dienen. In dieser Struktur steckt das Wissen um Beziehungen bzw. **Relationen** zwischen den Daten. Über ein derartiges in-Beziehung-setzen von Daten zueinander läßt sich die Bedeutung (**Semantik**) der Daten beschreiben. So ein inhaltlich-semantisches Repräsentationskonstrukt wird auf Implementierungsebene auch **Datenstruktur** oder Datenmodell genannt. Eine Wissensrepräsentation ist quasi ein Metamodell zur systemischen Formulierung von Wissen. Eine Wissensrepräsentation (*knowledge representation, KR*) besteht aus mehreren Bestandteilen (sog. **KR-Ideomen**). Diese Ideome werden in einer festgelegten Syntax ausgedrückt und stellen KR-Sprachen mit denen Wissensrepräsentationen unterschiedlicher Ausdrucksstärke (Semantik) beschrieben werden können. Besonders Ausdrucksstarke, aber dennoch formale und dadurch zur Weiterverarbeitung durch den Computer geeignete Wissensrepräsentationsformate werden über **Ontologien** bereitgestellt. KR- bzw. Ontologiesprachen stellen gewissermaßen semantische Metasprachen zur Datenannotation zu Verfügung.

2.1.1 Ontologiedefinitionen und Anwendungsgebiete

Der Begriff "**Ontologie**" stammt aus der Philosophie (gr. Onta "das Seiende", Logos "Lehre") und wurde im 17. Jahrhundert durch R. Goclenius geprägt. Schon Aristoteles untersucht in seiner Metaphysik "Seins-Kategorien" als die obersten Strukturen und Gesetzmäßigkeiten alles Aussprechbaren und Gegebenen und erstellt eine semantische Gliederung der Natur und Organisation der Welt (siehe Vorsatz-Blatt). Wurden Ontologien noch Mitte des letzten Jahrhunderts fast ausschließlich unter rein philosophischen Aspekten diskutiert [7, 8, 9, 10], so sind sie seit etwa zehn Jahren auch ein populäres interdisziplinäres Forschungsthema in den

Informationswissenschaften [11]. Die Perzeption des Ontologiebegriffs und deren Anwendung verlagert sich zunehmend auf die Gebiete Künstliche Intelligenz, Computerlinguistik und Wissensmanagement [12]. Aus anwendungsorientierter Perspektive beschreiben Ontologien objektorientierte Metadatenbank-Schemata, welche Syntax und Semantik in Daten einer Wissensdomäne formalisieren, dadurch vereinheitlichen und durch verschiedene Agenten interpretierbar machen. Eine Ontologie liefert eindeutige Beschreibungen prinzipieller Entitäten einer Wissensdomäne sowie deren Eigenschaften und Relationen zueinander. Sie ordnet systematisch relevante Konzepte bzw. Begriffsklassen eines Fachgebietes in einer hierarchischen Generalisierungs- / Spezialisierungs-Hierarchie, einer sogenannten Taxonomie. Eine Ontologie einschließlich der hierüber beschriebenen Daten (Instanzen, s.u.) nennt man eine **Wissensbank (WB)** oder *knowledge base, KB*. Der Begriff Ontologie ist nicht einheitlich definiert und wird je nach Umfang bzw. Komplexität der formalisierten Semantik unterschiedlich weit gefaßt [13, 14]. Eine gängige Definition des Ontologie-Begriffs ist die nach J. Sowa [15], die auch der vorliegenden Arbeit zugrunde liegt (siehe Abb. 1).

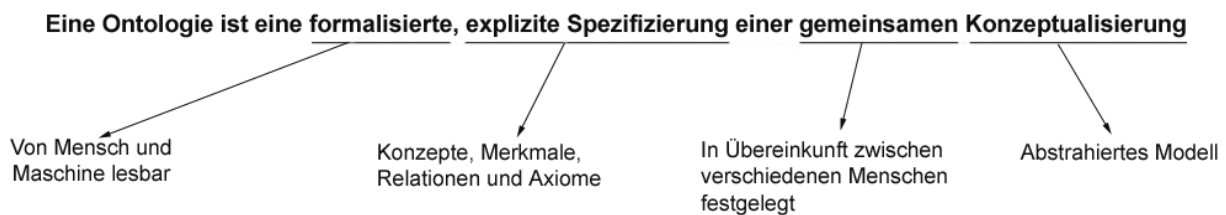


Abb. 1: Eine Ontologiedefinition nach Sowa. Die einzeln erläuterten Kriterien werden in unterschiedlichen Definitionen unterschiedlich weit gefaßt.

Auch N. Guarino [14] gibt je nach Interessenperspektive und semantischer Komplexität unterschiedliche Definitionen:

- Eine Ontologie ist eine formale Beschreibung der Gegenstände und Beziehungen, die für eine Gruppe von Personen als Begriffe verbindlich sind.
- Eine Spezifikation von Konzeptualisierung (Konzepte, Relationen, Objekte und Beschränkungen (*constraints*)), um Wissen für Menschen oder Computer darzustellen.
- Eine Spezifikation der Architekturkomponenten einer Domäne auf der Meta-Ebene in logischer Form.

Eine anwendungsbezogene Definition ist folgende [16]:

- Eine Ontologie ist eine hierarchisch strukturierte Menge an Begriffen zur Beschreibung einer Sachdomäne, die als Gerüst zur Erstellung einer Wissensbank genutzt wird.

2.1.2 Ontologie als Kommunikationsstandard und Modell

Agenten müssen, um ihre Wissensmodelle abzugleichen und zu aktualisieren, miteinander kommunizieren. Oft geschieht es, daß ein Konzept von unterschiedlichen Agenten mit unterschiedlichem Hintergrundwissen und aus verschiedenen Perspektiven betrachtet wird. Ein Konzept kann dann in verschiedenen Agenten unterschiedliche Bedeutungen erlangen. Interessen- und kontextabhängig können unterschiedliche Interpretationen und Assoziationen Mißverständnisse hervorrufen. Das **Homonym** "Kiefer" hat z.B. beim Mediziner eine andere Bedeutung als beim Botaniker. Andererseits kann dasselbe Ding verschiedene Namen haben (**Synonym**). Ontologien als Übereinkunft zwischen Agenten einer Domäne erleichtern über ihre gemeinsam definierte und vereinheitlichte Terminologie die Kommunikation, d.h. den Austausch und die Interpretation von eindeutigem modellhaftem Wissen bzw. von dessen Bedeutung.

Ontologien sollen ein systemisches und modellhaftes Verständnis von Daten ermöglichen. Verstehen heißt, daß man anhand der Repräsentation neue Schlüsse ziehen und semantische Äquivalenzen bei gegebenenfalls syntaktisch unterschiedlicher Repräsentation feststellen kann. Ein Modell ist ein abstraktes konzeptionelles Abbild von einem Teil eines umfassenderen realen Systems, wobei das System eine Gesamtheit von Elementen darstellt, die miteinander durch Beziehungen verbunden sind und gemeinsam einen bestimmten Zweck erfüllen [17]. Um ein Modell zu schaffen, benötigt man also neben Daten dieses Systems auch Kenntnis der Zusammenhänge dieser Daten bzw. eine allgemeine Vorstellung davon, wie sich die Realität zusammensetzt und wie man dies darstellt. Ontologien definieren diese Modellbestandteile über die KR-Ideome Konzepte und Relationen. Ähnlich wie die Formelsprachen mathematische Modelle von Systemen repräsentieren, ist die Ontologie eine Darstellungsmethode für sprachliche Modelle. Ontologie-annotierte Daten sind also *per se* Modelle, da sie Annahmen über reale Vorgänge beinhalten.

2.1.3 Bestandteile ontologischer Formalisierung (KR-Ideome)

Zunächst sollen die Bestandteile ontologischer Repräsentationen (**KR-Ideome**) genauer vorgestellt werden. In Abschnitt 2.1.6 wird dann das hier verwendete CLIPS KR-Format (siehe Abb. 4) erläutert.

2.1.3.1 Konzepte

Ein **Konzept** (lat. Conceptus "Begriff, Bewußtseinsinhalt, auch **Klasse**, **Kategorie** oder **Type**") ist ein abstrakter Repräsentant zur Beschreibung einer Gruppe von Datenobjekten (Instanzen, s.u.) mit gemeinsamen Eigenschaften (Slots, s.u.). Ein Konzept kann in einfachen Ontologien,

z.B. reinen Taxonomien, atomar, d.h. ohne interne Struktur, sein, oder es besteht wie in unserem Falle aus Konzeptname, Definition in natürlicher Sprache, Eigenschaften und gegebenenfalls aus **Axiomen**, expliziten generellen Regeln, die alle Instanzen eines Konzepts erfüllen müssen.

In Taxonomien werden Konzepte über eine *is-a* Relation in einer Hierarchie bzw. Baumstruktur entsprechend ihrem Abstraktheitsgrad von "allgemein" nach "speziell" geordnet. An der Wurzel einer Konzept-Taxonomie findet sich das allgemeinste Konzept (z.B. *thing*), darunter allgemeine abstrakte **upper-level-Konzepte**, und als "Blätter" der Baumstruktur, die speziellsten (*leaf*-)Konzepte. Das aktuell betrachtete Konzept heißt einfach Konzept. Die in der Hierarchie eine Ebene darüber liegenden, also direkt zugeordneten, generelleren Konzepte heißen direkte *parent*-, Eltern- oder **Superkonzepte**. Die zum betrachteten Konzept in der Hierarchie eine Ebene tiefer gelegenen, subsumierten, also spezielleren Konzepte heißen *child*-, Kind- oder **Subkonzepte**. Alle Superkonzepte zusammen, die über dem betrachteten Konzept stehen, heißen aufgrund der Vererbung ihrer Eigenschaften (siehe Abschnitt 2.1.4) in Analogie zu den Verwandtschaftsbeziehungen beim Menschen *ancestors* oder **Vorfahren**. Konzepte aller Ebenen unter dem betrachteten Konzept zusammen heißen *descendants* oder **Nachfahren**. Konzepte, die nebeneinander in einer Ebene stehen, heißen *sibling*- bzw. **Geschwister-Konzepte**.

2.1.3.2 Instanzen

Die konkreten Objekte oder Datenbank-Einträge, die über die Ontologie formal annotiert werden sollen, heißen **Instanzen** (auch Individuen oder Entitäten). Das Gen "β2-adrenergic receptor 34532" beispielsweise ist ein Argument bzw. eine Instanz des allgemeinen Konzepts "Adrenergic_Receptor". Die Instanz ist also ein bestimmtes ontologisch annotiertes Datenobjekt, im folgenden meist ein auf dem Microarray repräsentiertes und über seine probe set ID referenziertes Gen bzw. Genprodukt.

2.1.3.3 Slots, facets und constraints

Die allen Subkonzepten und Instanzen eines Konzepts gemeinsamen Eigenschaften nennt man Attribute, oder im Bezug auf ihre graphische Darstellungsweise **Slots**. Beschreiben diese Slots inhaltliche Verbindungen bzw. **Beziehungen** zwischen den Konzepten, also die semantischen Verhältnisse der Konzepte zueinander, nennt man sie **Relationen**. Sie repräsentieren die **Metadaten**, die aus einfachen Daten Wissen machen. Über die Relationen der Konzepte wird der **Kontext**, d.h. die Struktur eines Wissensmodells, in Form eines **semantischen Netzwerks** erstellt. Dabei stellen Konzepte und Instanzen die Knoten dieses Wissens-Netzes dar und die Slots die Kanten. Slots sind unabhängig von Konzepten definierte eigene Objekte und können

verschiedenen Konzepten gleichzeitig zugewiesen werden. Die Gesamtheit der Konzepte, denen ein Slot zugewiesen ist, nennt man seine **Domäne** (*domain*). Zum Slot gehört der mögliche **Werttyp** (*valuetype*), der neben den **primitiven Datentypen** wie *string*, *float*, *boolean*, *symbol* oder *integer*, im Falle der Relation auch Konzepte und Instanzen, sog. **komplexe Datentypen**, beinhalten kann. Diese werden in ihrer Gesamtheit für einen Slot **range** genannt. Da die Konzepte und Instanzen der *range*, gegebenenfalls über weitere Slots, selber definierbar sind, bietet die Ontologie hier die Möglichkeit, eigene bedarfsoptimierte Datentypen zu erstellen. Slots haben ihrerseits Eigenschaften (**facets**): Über *facets* werden erlaubte bzw. mögliche Slotwert-Typen eingegrenzt (**constraints**). Zu den Slot-*facets* gehören neben dem Werttyp und der *range* des Slot die **Kardinalität**, die angibt, ob dem Slot mehrere Werte (*multiple*) zugewiesen werden können oder müssen (*required*). Sie legt weiter fest, wie viele Werte der Slot mindestens und maximal haben kann. Für Slot-Werttypen wie *float* oder *integer* können Wertbereichsgrenzen (Min. und Max.) angegeben werden (siehe Abb. 9 und 4). Weiter können voreingestellte Slotwerte vorgegeben werden. Diese Werte werden bei Instantiierung dann automatisch erzeugt, können aber später "überschrieben" werden. Bei der Wissensakquisition, also dem Formulieren neuer Instanzen, werden diese *facets* überprüft. Für bestimmte Slots innerhalb ihrer **facet-constraints** festgelegte Beschränkungen werden bei der Wertefüllung der Instanz-Slots erzwungen. Ist z.B. für einen `has_Ligand`-Slot eines "Katecholamine_Receptor"-Konzepts ein Werttyp "Instanz des Konzepts "Small_Molecule"" definiert, so merkt und verhindert das System, falls man diesen Slot für einen konkreten Katecholamin Rezeptor mit einer "Neuropeptid"-Instanz als Wert füllen will und schlägt statt dessen eine Liste mit erlaubten "Small_Molecule"-Instanzen vor. Auch kann festgelegt werden, daß ein bestimmter Slotwert für alle Instanzen ausgefüllt werden muß. In der Frame-Darstellung der Instanz wird der Slot dann rot umrandet, wenn er falsch oder noch nicht gefüllt ist. Hierüber können also konsistente und in gewissem Rahmen "richtige" Eingaben bei der Instantiierung und Annotation gefördert werden. Derartige den Nutzer unterstützende Hilfsmittel sind besonders praktisch, wenn - wie im vorliegenden Falle - der Domänenexperte selbst die Ontologie und Wissensbank nutzen, verändern und erweitern soll. Für die semantisch weitergehende Spezifikation axiomatischer und logischer *constraints* (z.B. in *first order predicate logic*, FOL) kann die KIF verwandte Sprache *protégé axiom language (PAL)* genutzt werden (http://Protege.stanford.edu/plugins/paltabs/PAL_tabs.html). *PAL-constraints* wurden mit Rücksicht auf die Nutzergruppe vorerst nicht implementiert.

Die einem Konzept direkt zugewiesenen Eigenschaften nennt man *direct-slots*, die von Vorfahren-Konzepten automatisch übernommenen Slots nennt man *inherited-* oder *geerbte Slots*

(Vererbung, s.u.). Die Mehrheit der Slots sind **asymmetrisch**, d.h. der `slot (A, B)` impliziert nicht `slot (B, A)`. Seltener sind symmetrische Slots wie der "liegt_neben"-Slot, bei dem der `slot (A, B)` den `slot (B, A)` impliziert. Slots können selbst wie Konzepte taxonomisch hierarchisiert werden.

2.1.3.4 Frames und Forms

Die graphische Darstellungsweise der Konzepte, Slots und Instanzen über eine Art Formular mit Eintragungsmöglichkeiten bezeichnet man als **Frame** (engl. Rahmen). Frames stellen das verbreitetste objekt-orientierte Schema zur expliziten deklarativen Wissensrepräsentation dar. Ein Frame ist ein Fenster, das Eigenschaften von KR-Ideomen in einheitlicher Weise graphisch repräsentiert. Die Slots eines KR-Ideoms werden im Fenster des Frame als interaktive Eintragsfelder dargestellt und sind über sog. **Forms** frei konfigurierbar. Ein Frame kann Slots enthalten, die wiederum andere Frames, also Konzepte oder Instanzen, enthalten können.

2.1.4 Objektorientierung und Vererbung

In taxonomischen Konzepthierarchien werden Slots automatisch von allgemeineren Superkonzepten, für die sie definiert wurden, an speziellere Subkonzepte übertragen, d.h. **vererbt** (siehe Abb. 2). Das Vererbungsprinzip ist analog zum Vererbungs-Begriff beim OOP, wobei den Konzepten die Objekt-Klassen und den Slots die geerbten Attribute und Methoden entsprechen. Wenn für ein generelles Konzept "Enzym" der Slot `hat_EC_Nummer` definiert wurde, so gilt für alle Subkonzepte von "Enzym", z.B. für "Tyrosine_Kinase", daß es den `hat_EC_Nummer`-Slot und alle anderen auf der "Enzym"-Konzept-Ebene definierten Slots erbt. Slots können linear, d.h. über mehrere direkte Superkonzepte, auf ein Konzept vererbt werden (**single inheritance**) oder in Mehrfachvererbungs- (**multiple inheritance**) Hierarchien von mehreren Superkonzepten an ein Konzept vererbt werden.

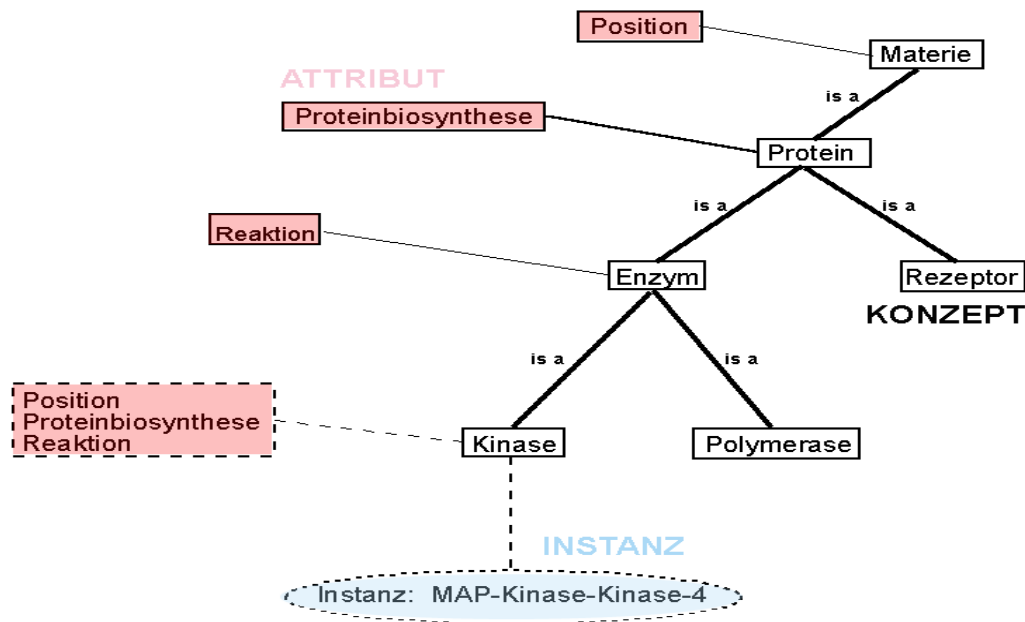


Abb. 2: Beispiel für die Vererbung von Konzept-Eigenschaften (Attributen bzw. Slots, rot) in einer Konzept-Taxonomie. Das leaf-Konzept "Kinase" und ihre Instanz *MAP-Kinase-Kinase-4* (hellblau) erbt alle für die Superkonzepte "Enzym", "Protein" und "Materie" explizit angegebenen Slots (rot).

Taxonomien, in denen ein Konzept mehrere Superkonzepte haben kann, nennt man auch **directed acyclic graph (DAG)**.

2.1.5 Datentypen und ihre Repräsentation durch Slot-*widgets*

Ein großer Vorteil gegenüber tabellenorientierten Repräsentationsformen bieten die vielfältigen repräsentierbaren Datentypen, die um selbstdefinierte Konzepte und Instanzen erweitert werden können. Gegenüber den in Tabellen möglichen Datentypen bietet die Ontologie zusätzlich Datentypen wie Instanz, Konzept und Hyperlink, die Relationen zwischen Daten erfassen. Den Datentypen wird vom System oder über wählbare sog. **Slot-widgets** eine entsprechend angepaßte Darstellungsart innerhalb des Frames zugewiesen (siehe Abb. 3).

Datentyp	Widget-Aussehen	Widget-Beschreibung	Anfrage-Constraints
Boolean	<input checked="" type="checkbox"/> Urgent	Eine wahr/falsch Checkbox.	is
Class	Present In <input type="text" value="Cytosol"/> <input type="button" value="V"/> <input type="button" value="+"/> <input type="button" value="-"/>	Eine Konzept-Liste, die um wählbare Konzepte als Slotwerte ergänzt werden kann. Über das "+"-Zeichen können Konzepte aus der Konzepthierarchie ausgewählt werden.	contains does not contain
Float	HPCAvgDiff <input type="text" value="-147.6"/>	Einfache Fließkomma-Slotwerte werden direkt in ein entsprechendes Feld eingegeben.	is is greater than is less than
Instance	St Predecessor <input type="button" value="V"/> <input type="button" value="C"/> <input type="button" value="+"/> <input type="button" value="-"/> <input type="text" value="apoptosis, IAP, cIAP1=36578_d"/> <input type="text" value="receptor, TNF, TRAP, TRAF2=3"/>	Instanzen als Slotwerte werden über das "+"-Zeichen hinzugefügt, oder über das "C"-Zeichen neu erstellt.	contains does not contain
Integer	Page Number <input type="text" value="14"/>	Einfache ganze Zahlen werden wie Fließkomma-Zahlen direkt in ein entsprechendes Feld eingegeben.	is is greater than is less than
String	ProbelDv2 <input type="text" value="974_at"/>	Zeichenketten werden wie die Zahlen direkt in entsprechende Textfelder eingegeben.	contains does not contain is is not begins with ends with
Symbol	Reading Level <input type="text" value="High_school"/>	Diskrete definierte Symbole werden über eine Dropdown-Liste aus den möglichen Symbolen ausgewählt.	is is not

Abb. 3: Die zur Annotation möglichen Datentypen und ihre angepasste graphische Darstellung im Frame über *widgets*. Das Erscheinungsbild der *widgets* kann über das Forms-Tab angepasst und erweitert werden (siehe Abschnitt 3.5.1). Im Query & Export Tab kann dann über die Anfrage-constraints (rechts) nach den Werten der Datentypen gefragt werden (siehe Abb. 14).

Einige weitere Darstellungs-*widgets* seien kurz genannt: Ein *instance row-widget* und *instance table-widget* erlauben die tabellarische Darstellung der Slotwerte mehrerer Instanzen in einem Tabellenfeld innerhalb des Frames. Das *contains-widget* erlaubt die Darstellung eines Frames im Frame (verschachtelte Frames). Ein *slider-widget* erlaubt die analoge Balkendarstellung von über Min.- und Max.-Werte begrenzten *integer*-Wertebereichen.

2.1.6 Semantik der Ontologie (OKBC-CLIPS)

Um die Interoperabilität zu erhöhen, erfolgte die Implementierung der Gandr Wissensrepräsentation in standardisierter Syntax und Semantik. Entsprechend der Forderung nach einer Lesbarkeit durch Mensch und Maschine basiert das Speicherformat der Ontologie auf leicht zu interpretierendem ASCII-Text in **CLIPS 6.1**-Semantik (*c language integrated*

production system, [18]). Das CLIPS-Format ist abwärtskompatibel zum OKBC-Protokoll (*open knowledge base connectivity* [19]), einer gemeinsamen Anfrage- und Entwicklungsschnittstelle für framebasierte Wissensbanken. Das OKBC-konforme Protégé KR-Metamodell liegt in Form einer Metaklassen- bzw. Metakonzep-Architektur vor, welche CLIPS bzw. die interne Struktur des Protégé Wissensmodells implementiert (siehe Abb. 4). Die Metakonzep-Architektur beschreibt also die (Meta-)Ontologie der ausdrückbaren Semantik und liefert Schablonen für die Definition eigener Konzepte und Metakonzep- bzw. (Meta-)KR-Ideome. Über Metakonzep- definierte Konzepte sind dann Instanzen dieser Metakonzep-. Sie erlauben die Veränderung und Erweiterung des Protégé Metamodells, was die Erstellung neuer eigener Semantiken ermöglicht (siehe Abschnitt 3.6.2).

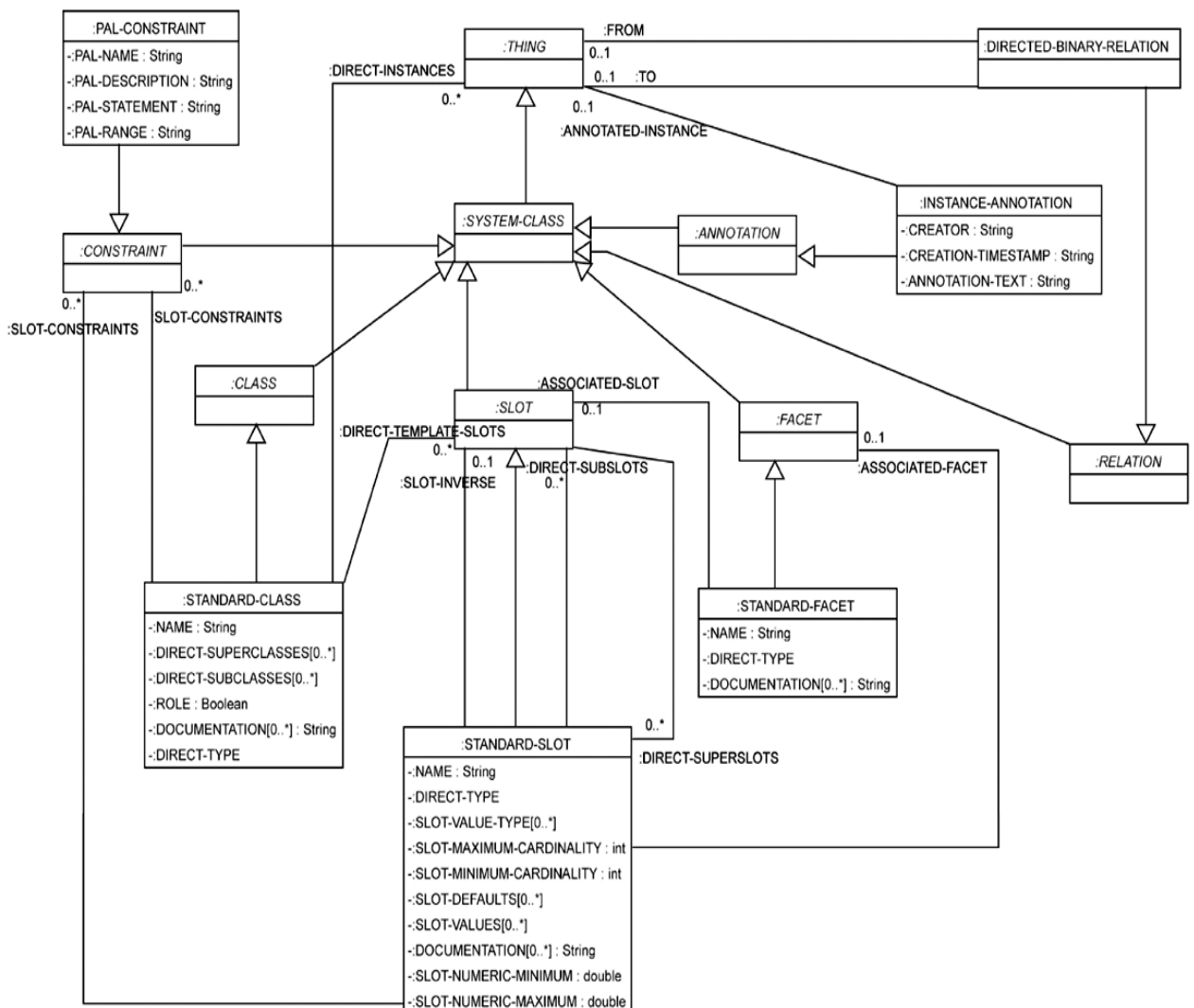


Abb. 4: Das Protégé CLIPS-Metamodell als UML-Klassendiagramm. Es beschreibt die (Meta-)Klassen-Architektur der (Meta-)KR-Ideome, die zum Aufbau von Clips-Ontologien verwendet werden. Eigene Metakonzep- werden von :STANDARD-CLASS, eigene Metaslots vom :STANDARD-SLOT abgeleitet.

2.1.7 Beschreibung des Wissensbank-Editors Protégé-2000

Die *de novo* Erstellung neuer Ontologien in Wissensrepräsentationssprachen ist technisch und inhaltlich anspruchsvoll und wird daher durch Software-Werkzeuge, sog. **Ontologie-Editoren** und **-Browser**, unterstützt [11]. Sie ermöglichen bei der Erstellung und Bearbeitung komplexer Ontologien den Überblick zu behalten. Der Ontologie- und Wissensbank-Editor dient der Kommunikation zwischen Nutzer und Wissensbank und repräsentiert die Schnittstelle zwischen einem System, das Informationen subjektiv, intuitiv, kreativ, aber langsam verarbeitet und einem System, das objektiv, logisch und schnell arbeitet. Entsprechend der Vielzahl heterogener Standards zur Erstellung und Repräsentation von Ontologien steht eine breite Palette verschiedener Ontologie-Editoren zur Verfügung [20]. Eine sehr komfortable Entwicklungsplattform für die framebasierte Ontologieerstellung bietet der Ontologie- und Wissensbank-Editor Protégé-2000 [21]. Die Hauptkriterien, die zur Wahl gerade dieses Werkzeugs geführt haben, waren folgende:

Die benutzerfreundliche intuitive graphische Oberfläche schirmt den Nutzer weitgehend von der semantischen Komplexität einer direkten Implementierung in einer Ontologiesprache ab. Der Benutzer braucht keine Wissensrepräsentationssprache zu erlernen, da die Ontologie über eine graphische Schnittstelle erzeugt und verändert wird. Die generierte Struktur wird dann von Protégé in die CLIPS-Sprache übersetzt und gespeichert. Das Werkzeug ist plattformunabhängig, frei verfügbar und besitzt eine große schnell wachsende Nutzergruppe. Durch die ständige Erweiterung und Verbesserung des Systems hält es auch auf Zeit den aktuellen Anforderungen stand. Die Dokumentation des gesamten Werkzeugs ist sehr ausführlich. Über eine sog. *discussion-group* steht eine große Nutzergruppe via e-mail zur Beantwortung von Fragen ständig zur Verfügung. Die Offenheit des Systems hat den Vorteil, daß die Ontologie und die Wissensbank auch direkt über die frei zugängliche *protégé knowledge model API* manipuliert werden können. Eigene und neu erstellte Java Anwendungen können dann über die API Bibliothek auf die Wissensbank zugreifen und gegebenenfalls als Plugin-Tab in die graphische Benutzeroberfläche integriert werden. Das Gesamte Anwendungspaket besteht aus der Protégé-2000-API, der graphischen Benutzeroberfläche mit dem Ontologie- und Layout-Editor sowie obligaten Plugins, die verschiedene zusätzliche Funktionalitäten bereitstellen. Die Protégé-API greift auf die Ontologie und die zu strukturierenden bzw. annotierten Daten zu, die aus verschiedenen Text-Dateien geladen oder aus anderen Formaten importiert werden (siehe Abb. 5).

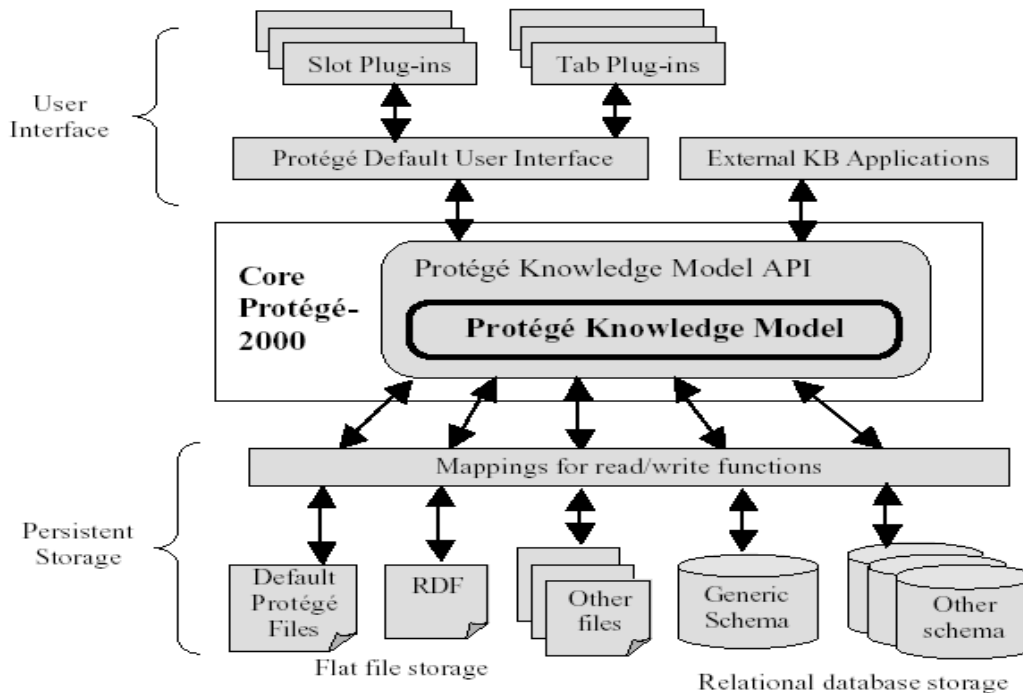


Abb. 5: Die offene Architektur der Protégé-API erlaubt die Erweiterung der Anwendung und der Nutzer-Schnittstelle um eigene Plugins (oben). Über Abbildungen auf Repräsentationssprachen ähnlicher Semantik wird der Zugriff auf diverse Speicherformate bzw. Repräsentationssemantiken ermöglicht (unten).

Die Protégé-2000 GUI ist in verschiedene sog. Tabs aufgeteilt, die verschiedene Arbeitsbereiche und Sichtweisen auf die Ontologie repräsentieren. Die Ontologie wird in Form einer interaktiven hierarchischen Verzeichnisstruktur dargestellt. Darin ausgewählte Konzepte werden als Frames mit den zugewiesenen Slots repräsentiert. Konzepte, Subkonzepte, Relationen und Instanzen können per "Mausklick" erzeugt und editiert werden. Automatisches Schlußfolgern ist in Form von Generalisierungen über Subkonzepte (Subsumption) möglich und kann über verschiedene Plugins aus dem KI-Bereich stark erweitert werden (siehe Abschnitt 3.4.8-9 und Anhang C). Weiter kann das System in breiten Grenzen kenntnis- und bedarfsorientiert für die Anwendung durch Mediziner und Biologen konfiguriert werden. Damit ist das Arbeiten mit Ontologien unter Protégé-2000 relativ schnell erlernbar. Die erstellten Annotationsschemata und Wissensbanken können über das Internet weltweit verfügbar gemacht werden. Der Zugang kann über Internet-Browser durch mehrere Nutzer gleichzeitig erfolgen. Damit die Ontologie und Wissensbank auch durch andere Anwendungen bearbeitet werden kann, wurde darauf geachtet, daß eine Transformation der Wissensbank in andere gängige Formate ohne viel Aufwand möglich ist. Diese Option bot von den in Betracht gezogenen Ontologie-Editoren lediglich das Protégé-System, genauso wie die Möglichkeit, Ontologien miteinander zu vergleichen und gegebenenfalls zusammenzuführen (siehe Abschnitt 3.4.9). Zur Datenabfrage ist eine

komfortable Anfrageschnittstelle enthalten, die eine graphische und wissensgestützte Anfragegenerierung ermöglicht. Der Fachwissenschaftler muß also keine Datenbankabfragesprache wie SQL erlernen und kann sich schneller auf die Wissensextraktion und die Annotation seiner Daten konzentrieren.

2.1.8 Erstellung von Ontologien und *ontology engineering*-Standards

Der allgemeine Ablaufplan zur Erstellung einer Ontologie besteht aus folgenden Punkten:

- Erstellen der Anforderungsspezifikation: Beschreibung der Domäne, welche die Ontologie abdecken soll, Anwendungszweck (Aufgaben, Ziele), für welche Anfragetypen (Kompetenzfragen) liefert die Ontologie Antworten, wer nutzt und pflegt die Ontologie ?
- Wissenssammlung (Wissensakquisition): Roh-Datensammlung und Datenanalyse durch Domänen-Experten. Terminologie und Ontologie-Sammlung.
- Informelle Repräsentation der Sachdomäne (in natürlicher Sprache und als Begriffs-Liste): Identifizierung der wichtigsten Konzepte und deren Eigenschaften und Relationen.
- Auswahl des Repräsentationsformalismus (Semantik).
- Entwicklung einer ersten Taxonomie der wichtigsten Konzepte (Hierarchisierung der *top-level*-Konzepte)
- Auswahl und Verwendung eines Ontologie-Editors.
- Formale Repräsentation und Konzeptualisierung: Kodierung in formaler, maschinenlesbarer Syntax (Wissensrepräsentationssprache), gegebenenfalls Integration vorhandener Ontologien oder DB-Schemata.
- Import der Daten, die als Instanzen repräsentiert werden und gegebenenfalls manuelle Erstellung von Instanzen
- Vernetzung: Verknüpfung und Ausbau zum Wissensnetz über vermehrte Bildung von Relationen.
- Erstellung einer exemplarischen ontologischen Daten-/Situationsbeschreibung durch probeweise Erstellung und Ausformulierung entsprechender Instanzen.
- Evaluation und Dokumentation.
- Bewertung, Veröffentlichung und Wartung.

Standardisierte *ontology engineering-Methodologien* definieren Ontologie-Entwicklungs-Stadien sowie Prinzipien und Richtlinien, die diesen Stadien zugrunde liegen. Ein sog. *ontology engineering life cycle* verdeutlicht die Zusammenhänge zwischen den Stadien. Er kann lineare Aspekte betonen, wie bei TOVE [12], oder eher den iterativen Aspekt betonen, wie bei der MethOntology [11]. Wir orientierten uns an einer Zwischenform, der V-Process-Methodologie, [22] aus der Softwareentwicklung.

2.2 Ontologiebasiertes Wissensmanagement in den Biowissenschaften

Die ersten Anfänge ontologiebasierten Wissensmanagements reichen zurück bis in die griechische Medizin unter Hippokrates. Die Interpretation Indices genannter Krankheits-Symptome stellte dort eine Semiotik "natürlicher" Zeichen, wobei auch der Terminus "Semiotik" (**Zeichenlehre**) bzw. "Semeiologie" der medizinischen Symptomatologie entstammt. Eine der ersten medizinischen Klassifikationssysteme war die "Nosologica Methodica" von Francois Bossier de Lacroix (1706-1777). Aus einem 1893 in England entwickelten kontrollierten Vokabular zur Beschreibung "Warum Menschen sterben" entwickelte sich die **International Classification of Diseases, ICD**, die heute über 16 000 manuell geordnete Begriffe enthält und weiteste Verbreitung findet. Über derartige Ontologien werden die medizinische Dokumentation und Auswertung standardisiert. In letzter Zeit werden vermehrt auch Phänotyp-Ontologien für das äußere Erscheinungsbild von Krankheiten entwickelt [23].

Der Ansatz, einheitliche Terminologien zur computergestützten Datenannotation einzuführen, ist nicht neu [24, 25]. Meist werden ontologische Konzepte zur standardisierten Kommunikation zwischen Wissenschaftlern verschiedener Fach-Domänen [26], häufiger jedoch als semantische Grundlage für Meta-Datenbanken verwendet, die über den gemeinsamen Zugriff den Zugang zu heterogenen Datenbanken integrieren [27, 28]. Bioontologien dienen häufig auch als Lieferanten semantisch definierter *classifier* für statistische Untersuchungen [29].

2.3 Anwendungsdomäne: *Toll-like Receptors* und dendritische Zellen

Am Beispiel der Toll-like Receptor-/NFkB-Signaltransduktion (TLR-ST) wird ein "*modelling by annotation*"-Ansatz vorgestellt. Über die hier erstellte Ontologie sollen begriffliche Konzepte zur formalen Annotation von Genen und Genprodukten auf DNA-Microarrays zur Verfügung gestellt werden. Dabei soll die Annotation an die fachorientierten Bedürfnisse der Nutzergruppe angepaßt sein. Die Anwendergruppe führt Expressionsanalysen im Hinblick auf Immunbiologie und Ontogenie dendritischer Zellen (DC) in Maus und Menschen durch. Der Erforschung von DC durch *expression-profiling* kommt im Rahmen der molekularen Medizin große Bedeutung zu, da sie eine Schlüsselrolle bei der Initiation der antigenspezifischen adaptiven Immunantwort spielen [30, 31, 32]. Eine Infektion mit exogenen Pathogenen löst über eine schnelle direkte Immunantwort Entzündungsreaktionen aus. Hier binden DC die pathogenen Antigene, phagozytieren sie und präsentieren die prozessierten Antigene über MHC I und II den T-Lymphocyten [33, 34, 35]. Die Erkennung der prozessierten Pathogene erfolgt über sogenannte *pattern-recognition receptoren* (PRR), die spezifische konservierte Motive der Pathogene, sog.

pathogen associated molecular pattern (PAMP) erkennen. Das bekannteste PAMP ist das bakterielle Endotoxin LPS, das bei besonders starken Infektionen einen *septic shock*, eine Überreaktion der Zelle und Apoptose auslösen kann. Zu den bekanntesten PRR gehören die für die Arbeitsgruppe besonders interessanten TLR. Diese aktivieren über das I κ B-NF κ B-Signaltransduktions-Modul (siehe Abb. 13, 15 und 17) die Genexpression von Entzündungsmediatoren wie TNF, Cytokinen, Interferonen und Interleukinen, wobei unterschiedliche TLR-Aktivierungsmuster unterschiedliche Antwort-Expressionsprofile auslösen. Ontogenetisch stammen dendritische Zellen von hämatopoetischen Stammzellen des Knochenmarks ab. Sie werden in unterschiedlichen Reifungsstadien und Untertypen gefunden, die über spezifische Zelloberflächenmoleküle und Funktionen charakterisiert werden [36, 37, 38]. Die Arbeitsgruppe untersucht, ob und wie sich diese verschiedenen DC-Stadien in ihren Expressionsprofilen insbesondere im Hinblick auf bestimmte Markergene unterscheiden [39, 40, 41, 42]. Langfristig soll untersucht werden, inwiefern sich funktionell modifizierte immunstimulierende DC im Rahmen einer zukünftigen Krebs-Immunotherapie nutzen lassen. So wäre denkbar, Krebspatienten Tumor-Antigen präsentierende DC zu applizieren, die dann das Immunsystem des Patienten gegen diese Tumore aktivieren [43]. In diesem Ansatz produzieren dann *ex vivo* hergestellte modifizierte DC nach Transfektion mit cDNA von Immunmodulatoren, wie z.B. Cytokine oder Chemokine, entsprechende Antigene und lösen eine therapeutische zelluläre Immunantwort aus.

Die Gandr-Ontologie soll Begriffe zur Modellierung und der Analyse des TLR- und NF κ B-Signaltransduktionsweges liefern. Die anhand der Ontologie erstellten Wissensmodelle sollen die Untersuchung dieser Signaltransduktionsnetze im Hinblick auf die Differenzierung hämatopoietischer und immunbiologisch relevanter Blutzellen unterstützen. Die Ontologie reflektiert diese domänenspezifischen Forderungen in ihren auf diese Gebiete genauer angepaßten Ontologie-Modulen. So enthält die Gandr-Ontologie eine Klassifizierung von Blutzellen und ist besonders detailliert auf den Gebieten der TLR-ST (siehe Abb. 17). Außerdem sind Daten zu DC-Markergenen (CD-Molekülen) in der Gandr-Wissensbank enthalten. Genaue Beschreibungen potentieller Anwendungs-Szenarien befinden sich im Abschnitt 3.4.

2.4 Expressionsanalyse mit dem Affymetrix® Human Genome U95Av2 GeneChip®

Der photolithographisch [44] hergestellte HG-U95 A v2 Genchip® repräsentiert 12626 humane, in ihrer Funktion bekannte, d.h. relativ gut annotierte, *full-length*-Gene. Er wird deshalb im Gegensatz zu ESTs repräsentierenden B- und C-Chips besonders häufig zur Identifizierung und Analyse funktioneller genetischer Zusammenhänge herangezogen. Die über den Identifier **probe set ID** identifizierten Gensequenzdaten entstammen Sequenzclustern der UniGene-Datenbank (*build 95*), deren Daten aus GenBank Ver. 113 und dbEST/10-02-99 stammen. Die ausführliche Annotation und die Tatsache, daß sich der HG-U95Av2-Chip im Aufbau nicht so schnell verändert wie andere Microarrays, lassen ihn als Ausgangsmaterial und Lieferant von Instanzdaten für eine ontologische Annotation besonders geeignet erscheinen.

3 Ergebnisse

In dieser Arbeit wird eine molekularbiologische Ontologie und Wissensbank vorgestellt. Sie soll der formalen Annotation und Modellierung von Genen und Geninteraktionen dienen. Hierüber erstellte Annotationsmodelle können interaktiv und graphisch visualisiert werden und sind ontologischen Anfrageschnittstellen zugänglich. Dieses ontologische Wissensmanagement unterstützt die systemische Datenanalyse im Rahmen des *expression-profiling*.

3.1 Erstellung der Gandr-Ontologie

Dieser Abschnitt beschreibt, wie die Gandr-Ontologie in Anlehnung an die in Abschnitt 2.1.8 beschriebene Methode aus verschiedenen terminologischen Quellen, und unter Einführung neuer *ontology engineering* Ansätze, erstellt und ausformuliert wurde.

3.1.1 Anforderungsspezifikation und Kompetenzfragen

In der ersten Phase der Ontologie-Erstellung wurde durch Expertenbefragungen erhoben, welche Anforderungen die Nutzer an die Ontologie und das Gesamtsystem haben.

Die Anforderungsspezifikation enthält eine Beschreibung der Domäne, welche die Ontologie abdecken soll (siehe Abschnitt 2.3), die Anwendungszwecke der Ontologie im Gesamtsystem und bildete die Wissensgrundlage für die Erstellung einer ersten prototypischen Konzept-Taxonomie. Die Anforderungsspezifikation sollte später als gemeinsame Referenz der Evaluierung der Ontologie und des Gesamtprojekts dienen. Wichtige Anforderungen an das Gesamtsystem waren folgende:

- a) Das System soll dem Nutzer eine laborspezifische Annotation von Genen und Microarray-Expressionswerten mit ontologisch fundierten Konzepten ermöglichen. Über die Ontologie als Wissensrepräsentationsformat sollen dem Nutzer moderne Wissensmanagementtechniken verfügbar gemacht werden. Die wichtigsten sind die ontologiebasierte Annotation bzw. Wissensmodellierung, ontologische Abfragen auch nach implizitem Wissen und die Visualisierung der Annotationsmodelle als semantische Netzwerke.
- b) Inhaltlich verwandte Daten sollen über eine Art Hyperlink frei und assoziativ zugänglich sein und die Kontextexploration von KR-Ideomen bzw. Genen in einer Art Browser-Paradigma intuitiver, schneller und einfacher gestalten.
- c) Über einen integrierten Zugang zu verschiedenen webständigen Datenbanken sollen auch informale molekularbiologische Hintergrund- bzw. Kontextdaten als Entscheidungshilfen zur formalen Annotation zugänglich gemacht werden.
- d) Das System soll selbsterklärend sein und den Nutzer bei Annotation und Datenabfrage aktiv unterstützen.

- e) Das System soll verschiedenste Formate erkennen, importieren, ineinander umwandeln und exportieren können, um dem Nutzer eine möglichst breite Palette an integrierbaren Daten zu bieten und eine spätere Weiterverarbeitung mit verschiedensten Werkzeugen zu erlauben.
- f) Die Trennung der Ontologie von den Daten sollte der Verbreitung des Schemas bzw. der Ontologie dienen bzw. deren Wiederbenutzung in anderen Gruppen und mit anderen Werkzeugen ermöglichen.
- g) Das System soll eine graphische Benutzeroberfläche haben.
- h) Eine Trennung der Projekteinstellungen von den Daten sollte nutzerspezifische Anwendungen und Einstellungen an der GUI ermöglichen.
- i) Das System soll einfacher zu erlernen und zu installieren sein als konventionelle Datenbanksysteme ähnlicher Expressivität.
- j) Die Erweiterung, Wartung und Aktualisierung (Integration weiterer Daten) sollte durch den Endnutzer selbst möglich sein.

Bei der Erstellung der Anforderungsspezifikation wurden die Domänenexperten gefragt, welche typischen Anfragen sie zur Unterstützung ihrer Aufgaben durch das System beantwortet haben möchten. Die Fähigkeit, Fragen über einen bestimmten Themenbereich zu beantworten, wird als Kompetenz der Ontologie, die Anfragen als Kompetenzfragen bezeichnet. Diese befinden sich als formalisierte Beispiele in der Anfragenbibliothek des GandrKB Projekts (siehe Anhang B.).

3.1.2 Wissensakquisition

Da die Entwicklung von Ontologien ein hinreichend detailliertes Wissen über die wichtigsten Begriffe und Beziehungen innerhalb des Anwendungsgebiets erfordert, ist einer der ersten Prozesse im *knowledge engineering* die Einarbeitung in das Sachthema und die Erfassung des domänenspezifischen Wissens (die **Wissensakquisition**). Es folgt eine Übersicht über vorhandene Wissensrepräsentationen, die als Strukturgeber einer ersten von Hand zu erstellenden kleinen *top-level*-Ontologie untersucht wurden. Die tatsächlich verwendeten Quellen sind mit einem "+" markiert:

- Vokabularien und Ontologien allgemeiner Standardisierungs-Organisationen (National Institute of Standards and Technology, NIST, <http://www.nist.gov/>)
- +Vokabularien und Ontologien verschiedener sachdomänenorientierter Ontologie-Konsortien und Bibliotheken (E.C.-Kommission, PROW, R. Stevens Internetseite, Protégé Internetseite: <http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary>)
- GO *external mapping* Internetseite: <http://www.geneontology.org/GO.indices.html>
- XML-basierte *mark-up* Sprachen (MIAME und BSML)
- +Datenbankschemata (*Transfac-TF*, *Transcription Factors*)
- +Verschiedene Tabellen mit Annotationen (Netaffx)

- +Schlüssel-Veröffentlichungen, *Reviews* und Standardwerke der Literatur zum Sachgebiet
- +*Abstracts* und *keywords* aus Bibliographie-Programmen bzw. Referenzbibliotheken (*Endnote*[®] *library*)
- +Textdatenbanken (Medline), die über *textmining*- und NLP-Software analysiert werden
- +Ein von der Anwendergruppe (Prof. M. Zenke, Institut für Biomedizinische Forschung, RWTH, Uni-Aachen) genutztes Annotations-Vokabular (Wortliste als zenke.xls)
- +Expertenbefragungen

3.1.3 Manuelle Erstellung einer prototypischen Ontologie

Der weitaus größte Teil an Domänenwissen liegt in Form von Fachveröffentlichungen vor, die teils in Literaturdatenbanken als frei zugänglicher Text vorliegen. Es liegt nahe, diese als Begriffs-Quellen zur Erstellung von Ontologien heranzuziehen. Dies sichert weitestgehende ontologische Übereinkunft, da die Namen der KR-Ideome direkt aus der Fach-Terminologie abgeleitet werden. Dieses sog. *ontological commitment* sichert die konsistente Interpretation verschiedener auf die Ontologie zugreifender Agenten [14, 45]. Ausgangsbasis für die manuelle Erstellung der ersten Konzept- und Slot-Liste waren drei vom Domänenspezialisten und späteren Anwender ausgesuchte Schlüssel-Reviews zur Sachdomäne [36, 38, 46]. Darin wurden von Hand domänenspezifische Nomen, Adjektive und Verben der Sachdomäne als potentielle KR-Ideome markiert und zur Erstellung einer ersten kleinen Konzept- und Slot-Liste herangezogen. Hieraus wurde manuell eine prototypische Ontologie erstellt. Die Strukturierung der *top-level*-Konzepte dieser Ontologie erfolgte in Anlehnung an die *Gene Ontology*, GO von Ashburner [47] und die *Signal Transduction Ontology*, STO von Fukuda [48] und die *Molecular Biology Ontology*, MBO von Schulze Kremer [24].

3.1.4 Erweiterung der Ontologie um domänenspezifisches Vokabular

3.1.4.1 Erstellung des Ausgangs-Textkorpus zur KR-Ideom-Extraktion

Zunächst wurde eine Sammlung von Textdokumenten angelegt, die möglichst viele arbeitsgruppenspezifische Begriffe enthalten sollte, aus denen dann fachrelevante Konzepte und Slots erstellt werden konnten. Dieser möglichst domänenspezifische Textkorpus wurde über serielles Ausführen verschiedener Textmining- und NLP-Programme erstellt und bestand aus folgenden Text-Dokumenten:

- Aus der *Endnote*[®] *Reference Library* (NFkB.enl) des Arbeitsgruppenleiters wurden Titel, *abstracts* und *keywords* extrahiert.
- Über das Internet-Werkzeug XplorMed [49] wurde eine domänenspezifische Medline-*abstract* Sammlung extrahiert. Die *seed*-Wörter waren "*dendritic cell development*".

- *Reviews* und Schlüssel-Veröffentlichungen zur Sachdomäne.

3.1.4.2 POS-tagging und Extraktion potentieller KR-Ideome

Die Weiterverarbeitung des Textkorpus geschah über eine *part of speech (POS)-tagging-software* (*Tagger*, NLM, A. Zamora, persönliche Kommunikation, e-mail 14.06.2003). *Tagger* führt sowohl *chunking*, als auch morphologisches *tagging* durch. Dabei werden alle Wörter des Korpus automatisch mit ihren Wortarten (Nomen, Verben und Adjektive) annotiert. *Tagger* nutzt dabei die Wörterbücher *usuk.dic* und *medical.dic*, die allgemeine bereits "*getaggte*" englische Wörter und medizinische Fachausdrücke enthalten. Allgemeine umgangssprachliche Begriffe ohne ontologische Relevanz wie "and", "a", "the" wurden über eine Stopliste herausgefiltert, so daß nur domänenspezifische Begriffe übrig blieben. Zur Tokenbildung, also der Trennung der Worte im Text wurden nur das Leerzeichen und die Satzschlußzeichen verwandt.

Beispiel Tagger Input:

"A conditional v-Rel estrogen receptor fusion protein, v-RelER, causes estrogen-dependent but otherwise unaltered v-rel-specific transformation of chicken bone marrow cells."

XML Output format (\$op=11):

```
<azParag>
<azSent><azPhr p="N"><azPOS p="T">A </azPOS><azPOS p="J">conditional </azPOS><azPOS p="J">v-Rel
</azPOS><azPOS p="N">estrogen </azPOS><azPOS p="N">receptor </azPOS><azPOS p="N">fusion
</azPOS><azPOS p="N">protein</azPOS>, <azPOS p="J">v-RelER</azPOS></azPhr>,
<azPhr p="V"><azPOS p="V">causes </azPOS></azPhr><azPhr p="N"><azPOS p="J">estrogen-dependent
</azPOS><azPOS p="C">but </azPOS><azPOS p="J">otherwise </azPOS><azPOS p="J">unaltered
</azPOS><azPOS p="J">v-rel-specific </azPOS><azPOS p="N">transformation </azPOS></azPhr><azPhr
p="P"><azPOS p="R">of
</azPOS><azPOS p="J">chicken </azPOS><azPOS p="N">bone </azPOS><azPOS p="N">marrow
</azPOS><azPOS p="N">cells</azPOS></azPhr>.</azSent>
</azParag>
```

Tabelle 1: Erklärung der p-Attribut-Werte des azPOS-Tags (POS-*mark-up* der *Tagger*-Software)

Wortart	Tag <azPOS p="...">	Beispiele
ARTICLE	"T "	a the
AUXILIARY VERB	"X "	is a where be was
ADVERB	"A "	rather therefor now well here in
CONJUNCTION	"C "	that and or but while
PREPOSITION	"R "	with for on of under in after by
PRONOUN	"P "	this these we both it those our
NOUN	"N "	periphery antigens lymphocyte organs cytokines responses
VERB	"V "	tolerize obtained allow comes use presented enhanced
VERB USED AS GERUND	"G "	secreting including supporting using involving activating
VERB USED AS PARTICIPLE	"L "	induced activated phosphorylated associated expressed
ADJECTIVE	"J "	flexible unlinked high haematopoietic dependent induced
INTERJECTION	"I "	to
DETERMINER	"D "	many both all some such no
...		
Noun phrases _____	<azPhr p="N">	
Prepositional phrases++++++	<azPhr p="P">	
Verb phrases=====	<azPhr p="V">	
Infinitive phrases ~~~~~	<azPhr p="I">	

Die POS-Information der Begriffe (siehe Tabelle 1) lieferte dann Informationen über den potentiellen KR-Ideom-Typ, in den dieses Wort transformiert werden sollte. Über ein XSLT-Programm wurden die PCDATA-Elemente der azPos-Tags, je nach den Werten des Tag-Attributes p, in verschiedene Textdateien geschrieben. Im Textkorpus vorkommende Nomen, also PCDATA aus azPos-Tags mit dem Attributwert p=N wurden in eine Liste mit Nomen, also potentiellen Konzepten, geschrieben. PCDATA aus azPos-Tags mit dem Attributwert P=V, J, G und A wurden in eine Liste mit Verben, Adjektiven, Gerundien und Adverben bzw. potentiellen Slots geschrieben. Adjektive können jedoch zu Slots und zu Subkonzepten transformiert werden (z.B. Slot: "Receptor" *has-localisation* "Membrane" oder Subkonzept: "Membrane_receptor" *is-a* "Receptor").

Die so erhaltenen Listen potentieller Konzept- und Slotnamen wurden nach Worthäufigkeit sortiert und alle mehr als dreimal auftauchenden Wörter manuell, im *middle-out*-Ansatz, zum *ontology refinement* der prototypischen Ontologie herangezogen (siehe Abschnitt 3.1.5). Seltener auftauchende Begriffe repräsentierten oft unwichtigere, domänenfremde Begriffe oder Instanz-Namen. Die aus besonders häufig vorkommenden Begriffen abgeleiteten Konzeptnamen enthielten später oft mehr Subkonzepte bzw. Instanzen, als seltener auftauchende und lagen meist auch weiter oben in der Konzepthierarchie.

3.1.5 Taxonomische Integration der Konzepte unter die prototypische Ontologie

Zum Konzeptionalisieren konnten in Anlehnung an gängige *software-engineering*-Methoden *top-down*-, *bottom-up*- oder *middle-out*-Strategien eingesetzt werden. *Top-down* bedeutet, ausgehend von den generellen Konzepten über schrittweise Verfeinerung speziellere Subkonzepte zu formalisieren. *Bottom-up* bedeutet umgekehrt, ausgehend von speziellen Konzepten über schrittweise Verallgemeinerung und Abstrahierung generellere Superkonzepte zu formalisieren. In unserem Falle wurde zunächst mit der *top-down*-Strategie begonnen und dann mit der *middle-out* Strategie eine Kombination aus *top-down*- und *bottom-up*-Verfahren angewandt. Ausgehend von den wichtigsten Konzepten wurde also über schrittweise Verfeinerung und Verallgemeinerung in beide Richtungen formalisiert. Die Konzeptionalisierung wurde nur so detailliert durchgeführt, wie es der Verwendungskontext vorgab. Dabei wurde der Umfang der Wissensbank gegen den benötigten Kodierungsaufwand und semantische Redundanz gegen Durchsuchbarkeit abgewogen. Slots, deren Werte über *deeplinks* einsehbar waren, wurden nur dann als KR-Ideome formalisiert, wenn sie ontologischen Anfragetechniken zugänglich sein sollten.

Die taxonomische Gliederung der Konzepte soll möglichst viele potentielle Suchattribute subsumieren und dadurch eine umfassendere Antwort auf Datenabfragen unterschiedlicher Abstrahierungsniveaus ermöglichen. Durch die einmalige Platzierung eines Konzepts "TNF" in die Taxonomie wird von dem System geschlossen, daß eine TNF-Instanz, ein zuvor mit dem TNF-Konzept annotiertes Gen der probe set ID *1715_at* ein Tumor-Necrosis-Factor, ein Apoptose-Protein, ein Protein und damit eine zelluläre Komponente ist. Über die Vererbung wird geschlossen, daß die TNF-Instanz alle Eigenschaften bzw. Slots einer zellulären Komponente, eines Proteins und Apoptose-Proteins haben muß. Derartiges implizites Wissen wird bei korrekter Instanz-Positionierung über die Gandr-inherente Taxonomie expliziten Anfragen zugänglich.

Wenn zur Taxonomisierung der Konzepte das eigene Wissen nicht ausreichte, so wurden folgende Quellen zur weitergehenden Wissensakquisition genutzt:

- Andere Ontologien (dortige Position und Definitionen, z.B. aus UMLS)
- Eine Text-Konkordanz des Ausgangs-Textkorpus konnte genutzt werden, um den unmittelbaren Wissenskontext, in dem das Wort in der Fachliteratur auftaucht, zur Positionierung heranzuziehen (siehe Abschnitt 3.1.5.2)
- Internet-Recherchen mit herkömmlichen Suchmaschinen (z.B. Google-Suchen)
- Entsprechende Anfragen beim Domänenexperten

Bei der Taxonomisierung wurde darauf geachtet, daß für ein Konzept in der angestrebten Position die Konzepte seiner gesamten Vorfahren Gültigkeit haben. Nur so konnte korrekte Subsumption und die konsistente Vererbung von Konzept-Eigenschaften gewährleistet werden. Generell wurde versucht, Redundanz zu vermeiden, d.h. es sollten keine Konzepte semantischer Äquivalenz mit verschiedener Bezeichnung vorkommen. Bei Geschwisterkonzepten wurde Zugehörigkeit zur selben Spezialisierungs-Ebene, d.h. ein ähnlicher Generalisierungslevel angestrebt. Es wurde versucht, nicht unnötig viele Konzepte und Slots zu erstellen (*scope limitation*), sich also auf die im Rahmen der Anforderungsspezifikation geforderten und hierfür sinnvoll nutzbaren KR-Ideome zu beschränken. Während des abschließenden *ontology refinements* wurden daher verschiedene instanzlose Konzepte und *domain*-lose Slots wieder gelöscht. Am Ende des *ontological engineering* wurde die Ontologie nochmals überprüft: Enthielt ein Konzept mehr als zwanzig direkte Subkonzepte, wurde geprüft, ob nicht die Eigenschaften zur Konzeptdefinition zu weit gefaßt waren und die Struktur hier über neue Konzepte und Slots verfeinert werden konnte.

Da Ontologien, die zur Wissensrepräsentation genutzt werden sollen insbesondere auf *leaf*-Konzept-Ebene nie "fertig" zu nennen sind, war es schwierig zu entscheiden, wann die Konzeptualisierung beendet werden sollte.

3.1.5.1 Benennungs-Konventionen für Konzepte und Slots

Die zur funktionellen Beschreibung und Annotation der probe set IDs verwendeten ontologischen Konzepte beschreiben nicht Genfunktionen im engeren Sinne, sondern die Funktionen der durch sie kodierten Proteine. Um als Suchattribut verwendbar zu sein, sollten die vergebenen Konzeptnamen kurz, intuitiv und leicht zu merken sein. Protégé-2000 ordnet neu erstellten Konzepten einen - automatisch aus dem KB-Namen und einer Nummer bestehenden - internen Identifier zu. Um das Speicherformat möglichst klein zu halten und die Performanz zu verbessern, wurde darauf geachtet, daß der KB-Projekt-Name möglichst kurz ist.

Konzepte, deren Positionierung zweifelhaft war, wurden zunächst mit einem "?" als Präfix des Konzeptnamens gekennzeichnet. Wo sie völlig unsicher war, wurde ein "Orphan"-Konzept erstellt und das entsprechende Konzept zur späteren Positionierung dort abgelegt.

Wo in der Sachdomäne eine Abkürzung wesentlich häufiger benutzt wird als der ausgeschriebene Name, oder wo es sich um *leaf*-Konzepte handelte, fanden gelegentlich Abkürzungen, meist *genesymbols* als Konzeptnamen Verwendung. Dies erschien legitim, da *genesymbols* tatsächlich Genklassen beschreiben, die meist auch für Orthologe anderer Organismen (z.B. Maus) gelten. Das Protégé-System unterscheidet zwischen Groß- und

Kleinschreibung. Konzeptnamen wurden immer groß geschrieben und im Singular angegeben. Leerzeichen und Bindestrich in Konzeptnamen wurden durch Unterstrich ("_") ersetzt. Slot-Namen wurden immer klein geschrieben. Nur die über das Datagenie-Plugin (siehe Abschnitt 3.3.2) importierten Slots, Eigenschaften also, die aus anderen Datenbanken stammten, wurden groß geschrieben, um diese für eine spätere Aktualisierung schneller erkennbar zu machen. Wörter aus dem CLIPS-Metadaten Beschreibungsformat (wie `class`, `property`, `slot`) fanden in KR-Ideom-Namen keine Verwendung. Namen von Subkonzepten, die als Teil den Superkonzeptnamen enthalten, wurden möglichst konsistent unter Integration dieses Superkonzeptnamens benannt: Beispielsweise bekommen alle unter dem "Receptor"-Konzept stehenden Subkonzepte einen Namen der Form "XY-Receptor", wobei der Präfix XY den Liganden oder die "Cellular_Localisation" des Rezeptors kennzeichnet. Da aber auch hier die Häufigkeit des Begriffsnamens in der Sachdomäne oberste Priorität hatte, konnte diese Systematik nicht streng durchgehalten werden.

3.1.5.2 Nutzung einer Text-Konkordanz zur Konzeptpositionierung

Da der Textkorpus aus der Endnote[®]-Referenz-Bibliothek und den Schlüssel-Veröffentlichungen das selektiv domänenspezifische Wissen über die Konzepte und Slots enthielt, konnte er als Wissensquelle zur Konzeptpositionierung herangezogen werden. Da die ontologischen Konzepte teilweise aus diesem Text extrahiert wurden (siehe Abschnitt 3.1.4), konnte über eine **Text-Konkordanz** jederzeit unmittelbar auf den laborspezifischen semantischen Kontext der Begriffs-Konzepte zugegriffen und dieser zur Konzeptpositionierung und -überprüfung herangezogen werden. Zur Erstellung der Konkordanz wurde das korpuslinguistische Programm *Concordance*[®] von R. J. C. Watt (<http://www.concordancesoftware.co.uk/>) verwendet (siehe Abb. 6).

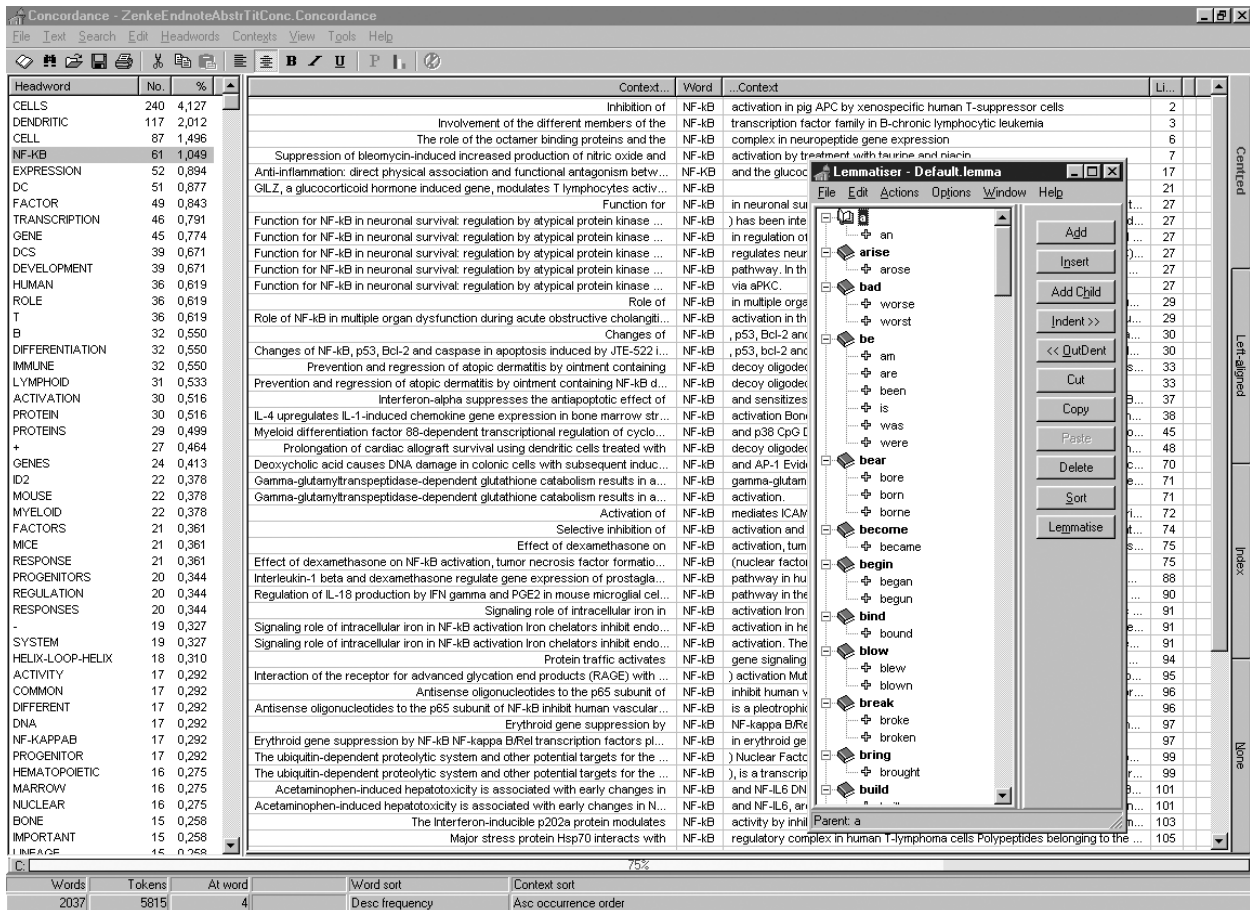


Abb. 6: Zugang des textualen Kontexts von potentiellen Konzepten entsprechenden Wörtern im Ausgangstextkorpus. In der linken Spalte sind die *tokens* nach ihrer Häufigkeit im Ausgangstextkorpus sortiert dargestellt. Bei Auswahl eines Wortes "NF-kB" (links) wird rechts die Wortumgebung (Kontext) und die Fundstelle innerhalb des Ausgangstextkorpus angezeigt. Dies ermöglicht einen schnellen Zugriff auf zur Taxonomisierung des diesem Wort entsprechenden Konzepts. Der Lemmatiser (Fenster rechts) wurde nicht genutzt.

3.1.5.3 Gliederung in *top-level-Module*

Die Ontologie ist in verschiedene mehr oder weniger abgeschlossene Oberkategorien, sog. *top-level-Module* gegliedert. Diese dienen der Orientierung in der Taxonomie bzw. der leichteren Auffindbarkeit von Konzepten bestimmten Typs und sollen ferner die Wiederverwendung einzelner Ontologie-Module in anderen Anwendungen erleichtern. Die wichtigsten die Immunbiologie- und Entwicklungsvorgänge dendritischer Zellen beschreibenden Konzepte sind in den Ontologie-Modulen "Biological_Process", "Cellular_Compound", "Cellular_Localisation", "Extracellular_Compound" und "Cell_Types" enthalten. Die ersten drei Module entsprechen den GO-Modulen *Biological Process*, *Molecular Function* und *Cellular Component*.

3.1.5.4 Partonomien und Prozess-Taxonomien

Neben der überwiegenden *is-a*-Taxonomie wurden auch Partonomien formuliert. Gene Ontology enthält *is-a*- und *part-of*-Relationen in einer Hierarchie z.B.

```
Organ          Kind-of
  Heart        Part-of
    Heart valve Kind-of
      Aortic valve Part-of
```

Solche gemischten Hierarchien können zwar von Menschen, schwer aber vom Computer bearbeitet werden. Sie stören die konsistente Slot-Vererbung und wurden daher in der Gandr-Ontologie vermieden. Des weiteren wurde eine "Biological_Process"-Taxonomie auf *top-level*-Ebene erstellt. Sie dient der Annotation von probe set ID-Instanzen mit einem bestimmten Stoffwechsel-Kontext bzw. mit biologischen Prozessen, die bestimmte probe set IDs vermehrt entstehen lassen (*produced_by*-Slot) oder verbrauchen (*consumed_by*-Slot). Als Wert für den *produced_by*-Slot wird ein unter dem "Biological_Process"-Konzept stehendes Konzept und keine Instanz erwartet, da man hier pragmatisch schnell annotieren will und nicht jedesmal "Biological_Process"-Instanzen erzeugen will. Die entsprechenden inversen Slots des "Biological_Process"-Konzepts sind *produces* und *consumes*.

3.1.5.5 Formalisierung als Konzept, Slot oder Instanz

Beim *ontology refinement* wurden neu zu formalisierende Konzept-Eigenschaften entweder über das Erstellen und Hinzufügen neuer Slots zu einem vorhandenen Konzept oder über ein neu erstelltes Subkonzept erfaßbar gemacht. Eine Dopamin-Rezeptor-Instanz z.B. konnte über ein vorhandenes Konzept "Rezeptor" definiert werden, zu dem lediglich der Slot *has_Ligand* vom Typ `[Instance of Small_Molecule]` hinzugefügt wurde. Alternativ konnte einfach ein Rezeptor-Subkonzept "Dopamine_Receptor" erstellt und instantiiert werden. Die Entscheidung, ob eine neue Eigenschaft unter Erstellung eines neuen Slots für ein vorhandenes Konzept oder unter Erstellung eines neuen Subkonzepts modelliert wurde, war anwendungsabhängig. War die Eigenschaft konzeptbildend, erzeugte also diese Eigenschaften im Menschen den Eindruck eines eigenen Konzepts und wurde in der Domänensprache auch so verwandt, so wurde sie auch als selbständiges Konzept modelliert. Diese neuen Konzepte sollten sich durch neue Slots vom Superkonzept unterscheiden. Dies ist in der Gandr-Ontologie dort nicht der Fall, wo Konzepte lediglich Begriffs- oder Suchattribut liefernde Funktion haben (siehe Schachtelkonzepte, Abschnitt 3.4.1). Auch die Granulいた und die Entscheidung, auf welchem Detaillevel die Formalisierung über Konzepte endet bzw. wo weitere Spezialisierung über das Erstellen von

Instanzen formalisiert wird, ist anwendungsabhängig. Sie wurde für die Gandr-Ontologie zunächst bis zu einem Detailgrad durchgeführt, der durch das gelieferte Vokabular vorgegeben war; die Granulいた kann jedoch beliebig erhöht werden.

3.1.5.6 Nutzung von Mehrfachvererbung

Im Gegensatz zu Konzepten können Instanzen in der GandrKB lediglich einem Konzept zugeordnet sein. Das bedeutet jedoch nicht, daß sie deshalb nur über ihre direkten Superkonzepte gefunden werden können. Um Instanzen zum einen unter verschiedenen Konzeptnamen auffindbar zu repräsentieren und zum anderen die parallele Vererbung von Eigenschaften von verschiedenen Superkonzepten (Mehrfachvererbung) zu ermöglichen, wurden den Konzepten einiger Instanzen mehrere Superkonzepte gleichzeitig zugeordnet. Das "G_Protein_Coupled_Receptor"-Konzept beispielsweise ist sowohl dem Konzept "Surface_Protein", also auch dem "Receptor"-Konzept, untergeordnet. Wird nun in der Wissensbank nach "Surface_Protein", also nach Instanzen dieses Konzepts, gesucht, so werden alle Instanzen des Konzepts "G_Protein_Coupled_Receptor" ebenfalls gefunden, ohne daß jemals eine "G_Protein_Coupled_Receptor"-Instanz explizit als "Surface_Protein" annotiert wurde. Die Mehrfachvererbung erhöht also die Trefferzahl bei Anfragen an die Wissensbank, da sie den Anfragen weitere implizite Annotierungen zugänglich macht bzw. diese über Subsumption erschließt (siehe Abschnitt 3.4.1).

3.1.6 Erweiterung der Semantik

Um auch detailliertere Anfragen beantwortbar zu machen, müssen zur Annotation benutzte Konzepte eine ausreichend große Anzahl Unterscheidungsmerkmale haben. Daher wurde ein Repräsentationsformat gewählt, das die Definition entsprechender Konzept-Eigenschaften und Relationen als Slots erlaubt. Die taxonomische *is-a*-Strukturierung wird so um Eigenschaften und Beziehungen zwischen Konzepten und Instanzen ergänzt. Der hier über CLIPS repräsentierte multidimensionale Funktionsraum (das Netz relationaler Slots) ist im Hinblick auf die Einbettung, also Situiertheit der Begriffe in ihrem Kontext, die geeignete Repräsentation.

3.1.6.1 Hinzufügen von Slots und Relationen

Die Slots wurden möglichst den generellsten Konzepten zugeordnet, ihre *domain* also möglichst allgemein gewählt, da so die größte Zahl von Subkonzepten den Slot erbt bzw. implizit über diesen annotiert wird. Verwies ein relationaler Slot auf ein Konzept und außerdem auf ein Subkonzept desselben, so wurde das Subkonzept der *range* entnommen, da dieses ja über das Superkonzept implizit enthalten war (Vermeidung von Redundanz). Es wurde versucht

möglichst viele **intrinsic Slots** zu erstellen, also Slots die Eigenschaften repräsentieren, welche die Instanz aus sich selbst heraus beschreiben (z.B. `has_Ligand` für `Rezeptor`). **Extrinsic Slots**, die Instanzen ohne inneren Bezug lediglich von außen zugewiesen bekommen, sind für den Wissenserwerb und die Wissensmodellierung weniger nützlich. Obwohl das Protégé System extrinsische interne Identifier erstellt, wurden der Ontologie Identifier aus bekannten Datenbanken als extrinsische Slots hinzugefügt, da diese entweder direkt als Parameter für cgi-Skripte dienen, die den Aufruf externer Internetseiten und deren Darstellung im Frame ermöglichen oder als Suchattribute Verwendung finden.

Relationen zwischen zwei Konzepten in beide Richtungen zu speichern wäre redundant, da sich bei symmetrischen Relationen der Wert der **inversen Relation** aus der anderen ergibt. Gibt z.B. der Nutzer für einen "Toll-Like_Receptor"-Instanz-Slot, genannt `ST_Successor`, als Wert die Instanz IRAK ein, so ergibt sich bei dieser Relation automatisch, daß dieser Toll-like Receptor ein `ST_Predecessor` von IRAK ist. Der Wert (die Instanz) der `ST_Predecessor`-Relation wird vom System automatisch erstellt, wenn der inverse `ST_Successor`-Slotwert vom Nutzer erstellt wird. Dem Slot `Annot_Ptrs`, der von den hierarchisierten (selbst annotierten und formalen) Instanzen zu den unter einem Konzept stehenden (importierten und weniger formalen) Annotations-Instanzen zeigt (**pointer**), wurde ein inverser Slot gegenübergestellt, um zu ermöglichen, auch von den Annotations-Instanzen jederzeit auf die korrespondierenden Instanzen in der selbst annotierten Konzept-Hierarchie gelangen zu können. Das ist besonders wichtig, wenn man z.B. im Search-Tab oder Queries & Export-Tab Ergebnisse erhält, die Instanzen des Annotations-Konzepts repräsentieren, und man von diesen aus sich die referenzierende Instanz in der Ontologie und das dieser zugewiesene Gandr-Konzept anschauen will.

3.1.7 Integration vorhandener Ontologien

Das ontologische Strukturieren ist zeitaufwendig. Um den Entwicklungsaufwand zu reduzieren und das ontologische *commitment* zu erhöhen, wurden Begriffe bzw. Konzepte aus folgenden bereits vorhandenen Vokabularen und Ontologien in die Ontologie integriert (*ontology reuse*). Die Integration erfolgte tief, für relevante domänenspezifische Bereiche und eher flach für weniger domänenspezifische Bereiche.

Wortliste der Domänenexperten: Eine Wortliste der Domänenexperten (ZenkeVok), die auf Grundlage zeitintensiver Literaturrecherchen manuell durch Frau Dr. Hacker [41] erstellt und bisher zur Annotation der Affymetrix[®] probe set IDs in Excel[®]-Tabellen genutzt wurde, diente als Lieferant wichtiger Konzepte und Slots, die in jedem Falle in die Ontologie aufzunehmen

waren. Semantische Beziehungen waren in dieser Wortliste nicht definiert, den Begriffen waren keine Eigenschaften zugewiesen, und die Wortliste enthielt Redundanzen und Inkonsistenzen.

Gene Ontology: Aus der Gene Ontology [50] wurden hauptsächlich *top-level*-Module und allgemeinere *upper-level*-Konzepte übernommen (siehe Abschnitt 3.1.5.3). Detailliertere Begriffe wurden lediglich aus den in der Anwendungsdomäne geforderten Themenbereichen "Dendritische Zellen" und "Signaltransduktion" übernommen. Es wurden nur kurze prägnante bzw. intuitive Begriffe übernommen. Einige teilweise satzlangen GO-Begriffe wurden entsprechend transformiert.

EVOC Cell-Type Ontology: eVOC und evokeTM (<http://www.evoontology.org/>) dienen der vereinheitlichten Beschreibung von Gewebe-Expressionsprofilen [51]. EVOC enthält 1007 Begriffe (März 2005), die sich auf die *upper-level*-Module Anatomical System, Cell Type, Development Stage, Experimental Technique, Microarray Platform, Pathology, Pooling, Tissue Preparation und Treatment verteilen. Für die Gandr-Ontologie wurden Konzepte der Module "Cell_in_vivo", "Cell_by_class" und "Cell_by_function" übernommen. Integriert wurde weiter das "Defensive_cell"-Modul inkl. aller Subkonzepte, jedoch ohne "sensu Invertebrata"-Konzepte. Im Konzept "B_lymphocyte" wurden die EVOC-Subkonzepte als Symbol-Slot I_g mit den vier Werte-Symbolen IgE, IgG, IgM, IgA integriert. Aus dem Modul "cell_by_lineage" wurde das "hematopoietic_stem_cell"-Konzept inkl. Subkonzepte integriert.

Transfac TF-Schema: Zur Beschreibung von Transkriptionsfaktoren wurden 49 Begriffe aus der Transfac TF-Klassifizierung als entsprechend strukturierte "Transcription_Factor"-Subkonzepte integriert (<http://www.gene-regulation.com/pub/databases/transfac/cl.html>).

EC-Enzym Klassifikation: Zur Strukturierung der Enzym-Konzepte wurde das *upper-level*-Schema der EC-Enzym Klassifikation (<http://www.chem.qmul.ac.uk/iubmb/enzyme/>) übernommen.

3.1.8 Entkopplung einer molekularbiologischen *upper-level-Ontologie*

Aus der fertigen Gandr-Ontologie wurde eine allgemein nutzbare ***upper-level-Ontologie*** entkoppelt. Hierzu wurden alle speziell auf die Gandr-Nutzergruppe zugeschnittenen KR-Ideome entfernt. Zunächst wurden Konzepte inkl. Subkonzepte über das Classes-Tab gelöscht, wobei bei den über Mehrfachvererbung enthaltenen Subkonzepten darauf geachtet wurde, daß diese erhalten blieben. Zur Auffindung solcher Konzepte erwies sich die Darstellung als Graph im OntoViz-Tab (siehe Abschnitt 3.4.5.1) als sehr hilfreich. Die abgekoppelten nicht mehr benötigten, d.h. nun *domain*-losen, Slots wurden anschließend über das Slot-Tab gelöscht. Hierdurch wurde die Ontologie vom 20.1.05 von 1200 Konzepten und 110 Slots auf 590

Konzepte und 45 Slots verkleinert. Diese wiederverwendbare *upper-level*-Ontologie liegt auf der Gandr-Internetseite zum Herunterladen bereit und kann für die Erstellung eigener molekularbiologischer Ontologien anderer Anwendungsschwerpunkte verwendet werden.

3.2 Beschreibung der Gandr-Ontologie

3.2.1 Grundlegende *top-level*-Module

Die Ontologie enthält verschiedene *top-level*-Module zu unterschiedlichen Domänenbereichen. Entsprechende Konzeptdefinitionen mit Beschreibung der charakteristischen Slots sind der Abb. 7, oder für die gesamte Ontologie der interaktiv introspektierbaren HTML-Klassenarchitektur zu entnehmen (siehe Gandr-Internetseite). Hier die wichtigsten *top-level*-Konzepte und ihre Verwendung in der GandrKB.

Probe_Set_Container: Für dieses Container-Konzept wurden Slots definiert, die Instanzen aller Subkonzepte zueigen sein sollen. Nur innerhalb dieses Konzepts können probe set ID-Instanzen bzw. Gene per *drag and drop* verschoben (annotiert) werden, ohne daß Slotwerte verloren gehen.

Cellular_Compound: Dies ist das wichtigste Konzept zur Annotation. Es wurde *abstract* gesetzt, kann also nicht direkt instantiiert werden, um eine weitergehend charakterisierende Annotation mit funktionell aussagekräftigen Subkonzepten zu erzwingen (siehe Abb. 8). Dieses Konzept enthält das Protein-Konzept, welches die meisten Subkonzepte und Instanzen enthält.

Signal_Description: Cellular_Compound-Instanzen können über Mehrfachvererbung auch mit Signal_Description-Subkonzepten annotiert werden. So kann z.B. das Cellular_Compound-Konzept MHC sowohl unter dem Konzept Surface_Proteine als auch unter dem Signal_Description-Subkonzept Immunological_Role stehen.

Cellular_Localisation: Über dieses Konzept werden mögliche Aufenthaltskompartimente der Genprodukte annotiert. Es wird nicht instantiiert, sondern nur durch Cellular_Compound-Instanzen über deren `present_in`-Slot als Konzept referenziert.

Biological_Process: Dieses Konzept dient der Annotation von Cellular_Compound-Instanzen über deren `starts-` oder `produced_by-` Slots mit Annotationskonzepten, die beschreiben, an welchen Stoffwechselprozessen ein Genprodukt bzw. probe set ID beteiligt sein kann.

Biocarta: Dieses Modul enthält Konzepte zur Annotation von Cellular_Compound-Instanzen mit Verweisen auf Biocarta_Map-Instanzen. Diese zeigen bei bestehender Internetverbindung die Biocarta[®] Pathway-Maps zu verschiedenen Stoffwechsel- und Signaltransduktionswegen

(z.B. TLR_ST oder Hematopoiesis). Man kann seine probe set IDs also mit einer entsprechenden Graphik versehen.

Small_Molecule: Hier sind Beschreibungskonzepte enthalten, die der Erfassung niedermolekularer, nicht-proteinerger Substanzen dienen.

Exogenous_Compound: Dieses Konzept dient der Beschreibung von Substanzen, die nicht endogen im Körper vorhanden sind. Es enthält z.B. exogene Antigene, die für die Aktivierung von TLR-Rezeptoren wichtig sind.

Sample_Origin: Mit den hier enthaltenen Konzepten kann annotiert werden, aus welchen Organismen, aus welchem Körperteil bzw. Zelltyp oder Zelllinie die mit dem Microarray untersuchte Probe stammt. Das hier enthaltene Cell_Type-Konzept enthält z.B. eine Ontologie hämatopoietischer Zellen und ihrer Differenzierungsstadien.

Context_Annotation: In diesem Konzept befinden sich aus Datenbanken importierte Zusatz-Annotationen zu den probe set IDs, die größtenteils aus NetAffx 04.2004 [52] stammen. Hierin sind auch die Gene Ontology-Beschreibungen enthalten. Instanzen dieses Konzepts enthalten Verweise auf externe Internetseiten verschiedener für die Arbeitsgruppe relevanter Datenbanken (Genesymbol, LocusLink, GeneCards und Kegg-Maps).

Array_Experiment: Dieses Konzept enthält Instanzen, die ein konkretes Affymetrix[®] Microarray-Experiment, also die Expressionsstärke eines über die probe set ID identifizierten Gens, beschreiben. Es wurde exemplarisch für die Arbeitsgruppe mit in die GandrKB aufgenommen.

Prow_CD: Für die Arbeitsgruppe relevant sind insbesondere die CD-Moleküle, die als *marker* für Blutzelltypen dienen. Sie wurden aus der PROW-CD Internetseite integriert.

Die letzten drei Konzepte wurden in den Projekteinstellungen als "*hidden*" markiert, um den Nutzer nicht durch allzu viele Konzepte zu verwirren. Die Instanzen dieser Konzepte sind jedoch über relationale inverse Slots von den Instanzen der probe set ID-Hierarchie aus und über die Anfrage-Schnittstelle weiterhin erreichbar.

3.2.2 Taxonomische Organisation und Kontext-Einbettung über relationale Slots

Neben der taxonomischen *is-a*-Relation wurden die *top-level*-Module über relationale Slots zueinander in Beziehung gesetzt. Diese Slots und die sich daraus ergebende Struktur sind folgender Graphik (Abb. 7) zu entnehmen:

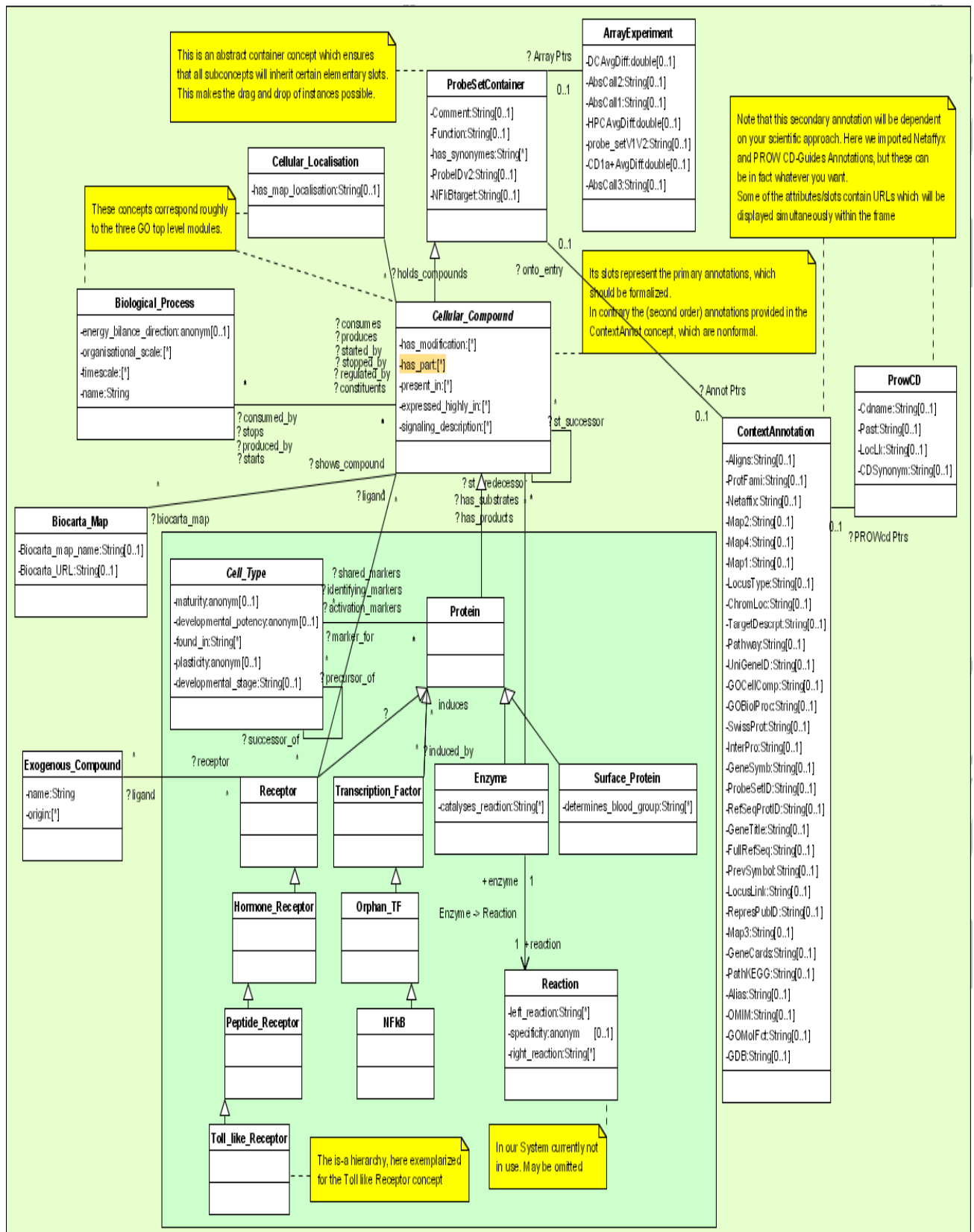


Abb. 7: UML-Klassendiagramm der wichtigsten Gandr-*top-level*-Konzepte und ihrer Slots. Die gelb unterlegten Kästchen (*notes*) beschreiben die Konzepte in ihrem Anwendungskontext. Das unterste Feld der UML-Klassen bleibt leer, da Gandr-Konzepte derzeit keine Methoden erfassen.

3.2.3 Abbildung ontologischer Ideome auf Protégé-GUI und CLIPS-Format

Im folgenden soll das zur Speicherung der Wissensbank verwendete CLIPS-Datenformat, seine Syntax und die graphische Repräsentation der CLIPS-KR-Ideome in der Protégé-GUI an einem konkreten Beispiel erläutert werden. Ein Konzept "Protein" hat in der Ontologie die Slots `has_Synonym` vom Datentyp *string* und `present_in` vom Datentyp "Cellular_Localisation"-Konzept. Für eine konkrete Instanz dieses Konzepts, z.B. der Protein-Instanz NPHS1 mit der probe set ID `31967_at` sind die entsprechenden Slotwerte der *string* "CNF" für den Slot `has_Synonym` und das Konzept "Plasma_Membrane" für den Slot `present_in`.

Konzepte und Slots werden in einer *.pont-Datei (von Protégé ontology) wie folgt gespeichert:

Beispiel Cellular_Compound Konzept:

```
(defclass Cellular_Compound // Konzeptdefinition

  (is-a ProbeSetContainer) // Superkonzept

  (role abstract) // Dieses Konzept kann nicht direkt instanziiert werden

  (multislot st_preceder // Slot-Zuweisung mit Name und Kardinalität (multiple)

;+ (comment "The signal transduction compound that comes before the one in focus in a path.")

  (type INSTANCE) // Slot-Datentyp

;+ (allowed-classes Cellular_Compound) // Als Slotwert erlaubte Konzepte (range)

  (create-accessor read-write))

  (multislot st_successor

;+ (comment " The signal transduction compound that comes after the one in focus in a path.")

  (type INSTANCE)

;+ (allowed-classes Cellular_Compound)

  (create-accessor read-write))

  (multislot consumed_by

  (type INSTANCE)

;+ (allowed-classes Biological_Process)

  (create-accessor read-write))

  (multislot has_modification

  (type SYMBOL)

;+ (allowed-parents Modification)

  (create-accessor read-write))

  (multislot biocarta_map

  (type INSTANCE)

;+ (allowed-classes Biocarta_Map)

  (create-accessor read-write)) ...)
```

Dieses Konzept wird im Classes-Tab der Protégé-2000 GUI als Frame dargestellt (siehe Abb. 8).

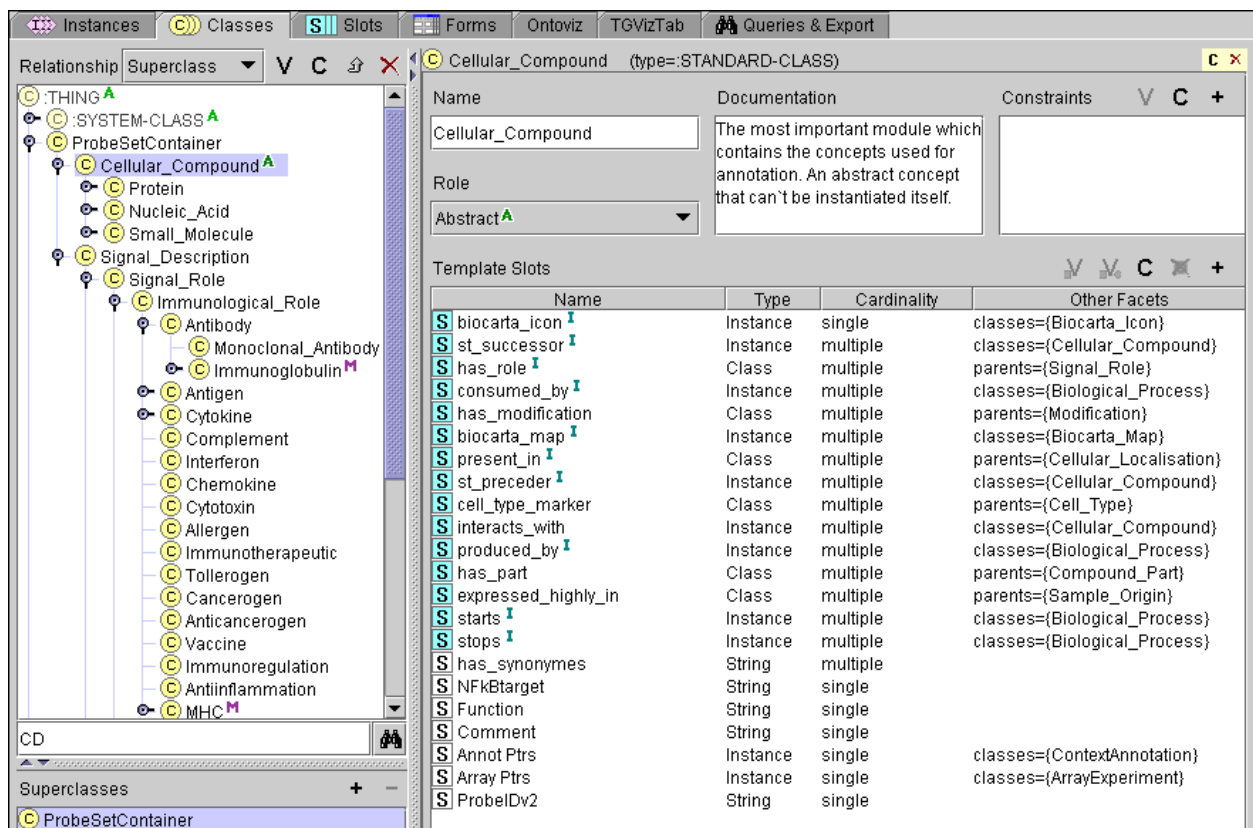


Abb. 8: Das Cellular_Compound-Konzept und die Konzepthierarchie der Ontologie (Taxonomie der Annotationsbegriffe, links) im Classes Tab. Das Cellular_Compound Modul ist markiert und wird rechts als Frame über seine Eigenschaften (Slots und deren facets) genauer beschrieben. Die vom Superkonzept Probe_Set_Container geerbten Slots sind schwarz-weiß, die auf Cellular_Compound-Ebene definierten Slots sind cyan dargestellt. Das "A" am Konzeptnamen kennzeichnet Konzepte, die *abstract* sind bzw. nicht instanziiert werden können; das "M" weist auf Mehrfachvererbung hin.

Die Slots werden zudem als vom Konzept unabhängige Objekte gespeichert. Als Beispiel hier die inversen, d.h. sich gegenseitig bedingenden `st_preceder`- und `st_successor`-Slots zur Beschreibung von Signaltransduktionskaskaden. Als Werte erhalten sie andere Gene bzw. Instanzen anderer Cellular_Compound-Konzepte.

Beispiel Slot `st_successor`:

```
(multislot st_successor // Die multiple Kardinalität gibt an, daß der Slot mehrere Werte erhalten kann
;+ (comment "Contains the next compound in a signal transduction path")
(type INSTANCE)
;+ (allowed-classes Cellular_Compound)
;+ (inverse-slot st_preceder) // Hier wird der komplementäre/inverse Slot angegeben
(create-accessor read-write))
```

Dieser `st_successor`-Slot wird im Slot-Tab der Protégé-GUI als Frame dargestellt (siehe Abb. 9).

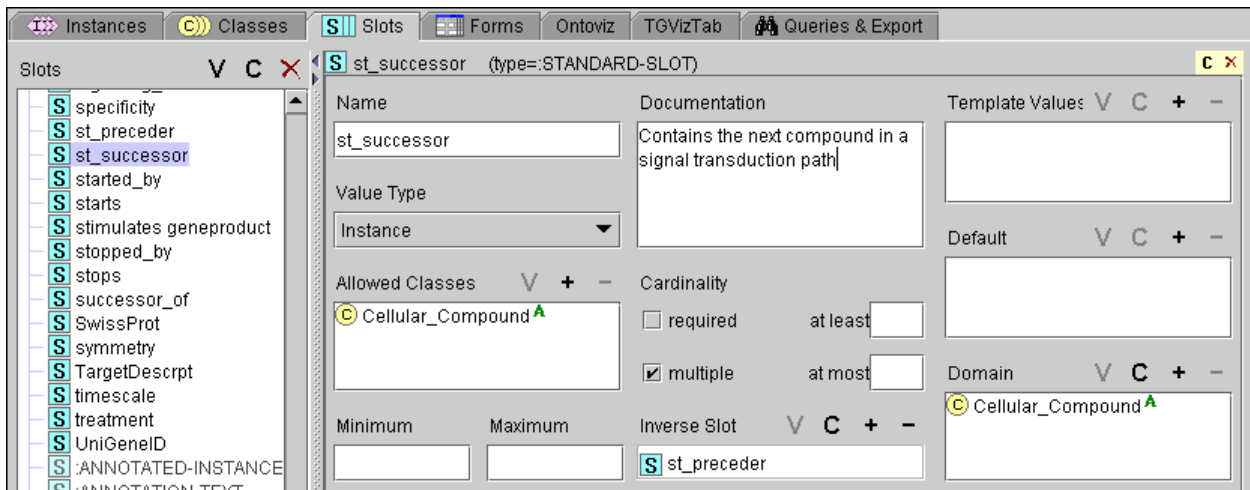


Abb. 9: Darstellung des `st_successor`-Slot als Frame im Slot-Tab. Die rechte Seite zeigt die Slot-facets und ihre entsprechenden Füller / Werte.

Ein `Cellular_Compound`-Subkonzept Protein wird wie folgt in der *.pont Datei definiert:

```
(defclass Protein
  (is-a Cellular_Compound)
  (role concrete)           // Das Konzept kann instanziiert werden (nicht abstract)
  (multislot induced_by     // Über diesen neuen Slot können induzierende Transkriptionsfaktoren
    angegeben werden. Die anderen Slots erbt das Konzept automatisch
    (type INSTANCE)
  ;+ (allowed-classes Transcription_Factor)
    (create-accessor read-write) ...)
```

Beispiel Protein-Instanz (annotierte probe set ID):

Die konkreten über die Ontologie annotierten und strukturierten Daten (z.B. eine Protein-Instanz der probe set ID 31967_at) werden in der *.pins-Datei (von protégé instances) wie folgt gespeichert:

```
([Funct_Instance_1985] of Protein // Interner Instanzname und zugeordnetes Konzept
  (Annot+Ptrs [Annot_Instance_2003]) // Der Annot-Slot enthält als Wert einen Pointer auf eine
  (Context_Annotation-)Instanz
  (Array+Ptrs [Array_Instance_1975]) // Der Array-Slot enthält als Wert einen Pointer auf eine
  (Array_Experiment-)Instanz
  (Function "adhesion") // Einfacher Function-Slotwert vom Datentyp string
  (present_in Plasma_Membrane) // Der present_in-Slot enthält als Wert das Cellular_Localisation-
  Subkonzept Plasma_Membrane
  (ProbeIDv2 "31967_at") ...)
```

Diese mit dem "Protein"-Konzept annotierte Instanz der Probe set ID 31967_at wird in der Protégé-GUI im Instances-Tab als Frame wie folgt dargestellt (siehe Abb. 10).

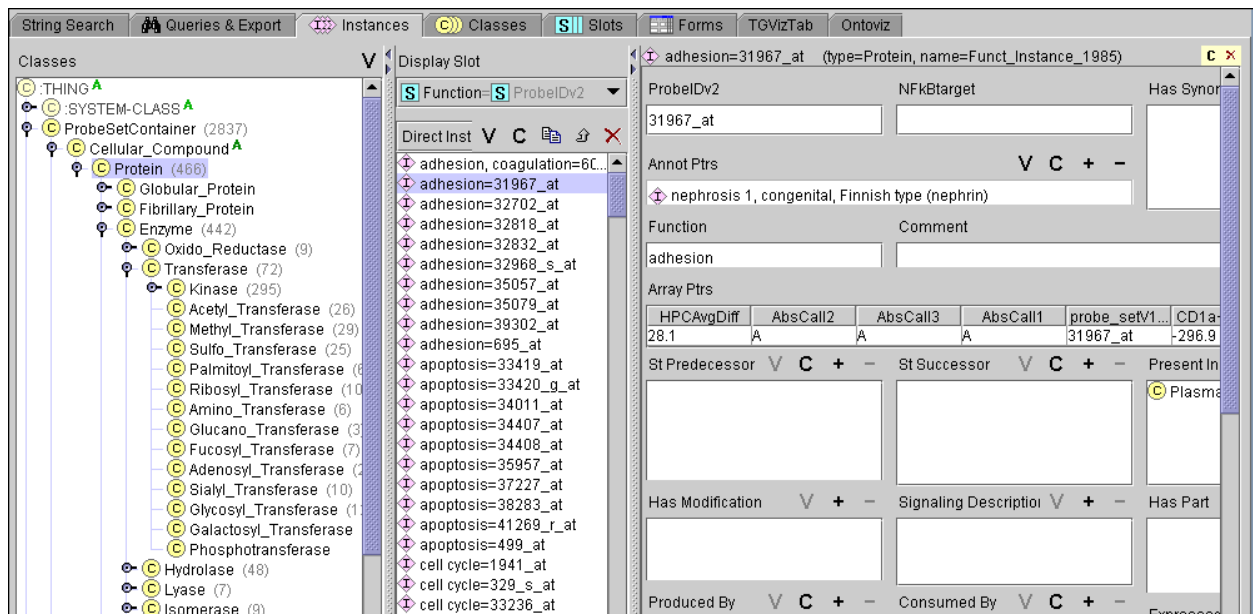


Abb. 10: Darstellung einer mit dem Konzept "Protein" annotierten Instanz 31967_at im Instances Tab. Links ist die Hierarchie der annotierenden Konzepte zu sehen. Hinter den Konzepten in Klammern steht die Anzahl der diesem zugewiesenen bzw. damit annotierten Instanzen. Wird das "Protein"-Konzept in der Hierarchie markiert, so werden die enthaltenen "Protein"-Instanzen als Instanz-Liste dargestellt (Mitte). Die Instanzen werden hierin über ihre Function- und ProbelDv2-Slotwerte sortiert und benannt. Diese selbst wählbaren (Display-)Slots bilden zusammen den browser-key (siehe Abschnitt 3.5.1).

Eine über den "Protein"-Slot `Annot+Ptrs` von `[Funcnt_Instance_1985]` of `Protein` aus referenzierte `Context_Annotation`-Instanz `[Annot_Instance_2003]` wird in der `.pins` Datei folgendermaßen gespeichert:

`([Annot_Instance_2003] of ContextAnnotation // Die Instanzen des Context_Annotation Konzepts beschreiben die probe set IDs genauer über die NetAffx Daten. Diese können dann zur Formalisierung und Erweiterung der Ontologie und Wissensbank verwandt werden.`

```
(Alias "CNF, NPHN")
(Aligns "chr19:41008995-41009285 (-)")
(ChromLoc "19q13.1")
(FullRefSeq "NM_004646 // nephrin")
(GDB "http://gdb.weizmann.ac.il/gdb-bin/genera/acno?accessionNum=GDB:342105")
(GeneCards "http://bioinfo.weizmann.ac.il/cards-bin/carddisp?NPHS1")
(GeneSymb "http://www.gene.ucl.ac.uk/cgi-bin/nomenclature/get_data.pl?match=NPHS1")
(GeneTitle "nephrinosis 1, congenital, Finnish type (nephrin)")
(GOBiolProc "7155 // cell adhesion // traceable author statement /// 7588 // excretion // traceable author statement")
```

```

(GOCellComp "5887 // integral to plasma membrane // traceable author statement")
(GOMolFct "5194 // cell adhesion molecule activity // inferred from electronic annotation")
(InterPro "IPR003961 // Fibronectin, type III /// IPR007110 // Immunoglobulin-like ///
IPR003598 // Immunoglobulin C-2 type /// IPR008957 // Fibronectin, type III-like fold")
(LocusLink "http://www.ncbi.nlm.nih.gov/LocusLink/LocRpt.cgi?l=4868")
(Netaffix "31967_at")
(OMIM "602716")
(ProbeSetID "31967_at")
(RefSeqProtID "NP_004637")
(RepresPublD "AF035835")
(SwissProt "O60500")
(TargetDescript "Cluster Inkl. AF035835:Homo sapiens nephrin (NPHS1) mRNA, complete cds
/cds=(0,3725) /gb=AF035835 /gi=3025698 /ug=Hs.190311 /len=4285")
(UniGeneID "Hs.122186"))

```

3.2.4 Größe und Metrik der Ontologie

Die Ontologie besteht derzeit (Juni 2005) aus 1061 Konzepten und 118 Slots, von denen 35 komplexe Slotwerte haben, d.h. auf andere Konzepte oder über Instanz-Pointer auf andere Instanzen verweisen (relationale Slots). Die restlichen Slots sind primitiven Datentyps. Jedes Konzept hat im Durchschnitt 1.03 Superkonzepte; maximal jedoch drei Superkonzepte. Die größte Anzahl Subkonzepte, die ein Konzept hat, beträgt 46. Jedes Konzept hat im Durchschnitt 16.91, maximal jedoch 32 Slots. Jedes Konzept hat im Durchschnitt 6.32, maximal jedoch 14 relationale Slots. Die Anzahl direkter Instanzen beträgt im Durchschnitt 1.12, maximal jedoch 1072 Instanzen. Jeder Slot hat im Durchschnitt 1,07 maximal jedoch drei Konzepte in seiner Domäne.

3.2.5 Zugänglichkeit und Veröffentlichungsstatus der Ontologie

Die Ontologie wurde frei zugänglich im Internet hinterlegt (siehe Gandr-Internetseite: <http://www.bioinf.mdc-berlin.de/~schober/GandrIntro/>). Der Zugriff auf die Ontologie in der Anwendung erfolgt *client*-basiert über das *open-source* Wissensmanagement-System Protégé-2000. Weiter steht eine auf *Tomcat*-Servertechnologie basierte Internetanwendung des Systems unter der Internetadresse <http://www.bioinf.mdc-berlin.de/> zur Verfügung, welche die komplette Wissensbank über das www Standard-Browsern zugänglich macht (siehe Abschnitt 3.7). In der Web-Version können allerdings nicht die Visualisierungs-Werkzeuge und die graphische Schnittstelle für komplexere Anfragen genutzt werden. Die Ontologie liegt in verschiedenen Repräsentationsformaten (HTML, XML, RDF und XMI) auf der Gandr-Internetseite zum

Herunterladen bereit. Eingesetzt wird die Ontologie primär in der AG Zenke am Institut für Biomedizinische Forschung, Abt. Zellbiologie, am RWTH des Universitätsklinikum Aachen.

3.3 Erstellung und Beschreibung der Gandr-Wissensbank

Die erstellte Ontologie wurde mit domänenspezifischen Gendaten (Instanzen) "gefüllt" und hierdurch zu einer Wissensbank erweitert.

3.3.1 Beschreibung der integrierten Daten (Instanzen)

Über Experten-Interviews wurde festgelegt, welche Daten neben den Expressionswerten in der Wissensbank enthalten sein sollten. Dies waren solche, nach denen in der Wissensbank durch die Anwender im Rahmen ihres Forschungsprojektes häufig gefragt werden würde:

ZenkeVok: Das von der Nutzergruppe ursprünglich zur Annotation von Genen in Tabellen genutzte ansatzweise kontrollierte Vokabular wurde als Wert des `Function-Slots` in die Wissensbank importiert. Diese Information wurde später genutzt, um die Gen-Instanzen entsprechenden formalen Gandr-Konzepten zuzuordnen (siehe Abschnitt 3.3.3).

Affymetrix[®] Chip-Annotation und NetAffx: Umfassende probe set ID-spezifische Informationen bietet die Affymetrix[®] eigene NetAffx Datenbank [52]. NetAffx ist ein Internet-Service, der zu wählbaren GeneChips textuelle probe set ID-Annotationen aus den gängigsten Datenbanken zur Verfügung stellt. Zugang erfolgt entweder über URLs, cgi-Skript oder per *download* der vollständigen Annotationen als Excel[®]-Tabelle.

Die NetAffx[™] Annotationen enthalten statische, extrinsische sowie intrinsische Informationen zu allen probe set IDs auf den Chips. Es wurden nach Möglichkeit intrinsische Informationen integriert, da diese im Gegensatz zu den extrinsischen funktionelle Informationen liefern, die im Verlauf des *ontology refinements* zur genaueren formalen Annotation und dann im Rahmen späterer Abfragen verwandt werden können. Auf extrinsische Informationen (meist Identifier anderer Datenbanken) konnte nicht verzichtet werden, da einige Instanzen über diese als gemeinsame Referenz-IDs (*primary key*) vernetzt wurden. So verbindet z.B. der LocusLink-URL die Instanzen aus dem Konzept `Context_Annotation` mit denen des `Prow_CD`-Konzepts. Aus der Netaffx Annotation vom 09.04.2004 für den Chip HU95Av2 wurden die probe set ID-Informationen folgender Spalten integriert (Informationen zu diesen Daten sind der Affymetrix-Internetseite zu entnehmen):

Probe Set ID, Transcript ID, Target Description, Representative Public ID, Archival UniGene Cluster, Alignments, Overlapping Transcripts, Gene Title, Genesymbol, Chromosomal Location, UniGene ID, Ensembl, LocusLink, SwissProt, EC, OMIM, RefSeq Protein ID, RefSeq Transcript ID, GO Biological Process, GO Cellular Component, GO Molecular Function, Pathway, Protein Families, Protein Domains, InterPro, Trans Membrane.

Genesymbols: Als für die Nutzergruppe wichtige Informationen sollten die Genesymbols des HUGO Nomenklatur Komitee (<http://www.hugo-international.org/hugo/>) in der Wissensbank enthalten sein. Das HUGO Nomenklatur Komitee stellt im Einklang mit den *Guidelines for Human Gene Nomenclature* genehmigte, nicht-redundante Bezeichner für menschliche Gene, die häufig aus Akronymen der vollständigen Gennamen bestehen.

PROW-CD-Guides: Die Nutzergruppe interessiert sich insbesondere für CD-Moleküle, da diese als Marker im Rahmen der Blutzelldifferenzierung eine besondere Rolle spielen. Die Prow-Internetseite (<http://www.sciencegateway.org/resources/cd.htm>) stellt standardisierte Namen und Beschreibungen von humanen CD-Proteinen zur Verfügung. Hieraus wurde eine Excel[®]-Datei extrahiert und mit einem VBA-Makro die Locus-Link-URL hinzugefügt.

Biocarta[®]-Stoffwechsel- und Signaltransduktions-Karten: Zur schnelleren Beurteilung der Funktion einer probe set ID im Rahmen von Stoffwechselwegen, welche die Nutzergruppe besonders interessieren, sind Stoffwechselkarten wie sie Biocarta[®] liefert, besonders nützlich (<http://www.biocarta.com/>). Die Biocarta *pathway-maps* sind aus einheitlich gestalteten *icons* zusammengesetzte, interaktive und einheitlich interpretierbare Stoffwechsel- und Signaltransduktions-Diagramme. Diese Protein-Interaktionsdiagramme sind übersichtlich und besonders für Mediziner und Biologen schnell erfaßbar. Sie werden über die Nutzergemeinschaft im *open-source*-Ansatz ständig weiterentwickelt und aktualisiert. Die in diesen Graphiken visualisierten Komponenten sind mit einer Biocarta[®]-Datenbank verlinkt, so daß Nutzer sich weiter über Literaturreferenzen und Produkte im Zusammenhang mit den visualisierten Genen informieren können. Biocarta[®] *pathway-maps* wurden über Hyperlinks in die Wissensbank integriert. Jede probe set ID kann so über die Relation `biocarta_map` mit externen Biocarta Pathway-Karten annotiert werden.

3.3.1.1 Verknüpfungen zu externen Daten über Hyperlinks

Um einen schnellen Zugang zu weiteren aktuellen Kontext-Informationen zu ermöglichen, wurden Verweise als Hyperlinks (sog. *deeplinks*) auf externe Internetseiten in die Context_Annotation-Instanzen eingebaut. Diese Internetseiten werden dann bei Aufruf der Instanzen über das *URL-widget* direkt im Frame angezeigt. Der Vorteil der *deeplinks* ist, daß die Daten extern gespeichert sind und so die Speichereffizienz verbessert wird. Weiter können *updating*-Probleme vermieden werden, da inhaltliche Aktualisierungen auf den externen Internetseiten keine Veränderung des Zugangs erfordern. Außerdem entfallen dank der Darstellung mehrerer Internetseiten innerhalb eines Frames zusätzliche zu öffnende Browserfenster. Die aktuellen Informationen aus den externen Internetseiten können direkt zur

weiteren Formalisierung bzw. genaueren Annotation der probe set IDs verwendet werden (siehe Abschnitt 3.4.1). Als *deeplinks* wurden in Absprache mit der Nutzergruppe folgende Internetseiten integriert:

Biocarta: http://www.biocarta.com/pathfiles/h_tollPathway.asp

Genesymbol: http://www.gene.ucl.ac.uk/cgi-bin/nomenclature/get_data.pl?match=CYP2C19

Locus Link: <http://www.ncbi.nlm.nih.gov/LocusLink/LocRpt.cgi?l=5595>

Entrez: <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=nucleotide&term=X68149>

KEGGPath: http://www.genome.ad.jp/dbget-bin/show_pathway?MAP00380+1.14.14.1

Die URLs für diese *deeplinks* wurden über VBA-Makros erstellt.

3.3.2 Datenimport mit dem Datagenie-Plugin

Das "Füllen" der Wissensbank mit Instanz-Daten kann auf verschiedene Weisen erfolgen:

- Instanzen können von innerhalb des Systems über sog. *knowledge acquisition* (KA-)Forms, das sind über die Ontologie formatisierte Eingabemasken, manuell und einzeln erstellt (**direkte Instanziierung**) und verändert werden.
- Die Protégé-GUI erlaubt **copy und deep copy von Instanzen**. So können schon vorhandene Instanzen vervielfältigt und dann verändert werden. Das ist nützlich, wenn man viele ähnliche Instanzen, die sich nur marginal unterscheiden, von einer prototypischen *template*-Instanz ableiten will.
- Instanzdaten können von außerhalb des Systems, zum Beispiel als Zeilen einer Tabelle, **importiert** werden. Konzepte und Slots werden dann entsprechend den Tabellen- und Spaltennamen automatisch erzeugt und mit den entsprechenden Daten "gefüllt". Für diesen Hochdurchsatz-Datenimport wird das Datagenie-Plugin genutzt, das tabellarische Daten aus EXCEL[®]-Tabellen und ODBC/JDBC-kompatiblen relationalen Datenbanken (MySQL- oder Access[®]) auf CLIPS-KR-Ideome abbildet und importiert. Über ein XML-Tab-Plugin können **XML-Daten** importiert werden (siehe http://Protege.stanford.edu/plugins/xmltab/xml_tab.html).

Die in Ultrahochdurchsatzexperimenten gewonnenen großen Datenmengen konnten nicht manuell in die Wissensbank eingebunden werden. Die meisten Daten wurden aus zunächst zusammengestellten Tabellen bzw. relationalen Datenbanken importiert. S erfolgte die Instanzzuweisung der durch die Affymetrix[®]-HU95A-Microarrays generierten Expressionsdaten zum Array_Experiment-Konzept und die Instanzzuweisung der Netaffx-Genannotationen zum Context_Annotation-Konzept automatisiert über das Datagenie-Plugin (<http://faculty.washington.edu/gennari/Protege-plugins/DataGenie/index.html>).

Nach Absprache, welche Daten von besonderem Interesse bezüglich der späteren Anwendung in der Nutzergruppe wären, wurden diese für den Datenimport vorbereitet. Die Rohdaten lagen zunächst verteilt in verschiedenen Datenbanken vor. Die zu importierenden Daten wurden in

Gandr-Konzepten entsprechende Excel[®]-Tabellen transformiert. Diese wurden anschließend in eine Access[®] DB importiert und in Relation zueinander gesetzt (ER-Diagramm, siehe Anhang A). Nachdem *primary keys* und die Datentypen für die Tabellenspalten festgelegt wurden, konnten diese Daten, d.h. die gesamte Access[®] DB unter Beibehaltung der Datentypen und Relationen (Instanz-Pointer), über das Datagenie-Plugin importiert bzw. zu Instanzen transformiert werden (siehe Abb. 11).

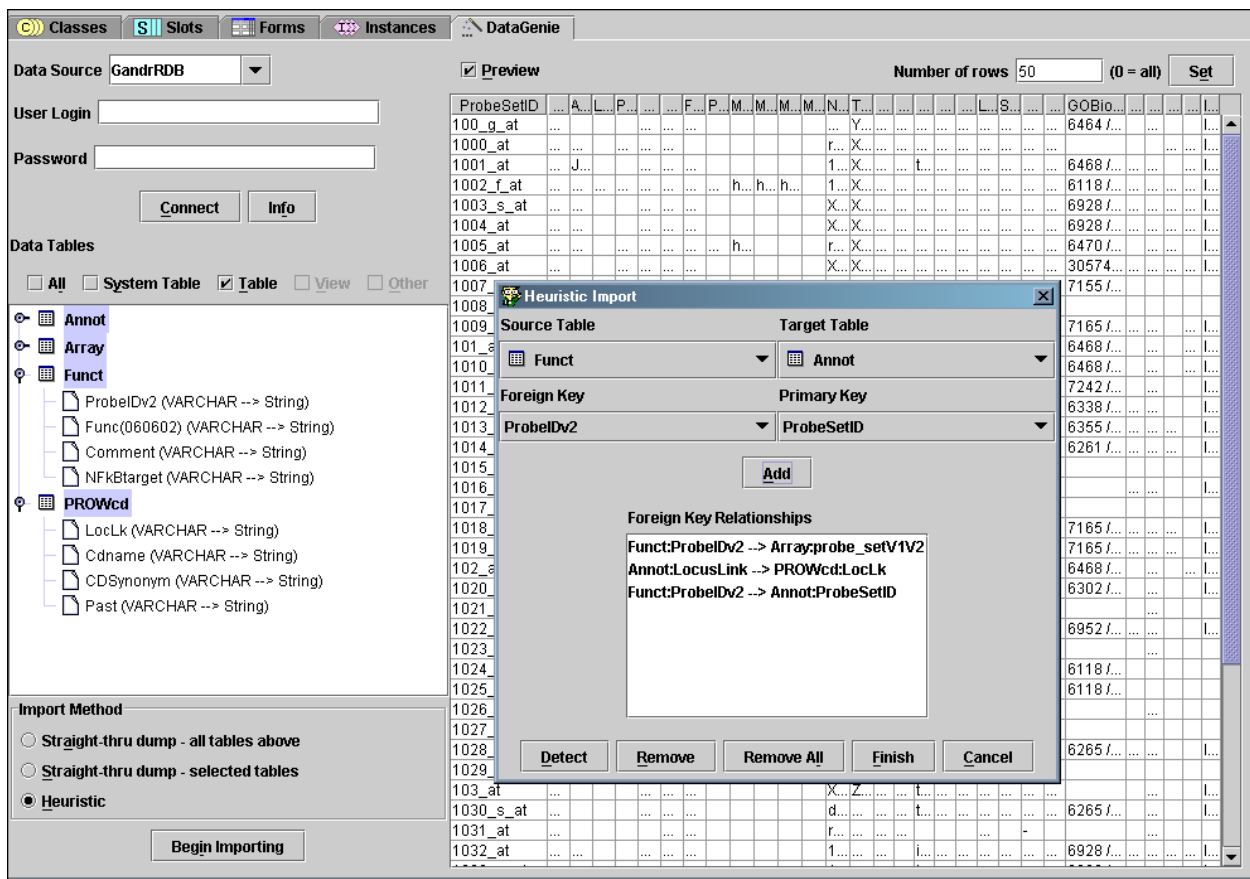


Abb. 11: Der Import der Primärdaten aus einer relationalen Access[®]-Datenbank (ER-Schema, siehe Anhang A.) über das Datagenie-Plugin erfolgt unter Angabe der als Instanz-Pointer zu importierenden *primary*- und *foreign key*-Relationen (Einschubfenster, rechts).

3.3.3 Genannotation durch Verschieben (*drag and drop*) von probe set IDs

Die über das Datagenie-Plugin importierten Genannotations- und Expressionsdaten standen zunächst als Instanzen neu generierter *top-level*-Konzepte bereit. Diese vom Datagenie generierten Konzepte erhielten die gleichen Namen wie die Access[®]-Tabellen aus denen sie importiert wurden. Nach dem Import standen die Instanzen noch in keiner weiteren Hierarchie, waren also nicht mit formalen Gandr-Konzepten annotiert. Um diese Gendaten den ontologischen Konzepten zuzuordnen, wurde folgendermaßen verfahren:

Das durch den Import erzeugte *top-level*-Konzept U95Av2(060602), welches die probe set IDs und die von der Nutzergruppe übernommene funktionale Annotation (Slot `Funktion (060602)`) enthielt, wurde in "Probe_Set_Container" umbenannt und dann alle ontologischen Konzepte, mit denen zukünftig annotiert werden sollte, unter dieses **Container-Konzept** gestellt. Da das Container-Konzept Slots für alle Informationen, die importiert wurden, besitzt und diese Slots an alle Subkonzepte der Gandr-Ontologie vererbt werden, können die nach dem Import ursprünglich alle unter diesem einen *top-level*-Konzept stehenden probe set ID-Instanzen nun per *drag and drop* in alle Subkonzepte des Container-Konzepts verschoben werden, ohne ihre Slots und Relationen zu verlieren. Diese Verschiebung einer Gen-Instanz aus dem Probe_Set_Container-Konzept in ein spezifischeres beschreibendes Gandr-Konzept entspricht dann der formalen **ontologischen Annotation** dieser Gen-Instanz. Dazu wurden die U95Av2(060602) Konzept-Instanzen, die dann Probe_Set_Container-Instanzen darstellten, im Instances-Tab nach dem Slot `Funktion(060602)` sortiert (siehe Abschnitt 3.5.1, *browser key*) und dann per *drag und drop* in "Bündeln" von meist ca. hundert Stück gleichzeitig in das dem gemeinsamen Wert des `Funktion(060602)`-Slot (dem ZenkeVok-Begriff) entsprechende Gandr-Konzept verschoben. Verschiebt man Instanzen per *drag und drop* in Konzepte außerhalb des Probe_Set_Container Konzepts, also in Konzepte, die nicht alle Slots des Ausgangskonzepts haben, so bleiben die betroffenen Slotwerte zwar (bis zum Abspeichern) erhalten, werden jedoch nicht mehr angezeigt. Um diese Slotwerte wieder sichtbar zu machen, fügt man entweder dem neuen, oder wenn möglich einem Superkonzept davon, die entsprechenden Slots hinzu, oder verschiebt das zur Annotation verwandte Konzept unter das Ausgangs- / Container-Konzept, wodurch die benötigten Slots über Vererbung zur Verfügung gestellt werden.

Bei der weiteren Annotation von probe set IDs mit den Konzepten der Gandr-Ontologie konnten nicht nur die Netaffx- und Internetseiten-Informationen aus der Wissensbank genutzt werden, sondern weiter auch analog der Beschreibung in Abschnitt 3.1.5.2 die Text-Konkordanz. Da wichtige Gen-Namen und Gensymbole in der Endnote[®]-Bibliothek der Nutzergruppe auftauchten, stand über die Konkordanz der Wort-Kontext dieser Begriffe zur genaueren konzeptuellen Annotation unmittelbar zur Verfügung.

3.3.4 Größe der Wissensbank, Systemanforderungen und Performanz

Das System läuft dank Implementierung in Java auf allen Plattformen, für die eine Java Virtual Machine ab JDK Version 1.3 existiert. Es läuft also auf fast alle Rechnertypen, insbesondere den verbreiteten 32 Bit Rechnern unter MS Windows (95/98/NT/2000), auf Mac OS X und auf Unix,

Linux und Solaris. GandrKB und Protégé-2000 laufen nicht unter Windows 3.1 und auf Apple Mac "Classic" OS (Mac OS 9.x) Rechnern.

Die gesamte Wissensbank besteht aus 39677 Frames bei einer Dateigröße von 115 KB für die *.pont Datei (Gandr-Ontologie) und 21 593 KB für die *.pins Datei (GandrKB). Als Minimalanforderungen benötigt das GandrKB System einen Rechner mit Pentium IV 2,6 GHz, mindestens 512 MB RAM und mindestens 1 GB Festplattenplatz. Das Laden der Ontologie und Wissensbank benötigt unter WinXP in derartiger Konfiguration 67 Sek. und 35 Sek. für die GUI (abhängig von Projekteinstellungen und Plugin-Konfigurierung). Protégé kann Wissensbanken mit mehr als 150 000 Frames (Konzepten und Instanzen) verwalten. Ab 80 000 Frames wird die Verwendung des *database-backends* empfohlen, da CLIPS-, XML- und RDF-Datei-basierte Protégé-Projekte sehr viel RAM verbrauchen und das Laden so großer Projekte dann mehrere Minuten dauern kann (http://protege.stanford.edu/doc/design/jdbc_backend.html). Das *database-backend* löst diese Probleme, da es über *cacheing* nur die gerade benötigten Frames (dynamisch) lädt.

Die GandrKB ist in der Vollversion sehr groß. Sie wird mit vielen Daten und Funktionalitäten geliefert, die nicht jeder Nutzer benötigen wird. Das Laden des vollen Projektes dauert deshalb relativ lange. Die Ladezeit kann erheblich verkürzt werden, wenn der Nutzer die von ihm nicht benötigte Instanzen und Slotwerte löscht und nicht benötigte Plugins deaktiviert.

3.4 Anwendungen der Gandr-Wissensbank

Die Hauptanwendungen und Vorteile des hier vorgestellten ontologiebasierten Datenmanagements bestehen in der schnelleren Wissensakquirierung und formalen Annotation von Genen mit einem laborspezifischen Fachvokabular. Das steigert die Interoperabilität der Daten und die Konsistenz ihrer Interpretation. Eine Einbindung des funktionalen Kontexts der Gene in ein Annotations-Modell, die fortschrittlichen graphischen Anfrageschnittstellen und interaktiven Visualisierungsmöglichkeiten sind weitere Nutzungsmöglichkeiten der Gandr-Wissensbank (siehe Abb. 12).

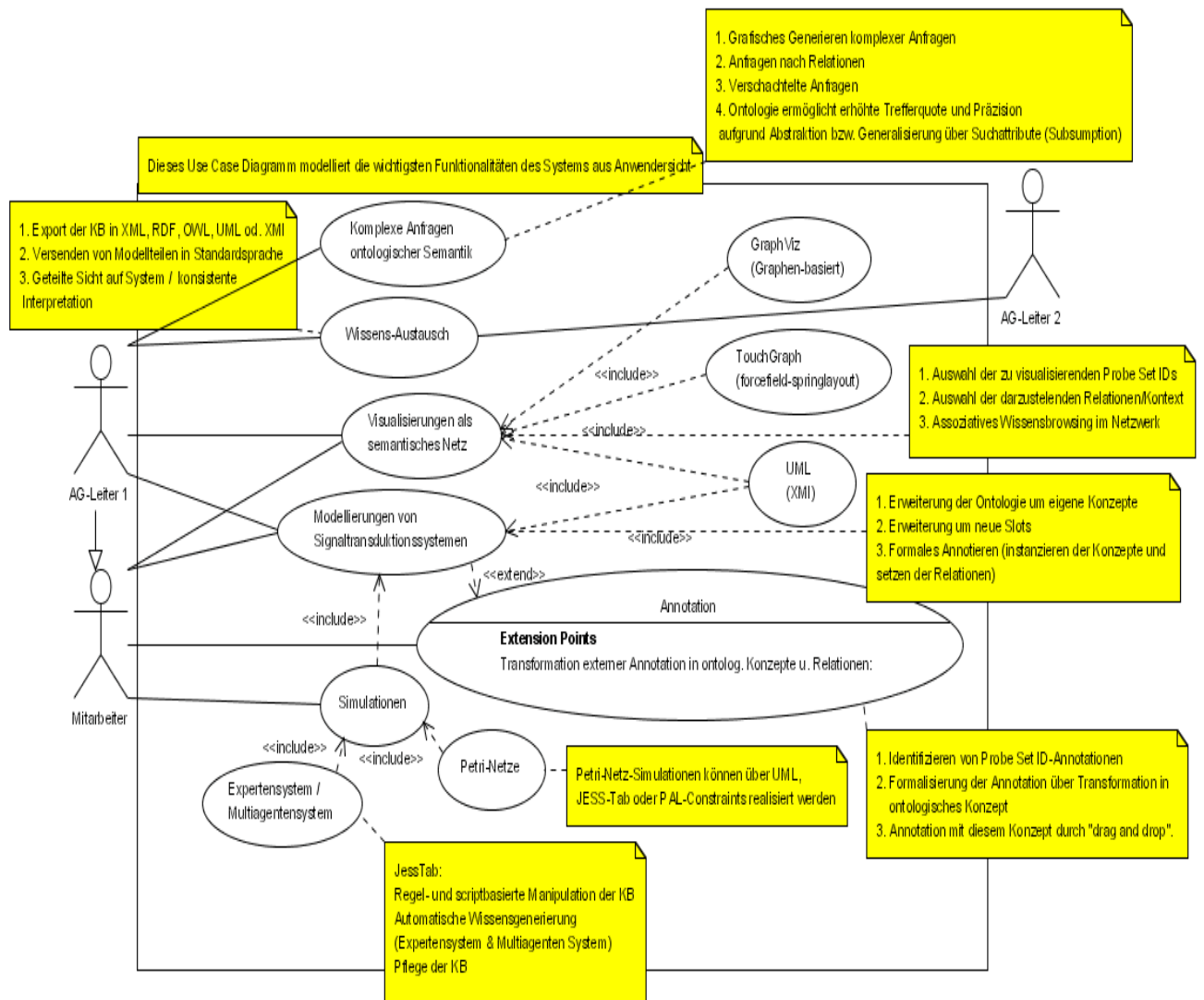


Abb. 12: Die Funktionalitäten bzw. Anwendungsfälle des GandrKB-Systems dargestellt als UML-Anwendungsdiagramm (use case).

Bis auf den Simulations-use case, der im Anhang C.5 erläutert wird, werden diese Anwendungen im folgenden genauer vorgestellt. Eine detaillierte Einführung anhand von weiteren praxisorientierten Beispielen befindet sich im GandrKB User Guide (siehe Anhang D.).

3.4.1 Formale Annotation von probe set IDs unter Nutzung von Mehrfachvererbung

Die ontologische Genannotation erfolgt wie in Abschnitt 3.3.3 beschrieben über einfaches *drag and drop* der probe set ID-Instanzen in ein entsprechend spezifischeres Konzept der Gandr-Annotations-Ontologie. Soll eine probe set ID-Instanz mit mehreren formalen Konzepten gleichzeitig annotiert werden, so erstellt man ein **Schachtelkonzept** für die Instanzen und setzt dieses unter alle weiteren zur Annotation benötigten Konzepte. Diesen Umweg muß man gehen,

da eine Instanz per Definition nur einem Konzept direkt zugeordnet sein kann, also nur Konzepte, nicht Instanzen, direkt unter mehreren Superkonzepten stehen können. Da das Schachtelkonzept lediglich die Weitervererbung der Slots aller Schachtelkonzept-Superkonzepte (Mehrfachvererbung) und die Subsumption unter diese Konzepte ermöglichen soll, braucht man ihm selbst keine neuen Eigenschaften zuzuweisen; es dient selbst nur als Vermittler. Sollen z.B. TRAP-Instanzen sowohl mit dem Konzept "Chaperone", also auch mit dem Konzept "Peptide_Receptor" annotiert werden, so fügt man zur Domäne eines erstellten TRAP-Schachtel-Konzepts das unter dem "Peptide_Receptor"-Konzept steht, einfach das zusätzliche Superkonzept "Chaperone" hinzu. Die Mehrfachvererbung kann dann bei Anfragen genutzt werden, um auch nach implizitem Wissen und Neben-Hierarchiestämmen zu suchen, die nicht direkt über einem bestimmten Anfragekonzept stehen (siehe Abschnitt 3.1.5.6). Die TRAP-Instanz z.B. wird so über Anfragen nach "Chaperonen" als auch nach "Rezeptoren" gefunden.

3.4.2 Integration des Speicher- und Modellierungs-Formates

Wie im Rahmen der **Funktionalen Genomik** gefordert, sollten moderne Wissensmanagement-Werkzeuge das Arbeiten unter einem systemischen bzw. holistischen Paradigma ermöglichen. Die Datenrepräsentation über eine ontologische Semantik und der über die Datenannotation erfolgende schrittweise Auf- und Ausbau der Gandr-Wissensbank ist eine Form der Modellbildung. Der vorgestellte Annotations-Ansatz trennt nicht mehr so strikt wie üblich die Datenspeicherung von der Wissensspeicherung bzw. Modellierung, sondern ermöglicht beides zugleich in einem holistischen Ansatz. Hier werden Systeme (z.B. die im Rahmen der Wissensdomäne untersuchten TLR- / NFkB-Signalwege) über ihre "objektierten" Komponenten (Genen/Instanzen) und deren Beziehungen zueinander (Relationen) kollektiv repräsentiert und dargestellt. Relationale Slots können Interaktionen zwischen Signaltransduktions-Komponenten beschreiben, wie z.B. der `ST_successor`-Slot. Die Abb. 13, 15, 16 und 17 zeigen einen Ausschnitt des hier exemplarisch modellierten Toll-like Receptor Signaltransduktionsweges.

3.4.3 Assoziative und kontextsensitive Navigation

Die Gandr-Benutzeroberfläche ermöglicht interaktive und graphische Strategien, um das in der Wissensbank enthaltene Wissen zu erschließen. Durch Nutzung einer Art "Hyperlink-Paradigmas" wird das schnelle und assoziative Erschließen des unmittelbaren aber auch des mittelbaren Wissens-Kontexts an jedem Punkt der Wissensbank ermöglicht. Inhaltlich verwandte Daten können sofort über ihre semantischen Verknüpfungen bzw. Relationen erschlossen werden. Dabei werden im Instances-Tab über einfaches Anklicken von Slotwerten entsprechende Konzepte oder Instanzen geöffnet und als Frame dargestellt. In diesen Frames stehen wiederum

Relationen bzw. Pointer zu weiteren Frames bereit usw. usf. Instance Tree-Instanzen können so über all ihre Relationen als interaktive Baumstruktur dargestellt werden, was den gesamten relationalen Kontext einer Instanz im Hinblick auf eine bestimmte Interessenperspektive unmittelbar zugänglich macht (siehe Abb. 13).

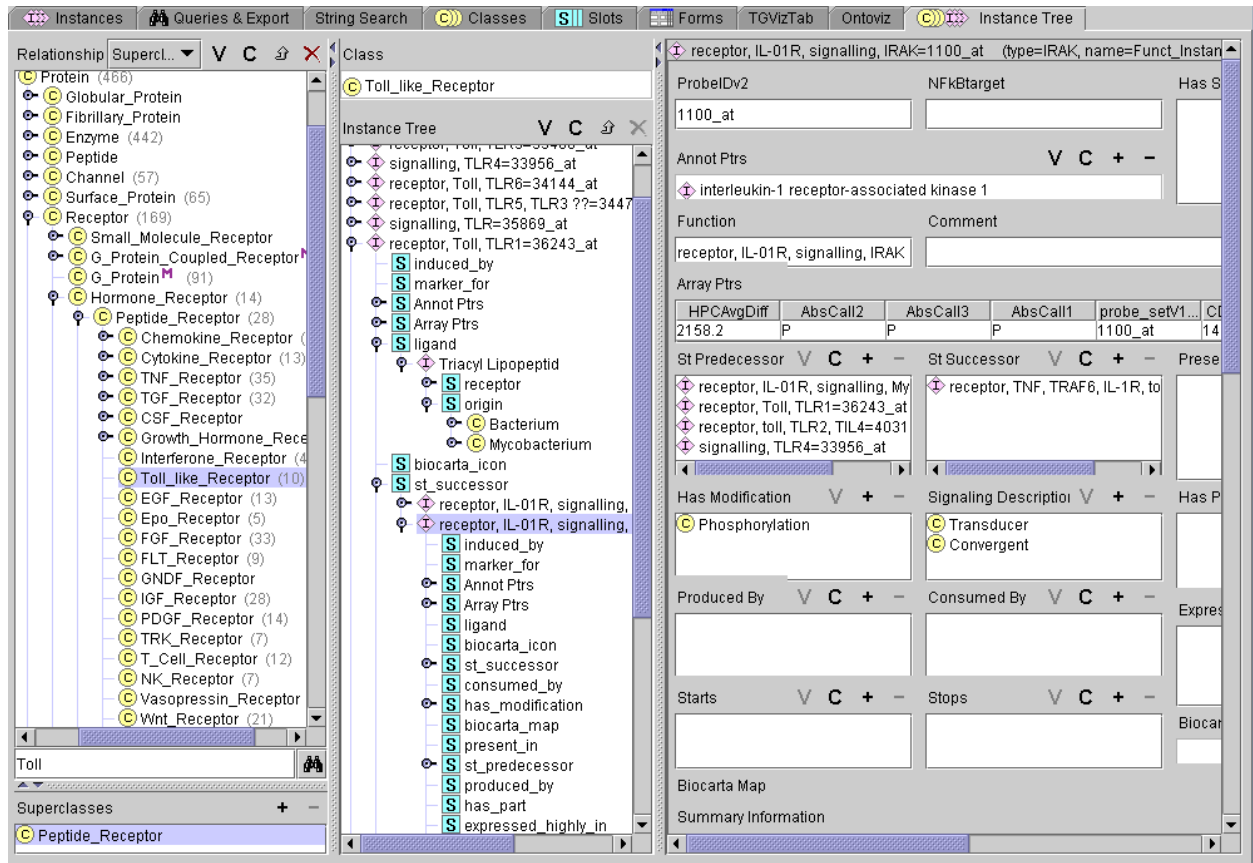


Abb. 13: Die kontextbasierte Darstellung von annotierten Genen über das *Instance Tree Tab*. Über das Menü "Relationship" kann gewählt werden, welche Relation zum Aufbau der hierarchischen Konzept-Struktur genutzt werden soll (links oben). Die Instanzen werden als Instanz-Baum über ihre relationalen Verknüpfungen d.h. ihren unmittelbaren und mittelbaren Kontext zugänglich gemacht (Mitte). Man sieht sofort, daß ein Ligand des TLR1 ein Triacyl Lipopeptid ist (unmittelbarer Kontext), welches aus (Myco-) Bakterien stammt (mittelbarer Kontext). In diesem Beispiel wurde über die Toll-like Receptor Instanz TRL1 das in der ST nachfolgende Glied, IL-01R/IRAK, ausgewählt und diese Instanz als Frame dargestellt (rechts).

Nutzt man Plugins wie Touch Graph oder OntoViz (siehe Abschnitt 3.4.5) wird dieses "Wissensbrowsing" noch komfortabler und intuitiver, da hier nicht mehr in der Hierarchie oder framebasierten Darstellung, sondern direkt in interaktiven Graphen navigiert wird.

3.4.4 Ontologische Informationsextraktion mit dem Queries & Export-Tab

Die Wissensbank soll dem Nutzer einen möglichst umfassenden inhaltlichen Zugang zu seinen Daten ermöglichen. Das Gandr-System erlaubt semantisch komplexe Anfragen an den Datenbestand bei dennoch einfacher und intuitiver Anfrageschnittstelle. Über die Anfragen

selektiert der Nutzer ihn besonders interessierende Teilmengen von Genen oder experimentellen Daten und reduziert so die weiter auszuwertenden Daten im Hinblick auf eine spezielle Fragestellung auf ein jeweils überschaubares Maß. Es kann nicht nur nach Generalisierungen und Spezialisierungen bzw. explizit und implizit annotierten Konzepten gefragt werden kann, sondern auch nach relationalem Wissen, also kontextuellen Bezügen zwischen den annotierten Genen. Komplexe Anfragen können durch Auswahl, logische Verknüpfung und Verschachtelung verschiedener Konzepte und Slots zusammengestellt werden. Das Queries & Export-Tab bietet eine ontologiebasierte Anfrageschnittstelle für die umfassende Informationsrecherche im Datenbestand. Die Vorteile der ontologischen Anfrageschnittstelle des GandrKB-Systems sind folgende:

- **Graphische Anfragegenerierung:** Die Anfragen im Gandr-System werden graphisch aus den vorhandenen ontologischen Elementen erstellt. Dabei werden gegebenenfalls vom System vorgeschlagene KR-Ideome als Suchkriterien ausgewählt und kombiniert.
- **Schnelle Erlernbarkeit:** Die Erstellung von Anfragen im Gandr-System ist aufgrund ihrer Strukturtreue zur menschlichen Sprache recht intuitiv und daher leicht erlernbar. Dabei kann die Anfragesemantik sukzessiv, von einfach zu komplex, erweitert werden. Zunächst werden eher einfache Anfragen nach bestimmten Konzepten gestellt und später, bei genauerer Kenntnis der zugrundeliegenden Semantik, zunehmend komplexere Anfragen.
- **Anfragen nach Relationen und Slotwerten:** Die Ontologie macht den vollen semantischen Kontext zugänglich und ermöglicht so "schärfere" Anfragen, als sie in Tabellenkalkulationsprogrammen möglich sind.
- **Anfragen nach ontologischen Metadaten:** Da auch nach Meta-KR-Ideomen gefragt werden kann, steht dem Nutzer neben dem Zugang zu Instanzdaten der volle Zugang zur Ontologie und zur CLIPS-Semantik zur Verfügung. Hierüber werden Anfragen nach ontologischem Meta-Wissen möglich.
- **Subsumption bzw. Anfragen nach implizitem Wissen:** Die taxonomische Gliederung der Annotations-Konzepte erlaubt die Subsumierung aller annotierten Gene unter Anfragen nach generellen Superkonzepten und Anfragen nach Annotationen auf verschiedenen Abstraktions-Niveaus unter Nutzung von Mehrfachvererbung: Das "Receptor"-Konzept repräsentiert annotierte Instanzen von Proteinen mit Rezeptor-Funktion. Die Ausgabe der direkten "Receptor"-Instanzen entspräche der einfachen Beantwortung der Anfrage nach allen Genen die explizit diesem Konzept zugeordnet sind. Das System ermöglicht jedoch bei einer derartigen Anfrage auch Gene zu finden, die nicht direkt dem "Receptor"-Konzept angehören, sondern diesem lediglich implizit angehören. Dies sind alle Instanzen, die einem in der Hierarchie tiefer stehenden detaillierteren, "Receptor"-Subkonzept zugeordnet wurden, wie z.B. "G_Protein_Coupled_Receptor", "Nuclear_Receptor" oder "Hormone_Receptor".

- **Verschachtelte Anfragen:** Es können verschiedene Anfragen ineinander "verschachtelt" werden. Hierbei baut eine Anfrage auf den Ergebnissen einer anderen auf. Die hierüber formulierbaren Anfragen teils hoher semantischer Komplexität wären über Skriptsprachen relativ schwer zu formulieren.
- **Wissensbasierte Hilfestellung und Überprüfung von Anfragen:** Die ontologiebasierte Anfrageschnittstelle erzwingt die Einhaltung ontologisch formalisierter semantischer Beschränkungen (*constraints*, siehe Abschnitt 2.1.3.3). Die Echtzeit-Verifizierung der gestellten Anfragen auf semantische Rigidität verhindert semantisch sinnlose Anfragen. Das System unterstützt den Nutzer aktiv bei der Formulierung von Anfragen komplexer Semantik und gibt Hilfestellungen, indem im jeweiligen Zusammenhang semantisch mögliche bzw. inhaltlich sinnvolle KR-Ideome über Auswahllisten vorgeschlagen werden. So hilft es z.B. bei der Präzisierung einer Anfrage nach "Kinasen", indem es alle "Kinase"-Subkonzepte ("Protein_Kinase", "ATPase", "Hexo-Kinase", "GTPase", ...) als potentielle Spezialfälle in einer über die Ontologie generierten Auswahlliste vorschlägt. Bei Anfragen nach Eigenschaften schlägt es entsprechend der Ontologie korrekte bzw. inhaltlich sinnvolle Datentypen (primitive Datentypen, passende Instanzen oder Konzepte) vor.
- **Aufbau von Anfragebibliotheken:** Einmal erstellte Anfragen können in einer Anfragebibliothek gespeichert werden. Sie sind dann später erneut einzeln, in Kombination mit anderen Anfragen oder in verschachtelten Anfragen nutzbar. Eine Fragenbibliothek ist ein Fragenkatalog besonders häufig vom Nutzer an die Wissensbank gestellter Anfragen zu den probe set IDs, z.B. Expressionswerten, Gen-Gruppen oder Gen-Eigenschaften. Sie dient hier ferner als Anfragebeispiel-Lieferant im Rahmen der Systemdokumentation.

Hier zwei einfache Anfrage-Beispiele und eins was die Anfrage nach ontologischen Metadaten beinhaltet:

- Zeige mir alle Instanzen (probe set IDs) des Konzepts "Protein", die im Prozeß "NFkB-Signalling" eine Rolle spielen und die Transkriptionsfaktoren sind.
- Zeige mir alle Instanzen des Konzepts "Receptor", deren Expressionswert >1200 ist UND deren Absent Call=P ist UND die sich in der Zellmembran befinden ODER die CDs kodieren.
- Zeige alle Konzepte ohne den Slot `hasLigand`, die nicht instantiiert werden können und deren Konzeptname mit "_Receptor" aufhört.

Ein Beispiel für die Realisierung einer komplexer Anfrage über die Protégé-Anfrageschnittstelle und die exemplarische Anfragebibliothek zeigt die Abb. 14.

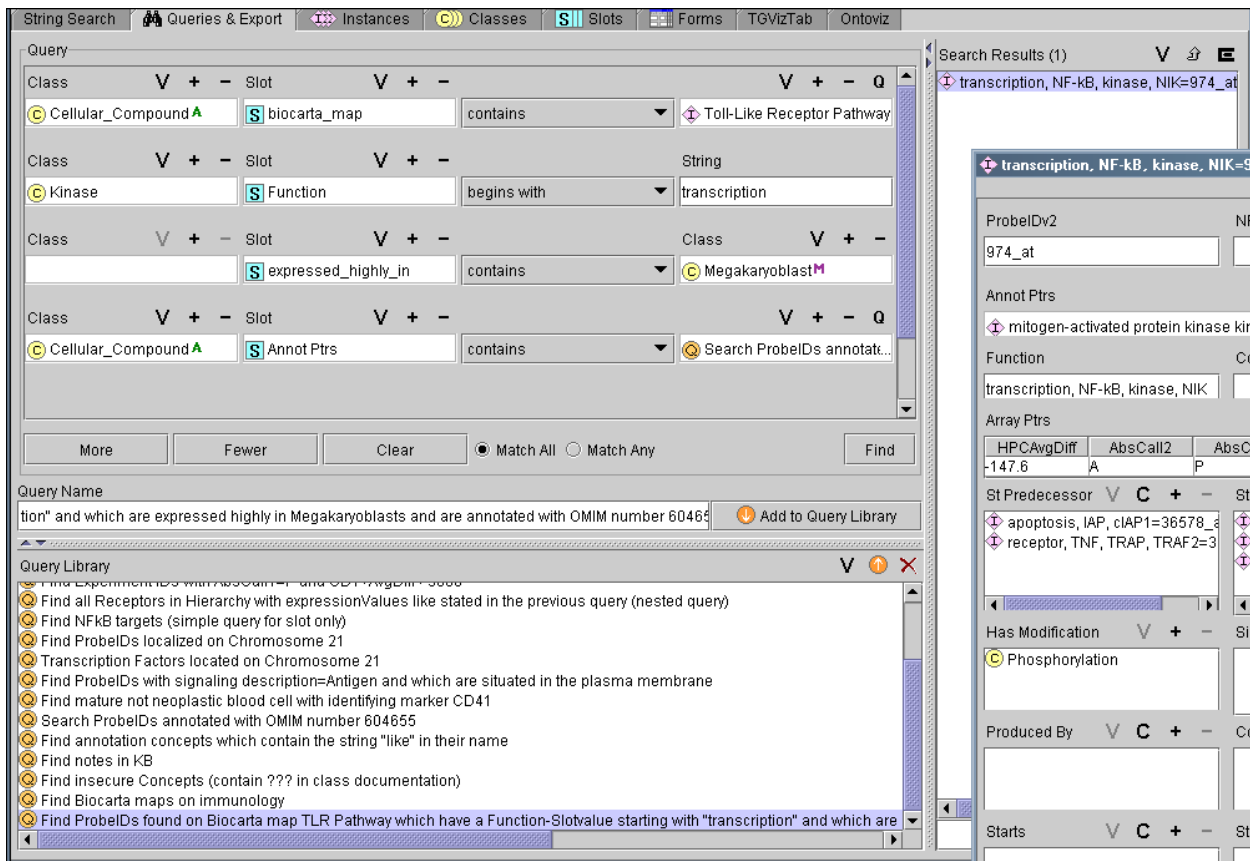


Abb. 14: Die ontologische Anfrageschnittstelle des Gandr-Systems. Hier ist die Anfrage-Syntax einer komplexen Anfrage an die GandrKB (oben links) aus der Anfragebibliothek (unten) im Queries & Export-Tab dargestellt. Oben ist die unten markierte Anfrage in ihrer vollständigen Syntax dargestellt. Die Anfrage ist verschachtelt, greift also auf die Ergebnisse einer weiteren Anfrage der Anfragebibliothek zu. Es werden probe set IDs gesucht, die auf der Biocarta-Map TLR-Pathway stehen, die in der Transkription eine Rolle spielen, in Megakaryoblasten exprimiert werden und deren Annotation die OMIM Nummer 604655 enthält. Eine diesen Suchkriterien entsprechende Instanz ist rechts dargestellt und unten als Frame angezeigt.

3.4.4.1 Export von Anfrageergebnissen

Die Queries & Export-Tab-Schnittstelle erlaubt den Export aller Daten in der Wissensbank und der Anfrageergebnisse zur weiteren Bearbeitung in externer Software. Als Exportformat wurde ein einfaches, als Tabulator-getrennte Text-Datei gespeichertes Tabellenformat gewählt, das direkt in die gängigsten Werkzeuge (Statistik-, Tabellenkalkulationsprogramme, Datenbanken und das Affymetrix-DMT[®]) importiert werden kann. So kann man die GandrKB für das IR nutzen und dann die Möglichkeiten anderer Werkzeuge ausnutzen, um diese Daten weiter zu verarbeiten. Es ist zum Beispiel möglich, sich aus einem Microarray-Experiment zunächst alle Gene mit einer bestimmten Funktion anzeigen zu lassen, zu exportieren und dann mit externen Werkzeugen zum Beispiel über *self organizing maps* auf ähnliche Expressionsprofile zu überprüfen [53]. Man kann auch über die Anfrageschnittstelle generierte probe set ID-Listen in

das Affymetrix DMT[®] importieren und hieraus Filter zur weiteren Bearbeitung erstellen oder Cluster-Analysen durchführen. Man kann sich dann homologe Ergebnisse aus verschiedenen anderen Experimenten zu den probe set ID Listen anzeigen lassen und vergleichen.

3.4.5 Visualisierungen der Wissensbank

Das CLIPS-Repräsentationsformat erlaubt die automatische datengetriebene Erstellung von interaktiven graphischen Darstellungen des relationalen Wissens in der Wissensbank. Die Darstellung erfolgt als semantisches Netzwerk [11] aus einer Menge von Knoten (Begriffseinheiten wie Konzepte oder Instanzen), die durch gerichtete und beschriftete Kanten (relationale Slots) miteinander verbunden sind. Der Vorteil liegt darin, daß ein semantisches Netz die semantische Nähe zwischen Konzepten graphisch widerspiegelt. Inhaltlich ähnliche und Bezug aufeinander nehmende Konzepte liegen nah beieinander, während unterschiedliche Konzepte durch eine Reihe von Zwischenknoten verbunden sind und weiter entfernt liegen. Diese Vernetzung ermöglicht einen besonders schnellen und assoziativen Zugriff auf mit einer Anfrage inhaltlich verwandte Informationen. Anwendungsfälle für vernetzte Visualisierungen sind:

- Visuelles interaktives und assoziatives Browsing des komplexen multidimensionalen ontologischen Raumes. Schnelles Erfassen und Evaluieren der ontologischen Struktur, z.B. durch Nutzer, welche die Ontologie nicht selbst erstellt haben (GraphViz)
- Visualisieren von Instanzen und diese verbindenden Relationen (Nutzung von Mustererkennung zur Datenanalyse in der Wissensbank)
- Visualisieren von Unterschieden zwischen verschiedenen Ontologien oder der evolutiven Veränderungen verschiedener Ontologie-Versionen derselben Ontologie (siehe Abschnitt 3.4.9)

Hierzu werden fortschrittliche Visualisierungs-Werkzeuge genutzt, die als Plugins in das GandrKB System integriert wurden. Die wichtigsten dieser Werkzeuge sind TouchGraph und ATT's GraphViz, die im folgenden am Beispiel der TLR-ST-Visualisierung näher erläutert werden.

3.4.5.1 DAG-basierte Visualisierung mit GraphViz und dem OntoViz-Plugin

Das über das OntoViz-Plugin integrierte Graphik-Layout Werkzeug GraphViz von ATT [54] ermöglicht die Darstellung der Wissensbank als semantisches Netzwerk. Strukturelle Informationen wie Signaltransduktionswege oder Gen-Interaktions-Netze werden nach Auswahl entsprechender Starter-KR-Ideome automatisch als interaktive geometrische Repräsentationen in einem intuitiven graphischen Layout dargestellt. In Verbindung mit der Gandr-Ontologie und Wissensbank kann GraphViz so als *pathway editor* genutzt werden, der - da ontologie-basiert -

"gezwungenermaßen" semantisch sinnvolle Graphen erzeugt. Diese Visualisierung erleichtert das schnelle Auffinden und Erfassen zu analysierender KR-Ideome, indem diese über frei wählbare charakteristische Labels, Farben und Formen und in ihren inhaltlichen Kontext eingebettet dargestellt werden. Der Algorithmus dieses CAD-Werkzeugs erlaubt die Erstellung sehr großer Graphen aus mehreren hundert Knoten unter Vermeidung von verdeckten Knoten oder sich überkreuzenden Kanten. Die generierten Graphen können als GIF-Datei gespeichert oder als DOT-Datei nachträglich verändert und weiterverarbeitet werden (siehe Abb. 15).

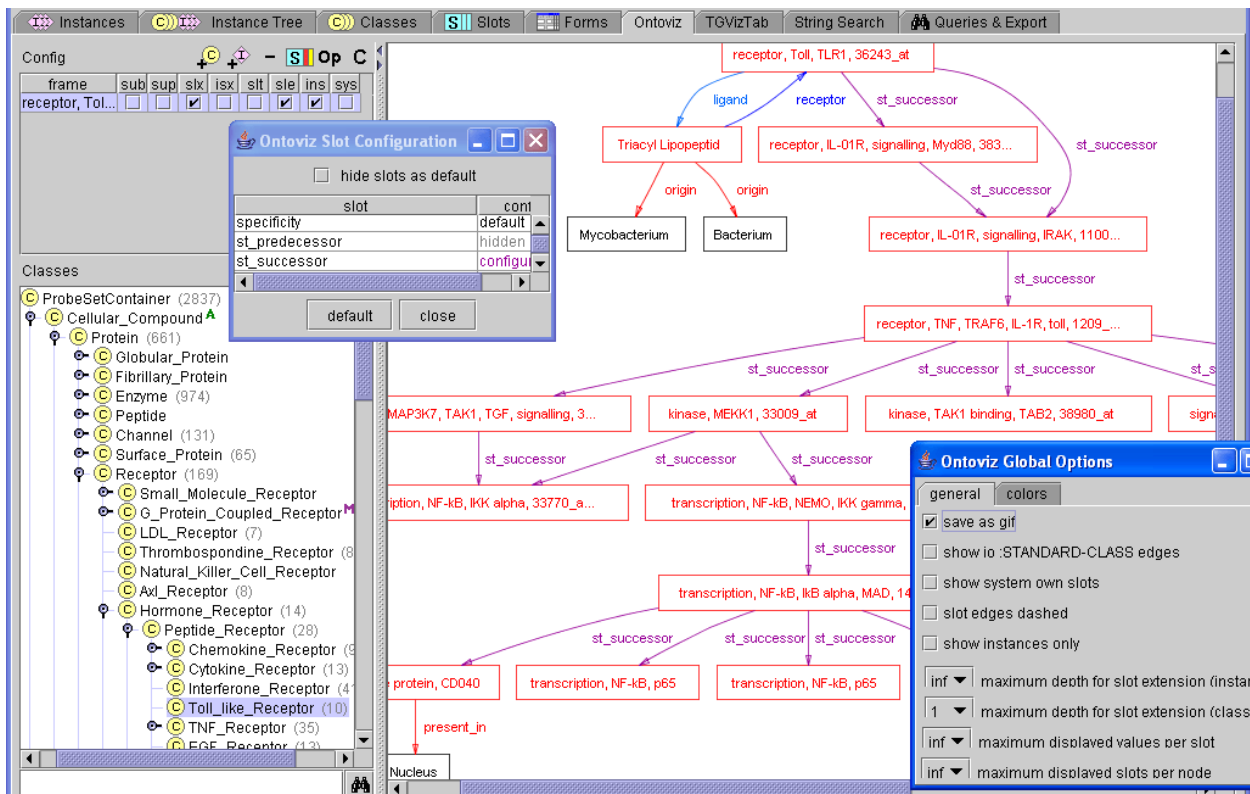


Abb. 15: Mit OntoViz erstellter Graph des TLR-Receptor Pathways, wie er über die Gandr-Annotation in der Wissensbank repräsentiert ist. Ausgangs-Instanz (Starter KR-Ideom, oben links) war die TLR1-Instanz; der volle Pathway wurde davon ausgehend automatisch aus der Wissensbank "erschlossen". Der dargestellte ST-Kontext entspricht in etwa dem von Abb. 16 und 17.

3.4.5.2 Spring-Layout-Visualisierung mit Touch Graph und dem TGViz-Plugin

Ein weiteres Visualisierungswerkzeug, Touch Graph [55], wird über das TGViz-Tab zur Verfügung gestellt (siehe Abb. 16). Es ermöglicht ähnlich dem OntoViz Tab, Teile der Wissensbank graphisch und interaktiv mit Schwerpunkt auf die relationale Vernetzung der Daten zu visualisieren.

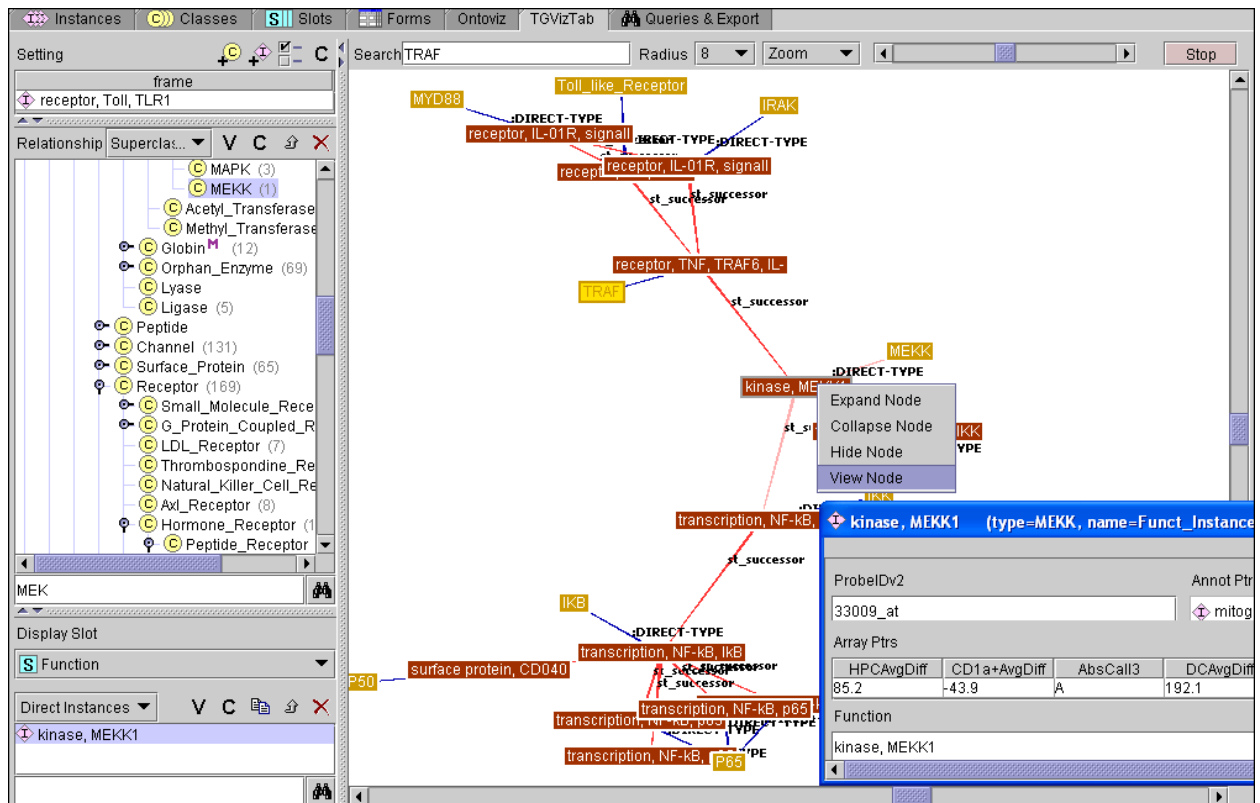


Abb. 16: Netzwerk-basierte (*force field spring layout*)-Visualisierung einer kontextual eingebetteten TLR-Instanz im TGViz-Tab. Dargestellt ist derselbe Toll-like Receptor-ST-Kontext wie in Abb. 15 und 17. Eine MEKK-Instanz wurde über die *view node*-Funktion als Frame geöffnet, um ihre Expressionswerte einzusehen (unten rechts).

3.4.6 Wissensakquisition und Konsistenzprüfung über *constraints*

Im Gegensatz zu Tabellenkalkulationsprogrammen wie Excel[®] assistiert die Protégé Benutzeroberfläche bei der Wissensakquisition bzw. hilft, die Wissensangabe sicherer und dabei komfortabel zu gestalten. Über an den Annotationsgegenstand angepasste, vom System erstellte Eingabeformulare wird die Annotation mit Eigenschaften beschleunigt. Soll z.B. eine neue Instanz erstellt oder eine vorhandene verändert werden, so wird basierend auf dem ontologischen Datenmodell eine entsprechend angepasste Eingabemaske zur ontologisch korrekten Datenerfassung generiert bzw. über das ontologische Datenmodell induziert. Die Eingaben des Nutzers werden nach den in der Ontologie definierten Beschränkungen (*constraints*, z.B. bestimmte Konzepttypen, Datentypen oder Wertebereiche) überprüft. Bei semantischen Fehlern werden formal korrekte Eingaben erzwungen (siehe Abschnitt 2.1.3.3). Durch dieses ständige Überprüfen der Eingaben und Annotationen mit dem ontologisch definierten Domänenmodell wird die Konsistenz der Annotation und der Formalisierung des Wissens verbessert. Es gilt selbstverständlich, daß die zugrunde liegende Ontologie die Qualität

und Konsistenz der Annotation bestimmt, eine inkonsistente Ontologie also zu einer inkonsistenten Wissensbank führen kann.

Die Wissensakquisition wird weiter über die in der Wissensbank selbst enthaltenen Hyperlink-Informationen und die Webbrowser-Funktion des Systems vereinfacht. Der Zugang auf vom Nutzer häufig genutzte Internet-Seiten wird direkt innerhalb des Systems ermöglicht. Dabei bieten die in der Wissensbank enthaltenen *deeplinks* auf externe Internetseiten funktionale Gen-Informationen, die unmittelbar zur Annotation und weiteren ontologischen Formalisierung genutzt werden können.

3.4.7 Wissensaustausch und Ontologieexport

Die Gandr-Ontologie stellt ein gemeinsames Vokabular zur Genannotation bereit. Als Spezifikation einer Fachterminologie vereinheitlicht die Ontologie den gemeinsamen Zugriff auf Labordaten und steigert die Interoperabilität. Dabei sichert die Formalisierung über die Ontologie ein gemeinsames Verständnis der Bedeutung dieser Annotationen zwischen verschiedenen Agenten. Das Annotations-Vokabular kann getrennt von den Daten verschickt und wiederverwendet werden.

Über eine Abbildung von CLIPS- auf andere KR-Ideome kann die Gandr-Ontologie und Wissensbank in verschiedenste KR-Formate wie HTML, XML, UML, RDFS, OWL oder JDBC (SQL-Datenbanken) konvertiert werden (siehe <http://protege.cim3.net/cgi-bin/wiki.pl?ProtegePluginsLibraryByType>). Dies ermöglicht die Nutzung einer Vielzahl weiterer moderner Software-Werkzeuge zur Bearbeitung der Wissensbank.

Ein die Konzept-Hierarchie oder die Wissensbank darstellendes HTML-index file mit Hyperlinks auf HTML-Seiten, welche die ontologischen Konzepte, Slots und gegebenenfalls Instanzen beschreiben, kann automatisch von Protégé aus generiert werden. Über die HTML-Version ist ein universeller Zugang zur Ontologie mit jedem Browser über das Internet möglich (siehe GandrKB Internetseite). Als allgemeiner Standard zur Darstellung objektorientierter Systeme hat sich in den letzten Jahren die *unified modelling language*, UML etabliert. Da UML im Informatik-Studium gelehrt wird und breite Anwendung in der Software-Industrie findet, gibt es ein reichhaltiges Angebot an ausgefeilten Graphik-Werkzeugen zur Bearbeitung und Visualisierung von UML-Modellen. Diese kann man für die graphische Bearbeitung der eigenen Ontologie nutzen. Dank einer Abbildung der CLIPS-KR-Ideome auf homologe UML-Ideome über die UML- und XMI-Export-Backends (<http://Protege.stanford.edu/plugins/uml/> und <http://Protege.stanford.edu/plugins/xmi/>) kann die Gandr-Ontologie als UML-Diagramm oder als XMI-Datei exportiert und so z.B. mit Poseidon CE2[®] weiterbearbeitet werden (siehe Abb. 7,

<http://www.gentleware.com/index.php>). Das XMI-Format der Gandr-Ontologie liegt auf der GandrKB Internetseite zum Herunterladen bereit.

3.4.8 Programmgesteuerte Manipulationen der Wissensbank (JessTab)

Außer der Manipulation durch den Menschen, welche die klassische Nutzung dieser Wissensbank darstellt, kann diese auch programmgesteuert bearbeitet werden. Dies ist besonders bei großen Wissensbanken, wie der vorliegenden, sinnvoll, da zeitintensive und monotone Modifizierungen automatisiert und dadurch beschleunigt werden können. Dank des CLIPS-Formalismus kann man neues Wissen aus vorhandenem Wissen herleiten. Dies kann beispielsweise über regelbasierte Simulationen geschehen, wobei die Wissensbank der formalen Beschreibung der "Ist-Situation" dient. Eine wichtige Voraussetzung hierfür ist die Möglichkeit der Programmierung des Wissensbank-Editors über das JessTab. Das freie Werkzeug *java expert system shell* (Jess, <http://herzberg.ca.sandia.gov/jess/>) stellt eine Java-API und CLIPS-verwandte Skriptsprache, die den Aufbau eines Expertensystems über eine Menge von Regeln (*production rules*) ermöglicht. Diese Regeln überprüfen die Fakten, also "Ist-Situationen" bzw. Zustände in der Wissensbank und können diese aufgrund der definierten Regeln verändern. Danach werden u.U. andere Regeln aktiv, die wiederum die Faktenlage verändern usw., bis eine Faktenlage vorliegt, bei der keine Regeln mehr feuern, das System also in einem Endzustand (einer Problemlösung) angelangt ist. Das JessTab (<http://www.ida.liu.se/~her/JessTab/>) integriert einen Jess-Editor als Plugin in die Protégé-GUI und ermöglicht die Gandr-Wissensbank direkt über eine Makro-ähnliche Skriptsprache oder regelbasiert automatisch zu verändern. Jede selbst zu definierende Regel besteht aus einer *left hand side*, LHS (*IF*-Kondition) und einer *right hand side*, RHS (*THEN*-Aktion). In der LHS werden die Bedingungen bzw. eine Faktenlage in der Wissensbank, welche die Regel zum Feuern bringen soll, als Instanz-Muster definiert. In der RHS wird die Aktion definiert, die beim Feuern der Regel ausgeführt wird. Diese Aktionen sind Funktionen oder Methodenaufrufe wie "*defclass*", "*make-instance*", "*slot-set*", welche die Wissensbank verändern.

Beispiel einer Jess-Skript-basierten Erzeugung und Instantiierung eines Konzepts:

Ein Konzept mit Slots und Slot-Datentypen definieren:

```
(defclass Protein (is-a:THING) (slot swissprotID (type string)) (slot phosphorylations (type integer)))
```

Das Konzept instantiieren:

```
(make-instance APOE of Protein)
```

Slotwert der neuen Instanz setzen:

```
(slot-set APOE swissprotID "A6472")
```

Beispiel für die Definition einer Jess-Regel:

```
(defrule twentyone (object (is-a Protein) (name ?n) (Funktion ?a&:(>= ?a proteolysis))) =>
  (printout t "Das Protein mit Namen=" ?n " ist ein Proteolyse-Protein " crlf))
```

Ausführen der Regel und Ergebnis:

```
Jess> (run)

Das Protein mit Namen= 1191_s_at ist ein Proteolyse-Protein
Das Protein mit Namen= 1614_s_at ist ein Proteolyse-Protein

2 Facts
```

Die folgende Regel setzt alle instanzlosen Konzepte *abstract*. So eine Regel kann nach "Abschluß" der Wissensbank-Entwicklung sinnvoll sein.

```
(defrule setze-konzepte-abstract
  "Setze nicht-instantiierte Konzepte abstract"
  ?c <- (object (:NAME ?n) // LHS (Bedingung bzw. Faktenlage)
          (:ROLE Concrete)
          (:DIRECT-INSTANCES ))
  (not (object (:NAME ?n) (:DIRECT-SUBCLASSES)))
  => (slot-set ?c :ROLE Abstract) // RHS (Auszuführende Aktion)
```

3.4.9 PROMPT *ontology-versioning* und *-merging*

Zur Analyse des Fortschritts während der Ontologie-Erstellung wurde die aktuelle Ontologie von Zeit zu Zeit mit älteren Versionen verglichen. Dabei wurden quantitative und qualitative strukturelle Unterschiede der verschiedenen Ausbaustufen der Ontologie über die PROMPT-Diff-Funktion des PROMPT-Plugins untersucht [56, 57]. PROMPT ist ein semiautomatischer Algorithmus, der den Vergleich, die Analyse und sogar ein Verschmelzen (*ontology merging*) verschiedener Ontologien und Ontologie-Versionen erlaubt (<http://protege.stanford.edu/plugins/prompt/prompt.html>). Wird die Ontologie in anderen Anwendungen oder von verschiedenen Anwendern weiterverwendet und dort verändert, so können sich den Interessenperspektiven der verschiedenen Anwender entsprechend unterschiedliche Versionen der Ontologie entwickeln. Durch PROMPT wird der Nutzer dabei unterstützt, Ähnlichkeiten zwischen verschiedenen Ontologien zu finden und diese in einer Ontologie vereinheitlicht zu übernehmen. Dieser Ansatz eignet sich auch für den Import von KR-Ideomen aus anderen Ontologien. *Ontology merging* und *reuse* können nur bedingt automatisiert durchgeführt werden, da für eine automatische Integration über Konzeptähnlichkeiten den meisten geeigneten Ontologien die Slots fehlen. Lediglich auf Konzeptnamen beruhende Ähnlichkeiten wären semantisch zu wenig definiert und könnten leicht zu fehlerhaften Integrationen führen.

3.5 Anpassungen der Benutzeroberfläche (GUI)

3.5.1 Darstellungsoptimierung über Slot-*widgets* und *browser keys* im Forms-Tab

Bei Erstellung oder Veränderung eines Konzepts wird im Forms-Tab automatisch ein Formular zur framebasierten Darstellung der Instanzen dieses Konzepts erstellt. Die Formulare (**forms**) legen die Gestaltung und Größe der Frame-Komponenten bzw. Slotwert-Darstellungen fest und können eigenen Gestaltungsvorstellungen angepaßt werden. Im Forms-Tab können z.B. die Positionen, wo innerhalb des Frames bestimmte Slotwerte dargestellt werden, und in welcher Weise diese Darstellung im Bezug auf den darzustellenden Datentyp erfolgen soll, eingestellt werden. Ein persönlich gestaltetes Formular beschleunigt das Auffinden und Interpretieren von Informationen und beschleunigt die Dateneingabe bei der Erstellung neuer Instanzen (siehe Abschnitt 3.4.6). Wichtige Slotwerte wurden so z.B. ganz oben im Frame positioniert, so daß sie sofort ins Auge fallen und schnell zugänglich sind, während unwichtigere weiter unten stehen und gegebenenfalls nur durch Scrollen innerhalb des Frames erreichbar sind. Forms bzw. Frame-Einstellungen, die für ein Konzept personalisiert wurden, werden auf alle Subkonzepte und subsumierte Instanzen vererbt, können für diese jedoch "überschrieben", d.h. erneut angepaßt, werden.

Für jedes Konzept ist ferner wählbar, über welche Slotwerte die entsprechenden Instanzen im Instances-Tab bezeichnet und referenziert werden sollen. Das bedeutet, man kann sich die Instanzen eines Konzepts nach verschiedenen Aspekten und Eigenschaften, als Liste sortiert, darstellen lassen. Dieser einstellbare Instanz-Bezeichner wird **browser key** genannt und wird im Forms-Tab für ein Konzept aus seinen Slots ausgewählt bzw. aus mehreren Slots zusammengestellt. Der *browser key* bietet die Möglichkeit, eine Instanz-ID bzw. aussagefähigen Bezeichner aus Instanz-Slotwerten abzuleiten. Als besonders praktisch erweist sich der *browser key*, wenn man Instanzen eines Konzepts nach gemeinsamen Eigenschaften gruppieren bzw. sortieren will, um dann ähnliche Instanzen zusammen entsprechend ihrer gemeinsamen Kriterien per *drag and drop* gemeinsam zu annotieren (siehe Abschnitt 3.3.3). Für die allgemeine Darstellung der Probe_Set_Container-Instanzen wurde hier ein kombinierter *browser key* aus den Slots `ProbeSetID` und `Function` erstellt (siehe Abb.13, mittlere Spalte). In der Instanz-Liste sieht man sofort, ohne die Instanzen als Frame öffnen zu müssen, welche Instanz gemeint ist (`ProbeSetID`), und welche Funktion (`Function`) diese hat.

Im Forms-Tab können für die darzustellenden Slotwerte verschiedene Darstellungsmodalitäten, sog. **Slot-widgets**, ausgewählt werden (siehe Abschnitt 2.1.5). Diese passen die Darstellung der Slotwerte im Instanz-Frame ihrem Datentyp an. So wird z.B. ein Slotwert der in der Ontologie

als *boolean* definiert wurde, automatisch als markierbares Kästchen dargestellt und für den Datentyp *string* entsprechend ein Texteingabefeld erzeugt. Auch eine analoge Darstellung von wertebegrenzten *integer*-Slots durch ein *slider-widget* ist möglich. Das *URL-widget* wurde bereits in Abschnitt 3.3.1.1 beschrieben. Über das *image-widget* können im Frame Bild-Dateien dargestellt werden. Dazu konfiguriert man den entsprechenden Slot als String-Slot und gibt als Slotwert den Bild-Dateinamen an. Dann konfiguriert man den Slot im Forms-Tab als *image-widget* und stellt dort den Pfad zum entsprechenden Bild-Verzeichnis ein. Analog können Ton- und Film-Dateien eingebunden werden. Ist ein Slot relational, d.h. besitzt er als Wert ein Konzept oder eine Instanz, kann die Darstellung über das *contains-widget* so konfiguriert werden, daß nicht bloß der *browser key* der entsprechenden Instanz angezeigt wird ("Durchreichen" des *browser keys*), sondern die referenzierte Instanz mit ihren Werten als Tabelle oder Frame im Frame der referenzierenden Instanz erscheint (siehe Abb. 10).

3.6 Modifizieren und Erweitern der Ontologie

Im Zuge der sukzessiven Erweiterung der Wissensbank zu einem adäquaten Domänenmodell wird der Nutzer die Wissensbank und die zugrundeliegende Ontologie erweitern und seinen laborspezifischen Erfordernissen anpassen müssen. Das System erlaubt dem Nutzer eigenständige Veränderungen in der ontologischen Struktur als Anpassung an den jeweiligen Forschungsstand und Kontext der Wissensdomäne. Beispielsweise kann der Nutzer jederzeit ein fehlendes Konzept oder einen Slot ergänzen oder umbenennen falls ihm der Name in seiner Verwendung als Suchattribut zu lang oder nicht intuitiv erscheint. Alternativ kann der Nutzer ein ihm passender erscheinendes Konzept als neues Super- bzw. *container*-Konzept (ohne eigene Slots) hinzufügen. Sollen Slots in den *facet*-Werten verändert werden, so wählt man den entsprechenden Slot im Slot-Tab aus und kann im Frame für diesen Slot die *facets*, wie Kardinalität, *range*, *domain* und Wertetypen ändern.

3.6.1 Einführung von Slot-Hierarchien

Neben Konzept-Hierarchien können im GandkKB-System auch Slot-Hierarchien erzeugt werden. Auch hier repräsentiert die Struktur eine *is-a*-Beziehung. Slot-Hierarchien haben den Vorteil, daß man über Slots bzw. Eigenschaften generalisieren kann. Die Vorteile sind dieselben wie beim Generalisieren bzw. der Subsumption über Konzept-Hierarchien. Slot-Hierarchien werden auch genauso erstellt. Im Slot-Tab werden ein oder mehrere Slots selektiert und diese dann per *drag and drop* in einen Superslot verschoben. Slot-Hierarchien erlauben beispielsweise die Anfrage nach generellen Eigenschaften unter Subsumierung der hierunter hierarchisierten

spezielleren. Beispielsweise kann man die Slots `GOBiolProc`, `GOMolFunct` und `GOCellComp` alle unter einen generellen Superslot `GO_Description` zusammenfassen und dann über eine Anfrage nach dem generellen `GO_Description`-Slot alle Werte der Slots `GOBiolProc`, `GOMolFunct` und `GOCellComp` zugleich in einer Anfrage überprüfen.

3.6.2 Erweiterung der KR-Semantik (Slothierarchien, Metakonzepete und -slots)

Nicht nur die Ontologie und Wissensbank selber, sondern auch die flexible Datenstruktur der Protégé Repräsentationssprache CLIPS (siehe Abb. 4) kann durch den Nutzer angepaßt, verändert und erweitert werden. Dies geschieht durch Veränderung oder Ergänzung der Protégé Metaklassen-Architektur. Das System erlaubt also die Erweiterung der zur Repräsentation dienenden Semantik. Eigene neue Slot- und Konzept-*facets* können über deren Ableitung von entsprechenden **Metakonzepeten** und **Metaslots** erstellt werden. Will man z.B., daß die Slots, die man erzeugt, zusätzliche *facets* bzw. Eigenschaften haben, so definiert man diese Eigenschaften als neuen Slot für ein neu erstelltes, z.B. "new_Metaslot" genanntes Subkonzept des :STANDARD-SLOT-Metakonzepets (alle Protégé-internen Metaklassen beginnen mit einem Doppelpunkt). Für diesen Metaslot definiert man nun seine *facets* und leitet die zu benutzenden Slots (über *change metaclass*) von dem erstellten Metaslot ab. Die neu definierten Metaslot-Eigenschaften sind dann neue *facets* für die hierüber erzeugten Slots. Will man zusätzliche Eigenschaften für das Grundkonzept :STANDARD-CLASS definieren, so geschieht dies analog, indem von diesem ein Subkonzept erstellt und diesem neue Metaslots zugewiesen werden. Das Superkonzept von allen neuen Klassen, die unter dieser neuen "Konzept-Maske" definiert werden sollen, muß dann von dem neuen Metakonzepet abgeleitet werden. Entscheidet zum Beispiel der Anwender, er möchte zu jedem Gandr-Konzept wissen, zu welchem Gene Ontology-Begriff dieser homolog ist, so kann diese Information als neue Metakonzepet-Eigenschaft definiert werden. Diese erstellt man als Slot des Protégé Metakonzepets :STANDARD-CLASS von dem alle Gandr-Ontologie Konzepte abgeleitet sind. Zu den vorhandenen Slots `Name`, `Comment`, ... kann man z.B. einen neuen Slot `hasGOcorrespondence` des Datentyps *string* definieren und die *constraints* für diesen Slot festlegen. Diese neue Konzept-Eigenschaft, wird dann in allen Gandr-Konzept-Frames als zu füllendes Textfeld erscheinen. Alternativ kann der Slot auch zuerst im Slot-Tab erstellt und ihm als Domäne dann das Metakonzepet :STANDARD-CLASS hinzugefügt werden. Da alle Konzepte der Ontologie von der Protégé Metaklasse :STANDARD-CLASS abgeleitet werden, erbt jedes Gandr-Konzept diesen neuen `hasGOcorrespondence`-Slot und besitzt damit das Potential, entsprechendes neues Wissen formal zu erfassen. Diese zugegebenermaßen etwas komplexen Erweiterungen der

Gandr-Semantik wurden vorerst nicht implementiert, da sie als zu schwer verständlich für die Labor-Endnutzer angesehen wurden.

3.7 Die Wissensbank als Internet-Anwendung: WebGandr

Die Entscheidung, die Wissensbank primär *client*- und nicht generell webbasiert zu implementieren, erfolgte im Hinblick auf den Erhalt der Möglichkeit von Modifizierungen und individuellen Anpassungen durch den Nutzer. Bei der webbasierten Implementierung und Veröffentlichung der Wissensbank sind diese bedarfsorientierten Anpassungen derzeit nicht realisierbar. Um einen vollständigen (*read-only*) Mehrbenutzer-Zugang zur Ontologie und Wissensbank mit gängigen Browsern über das Internet zu ermöglichen, wurde ein Tomcat-Server eingerichtet und Protégé-2000 Version 2.1 hierauf als *webapplication* installiert. Für Anfragen an die Wissensbank steht hier das String Search-Tab zur Verfügung (siehe Abb. 17). Die Gandr Internet-Anwendung kann jederzeit aktualisiert oder um neue Wissensbanken erweitert werden, indem das aktuelle Projekt im Verzeichnis `/local/jakarta-tomcat-5.0.25/webapps/webProtégé/kb/` ersetzt oder eine neue Wissensbank hinzugefügt wird.

The screenshot displays the GandrKB web application. At the top, there is a browser window with the URL <http://krk.bioinf.mdc-berlin.de:8080/webprotege/browse.jsp?kb=GandrKB.ppt>. The application header includes the GandrKB logo, a search bar containing 'Toll-Like', and a 'logged in as: guest | logout' link. Below the header, there are two columns: a left sidebar with a hierarchical tree of concepts (e.g., Biocarta, Map_Adhesion, Map_CellActivation) and a central column with a list of biological concepts such as 'Signal transduction through IL1R', 'T Cell Receptor Signaling Pathway', and 'Toll-Like Receptor Pathway'. The 'Toll-Like Receptor Pathway' is highlighted. To the right, a large, detailed diagram illustrates the TLR signaling pathway, showing extracellular ligands (Lipoproteins, LPS, CpG DNA, Poly(I:c), Imiquimod, PGN) binding to receptors (TLR1/2, TLR2/6, TLR4, TLR3, TLR7, TLR9) and the subsequent intracellular signaling cascade involving adaptors (MYD88, IRAK, TRAF6, TAK1, IKK, NIK, IKKβ, IKKα), kinases (MEK1/2, ERK1/2, JNK1, p38), and transcription factors (NF-κB, AP-1, ELK-1) leading to the production of pro-inflammatory cytokines (IL12, IL1, TNF-α) and outcomes like cell-mediated immunity and bacterial death.

Below the diagram, there is a text box with the following content: "To search a ProbelID or gene-name use the search field (i.e. *MAPK*). To browse through the hierarchy, click on the + in front of the concepts. Most ProbelIDs are found under :THING/ProbeSetContainer/Cellular_Compound/Protein. To see full Annotations click onto the description in the Annot field. Generated by [Protege Web Browser](#) using [Protege](#). Send Feedback and Comments to [Daniel Schober](#)".

Abb. 17: Die GandrKB als Internet-Anwendung (Tomcat-Server *webapplication*). Die TLR-Pathway-Instanz des Biocarta Map_Immunology-Konzepts ist markiert (Mitte) und rechts unter Anwendung des *URL-widgets* als integrierte externe Internetseite im Frame dargestellt. Oben links steht ein Eingabefeld für *string*-basierte Anfragen zur Verfügung.

3.8 Aufbau der Gandr-Internetseite, Schulungsfilm und Dokumentation

Ein Internetportal wurde eingerichtet, wo das GandrKB-System inkl. Plugins vollständig beschrieben ist und all seine Komponenten frei heruntergeladen werden können. Die Internetadresse lautet: <http://www.bioinf.mdc-berlin.de/~schober/GandrIntro/> Neben einer Installationsanleitung und multimedialen Lehr- und Schulungs-Unterlagen zum System werden hier in entsprechenden Tonfilmen alle GandrKB-Anwendungsmöglichkeiten erläutert und an einfachen Beispielen vorgeführt.

Auf das GandrKB-Portal wird vom Protégé-2000-Server aus verwiesen:

<http://protege.cim3.net/cgi-bin/wiki.pl?ProtegeOntologiesLibrary>

4 Diskussion

4.1 Lexikalische Eigenschaften molekularbiologischer Terminologien

Die Gandr-Ontologie formalisiert molekularbiologische Begriffe, deren Bedeutungen, entsprechend dem molekularen Maßstab der beschriebenen Komponenten, abstrakt und für den Menschen oft nicht intuitiv verständlich sind. Keine der über diese Terminologie zu beschreibenden Dinge und Sachverhalte kann der Mensch direkt über seine Sinne erfahren. Was Komponenten und Prozesse z.B. die Signaltransduktion angeht, haben wir weder ein adäquates Vokabular aus Nomina, Verben und Adjektiven, noch liegen diese in formaler Form strukturiert vor. Die existierenden Begriffsbildungen sind, wie in kaum einem anderen Sachgebiet, Neubildungen oder bestehen aus oft drastisch vereinfachenden, alltagssprachlichen Entlehnungen. Medizinische Texte bestehen zu ca. 95% aus normaler Alltagssprache und nur zu ca. 5% aus medizinspezifischen Nomina und Adjektiven wie z.B. "immunostain", "bioreduce", "xenograft" (persönliche Kommunikation, A. Zamora, NLM, 21.6.2003). Die umgangssprachlichen Begriffe sind oft mehrdeutig und lösen - da nach dem hermeneutischen Prinzip [58] der Empfänger die Bedeutung einer Aussage bestimmt - oft unadäquate Assoziationen beim Empfänger aus. Je nach Anwendungskontext und vorhandenem Wissen des Empfängers können durch denselben Begriff verschiedene Bedeutungen und Assoziationen erzeugt werden. Eine ontologisch definierte Terminologien kann dazu beitragen, derartige Kommunikationsprobleme zu reduzieren, da die Bedeutung ihrer Begrifflichkeit auch über ihren relationalen Kontext erschlossen werden kann. Die Molekularbiologie ist hierfür ein gutes Anwendungsgebiet, da hier komplexe interne Strukturen vorliegen. Ihre Komponenten sind über verschiedenartige Relationen zu Systemen und Subsystemen vernetzt [59]. Diesen Gegebenheiten tragen Objektorientiertheit, komplexe Datentypen, Vernetzung ontologischer Ideome untereinander und graphische Visualisierung des beschriebenen Wissens im vorgestellten Ontologie-System Rechnung.

4.2 Neurokognitive Grundlagen der Wissensakquisition

Neues "Wissen" wird durch Erlernen oder durch Deduktion aufgrund intern vorhandenen Wissens erworben. Beide Prozesse können durch Anwendung moderner informatischer Methoden wie der vorgestellten ontologischen Wissensrepräsentation unterstützt und bedingt automatisiert werden.

4.2.1 Strukturtreue und Kontext erhöhen Interpretationsgeschwindigkeit

Aus der Kognitionspsychologie ist bekannt, daß neue Informationen nicht einfach seriell im Gedächtnis gespeichert, sondern zunächst mit bereits vorhandenem Wissen, also einem eigenen Gedankenmodell des Sachverhalts, abgeglichen werden. Dieser **Interpretation** genannte Vorgang geschieht um so schneller, je ähnlicher die Information in bereits vorhandenen internen Modellteilen ist. Existiert eine gute Abbildung der aufzunehmenden Information zu eigenen sprachlichen Begriffen und Relationen, so werden die Informationen schneller aufgenommen, weiterverarbeitet und verstanden. Soll eine ontologische Repräsentation die Aufnahme von Wissen beschleunigen, so muß eine intuitive Abbildung ontologischer Konzepte auf Ideome des vorhandenen Gedankenmodells gewährleistet werden. Aus diesem Grunde wurden möglichst für den Domänenspezialisten intuitive Beschreibungstermini als Annotations-Konzepte formalisiert und dem Nutzer die Möglichkeit der Anpassung, d.h. späteren Umbenennung von KR-Ideomen nach eigenen Vorstellungen, gegeben.

Auch der gebotene **Kontext** der Daten beeinflusst interne Modellbildungs- und Verifikationsprozesse. So steigen Aufnahmekapazität und Erinnerungsvermögen mit der Ähnlichkeit der gebotenen Informationsstruktur zu vorhandenen internen Repräsentationen stark an [60]. Man merkt sich Daten leichter und erfaßt sie schneller im Kontext. Den großen Einfluß der lexikalischen Organisation von Daten auf die Verarbeitung von Wörtern und Sätzen demonstriert die **Priming-Methode** der Psycholinguistik. Diese weist nach, daß wenn einem zu interpretierenden Begriff ein zusätzlicher Begriff vorangestellt wird, das Erfassen des ersteren proportional zur semantischen Nähe zu letzterem beschleunigt wird. Das Wort "Hund" z.B. wird schneller erkannt, wenn zuvor ein semantisch verwandtes Wort aus dem selben Kontext, z.B. "Katze" dargeboten wird. Die semantische Verwandtschaft zwischen dem zunächst dargebotenen *prime*-Konzept und dem *target*-Konzept "bahnt" das Erkennen des Zielbegriffes über einen *spreading activation* genannten Prozeß [61]. Die interne Modellbildung ist um so schneller und genauer, je mehr schon vorhandene Entitäten bei der Interpretation aktiviert werden bzw. je umfangreicher der dargebotene Kontext ist [62]. Die neuronale Aktivierungsausbreitung geht wie eben erläutert vom semantischen Netzwerk aus, in dem assoziierte Begriffe miteinander verknüpft sind. Es werden also über Assoziationen semantisch verwandte Begriffe aktiviert, die ihrerseits wiederum neue Begriffe ins Gedächtnis rufen und gegebenenfalls neue Betrachtungs-Perspektiven zu einem Sachverhalt ermöglichen. Relativ freies Assoziieren über den Kontext ist eine Grundlage der **Kreativität** und wird durch das GandrKB-System besonders gefördert. Schnelle unbewußte Entscheidungen auf Grundlage vieler parallel abgewogener

Assoziationspfade bezeichnet man als **Intuition**. Das GandrKB-System stellt eben diese vielen parallelen Assoziationen explizit dar und erleichtert so ein intuitives kreatives Arbeiten. Wenn bei der internen Modellverifikation ein Gedächtnisinhalt nicht abgerufen werden kann, wird oft unbewußt aus mit ihm zusammenhängenden Sachverhalten auf dieses Zielfaktum geschlossen. Gandr bietet *priming*-Konzepte, die für derartige **Inferenzen** genutzt werden können.

Die Strukturtreue der über die GandrKB aufgebauten semantischen Assoziationsnetze zum neuronalen Korrelat der Modellbildung und das assoziative *priming* der über den Kontext gebotenen Konzepte erleichtern und fördern so die interne Modellbildung und -verifizierung.

4.2.2 Festigung und Erweiterung des Wissensmodells

Wird Information als schon intern repräsentiert erkannt, so wird sie zwar als **redundant** verworfen, bleibt jedoch nicht ohne Auswirkungen auf das gedankliche Modell. Redundante Informationen führen zu einer Aktivierung entsprechender neuronaler Pfade und verstärken das neuronale Korrelat dieser Begriffs- oder Relations-Repräsentation [63]. Auf der Ebene der Ontologie entspricht diese Redundanz dem wiederholten Heranziehen eines KR-Ideoms zur Annotation bzw. Instantiierung. Je häufiger mit einem Konzept annotiert wird, desto sicherer ist seine Legitimation. Paßt gebotene Information widerspruchsfrei in das gedankliche Modell und erweitert dieses, ist also nicht redundant, so wird die Information eingebaut, und wir haben neues Wissen gewonnen. Auf ontologischer Ebene entspricht diese Wissensakquisition der Erweiterung der Ontologie um neue Konzepte, Slots und Instanzen. Je widerspruchsfreier und einfacher neue Information in das gedankliche Modell oder entsprechend in die Ontologie eingefügt werden kann, desto "sicherer" ist sie.

Es kann sich aber auch aufgrund der neuen Informationen das interne Modell ändern. Dies ist der Fall, wenn man der neuen Information und dem neuen Modell oder von anderen übernommenen Modellteilen größere Wahrscheinlichkeit bzw. Richtigkeit beimißt als dem eigenen Modell. Unser Gehirn wägt jederzeit zwischen Ablehnung der Information aufgrund von Widersprüchen zum internen Modell und der Abänderung und Anpassung des internen Modells zwecks widerspruchsfreier Integration als sicher beurteilter neuer Information ab. Diese mentale Offenheit bzw. Bereitschaft zur Revidierung des Wissensmodells, muß sich in der Möglichkeit, die Ontologie neuen Erkenntnissen anzupassen und zu verändern, niederschlagen und trägt dem Eingeständnis Rechnung, daß Wissen insbesondere auf der von Fachleuten diskutierten Ebene - also auf der *leaf*-Ebene der Ontologie - nicht statisch, sondern dynamisch ist. Die hier verwandte Definition des Ontologie-Begriffs [15] macht keine Aussage über die Größe der Gruppe, die nötig ist, um ein Konzept bzw. Begriff zu legitimieren. Sie kann anfangs klein sein, da neues

Wissen zunächst immer wenig anerkannt und akzeptiert ist. Sind die damit erstellten Modelle überzeugend und brauchbar, wird die Gruppe größer und das Wissen bzw. KR-Ideom zunehmend anerkannt.

4.3 Anlehnung an *ontology engineering*-Methodologien (ONIONS)

Da Ontologien in der praktischen Anwendung ein recht junges Forschungsgebiet darstellen und die Ontologiedefinitionen im Bezug auf die repräsentierbare Semantik stark divergieren, gibt es kaum allgemein akzeptierte Standards. Auch die aufgrund der Interdisziplinarität der Entwicklungsbereiche der Ontologien heterogene Terminologie zur Bezeichnung ontologischer KR-Ideome wirkt sich auf die Ausbildung definierter Standards hemmend aus. Dieser Mangel an exakten Richtlinien und Methoden erschwert die Entwicklung einheitlicher Ontologien und in Folge deren Vergleichbarkeit, Wiederverwendung und Integration in andere Anwendungen. Bei der Anwendung und Umsetzung von *ontology engineering*-Methodologien kann man also nicht immer streng verfahren, da sie im Rahmen der Erstellung teils sehr spezifischer Ontologien bzw. Anwendungen entwickelt wurden oder sich auf unterschiedliche Ontologiedefinitionen stützen. Obwohl in letzter Zeit einige formale *ontology engineering*-Methodologien vorgestellt wurden [11, 64], nutzen viele Gruppen daher allgemeinere Leitlinien zur Entwicklung objektorientierter Software-Systeme. Verschiedene Methodologien wurden hier jedoch als Ideengeber betrachtet und Teilprozesse daraus angepaßt und in den eigenen Entwicklungsprozeß integriert. Hier wurden besonders Methodologien für thematisch und anwendungsbezogen verwandte Ontologien evaluiert. Der *ontology engineering*-Ansatz *ontological integration of naive sources*, ONIONS [64] definiert Richtlinien zur konzeptionellen Analyse und Zusammenführung bestehender Terminologien. Er wurde in Betracht gezogen, da er sich in langer Tradition (seit 1993) bei der integrierten Modellierung von Bioontologien bewährt hat. Folgende Punkte der ONIONS *domain analysis* wurden, sofern sie sich nicht auf axiomatisierungs- bzw. beschreibungslogische Strukturierungsaspekte bezogen, für das *Gandr-ontology engineering* übernommen:

1. Terminologische Quellensammlung.
2. Ontologie Architektur-Design bzw. Untersuchung der Anwendungsdomäne und Erstellung eines ersten *top-level*-Modul-Graphen.
3. Weitere Quellenanalyse: Begriffs-, Relations-, Synonym-Sammlung und textuelle Beschreibung dieser KR-Ideome.
4. Begriffsextraktion aus domänenspezifischen Quelltexten.

5. Import ontologischer Ideome bzw. Übersetzung in eigene formale Repräsentation. Wichtige KR-Ideome erhielten textuell dokumentierende Definitionen.
6. Formulierung der *core (upper-level) ontology* unter Einbeziehung bestehender Ontologien: Hier mit eigenem *top-level*-Schema als *core axiom schemata* der *Gandr-core ontology* und ohne Foundational ontology (siehe Abschnitt 4.5.1). Dem *core axiom schemata* entspricht die *Gandr-top-level*-Ontologie (Abb. 7).
7. Terminologische Analyse: Die Untersuchung textueller Beschreibungen über linguistische Regeln entfiel.
8. Einteilung der Begriffe in entsprechende KR-Ideome (Konzepte, Slots und Relationen).
9. Vertikale Integration: Einordnung der KR-Ideome unter die *top-level*-Konzepte des *core axiom schemata* und der *core ontology*. In der Praxis hier das Verschieben der Konzepte in die korrekten annotierenden Konzepte.

4.4 Probleme bei der Erstellung von Ontologien

4.4.1 Kommunikation mit den Experten und Wissensakquisition

Erfolgen die Ontologie-Erstellung und die Wissensbank-Nutzung durch verschiedene Personen, ist also der Wissensingenieur nicht der Endnutzer, so muß das zu formalisierende Wissen vom Domänenspezialisten auf den Wissensingenieur übertragen werden. Voraussetzung hierfür ist eine intensive und ausführliche Kommunikation zwischen beiden, wobei zu beachten ist, daß Wissensingenieure normalerweise mit dem Vokabular der Experten nicht vertraut und umgekehrt die wenigsten Experten mit der Terminologie des Wissensmanagements vertraut sind. Es kommt zu Problemen, wie sie bereits in Abschnitt 1.3.1 als ontologisch lösbar erwähnt wurden. Ein "Flaschenhals" der Wissensakquisition besteht darin, daß ein Großteil der Expertise der Experten implizit, als sog. *tacit knowledge*, vorhanden ist und dieses bei der Kommunikation im Rahmen der Wissensakquise oft ausgeklammert wird. Diese impliziten Grundvoraussetzung jedoch fehlen dem domänenfremden Wissensingenieur. Daher wurde in periodisch abgehaltenen *expert-interviews* und über *e-mail* ständig Kontakt zu den Experten gehalten und deren Wissen über detaillierte Befragungen einbezogen bzw. in Rücksprachen verifiziert. Das letzte *ontology refinement* und die Erweiterung um unsicheres Wissen obliegt dem Endnutzer.

4.4.2 Taxonomisierungs-Probleme

Bei der Taxonomisierung der Konzepte tauchten Probleme auf, die im folgenden in Anlehnung an die Klassifizierung nach Jones und Paton [59] erörtert werden. Eine detaillierte ontologische Fehlertyp-Klassifizierung gibt auch Hoekstra [65].

1. Atypische Subkonzepte oder Instanzen: Unklar zu klassifizierende KR-Ideome wurden nach ihrem wahrscheinlichsten Abfragekonzept klassifiziert. Es wurde darauf Wert gelegt, atypische

KR-Ideome nicht aufgrund einer nicht ausreichenden bzw. repräsentativen Menge ihrer Slots bzw. Slotwerte kontraintuitiv zu klassifizieren. So werden z.B. ausdifferenzierte Erythrozyten unter "Blood_Cell" und nicht unter "Prokaryotic_Cell" klassifiziert, obwohl sie die Prokaryonten-Eigenschaft "hat_keinen_Zellkern" haben.

2. Mehrfach-Geschwister-Instantiierung: Besitzt ein Konzept Charakteristika mehrerer Superkonzepte, fällt eine Klassifikation oft schwer. Kann ein Konzept als Subkonzept mehrerer Geschwister-Konzepte dieses Konzepts positioniert werden, so ist es meist atypisch im Sinne 1. für mindestens ein Superkonzept. Ein Konzept "Neuroendocrine_Cell" z.B. kann unter "Endocrine_Cell" und unter "Neuronal_Cell" stehen. Da eine Unterscheidung nach sortierenden und nicht-sortierenden Konzept-Eigenschaften nach Strawson [66] in der Molekularbiologie nicht immer leicht möglich ist, wurden bei unsicherer Klassifizierung in einer pragmatischen Lösung Konzepte und Instanzen einfach allgemeiner, d.h. dem Superkonzept zugeordnet, für das die Klassifizierung noch klar ausfiel.
3. Kontextabhängige Konzeptzugehörigkeit: Im Bezug auf verschiedene Abfrage-Kontexte kann ein Konzept oder eine Instanz unter verschiedene Konzepte eingeordnet werden. In einem bestimmten Anfragekontext kann ein Konzept richtig formalisiert sein, in einem anderen falsch.
4. Ausgeschlossene Zyklisierung: Ein Konzept kann kein Subkonzept seiner Subkonzepte sein. Diese Art von Fehler wird durch das Protégé-2000 System in Echtzeit entdeckt und so verhindert.
5. Trügerische Konzeptähnlichkeiten: Eine Konzeptdefinition über ihre Eigenschaften kann in mehreren Dimensionen ähnlich der Definition eines eigentlich unpassenden Konzepts sein, was zu fehlerhaften Konzeptpositionierungen führen kann. Ein Beispiel wäre die Fehlklassifizierung von Mitochondrien als Bakterien aufgrund der vielen Gemeinsamkeiten in den Eigenschaften. Derartige Ähnlichkeiten können auf evolutionäre Verwandtschaft hindeuten.

Ein bei Jones nicht genanntes Problem sind wiederholte Slot-Zuweisungen. Dabei weist man einen Slot, der einem Konzept schon zugewiesen wurde, einem Subkonzept noch einmal explizit zu. Die genannten Probleme wurden hier teilweise in Abweichung von Jones neu bezeichnet, da Jones eine in unserem Zusammenhang verwirrende Terminologie zur Beschreibung der KR-Ideome nutzt. So bezeichnet er das, was hier als Konzepte und Instanzen bezeichnet wird, kollektiv als Instanzen; die Instanzen unserer Terminologie bezeichnet er als *individuals*.

4.4.2.1 Konzept oder Instanz, Subkonzept oder Slot

Die Entscheidung, ob Gendaten als Konzept oder als Instanz repräsentiert werden, erfolgte in Abhängigkeit des Anwendungskontextes bzw. der in diesem Bereich geforderten Granulいた der Wissensbank. Für den Mediziner z.B., der sich nur nebenbei für den Lipidstoffwechsel interessiert, kann die Formulierung eines LDL-Rezeptor-Gens als Instanz eines allgemeineren Konzepts ausreichen, während der Fettstoffwechsel-Genetiker das Konzept "LDL-Rezeptor-

Gen" als recht grobes Konzept empfinden muß, da er mit vielen sein Detailwissen reflektierenden Subkonzepten des "LDL-Rezeptor-Gen"-Konzepts vertraut ist, wie z.B. *splice*-, SNP- und Methylierungsformen. Manchmal werden KR-Ideome, die einmal als Instanz formalisiert wurden, später mit detaillierterem Wissen als Konzept reformalisiert.

Neue Eigenschaften können entweder als neues Subkonzept formalisiert werden oder über neue Slots, die einem vorhandenen Konzept hinzugefügt werden. Bei Neurotransmitter-Rezeptoren z.B. kann man die Existenz eines Dopamin-Rezeptors einmal als Subkonzept von "G_Protein_Coupled_Receptor" mit Namen "Dopamin_Rezeptor" formalisieren oder als "G_Protein_Coupled_Receptor"-Instanz, in dem ein Slot `ligand` mit dem Wert "Dopamin" gefüllt ist. Letztere Alternative wird man wählen, wenn die Liganden bereits als Instanzen z.B. eines "Small_Molecule"-Konzepts vorliegen.

4.4.2.2 Konzept oder Instanz als Slotwert

Bei der Formalisierung der Slot-Eigenschaften, der sog. *facets*, sind die potentiellen Füller der Slots bzw. Slot-Wertetypen zu definieren. Repräsentiert ein Slot eine Relation, so muß definiert werden, ob der Slotwert eine Instanz oder ein Konzept repräsentiert. Diese Entscheidung hängt u.a. davon ab, mit wieviel Zeitaufwand die Annotation später ausgeführt werden soll und wie detailliert die Wissensbank in diesem Bereich werden soll. Definiert man den Slotwert der Relation als Konzept, kann er im Laufe der Annotation sehr schnell mit einem entsprechenden Konzept unterhalb des als potentiellen Wertes definierten Konzeptbereichs (der *range*) gefüllt werden. Definiert man den möglichen Slotwert als Instanz eines Konzepts und muß diese bei Füllung des Slotwertes aus gegebenenfalls sehr vielen Instanzen ausgewählt, oder neu erstellt werden. Dabei ist der Zeitaufwand wesentlich größer, da die zur Instanz gehörigen Slotwerte erst gefüllt werden müssen, bevor auf diese verwiesen werden kann.

4.4.2.3 Repräsentation von Transformationen und graduellen Zustandsübergängen

Eine adäquate Beschreibung molekularbiologischer Komponenten sollte auch Zustandsübergänge derselben erfassen können. Werden Proteine, z.B. durch Chaperone umgefaltet, dann wechseln sie u.U. die qualitative Kategorie/Klasse bzw. das Konzept, da ein anders gefaltetes Protein gänzlich andere Eigenschaften haben kann als das ursprüngliche Protein in nativer Konformation. Vom ontologischen Standpunkt kann man dann sagen, die Instanz `Healthy1` des Konzepts "Healthy_Enzym" hat aufgehört zu existieren, und eine neue Instanz `Prion1` des Konzepts "Toxic_Protein" ist entstanden. Das Material ist dasselbe, die SwissProt ID auch; beide haben dasselbe Gen, dieselbe AMS Sequenz, aber das funktional annotierende

Konzept muß verändert werden. Ähnlich radikalen Einfluß auf die Konzeptionierung können auch Eigenschaften des umgebenden Mediums, wie pH-Wert oder Temperatur, haben. Probleme bereiten auch Konzeptionalisierungen komplexer bzw. zusammengesetzter Substanzen, also Konglomerate und die sich aufgrund dieser neuen Zusammensetzung ergebenden synergetischen neuen Eigenschaften. So erfüllen einige Signalproteine ihre Funktion erst, wenn sie in einem Signalkomplex höherer Ordnung, dem Signalosom, situiert vorliegen. Eine adäquate Repräsentation derartiger Konglomerate und ihrer emergenten Eigenschaften setzt komplexe Partonomien voraus, die derzeit in den wenigsten Ontologien implementiert sind und aufgrund fehlender Detailkenntnisse gegenwärtig auch nicht formuliert werden können.

Problematisch ist auch die Modellierung von graduellen Unterschieden, also von Verläufen, die sich in Unschärfen in den Annotationen äußern: Ist ein Oligonukleotid, -peptid oder -lipid ein Mikro- oder schon ein Makromolekül? Im Bereich der Entwicklung dendritischer Zellen wandeln sich nach Zugabe von Wachstumsfaktoren Zellarten allmählich und kontinuierlich ineinander um. Ein Prozeßmodell, das wie eine *black box* Input- zu Output-Konzept-Instanzen transformiert, ist für die Modellierung gradueller Konzept- bzw. Klassenübergänge unzureichend. Symbolische Repräsentationen, wie die des GandrKB-Ansatzes, sollten nur für diskrete Zustandsbeschreibungen verwendet werden.

4.4.2.4 Kontextwandel, Synonyme, Redundanz und taxonomische Inkonsistenzen

Die Taxonomisierung kann sich bei verschobenem Anwendungskontext verändern. Hat der Nutzer bei der Formalisierung des Konzepts "Receptor" einmal eine konkrete Molekülklasse im Sinn und in einem anderen Kontext eine abstrakte Molekül-Funktion, so besteht die Gefahr, daß derselbe Begriff - oder ein Synonym davon - unbemerkt unter verschiedenen *upper-level*-Konzepten, z.B. einmal unter "Cellular_Compound" und ein anderes mal unter "Molecular_Function", formalisiert wird. Semantisch eng verwandte, oft synonym verwendete Konzepte, können also in einer Ontologie manchmal an entfernten Positionen stehen (z.B. "DNA" *is-a* "Physical_Object". "DNA-Sequence" *is-a* "Abstract_Object"). Dadurch kann es zu Redundanzen und letztlich zu Unschärfen in der Ontologie kommen.

Oft lassen sich aus den Namen gewisse Rückschlüsse auf das Konzept ziehen. Dies trifft insbesondere auf systematisch zusammengesetzte Konzeptnamen zu. Man darauf achten, die KR-Ideom-Benennung syntaktisch konsistent und definiert zu gestalten: Der Name der Rezeptor-Konzepte beispielsweise setzt sich nach Möglichkeit aus dem Liganden gefolgt vom *string* "Receptor" zusammen. Leider gibt es auch hier Ausnahmen, z.B. bei dem Namen "G_Protein_Coupled_Receptor", für den sich das "G_Protein" nicht auf den Liganden, sondern

auf das nachgeschaltete Signalmolekül bezieht. Ein anderes Beispiel ist "Nuclear_Receptor", für den sich das "Nuclear" auf den primären Aufenthaltsort des Rezeptors und nicht auf seinen Liganden bezieht. In diesen Fällen wurde der Nutzbarkeit und Intuitivität bzw. "Gängigkeit" des Konzeptnamens höhere Priorität eingeräumt als der syntaktischen Konsistenz der Begriffsbezeichnung (siehe Abschnitt 3.1.5.1). Die ständige sukzessive Erweiterung und Reformulierung von Ontologieteilen kann zu unbemerkten Inkonsistenzen führen [67]. Wenn z.B. ein Slot eines Konzepts gelöscht wird, welches Instanzen hat, so verlieren nach dem Abspeichern der Wissensbank alle Instanzen dieses Konzepts und alle Subkonzepte desselben ihre Slotwerte. Des weiteren können Flüchtigkeitsfehler der Form auftreten, daß man "Receptor" als Subkonzept von "Surface_Molecule" formalisiert und damit implizit auch unpassende Subkonzepte, wie "Nuclear_Receptor", unter das "Surface_Molecule"-Konzept fehlklassifiziert. Solche Fehler sind besonders tückisch, da schwer zu entdecken. Generell kann man fragen, inwieweit z.B. ein "Receptor"-Konzept überhaupt sinnvoll ist, da im Stoffwechsel letztendlich alle Moleküle, die mit anderen interagieren, Rezeptoren darstellen, und sei es nur im Hinblick auf angefügte chemische Gruppen oder Elektronen.

4.4.2.5 Gleiche Detailliertheit bei Geschwisterkonzepten

Im Hinblick auf die Traktabilität der KR-Ideome sollten alle direkten Subkonzepte eines Konzepts, also alle Geschwisterkonzepte, eine ähnliche Detailliertheit aufweisen bzw. sich auf gleichem Abstrahierungsgrad befinden [68]. Die Formalisierung definierter unterscheidbarer ontologischer Ebenen erleichtert es, Daten anhand von Suchbegriffen auf frei wählbaren Abstrahierungsebenen aufzufinden. Im Rahmen der Analyse von Anfrage-Ergebnissen kann umgekehrt festgestellt werden, wieviel Annotationen auf einem bestimmten Detaillevel gefunden werden. Das erleichtert die Ontologie-Analyse bzw. hilft festzustellen, wie differenziert Bereiche der Wissensbank modelliert wurden.

4.4.3 Fehler in zu integrierenden Ontologien

Manche Zweige der Gandr-Ontologie bestehen teilweise aus Modulen anderer Ontologien. Bei der Integration stellte sich heraus, daß diese nicht immer fehlerfrei, geschweige denn für unsere Anwendungsdomäne ausreichend vollständig waren. Bei dem Konzept "Eosinophil_Cell" aus der EVOC Ontologie z.B. lautet die Definition: "Granular **leukocytes** with a nucleus that usually has two lobes" Der Begriff ist dort aber nicht unter Leukocytes eingeordnet. Derartige Fehler wurden, auch für einige Gene Ontology-Konzepte, den entsprechenden Kuratoren übermittelt.

4.5 Beurteilung der Gandr-Ontologie

Die Entwicklung präziser, einheitlicher und objektiver Bewertungskriterien bzw. Referenzstandards zur Evaluierung von Ontologien und ihrer Adäquatheit im Hinblick auf ihre Anwendungsdomäne gestaltet sich schwierig, weil das Anwendungsspektrum für Ontologien breit und sehr uneinheitlich ist. Sachdomäne, Anwendungstyp, Semantik und Granulいた weichen zwischen verschiedenen Ontologien stark ab und sind daher schwer zu vergleichen. Es gibt nahezu ebenso viele "richtige" Möglichkeiten, eine Ontologie zu formalisieren, wie es unterschiedliche Perspektiven und Anwendungen dafür gibt. Was hier zählt, ist letztendlich die Brauchbarkeit der Ontologie in einer Anwendung [11].

4.5.1 Ontologie-Typ

Breuker et al. [69] klassifizieren Ontologien nach ihrer Verwendung als Domänen-, Kern- und *top-level*-Ontologien, wobei die Übergänge oft fließend sind, da die meisten Ontologien Charakteristika mehrerer Typen beinhalten. Aus der mittlerweile unüberschaubaren Menge an Ontologieklassifizierungen seien hier einige nach Domänenspezifität, Formalisierungsansatz und Anwendung unterschiedene Ontologie-Typen genannt und zur Gandr-Ontologie in Beziehung gesetzt:

- **Upper model (UM) Ontologien:** In diesen Ontologien wird *common sense*, also weitgehend domänenunabhängiges Weltwissen, wie Zustand, Ereignis, Prozeß, Zeit etc. repräsentiert. Die Ontologien sind deshalb universell wiederverwendbar. Beispiel: CYC-Ontology [70].
- **Middle model (MM) Ontologien:** Diese enthalten Konzepte mittlerer Spezialisierung. Beispiel: Tambis Ontology (siehe Abschnitt 4.8.5).
- **Domain model (DM) oder lower-level-Ontologien:** Diese enthalten viele Konzepte hoher Spezialisierung und sind daher nur in einer bestimmten Domäne wiederverwendbar. Beispiel: Gene Ontology (siehe Abschnitt 4.8.2).
- **Anwendungsontologien:** Sie beinhalten Konzepte und Definitionen, die notwendig sind, um das für eine bestimmte Anwendung bzw. Aufgabe benötigte Wissen zu modellieren. Sie sind meist nicht wiederverwendbar und können sehr heterogen zusammengesetzt sein. Beispiel: "Diagnose-Erstellungs-Ontologie".
- **Metaontologien:** Eine Art Ontologie der Ontologie, welche die Begriffe zur Repräsentation von Ontologien stellt (die Ontologie-Sprache).

Nach Gangemi [64] und Fensel [71] stellt die Gandr-Ontologie eine *domain model*-Ontologie mit einigen *middle model*-Konzepten (z.B. "Enzyme") dar, die dort *core ontology*-Konzepte genannt werden. Da die Ontologie im Hinblick auf die spezielle Domäne "*dendritic cell development*" und NFkB- bzw. Toll-like-Receptor-ST formalisiert wurde, stellen *domain-*

ontology-Ideome wie z.B. "Dendritic_Cell_Type" und Genfunktionen beschreibende Konzepte und Slots die meisten Gandr KR-Ideome. Vorteil der domänenspezifischen Ontologie ist, daß sie sich auf einen realisierbaren speziellen Teilbereich reduziert und nicht gleich "die ganze Welt" zu modellieren versucht. Man kann die Ontologie als Anwendungsontologie betrachten, da sie primär im Hinblick auf die spezielle Anwendung Genannotation und -visualisierung formalisiert wurde. Ein charakteristisches anwendungsbezogenes KR-Ideom der Gandr-Ontologie ist z.B. der `ST_successor`-Slot. Die Metaontologie der Gandr-Ontologie ist die Protégé-Metaarchitektur (siehe Abb. 4) bzw. das OKBC-kompatible CLIPS-Ontologieformat. Damit stellt die Protégé Metaontologie das *core axiom*-Schema der Gandr-Ontologie nach Guarino [14]. Seiner Terminologie folgend ist die Gandr-Ontologie eine *core ontology*.

Da von der vorgesehenen Nutzergruppe nicht erwartet werden konnte, ihre eigene "fehlerfreie" Ontologie von Grund auf selbst zu erstellen, wurde dem System die Gandr-Ontologie als Basis-Schema beigegeben. Sie stellt eine erste Leitstruktur dar, die dem Nutzer Hilfestellung zur weiteren Ausformulierung der Ontologie gibt. Die Ontologie erfüllt damit aus der Perspektive des Endnutzers die Funktion einer sog. *foundational ontology*, FO nach Gangemi [64]. Hierbei liefern die von Superkonzepten geerbten Konzepteigenschaften Anhaltspunkte dafür, ob eine durch den Nutzer durchgeführte neue Annotation oder Konzeptionalisierung sinnvoll ist. Die Gandr-Ontologie erhöht so grundlegende Konsistenz in der weiteren Ausdifferenzierung der Ontologie durch den Endnutzer. Über eine **Normalisierung** der Ontologie kann die Interpretationskonsistenz allgemeinerer *top-level*-Konzepte weiter verbessert werden. Dabei formalisiert man diese konform zu einer in allgemeiner Übereinkunft standardisierten FO. Alle Ontologien, bei denen Kompatibilität untereinander angestrebt wird, werden dann von diesem universellen domänenunabhängigen Standard abgeleitet [64]. Da die über die FO-Konzepte den *top-level*-Konzepten zufallenden Eigenschaften und *constraints* vererbt werden, sichert die Unterordnung eines *top-level*-Konzepts unter ein FO Konzept die Konsistenz aller Subkonzepte. So eine Normalisierung könnte über die Einordnung der Gandr *core ontology*-Konzepte unter entsprechende FO-Konzepte z.B. der SUMO-Ontologie [72] erfolgen. Das Gandr-Konzept "Biological_Process" stände dann z.B. unter dem SUMO-Konzept "Event", das Gandr-"Cellular_Compound"-Konzept dagegen unter dem SUMO-Konzept "Physical Object". Das Gandr-Konzept "Compound_Part" wäre ein SUMO-"Component Part"-Subkonzept und das Gandr-Konzept "Context_Annotation" ein SUMO-"Abstract Object". Der Zeitpunkt für eine FO-Normalisierung der Gandr-Ontologie erscheint jedoch zu früh, da sich auch unter den FO-Ansätzen kein allgemein akzeptierter Standard durchgesetzt hat. Die Namen der Gandr-*top-level*-Konzepte wurden so intuitiv gewählt, daß eine weitergehende Beschreibung durch FO-Konzepte

die meisten Endnutzer eher verwirren würde. Die Gandr-Konzepte sind über ihre Positionen in der Taxonomie, ihre Slots und textuellen Definitionen für den Nutzer ausreichend beschrieben.

4.5.2 Beurteilung der Kodierungssprache und Expressivität

Zur Evaluierung der ontologischen Repräsentationssprache wurden folgende Eigenschaften herangezogen:

- **Expressivität:** Kann die Sprache die benötigten Ontologiebestandteile eindeutig repräsentieren? Ist sie für den geplanten Anwendungsbereich flexibel genug? Kann sie gegebenenfalls selbst erweitert werden?
- **Komplexität:** Wie einfach ist das Erlernen und die Benutzung der KR-Sprache?
- **Übersetzbarkeit bzw. Abbildung auf andere Formate:** Können KR-Ideome gegebenenfalls auf andere Formate abgebildet bzw. ohne Informationsverlust in andere Sprachen ähnlicher Semantik übersetzt werden?
- **Entwicklungsstand und Verbreitung:** Ist die Sprache entwicklungsbedingt noch in ständigem Wandel begriffen oder ausgereift? Ist die Sprache ein akzeptierter Standard? Gibt es zu dieser Sprache qualifizierte Entwicklungsgruppen, gute Entwicklungswerkzeuge und Dokumentationen?
- **Zugänglichkeit:** Wie einfach kann man die Sprache erhalten (Kostenaufwand / Lizenzierung)?

Es wurde das CLIPS-Format als Implementierungssprache gewählt, da es bei hoher Ausdrucksstärke und guter Lesbarkeit (auch durch Nicht-Experten) eine klare Abbildung auf andere gängige Formate wie XML und UML erlaubt. CLIPS ist gut zugänglich, da es über den *Protégé ontology editor* kostenlos bereitgestellt wird. Als OKBC-Standard findet CLIPS weite Verbreitung, ist gut dokumentiert und durch eine Vielzahl von weiteren Werkzeugen bearbeitbar. Die in CLIPS realisierte Trennung von Ontologie-, Instanz-Daten und Konfigurierungseinstellungen erlaubt, daß diese Bestandteile der Wissensbank gegebenenfalls getrennt bearbeitet, wiederverwendet und verschickt werden können. Die semantische Ausdrucksstärke des CLIPS-Datenmodells ist für den Zweck der konzeptbasierten Genannotation mehr als ausreichend, was durch die exemplarische Formalisierung des TLR-/NFkB-ST-Weges (gezeigt in Abb. 13, 14 und 15) und durch die Fähigkeit der Wissensbank die ontologischen Kompetenzfragen zu beantworten (siehe Beispiel in Abb. 14), belegt wird. Im Bedarfsfall kann die Semantik sogar noch stark erweitert werden (siehe Abschnitt 2.1.6 und 3.6). Markupssprachen wie XML und XMI schieden aufgrund ihrer gegenüber den Ontologie-Modellierungssprachen eingeschränkten Semantik (wenig *Slot-facets*) als Ontologie-Formate aus. Die Modellierung von *constraints* wie Axiomen und Kardinalitäten wäre z.B. in UML nur über die Spracherweiterung *object constraints language*, OCL und dann auch nur über das provisorisch anmutende Anfügen an die Klassen als UML-*notes* möglich. Zudem gibt es für in

OCL formalisierte Axiome keine *reasoner*. XML und UML eignet sich eher für die Darstellung kleiner wenig komplexer Taxonomien. Einen umfassenderen Überblick über die Vor- und Nachteile beim Einsatz verschiedener KR-Semantiken in der Medizin liefert Biolchini [73], weitere Repräsentationsformate im Vergleich und eine Analyse in Bezug auf ihre Eignung für die Bioinformatik erörtern McEntire [74] und Rector [75].

4.5.2.1 RDB vs. CLIPS vs. OWL

Gegenüber dem RDB-Schema liegt der Schwerpunkt bei der Ontologie auf der Hierarchisierung, der relationalen Komplexität und der universellen Wiederverwendbarkeit des Schemas, wogegen ein RDB-Schema meist semantisch einfach und anwendungsspezifisch bleibt. RDB-Schema und Daten sind enger gekoppelt, während die Ontologie auch von den Daten getrennt genutzt, bearbeitet und verbreitet werden kann. Im Gegensatz zur Ontologie enthält ein RDB-Schema selten mehr als 100 Konzepte (Tabellen), die zudem nicht hierarchisch strukturiert sind, keinen Gebrauch von Vererbung machen und daher keine impliziten Informationen fassen können. Ontologien sind objektorientierte und damit dem Menschen ergonomisch zugänglichere Datenrepräsentationen.

Bei der Wahl des ontologischen Repräsentationsformalismus war zu entscheiden, ob nicht der zukunftssträchtige OMG-Standard OWL anstelle von CLIPS genutzt werden sollte. Der Vorteil von OWL ist, daß es die Definition neuer Konzepte als Kombination von vorhandenen mit zusätzlichen Eigenschaften, also in einer Art "Konzept-Lego", ermöglicht. Die Wahl fiel auf CLIPS, da die Implementierung und vor allem Erweiterung der Ontologie hierin sehr viel einfacher und nur so wie gefordert durch den Endnutzer zu bewerkstelligen ist. Die Erstellung größerer Ontologien in OWL ist aufgrund seiner komplexen schwer zu erlernenden und streng logisch definierten Semantik sehr aufwendig. Für umfassendere Ontologien, wie der vorliegenden, wäre der benötigte Kodierungs- und Zeitaufwand zu groß gewesen. Um die Vorteile der OWL-Semantik nutzen zu können, hätte man jedoch sehr präzise und ausführlich formalisieren müssen, was der Forderung nach einfacher Nutzung, auch durch "Nicht-Wissensingenieure", widersprochen hätte. OWL ist eher zur detaillierteren Formalisierung kleiner spezieller und gut bekannter Teilbereiche einer Wissensdomäne geeignet, in der *automatic reasoning* und komplexere KI-Methoden Anwendung finden sollen.

4.6 Beurteilung der Gandr-Wissensbank

Die Kenngrößen zur Bewertung der Wissensbank sind ähnliche wie die in 4.5.2 genannten, da die Ontologie die Anwendungsmöglichkeiten der Wissensbank größtenteils vorgibt. Im Hinblick

auf semantische Ausdruckskraft und Konsistenz ist jede Wissensbank nur so gut wie die zugrundeliegende Ontologie.

4.6.1 Formale Klassifizierung der Anwendung nach dem System Uscholds

Im folgenden soll die Klassifizierung und Bewertung ontologischer Anwendungen nach dem standardisierten System von Uschold [76] erörtert und auf das GandrKB-System angewandt werden. In Anlehnung an die *use-cases* von Ivor Jacobson [77] stellt Uschold drei verschiedene abstrakte Anwendungsszenario-Klassen vor. Hierüber sollen spezielle Anwendungsszenarien klassifiziert und in einheitlicher Terminologie beschrieben werden:

- a) **Neutrales Authoring**, d.h. Entwicklung ontologischer oder operationaler Daten in universellem Format zur Nutzung in verschiedenen Umgebungen. Die standardisierten KR-Ideome können über Abbildungen in andere Formate transformiert werden.
- b) **Gemeinsamer Zugang** und geteiltes Verständnis zu Informationen (Interoperabilität).
- c) **Ontologie als Suchindex**

In Erweiterung zu Uschold System wird hier noch folgende Anwendungsszenario-Klasse hinzugefügt:

d) **Ontologie als vernetzter Wissens- bzw. Modell-Speicher**

Seiner Wichtigkeit entsprechend sollte d) nicht nur als Unterpunkt von "Beabsichtigter Anwendungszweck/System engineering" in den unten genannten Anwendungscharakteristika, sondern als eigenständige Anwendungsszenario-Klasse repräsentiert werden. Jede Anwendungsszenario-Klasse kann mehrere Anwendungsszenarien beinhalten, die über wichtige Schlüssel-Dimensionen charakterisiert werden. Diese Schlüssel-Dimensionen werden bei Uschold auf standardisierte Diagramm-Darstellungsideome abgebildet und können so zur Darstellung einheitlich interpretierbarer Anwendungsszenario-Diagramme genutzt werden:

1. **Beabsichtigter Anwendungszweck:** z.B. Humankommunikation, System-Interoperabilität, *system engineering* (hier *re-usability*, Datenbestandssuche über Metadaten, Erhöhung der Zuverlässigkeit z.B. über *consistency checking*, IT-System-Spezifikation (heute MDA, siehe Anhang C.3), Wartungsvereinfachung, Wissensakquisition).
2. **Rolle der Ontologie:** Jede Anwendung enthält zwei Ontologien. Die domänenspezifische Ontologie, nach Uschold L_1 , hier die Gandr-Ontologie und die Ontologie der ontologischen Repräsentation, nach Uschold L_2 , hier das Protégé OKBC/CLIPS-Metamodell. Um Information auf Level L_n zu beschreiben benötigt man also eine Referenz auf L_{n+1} . Uschold unterscheidet noch einen Informationslevel L_0 als operationale Daten und meint damit die über L_1 beschriebenen Instanz-Daten. Der Begriff "operationale Daten" ist nicht besonders glücklich gewählt, da hierunter leicht auch vom System generierte, zur Laufzeit erzeugt, verändert und genutzte Daten verstanden werden können. Der Begriff "Instanz-Daten" wäre hier klarer.

3. **Rollen der Akteure/Agenten in den Szenarien:** Ontology Autor, OA (Schober, Hacker), Operational Data Author, DA (AG Zenke, Affymetrix[®], *science community*), Application Developer, AD (*protégé community*, Schober), Application User, AU (*public domain*, Zenke et al.), Knowledge Worker, KW (Akteur, der das Wissen nutzt, *public domain*, AG Zenke).
4. **Unterstützende Technologien:** Sprachen, Konvertierungswerkzeuge, *ontology merging*, *DB-backend*. Uschold unterscheidet hier Ontologie-Repräsentationssprachen und Wissensaustausch-Sprachen, ohne zu beschreiben, worin für ihn der Unterschied zwischen beiden besteht.
5. **Reife des Projekts:** z.B. Ideenlevel, *proof of concept*, kommerzielle Nutzung.

Als weiteres Charakteristikum wird noch die Semantik der Repräsentation erwähnt: Eindeutigkeit und Grad der Formalisierung (Interpretationsspielraum), z.B. hoch informal, strukturiert informal, semi-formal und hoch-formal.

Hier nun die Klassifizierung der GandrKB-Anwendungsszenarien nach dem System von Uschold. Die alphabetischen Überschriften beziehen sich auf Uscholds Anwendungsszenario-Klassen. Die nachfolgende Numerierung bezieht sich auf die oben genannten Schlüssel-Dimensionen der jeweiligen Anwendungsszenarien:

a) Neutrales Authoring

1. Beabsichtigter Anwendungszweck und 2. Rolle der Ontologien: Die Konvertierung des L₂-Metaontologie-Formates in andere L₂-Formate (XML, UML, RDFS, OWL) erleichtert die Wiederverwendung und Weiterverarbeitung der Gandr-Ontologie (L₁) und der Instanzdaten (L₀) in den verschiedensten Protégé-unabhängigen, also andere L₂ nutzenden Werkzeugen. Die Umwandlung von L₁ und L₀ in eine Anwendungsbezogene L₂ erfolgt über eine Abbildung der verschiedenen L₂-Ideome aufeinander.
3. Die Akteure für das Aufstellen der *mapping*-Regeln für die Abbildung der L₂-Formate sind die (Meta-)Ontologie Autoren (OA) und der (Meta-)Anwendungsentwickler (AD) Holger Knoblauch. Bei der späteren Konvertierung selbst sind es der Anwendungs-Nutzer (AU) bzw. die Protégé-Anwendung, welche die Abbildungs-Regeln enthält und umsetzt.
4. Unterstützende Technologien für die Konvertierungs-Funktionen sind die L₂-Semantiken bzw. Markupsprachen und die Java-XML Bibliothek.
5. Die Reife des Projekts im Rahmen dieses Anwendungsszenarios kann im Bezug auf OWL als Erprobungsphase, für csd, txt, XML und Dot als ausgereift und für XMI und UML als fortgeschritten betrachtet werden.

b) Gemeinsamer Zugang

1. Im Rahmen des Anwendungsszenarios dieser Klasse ist der beabsichtigte Anwendungszweck die verbesserte Kommunikation zwischen Menschen bzw. die einheitliche Interpretation der Annotations-Konzepte und der Instanzdaten. Auch die Gandr Web-Anwendung fällt in dieses Szenario.
2. Bezüglich der Rollen der Ontologien dient die Gandr-Ontologie L₁ der Formalisierung der L₀-Instanz-Daten (Netaffx-Genbeschreibungen und Expressionswerte).

3. Akteure innerhalb dieses Anwendungsszenarios sind die verschiedenen Microarray-Auswerter der AG Zenke als AU und KW.
4. Unterstützende Technologien stellen das Affymetrix-DMT für die anschließende statistische Untersuchung, die Java-Tomcat-Servertechnologie für die Webanwendung und die externen Internetseiten zur Darstellung der Kontext-Informationen über *deeplinks* dar.
5. Die Reife des Projekts ist hier als *work in progress* zu bezeichnen, da die L_2 je nach Detailtiefe und Wissensstand verändert bzw. ständig erweitert werden sollte.

c) Ontologie als Suchindex

1. Die Anwendung ermöglicht das Auffinden implizit in L_1 und L_0 vorhandener Informationen und Relationen zwischen den Daten. Metadaten werden als Suchattribute für Anfragen nutzbar gemacht.
2. Die L_2 Ontologie-Ideome werden zum Aufbau eines Wissensbank-Schema, das L_1 - und L_2 -inhärente Relationen repräsentiert, genutzt.
3. Beteiligte Akteure sind die verschiedenen Microarray-Auswerter als AU und KW.
4. Unterstützende Technologien sind das Queries & Export-Tab und die Visualisierungs-Plugins.
5. Die Reife des Projekts für dieses Anwendungsszenario ist als fortgeschritten zu bezeichnen.

d) Ontologie als vernetzter Wissens- bzw. Modell-Speicher (in Ergänzung zu Uschold)

1. Formale Wissensrepräsentation bzw. Modellierungsumgebung, (wie c).
2. Über L_2 formalisierte L_1 -Ontologie stellt Daten- bzw. Annotationsmodell, das L_0 -Daten mit Schwerpunkt auf relationalem Kontext, also als Wissen bzw. Domänenmodell speichert. Die L_1 -strukturierten, als semantisches Netzwerk visualisierten L_0 -Daten bieten eine formale und dadurch modernen WM-Werkzeugen zugängliche Ergänzung zu unformalen Domänen-(*pathway*-Diagrammen) und Wortmodellen (*paper*).
3. Beteiligte Akteure sind die verschiedenen Microarray-Auswerter als AU und KW.
4. Unterstützende Technologien stellen die Visualisierungs-Plugins und die Slot-*widgets* für relationale Slots.
5. Die Reife des Projekts für dieses Anwendungsszenario ist als fortgeschritten zu bezeichnen.

Uscholds Ansatz einer "formalisierten" Anwendungsbeschreibung muß als relativ unausgereift bezeichnet werden. Er hat weder weitere Verbreitung gefunden, noch ist dies für die nahe Zukunft wahrscheinlich, da das System nicht vollständig und relativ unformal ist. Eine Ontologie seiner Kriterien wäre formaler und in der Anwendung hilfreich gewesen. Einige von Uscholds Anwendungsklassen und Bewertungskriterien stellten sich als mehrdeutig heraus. Der *re-use*-Aspekt wird z.B. in allen drei Punkten betont. Des weiteren kann a) auch als Subtyp von b) bzw. a) als Methode, um b) zu erreichen, verstanden werden. Als Vorzüge von a) wird von Uschold die verbesserte Wartung genannt. Dieser Vorteil gilt genauso für die Punkte b) und c).

In der Praxis können Ontologie-Anwendungsszenarien sehr viel schneller als UML-*use cases* formal repräsentiert werden als durch Uscholds von keinem Design-Werkzeug unterstützten

Diagramme. UML bietet aufgrund seiner weiten Verbreitung den Vorteil, daß es sofort von einer großen Nutzergruppe angewandt und interpretiert werden kann, was die Abb. 7 und 12 zeigen.

4.6.2 Beurteilung der Anwendung anhand der Anforderungsspezifikation

Zur weiteren Kontrolle, ob die fertige Wissensbank in Anwendung und Funktion die Nutzererwartungen erfüllt, wurden die tatsächlich ermöglichten Anwendungen der Wissensbank mit den vorher festgelegten Anforderungsspezifikationen abgeglichen. Hierüber konnte festgestellt werden, ob die ursprünglichen Zielvorstellungen erfüllt und ob alle wichtigen Wissensquellen in die Ontologie und Wissensbank eingeflossen sind. Ein Vergleich der Anforderungsspezifikationen in Abschnitt 1.4 und 3.1.1 mit den entsprechenden Abschnitten im Ergebnis-Teil ergibt, daß die Anforderungsspezifikation in den meisten Punkten als erfüllt betrachtet werden kann. Die Ontologie-Kompetenzfragen konnten durch das System beantwortet werden (siehe Abschnitt 4.5.2). Es konnten einige über die ursprünglich geforderte formale Annotations- und Modellierungs-Plattform hinausgehende weitere Anwendungen vorgestellt werden. Hier sind insbesondere die Visualisierungs-Ansätze und das regelbasierte automatische Verändern der Wissensbank mit dem JessTab zu nennen. Als weniger gut stellte sich jedoch die *user-compliance* heraus (siehe Abschnitt 4.6.5).

4.6.3 Beurteilung der IR-Kapazität

Der Hauptanwendungszweck des Systems ist die inhaltsbasierte Abfrage annotierter Microarray-Gendaten über eine ontologische graphische Anfrageschnittstelle. Der Aufwand, der auf der Benutzerseite für die Anfrageformulierung erforderlich ist, wird im GandrKB-System über die gebotenen ontologie-gesteuerten Hilfsangebote wie Konzept- und Slot- Auswahllisten minimiert (Nutzung der *constraints*). Die in herkömmlichen LIMS-Systemen genutzten Anfrageschnittstellen liefern dem *retrieval-system* selten genügend Daten zur umfassenden und korrekten Interpretation der Anfrage-Intentionen. Daher liefern diese Systeme auf Anfragen auch irrelevante Ergebnisse, die der Intention des Anfragers nicht voll entsprechen. Die vom Nutzer formulierte Anfrage enthielt dann meist implizite Absichten, die dem *retrieval-system* über die Anfragesemantik nicht vermittelt wurden bzw. konnten. Diesem Problem versucht das GandrKB-System beizukommen, indem es eine starke Anfragesemantik bereitstellt, die den Nutzer bei der Anfrageformulierung unterstützt und implizites Wissen über die Ontologie nutzbar macht (siehe Abschnitt 3.4.4).

Zur Evaluation der Leistungsfähigkeit von Anfrageschnittstellen sind nach Cooper [78] die normalisierten Begriffe *precision* und *recall* heranzuziehen, da sie zwischen verschiedenen Such-Systemen vergleichbar sind. Der **recall** (Vollständigkeit) gibt an, ob alle relevanten

Informationen gefunden wurden, und ist definiert als das Verhältnis der Anzahl gefundener relevanter Daten zu der Gesamtzahl in der Datenbank vorhandener relevanter Daten. Der *recall* ist also 100%, wenn alle relevanten Daten gefunden wurden. Die **precision** (Brauchbarkeit) beschreibt, wie viele der gefundenen Ergebnisse wirklich im Sinne der Intention der Anfrage brauchbar und relevant sind. Sie ist definiert als das Verhältnis der Anzahl gefundener relevanter Dokumente zu der Anzahl aller gefundenen Dokumente. Ist die *precision* 100%, so ist jedes gefundene Ergebnis auch für den Nutzer relevant. Über die ontologische Datenrepräsentation im GandrKB-System wurde versucht, die *precision* möglichst nicht auf Kosten des *recall* zu erhöhen. Beide Faktoren können über die ontologische Anfrageschnittstelle direkt vom Endnutzer beeinflusst werden. Generalisierungen von Anfragen bzw. Subsumierungen erhöhen den *recall*. Spezialisierungen von Anfragen und Anfragen nach Slots und Relationen erhöhen die *precision*. Herkömmliche in Microarray-LIMS implementierte Suchstrategien haben gegenüber dem GandrKB-Ansatz eine geringe *precision* bei gutem *recall*. Sie liefern also viele irrelevante Ergebnisse unter generell sehr vielen Treffern. Die geringe *precision* wird dabei durch die meist lediglich *string*-basierten Anfragen verursacht, die weder Subsumption, noch Anfragen nach Eigenschaften und Relationen erlauben. Auch speichereffiziente Vererbungsprinzipien, welche die IR-Kapazität weiter erhöhen, sind in diesen Systemen nicht nutzbar. Zur weiteren Beurteilung der IR-Fähigkeiten ist auch die Position der gesuchten Information in der Liste der Ergebnisse heranzuziehen. Steht ein relevantes Suchergebnis direkt am Anfang der Ergebnisliste, so wird es schneller gefunden, als wenn es am Schluß steht. Viele unrelevante Ergebnisse am Anfang der Liste können den Nutzer sogar dazu bewegen, die Ergebnisse nicht bis zu Schluß durchzugehen, so daß das gefundene relevante Ergebnis u.U. gar nicht genutzt wird. Im GandrKB-System läßt sich die Reihenfolge in der Ergebnisliste explizit gestalten. Über Einstellung des *browser keys* können die Ergebnislisten nach gewünschten besonders relevanten Eigenschaften sortiert werden.

4.6.4 Dokumentation und Schulung

Eine ausführliche, an konkreten Anwendungsbeispielen orientierte Dokumentation der GandrKB-Anwendung ist Grundvoraussetzung für die Nutzung durch Domänenexperten, die von ihrer Ausbildung her keine Wissensingenieure sind. Dabei muß einerseits versucht werden, den Nutzer nicht durch zuviel theoretisches Detailwissen über L_1 -Repräsentationsfomalismen oder gar philosophische Betrachtungen zu überfordern, andererseits müssen ihm die Grundlagen objektorientierter Repräsentation vermittelt werden, ohne die ein sinnvolles Arbeiten mit dem ontologie-basierten System nicht zu gewährleisten ist. Ein grundlegendes Verständnis von

"Objektorientierung" und "Vererbung" ist für eine effiziente Nutzung ontologischen Wissensmanagements unbedingte Voraussetzung. Dieses Verständnis ist aber offenbar noch nicht so weit als Allgemeingut in den Naturwissenschaften anzusehen, als daß auf eine gründliche Erläuterung und Schulung dieses Prinzips verzichtet werden könnte. Deshalb wurde hierauf in Dokumentation und Schulung besonderes Gewicht gelegt. Ein erster Überblick über "Objektorientierung" und die grundlegenden Funktionen des GandrKB-Systems konnte der Nutzergruppe in einer zweistündigen Schulung soweit vermittelt werden, daß die Biologen der AG in der Lage waren, selbständig einfachere Anfragen an das System zu formulieren.

Was die Einstellung auf den Anwender anbelangt, sind die Dokumentationen existierender Ontologien oft wenig ausführlich und praxisnah. Für die Nutzer der Gandr-Wissensbank wurde daher besonderes Gewicht auf die Dokumentation und einen didaktischen Aufbau des Lernmaterials gelegt. Hier erleichtern umfassende multimediale Schulungsunterlagen den Einstieg in die Nutzung des Systems. So wurden z.B. Tonfilme erstellt, die verschiedene Anwendungen des GandrKB-Systems an beispielhaften Fragestellungen genau erläutern.

4.6.5 Akzeptanz beim Nutzer

Es stellte sich heraus, daß die Akzeptanz zur tatsächlichen Einarbeitung und Nutzung des Systems für die intendierte Endnutzergruppe, also Biologen und "Nichtinformatiker", trotz ausführlicher Dokumentation als relativ gering anzusehen war. Das Erlernen der effektiven Nutzung ontologischer Anfrage- und Visualisierungs-Techniken erfordert offenbar hohe Motivation, die bei Nutzern, die keine genaue Vorstellung bezüglich des zugrundeliegenden Paradigmas der Objektorientierung und vor allem der Vorteile ontologiebasierten Wissensmanagements haben, nicht gegeben zu sein scheint. Das Verständnis des Wortes "Ontologie" ist bei den meisten Nutzern durch die weit verbreitete Gene Ontology geprägt, die im Hinblick auf die zugrundeliegende Semantik sehr viel schwächer formalisiert ist und eher als einfache Taxonomie ohne Vernetzung und ohne vererbare Eigenschaften zu bezeichnen ist. Vor dem Hintergrund der omnipräsenten GO stellt die Mehrheit der Nutzer bei der Interpretation des Ontologiebegriffs auch eher den Standardisierungs- als den Wissensrepräsentationsaspekt in den Vordergrund.

Die Benutzeroberfläche der GandrKB ist relativ komplex und für den unerfahrenen Nutzer durch ihre Vielzahl von Konfigurierungsmöglichkeiten zunächst etwas verwirrend. Hier erweist sich die Möglichkeit einer an das Vorwissen des Nutzers anpaßbar konfigurierbaren GUI (siehe Abschnitt 3.5 und 3.6) als außerordentlich praktisch. Um die *user compliance* zu erhöhen, wurden den Nutzern zwei unterschiedlich komplexe Versionen der Wissensbank zur Verfügung

gestellt. So wurde zunächst eine einfache und übersichtliche "*lightweight*"-Benutzeroberfläche zur Verfügung gestellt (GandrKBs). Diese enthielt nur Funktionalitäten, die der Nutzer bereits von der GO kennt und die seinen unmittelbaren Nutzerwünschen am besten entsprechen. Nach Einarbeitung konnten den Nutzern dann nach und nach die komplexeren Anwendungsmöglichkeiten inkl. einer "*heavyweight*"-Benutzeroberfläche vorgestellt und nutzbar gemacht werden (GandrKB).

Über eine noch engere Absprache, die hier allerdings durch den räumlichen Abstand von der Nutzergruppe nicht leicht zu bewerkstelligen war, sollte das System Schritt für Schritt an mit dem Nutzer gemeinsam entwickelten Beispielfragestellungen vorgestellt werden. Im vorliegenden Fall wurde das System in einer einmaligen Schulung in seiner Gesamtheit vorgestellt. Des weiteren mag der Zeitverzug zwischen Systemplanung und Fertigstellung zu einem verschobenen Anwenderinteresse geführt haben, dem in der Systementwicklung nicht mehr Folge geleistet werden konnte. Abschließend betrachtet erscheint es, als würde die Bearbeitung von Microarraydaten generell zunehmend einen eigenen Typ von Biologen erfordern, der in erster Linie Informatiker und "Wissensingenieur" ist.

4.7 Beurteilung der Visualisierungsansätze

4.7.1 Datengetriebene und konfigurierbare GUI

Da der Mensch graphische Elemente schneller verarbeiten kann als textbasierte, wurde für das Annotationssystem eine eher graphik- als skriptbasierte angesprochene Benutzeroberfläche gewählt (für skriptbasierte WB-Manipulationen siehe Abschnitt 3.4.8). Die Protégé-GUI dient der Kommunikation zwischen Nutzer und Wissensbank. Ihre fensterbasierten Darstellungen mit fertigen Menüs und Auswahllisten, wie man sie vom Windows[®]-Betriebssystem bereits kennt, sind vergleichsweise übersichtlich und klar strukturiert. Der Nutzer wird dabei vom zugrundegelegten Repräsentationsformalismus weitgehend abgeschirmt. Trotz der Komplexität und Ausdrucksstärke der erstellbaren Anfragen, ist die Bedienung der Anfrageschnittstelle schneller erlernbar, als viele kommandozeilen- oder skriptbasierte Datenbank-Sprachen ähnlicher Ausdrucksstärke.

Als vorteilhaft erwies sich auch die ontologie- und dateninduzierte Anpassung der einzelnen GUI-Komponenten bei der Visualisierung von KR-Ideomen. So werden bestimmte Datentypen nach den ihnen eigenen Erfordernissen automatisch adäquat dargestellt und für ihre Eingabe entsprechend angepaßte Eingabeformulare, sog. *knowledge acquisition forms* kreiert (siehe Abschnitt 2.1.3.4). Ein weiterer Vorteil des vorgestellten Systems ist, daß die

Benutzeroberfläche bedarfsorientiert an die gegenwärtige Kenntnis- und Sachlage angepaßt werden kann. Das Gandr-System kommt damit der Forderung von Baeza-Yates und Ribeiro-Neto [79] nach, unterschiedliche Benutzersichten erfahrener und unerfahrener Benutzer bei der GUI-Konstruktion zu berücksichtigen.

4.7.2 Visualisierungen der Wissensbank-Inhalte

Die Annotation von Genen mit Konzepten der Gandr-Ontologie und die Verknüpfung von Genen untereinander über relationale Slots führt sukzessiv zu einem stark vernetzten Domänenmodell des Anwenderwissens. Da die Arbeitsspeicherkapazität des Gehirns begrenzt und bei der holistischen Verarbeitung von komplexen und großen systemischen Datenmengen an ihre Leistungsgrenzen stößt (siehe nächster Abschnitt), sollten computerbasierte Darstellungsverfahren den Menschen bei der Generierung, Vorauswahl und Verarbeitung kleinerer überschaubarer und so verarbeitbarer Subsysteme des Domänenmodells unterstützen. Das GandrKB-System ermöglicht dem Nutzer kontextabhängig, schnell und intuitiv einzelne KR-Ideome oder ganze Wissensnetze und Subnetze der Wissensbank automatisch auf frei wählbaren Abstrahierungsebenen darzustellen und die erzeugten Graphiken interaktiv zu erforschen. Die Repräsentation des Wissens in der Wissensbank erfolgt am besten graphisch, da der Mensch als ausgesprochenes "Augentier" hier ein großes Hirnareal mit starker Parallelverarbeitungs-Kapazität für *pattern matching*, also das Offenlegen von Symmetrien und Regelmäßigkeiten in den Daten, nutzen kann [60]. Von den sensorischen Gedächtnissen hat das visuelle Arbeitsgedächtnis [80] die größte Kapazität. Graphische, d.h. analog und parallel präsentierte, Wissensinhalte werden daher schneller verarbeitet und gespeichert als seriell präsentierte Zahlen und Wörter. Oft kann man durch einen Blick auf eine Graphik Zusammenhänge erkennen, die einem selbst nach längerem Studieren entsprechender Texte nicht offenbar werden (hier z.B. *feedback*-Schleifen). Datengetriebene und direkt aus der Wissensbank erzeugte graphische Visualisierungen bieten eine kognitiv besonders adäquate Hilfe zum schnelleren Erfassen komplexer Kontext-Strukturen in der Wissensbank. Sie sind bei der Erstellung und Pflege von Wissensbanken von Nutzen und erleichtern letztendlich über ein "ergonomisches *biofeedback*" die (Meta-)Analyse und Reflektion des eigenen Wissensmodells.

4.7.2.1 Vorteile der Frames gegenüber tabellarischen Darstellungen

Das Kurzzeit- bzw. Arbeitsgedächtnis speichert augenblicklich zu verarbeitende Informationen und erstellt Modelle, deren Teile dann nach entsprechend wiederholter Abrufung im Langzeitgedächtnis gespeichert (*encodiert*) werden [80, 81]. Da das Arbeitsgedächtnis nur ca. sieben unabhängige Entitäten parallel speichern und verarbeiten kann [63], sind alle darüber

hinausgehenden Informationen durch das Gehirn nicht parallel nutzbar. Eine Datenrepräsentation in tabellarischer Form ist zwar eine für Computer einfach zu bearbeitende Darstellung, wahrnehmungs- und kognitionspsychologisch betrachtet jedoch eine für den Menschen unadäquate und denkbar ungeeignete Repräsentation. Tabellen präsentieren dem Gehirn zu jeden Zeitpunkt sehr viele Informationen, die ungenutzt bleiben und das Auffinden tatsächlich gesuchter Informationen verlangsamen. Die Ressource **Aufmerksamkeit** (*alertness*) [82] kann nur wenig durch bewußte Anstrengung gesteigert, aber über die framebasierte kontextuale Darstellung im Gandr-System effizienter genutzt werden.

Ein Nachteil tabellarischer Repräsentationen besteht auch in den eingeschränkten Datentypen, die beispielsweise das Öffnen externer Browser erfordern, wenn ein Hyperlink-Inhalt dargestellt werden soll. Derartige "Distraktionsaufgaben" beanspruchen Aufmerksamkeits-Ressourcen, was Konzentration und freien Ideenfluß stören kann. Gandr minimiert derartige Distractionen über die integrativen Darstellungs-Modalitäten verschiedenster Datentypen. So werden Internetseiten-Inhalte und semantisch im Zusammenhang stehende Instanzen im Frame direkt angezeigt.

Ontologien zeichnen sich gegenüber Tabellen und ER-Diagrammen durch stärkerer Betonung der Vernetztheit bzw. kontextualen "Situiertheit" der Daten aus. Die Vernetzung von Daten untereinander ist in Tabellen gar nicht und in relationalen Datenbanksystemen in vergleichsweise geringem Maße ausgeprägt, u.a. weil sie dort recht umständlich zu erstellen ist. In objektorientierten und ontologiebasierten Managementsystemen mit ihren datentypinduzierten Eingabemasken und überprüfenden *constraints* können Vernetzungen über relationale Slots schnell und einfach erstellt werden. Die Vernetzung der Daten in ihrem funktionalen Kontext ist für ein holistisch-modellhaftes Verständnis im Rahmen der Systembiologie und **Funktionalen Genomik** erforderlich. Mit den unter Abschnitt 3.4.3 und 3.4.5 beschriebenen Visualisierungsmethodiken werden dem Anwender Darstellungsmittel gestellt, die der menschlichen Wahrnehmungspsychologie besser Rechnung tragen als die herkömmlich auf diesem Gebiet eingesetzten tabellarischen Darstellungen.

4.7.2.2 Vergleich mit Kohns *molecular interaction maps*

In den letzten Jahren wurden verschiedene ontologie-basierte Ansätze zur Darstellung biologischer Komponenten inkl. ihrer potentiellen Interaktionen vorgestellt [48, 83, 84, 85, 86, 87]. Diese Ansätze erlauben zwar dem Nutzer eine Ontologie anzuwenden, aber nicht diese selbst zu erstellen oder nach eigenen Maßgaben zu verändern. Weiter sind die Ontologien, sofern der Ausdruck für diese Schemata gerechtfertigt ist, fest in die jeweiligen Anwendungen eingebettet und daher schwer wiederverwendbar.

Kohns *molecular interaction maps* [88] beispielsweise (http://discover.nci.nih.gov/kohnk/interaction_maps.html) beschreiben Diagramm- und Darstellungs-Richtlinien zur eindeutigen Repräsentation und Visualisierung molekularbiologischer Netzwerke. Darstellbar sind u.a. Multiprotein-Komplexe, Proteinmodifikationen und Enzym-Substrat-Interaktionen. Hiermit können komplexe standardisierte Visualisierungen von z.B. Zellzyklus und DNA-Repair-Stoffwechselsystemen erstellt und gesichert interpretiert werden. Jeder Interaktions-Typ wird über einen spezifischen Kanten-Darstellungstyp, die jeweilige Interaktionsstärke über die Strichdicke kodiert. Jede Interaktion (Kante) wird in einer Legende genauer erläutert und um statische Informationen und Referenzen ergänzt. Vorteilhaft ist die Gruppierung zusammengehöriger Komponenten in funktionalen Subsystemen. Die generierten Karten verdeutlichen die Vielfältigkeit der Interaktionen zwischen den Subsystemen und ihrem umgebenden Kontext. Die Strategie, jede Komponente lediglich einmal pro Graphik zu repräsentieren, ist zumindest ambivalent zu beurteilen, da hier der Vorteil des sparsamen Platzverbrauches auf Kosten der Übersichtlichkeit erkaufte wird. An jedem Knoten entsteht ein komplexes Wirrwarr von sehr vielen verschiedenen Kanten, das nicht immer einfach zu interpretieren ist. Dabei helfen auch die indexierten 2D-Raster-Koordinaten nicht, welche die genaue Position jeder Komponente beschreiben. Daß die Darstellung der Diagramme eine Legende erforderlich macht, verdeutlicht, daß die Diagramme nicht aus sich selbst heraus verständlich sind und daß die Darstellung zu kompakt ist, um Bezeichner direkt an den Darstellungs-Ideomen anzubringen. Da die generierten Diagramme oft kontraintuitiv sind und man zunächst die sehr abstrakte nicht-intuitive Darstellungsweise erlernen muß, wozu der Laborbiologe weder Zeit noch Lust hat, ist fraglich, ob sich dieser Ansatz in der Praxis durchsetzen wird. Da dieser Repräsentation kein gängiger Formatstandard zugrunde liegt, können erstellte Modelle nicht in andere Formate transformiert werden, was die Interoperabilität erheblich einschränkt. Hier wäre zu überlegen gewesen, auf den sich zur Publikationszeit etablierenden universalen Standard UML aufzubauen. Einen derartigen Ansatz stellt z.B. die Gruppe um Kolpakov mit BioUML [89] vor (<http://www.biouml.org/>).

4.8 Vergleich mit anderen Annotations-Systemen und Ontologien

In den letzten Jahren wurden unterschiedlichste Bioontologien für diverse Anwendungen entwickelt [23, 51, 85, 90, 91, 92, 93, 94]. Einen guten Überblick mit Vergleich der bekanntesten Ontologien und Systeme untereinander geben Schulze-Kremer [95], Stevens [96] und Sklyar [97]. Einige zur Genannotation verwandte Ontologien bzw. ontologische Anwendungen sollen folgend mit dem GandrKB-Ansatz verglichen werden.

4.8.1 Affymetrix®-eigene Annotationsmöglichkeiten

Das Affymetrix® **Data Mining Tool, DMT** filtert und sortiert Expressionsergebnisse eines oder mehrerer auf dem Affymetrix LIMS-Rechner installierter GeneChip-Experimente. Es ermöglicht die statistische Untersuchung von Replikat-Ergebnissen und das *clustering* nach ähnlichen Expressionsprofilen. Ein enthaltenes *matrix analysis tool* erlaubt den Vergleich zweier probe set ID-Listen. In der Version 3.0 wurden die Annotations-Möglichkeiten um die Integration von Netaffx-Annotationen erweitert. Ein *NetAffx download utility* erlaubt das periodische Abfragen und automatische Herunterladen gewünschter Netaffx-Annotationen. Das DMT erlaubt auch das Hinzufügen eigener Annotationen, nach denen Abgefragt werden kann und deren Ergebnisse Suchfiltern hinzugefügt werden können. Die Annotationen sind jedoch lediglich textueller Art, erfolgen also im Datentyp *string* in Tabellenspalten. Sie auch nehmen keinerlei Bezug aufeinander oder auf ontologische Semantiken. Aus einer *drop-down* Liste können per *annotations-type*, einer Art kontrolliertem Vokabular, allerdings primitive Metadaten zur Genannotation genutzt werden. Die Annotation per DMT besitzt dennoch bei weitem nicht die Ausdruckskraft ontologiebasierter Annotationen, wie sie der GandrKB-Ansatz zur Verfügung stellt. Die Designprämissen des Affymetrix-Ansatzes liegen erkennbar auf der einfachen Bedienbarkeit durch den Endnutzer.

4.8.2 Gene Ontology, GONG und GO-Mining-Tool

Als Ontologie zur Annotation von Datenbankobjekten mit Genfunktionen hat sich die Gene Ontology, GO [47] als Standard durchgesetzt. GO liefert kontrollierte Textbeschreibungen zur Annotation eukaryotischer Genprodukte in einem taxonomisch strukturierten standardisierten Vokabular. GO ist in Form von drei Text-Dateien (*flatfiles*) oder als XML-Datei frei über das Internet erhältlich, die drei *top-level*-Modulen für jeweils bestimmte Teilaspekte der Genannotation entsprechen (Stand 9/2005):

- Molecular Function.txt enthält 7078 Begriffe für die molekularbiologischen Aufgaben und Funktionen, die Genprodukte erfüllen können.
- Biological Process.txt enthält 9820 Begriffe für die systemischen biologischen Aufgaben bzw. Wirkungs-Ziele, die Genprodukte durch ihr synergetisches Zusammenwirken erreichen können. Sie bezeichnen den biologischen Prozeß, der den globalen Kontext der Genfunktion beschreibt.
- Cellular Component.txt enthält 1576 Begriffe für die subzellulären Komponenten, also Orte innerhalb der Zelle, wo Genprodukte sich befinden können.

GO wird relativ zentral in mehreren Gruppen und größtenteils manuell erstellt, aktualisiert und verifiziert. Die Annotation von Genen mit GO-Begriffen erfolgt dann über

organismusspezifische Datenbankprovider und dezentral in den Anwendergruppen. Annotiert wird auch semiautomatisch über Abbildungen verschiedener Datenbank-Annotations-Schemata auf GO-Begriffe. Die mit einer GO-Nummer als Identifier versehenen molekularbiologischen Begriffe (*GO-terms*) werden hauptsächlich als Such-Attribute in Datenbank-Anfragen und zum *clustering* im Rahmen statistischer Analysen genutzt [50]. Im Gegensatz zur Gandr-Ontologie ist GO vom Anspruch universal und sehr breit angelegt; GO soll quasi die gesamte Zellbiologie abdecken. Die große Menge an Konzepten - "Begriffe" wäre in diesem Zusammenhang treffender - und die oft sehr langen und dadurch wenig intuitiven Bezeichnungen machen es dem Anwender weder leicht, die für ihn zur Datenannotation benötigten Begriffe zu finden, noch diese Begriffe als Suchattribute zu nutzen. Oft sind vom Domänenspezialisten geforderte Fachbegriffe auch nicht in der GO enthalten. Was die Semantik betrifft, ist die GO im Gegensatz zur Gandr-Ontologie eher als reine Taxonomie denn als ontologische Wissensrepräsentation zu bezeichnen. Ihr fehlt es an definierter formaler Struktur bzw. semantischer Ausdruckskraft. Da die GO-Taxonomie durch eine gemischte *is-a* und *part-of* Hierarchie gebildet wird, können den Konzepten keine Eigenschaften zugewiesen werden (siehe Abschnitt 3.1.5.4). Auch die Möglichkeit einer Vernetzung von Konzepten untereinander ist durch den primitiven Repräsentationsformalismus nicht zu realisieren. Aufgrund der Standardisierungspriorität, in der GO zugrundeliegenden Ontologiedefinition, können und sollen "GO-Konzepte" nicht selbst erzeugt oder verändert werden. Das Erstellen angepaßter Annotations- und Wissensmodelle ist also mit GO nicht zu bewerkstelligen.

In einem GO-Nebenprojekt "*gene ontology next generation*", GONG [98], wird GO in die semantisch definierte und KI-Methoden zugängliche Repräsentationsform *description logic*, DL transformiert. Hierbei werden die Ontologiesprache OWL eingesetzt und neue DL-Werkzeuge entwickelt. Eine von Affymetrix[®] angebotene Internetanwendung *gene ontology-mining tool* [99] dient der Abbildung von probe set IDs auf GO-Begriffe, stellt also eine GO-Anwendung dar. Durch "Hochladen" einer auf tausend Gene begrenzten probe set ID-Liste, Angabe des Affymetrix[®] Chiptyps und Auswahl des interessierenden GO-Moduls erstellt das *GO-mining tool* eine Baumstruktur der GO-Konzepte mit den entsprechend zugeordneten probe set IDs. Man erhält wahlweise diesen per GraphViz generierten interaktiven DAG-Graphen oder eine Annotations-Datei mit den GO-Beschreibungen der probe set IDs als Tabelle. Bei *click* auf einen GO-Konzept-Knoten in der DAG-Graphik erhält man eine Liste der hiermit annotierten, zuvor hochgeladenen probe set IDs. Diese Funktionalitäten stellt das GandrKB-System über das OntoViz-Tab bereit, jedoch ohne Begrenzung auf eine bestimmte probe set ID. Abschließend sollte betont werden, daß der GandrKB-Ansatz keinesfalls eine Art Konkurrenzunternehmen zur

GO darstellt, sondern eine Ergänzung unterschiedlichen Themenschwerpunktes. Nicht der Standardisierungs-, sondern der Repräsentations-Aspekt und die Möglichkeit einer laborspezifischen Annotation und Modellierung mit eigenen ontologischen Konzepten stehen im GandrKB-Ansatz im Vordergrund.

4.8.3 UMLS

Das an der *National Library of Medicine* entwickelte *Unified Medical Language System*, UMLS, [100] soll Aufbau, Verwaltung und Verbreitung medizinischer Ontologien und entsprechender Werkzeuge unterstützen. Die UMLS-Ontologie vereinheitlicht den Zugang zu über 60 Terminologien (u.a. MeSH, ICD und SNOMED) aus den Bereichen Anatomie, medizinische Phänotypen, Symptomatologie und Nosologie über eine gemeinsame Semantik. Es führt die Bezeichnungen aus den Einzelvokabularien über einen "begriffsbasierten Metathesaurus" zusammen. Findet sich eine Bezeichnung in verschiedenen Quellterminologien, so wird für diese ein Begriff (*term*) erstellt. Der UMLS-Metathesaurus strukturiert mehr als 2.100.000 verschiedene Bezeichnungen über ca. 800.000 Begriffe. Die Begriffe des Metathesaurus sind wiederum selbst in einem semantischen Netzwerk, einer Art *foundational-* oder *top-level-*Ontologie aus ca. 140 Konzepten (sog. *semantic types*) und 60 Relationen, verknüpft. Aus molekularbiologischer Perspektive ist UMLS eine *top-level-*Ontologie, da sich die meisten Konzepte auf klinische Wissensgebiete beziehen. Da in UMLS wenige molekularbiologische Terminologien enthalten sind, ist eine Verwendung zur Genannotation bisher nicht ratsam. Dennoch können Domänenexperten, die ihre Terminologie in UMLS repräsentiert sehen, für sie brauchbare Konzepte direkt aus UMLS in die Gandr-Ontologie importieren. Dies geschieht über das UMLS-Tab Plugin (<http://protege.stanford.edu/plugins/umlstab/>). Will man z.B. MeSH-Begriffe importieren, wird über das UMLS-Tab auf die UMLS-Datenbank zurückgegriffen und überprüft, ob das gesuchte KR-Ideom in UMLS existiert. Wenn ja, kann es unter Erhalt seiner UMLS-Metadaten für die eigene Ontologie übernommen werden. UMLS kann auch als Hilfe genutzt werden, um Gandr-Klassifikationen zu überprüfen. Ein automatisierter UMLS-Konzept-Abgleich oder Integration in die eigene Ontologie ist dagegen schwer zu realisieren, da man mit dem UMLS-Plugin u.U. Antonyme findet, die aufgrund fehlender Slots nicht automatisch erkannt werden können (siehe Abb. 18).

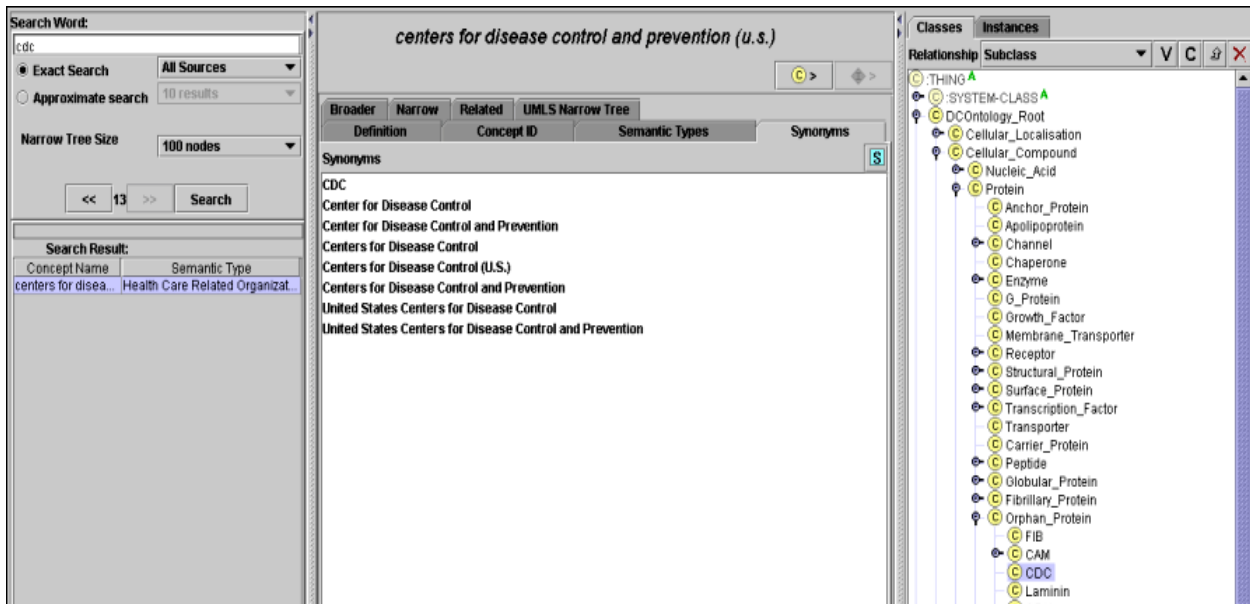


Abb. 18: Import von UMLS-Konzepten mit dem UMLS Tab Plugin. Darstellung eines zum Gandr "CDC" (*cell division cycle*)-Konzept antonymen UMLS-"CDC"-Konzepts. Dieses steht für *Center for Disease Control* und sollte mithin nicht als "CDC"(Cell_Division_Cycle)-Konzept in die Gandr-Ontologie (links) übernommen werden.

Da das UMLS-Plugin die Protégé Metakonzept-Architektur verändert und alle damit erstellten Konzepte zu Instanzen spezieller UMLS-Tab Metaklassen werden, läßt sich die damit erstellte Ontologie nicht mehr so leicht in andere Formate konvertieren. Von einer UMLS-compliance und Integration von UMLS-Konzepten wurde daher abgesehen.

4.8.4 MGED, MIAME und MAGE

Die vielen unterschiedlichen Microarray-Technologien mit jeweils wiederum einer Vielzahl von Parametern führen dazu, daß die über Microarrays gewonnenen Ergebnisse schwer reproduzier- und vergleichbar sind. Daher wird versucht, diese Daten über Ontologien zu standardisieren und in Onlinedatenbanken vereinheitlicht verfügbar zu machen. Das MGED-Konsortium entwickelt einen entsprechenden Datenstandard MIAME (*Minimum Information About Microarray Experiments*). Er stellt Richtlinien, welche Informationen zur eindeutigen Beschreibung von Microarray-Ergebnissen benötigt werden und soll Kerdatentypen, die allen Microarray-Experimenten gemein sind, semantisch einheitlich beschreiben [101]. Das entsprechende Microarray Datenaustauschformat MAGE-ML (*microarray and gene expression markup language* [102]) wird von den großen Expressionsbibliotheken *stanford microarray database*, SMD [103] und *array express* [104] sowie durch einige neuere Microarray-LIMS genutzt [4]. Was Umfang und Sachdomäne betrifft, divergieren Gandr und MIAME dennoch erheblich. MIAME strukturiert Eigenschaften von Microarray-Experimenten, enthält also keine Konzepte

zur Gen- oder Proteinannotation und besteht aus weniger als hundert Konzepten. Was die semantische Tiefe des Repräsentationsformalismus angeht, ist MIAME mit der Gandr-Ontologie vergleichbar. Bei Kenntnis der zugrundeliegenden Formate XML, XMI und UML ist es ferner möglich, Gandr-Konzepte über das XMI-Format in das MAGE-OM oder über das XML-Export-Tab in MAGE-ML zu integrieren. Hier exemplarisch die Zuordnung der Gandr-Konzepte "Sample_Origin" und "Exogenous_Compound" unter entsprechende MAGE-Konzepte:

```
DescriptionPackage: Description:OntologyEntry:← bioinf.mdc-berlin.de:Gandr:<object>
```

```
BioMaterialPackage: BioMaterial:BioSample ← Gandr:SampleOrigin
```

```
BioMaterial:Treatment:CompoundMeasurement:Compound ← Gandr:ExogenousCompound
```

4.8.5 TAMBIS

TAMBIS, *transparent access to multiple bioinformatics information sources* [28], ist ein intelligentes *information retrieval web-interface* für den einheitlichen Zugriff auf die heterogenen molekularbiologischen Datenbanken Swissprot, Enzyme, Cath, Prosite und die Anwendung Blast. Die Tambis-Ontologie (TaO) soll den Benutzer von den unterschiedlichen primären Quell-Datenbanken und Anwendungen bzw. deren Zugriffsmethodiken und Datentypen abschirmen. Der Benutzer erstellt, ähnlich wie in Gandr, graphische Anfragen an die Datenbanken, wobei der *tambis ontology browser* als visuelle Schnittstelle zur Auswahl und Verknüpfung von, einer Suche entsprechenden, TaO-KR-Ideomen dient [22]. TaO enthält ca. 1500 molekularbiologische aber auch relativ viele bioinformatische Konzepte und Relationen. TaO dient als *linker* zu Wissen in Quelldatenbanken, indem sie die Formulierung von Metadatenbank-Anfragen ermöglicht. Wie Gandr erlaubt TaO die ontologiebasierte Formulierung komplexer Anfragen in einer graphischen und intuitiven, d.h. schnellen und einfachen, Art und Weise. Die zunächst Quelldatenbank-unabhängigen Anfragen werden intern in der *description logic*-Sprache OIL (*ontology inference layer*) formalisiert. Aufgrund dieser Anfragen ermittelt TAMBIS dann ontologie- bzw. wissensbasiert die für die Anfrage geeigneten Quelldatenbanken, erstellt entsprechende Anfragen und generiert die Antwort. Der Hauptunterschied zu konventionellen *retrieval*-Systemen wie SRS ist, daß nicht der Nutzer, sondern das System die Quelldatenbanken und die Reihenfolge der *sub queries* auswählt. Wie Gandr bietet TAMBIS dem Nutzer Hilfestellung über Auswahlmöglichkeiten, die durch semantische *constraints* überprüft werden. Die TAMBIS-Anfrageschnittstelle macht dem Nutzer so Domänenwissen zugänglich, das direkt zur Anfragestellung genutzt werden kann. Von dem unterschiedlichen Anwendungsschwerpunkt (Integration des Zugriffs auf heterogene Datenbanken) einmal abgesehen, ist das Tambis-System, was die verwendeten ontologischen Technologien und Abfragemodalitäten betrifft, noch am ehesten mit dem GandrKB-System zu

vergleichen. Beiden Systemen liegt eine komplexe Ontologie zugrunde, die über eine Anfrageschnittstelle ein intelligentes und graphisches *query-building* ermöglicht. Da die TaO neben sequenzorientierten Gendaten viele Datenbank-Struktur-(Meta-)Informationen enthält, ist sie jedoch nicht für die Genannotation geeignet. Ein größerer Teil der TaO formalisiert zudem anwendungsbezogene Informationen zu DB-Zugangstechniken und Sequenzanalyse-Anwendungen, die für die Genannotation und damit für den GandrKB-Ansatz sekundär sind. Weiter ist die Ontologie in DL formalisiert, was ihre Nutzungsmöglichkeit durch Nicht-Experten erheblich einschränkt.

4.9 Ausblick

Hier sollen noch einige potentielle Anwendungen beschrieben werden, die weniger im unmittelbaren Anwendungsfokus "Genannotation" liegen. Einiges davon wurde bereits implementiert, anderes soll lediglich vorgestellt werden, da sich hier einige interessante Perspektiven für zukünftige Forschungen eröffnen.

4.9.1 Ontologie-induzierte Konzept-Ikonographien

Bilder sagen mehr als tausend Worte. Das gilt auch im Bezug auf Stoffwechsel- und ST-Karten; aber zu einem Bild gibt es oft auch tausend verschiedene Interpretationen. Bei den durch die vorgestellten Visualisierungsmethoden erstellten Graphiken werden semantische Beschränkungen lediglich über Bezeichner für Konzepte und Relationen dargestellt. Es wäre zu Überlegen, ob man den Interpretationsspielraum nicht noch weiter über ontologisch standardisierte und graphisch definierte Abbildungen (*icons*) bzw. deren Eigenschaften einschränken und einer einheitlicheren Interpretation zugänglich machen könnte. In Publikationen benutzt bisher jede Arbeitsgruppe andere Formen, um die Bestandteile eines zu beschreibenden Stoffwechselsystems graphisch darzustellen. Neben "Molecular Interaction Maps" [88] und Schaltplan-basierten Visualisierungen aus der Elektrotechnik [105] gibt es bisher kaum Ansätze für derartige standardisierte Visualisierungen von Genprodukten in Stoffwechsel- oder Signaltransduktions-Graphiken. Dies führt dazu, daß sich der Wissenschaftler bei jeder Darstellung neu auf die jeweilig benutzten Abbildungsbestandteile und Formalisierungen einstellen muß. Wenn man einen Standard etablieren könnte, der über eine Ontologie festlegt, wie bestimmte Komponenten-Typen in Graphiken abgebildet werden, so würde das zu schnellerem und aufgrund des engeren Interpretationsspielraumes genauere Verständnis unter den Anwendern führen. Solche "objektorientierten Graphiken" wären automatisch aus ontologischen Wissensbanken herleitbar, interaktiv bearbeitbar und

gegebenenfalls in andere Formate konvertierbar. So ein datengetriebener Darstellungs-Standard könnte sich aus der vorgestellten Gandr-Ontologie ableiten lassen. Konzepten einer bestimmten Hierarchiestufe würde ein *icon* bzw. *icon*-Element zugeordnet, das in seiner Grundform an alle Subkonzepte und Instanzen vererbt wird. Graduelle Unterschiede könnten über verschiedene analoge Graphik-Eigenschaften kodiert werden, z.B. über Größe, Farbverläufe oder Kontraste. Konzentrationsangaben, Mengenangaben oder die "Wichtigkeit" einer zellulären Komponente als Knotenpunkt bzw. deren "*hub*-Funktion" innerhalb eines Signalweges könnten über die Darstellungsgröße visualisiert werden. Der Ort der Darstellung eines *icons* könnte die Position der Stoffwechselkomponenten-Instanz bzw. ihre Relation zu einem "Cellular_Localisation"-Konzept kodieren. Visualisierungen in spezifischen "Cellular_Localisation"-Konzepten entsprechenden Bildbereichen gäben dem Nutzer unmittelbaren Aufschluß über Aufenthalts- und potentielle Reaktions- bzw. Interaktions-Kompartimente. Die den Molekularbiologen und Mediziner besonders interessierenden Genfunktionen und zellulären Lokalisationen der Genprodukte wären aus so einer Bilddarstellung unmittelbar ersichtlich. Die eindeutige Spezifizierung müßte weiterhin über Text, z.B. die probe set ID und den Gensymbol-Namen, erfolgen. Ausgangspunkt für derartige *icons* könnten die von der Firma Biocarta[®] zur Verfügung gestellten Freelance Graphics[®]-Objekte (<http://www.biocarta.com/genes/index.asp>) sein. Diese ikonographischen Gestaltungselemente implizieren bestimmte Funktionen, Strukturen oder Gen-Gen-Interaktionen und sollen der Erstellung von interaktiven und einheitlich interpretierbaren Pathway-Diagrammen, wie in Abb. 17 gezeigt, dienen. Probleme könnten bei derartigen Visualisierungen die Darstellung emergenter bzw. synergetischer Eigenschaften sowie größerer Mengen parallel zugewiesener potentieller Funktionen (z.B. Modifikationen und Multimerisierungen) bereiten, die ja zudem perspektivenabhängig wäre.

4.9.2 Internetseiten-Annotation im *semantic web*-Ansatz

Die Gandr Ontologie könnte als Beschreibungssprache zur Annotation von Texten und Internetseiten benutzt werden. Die Nutzung des Internet und das IR von dezentralisiert gespeicherten Informationsquellen, insbesondere Internetseiten, wird für Biowissenschaftler immer wichtiger [106]. Der *semantic web*-Ansatz soll, nach dem "www-Erfinder" Tim Berners-Lee, internetständige Daten für Menschen und Maschinen lesbar machen und die automatisierte Sichtung und intelligente, d.h. wissensbasierte Weiterverarbeitung verteilter Informationen ermöglichen (<http://www.semanticweb.org>). Er soll das Auffinden, Zusammenfassen und Vergleichen dezentraler Inhalte und ihre deduktive Weiterverarbeitung bzw. eine automatische Wissensgenerierung ermöglichen. Das Werkzeug GATE [107] erlaubt beispielsweise das *natural*

language processing von Internetseiten und Texten unter Anwendung der Ontologie innerhalb eines Gazetteers, der Textstellen als Instanzen ontologischer Konzepte zuordnet und hierüber generalisieren kann. Das erleichtert spätere IE-Prozessierungen wie z.B. *named entity recognition* und *coreference resolution*. Um diese Vorteile nutzen zu können, müssen internetständige Daten zunächst mit Wissen in Form von ontologischen Metadaten annotiert werden. Als strukturgebende Markupsprache für dieses *semantic web* schlägt die OMG Ontologien im OWL-Format vor. Momentan sind diese Ontologien zwar nur für fachspezifische Teildomänen verfügbar; es ist jedoch absehbar, daß über die zunehmende Etablierung des OWL-Standards und entsprechender Verarbeitungs- und *ontology-merging*-Werkzeuge (z.B. *reasoner* und PROMPT) nach und nach ein Netz aus ineinander verwobenen Ontologien entsteht, die später zusammen die semantisch definierte Beschreibung des gesamten Wissens im www ermöglichen könnten. Anfragen an Internet-Suchmaschinen werden dann nicht mehr einfach durch eine unzusammenhängende Folge von Stichwörtern, sondern in einer graphischen konzept- bzw. inhaltsbasierten Anfragesprache, wie im Gandr-System implementiert, gestellt. Dies ermöglicht auch komplexere Anfragen nach semantischen Verbindungen in einer intuitiven Weise, wie sie mit traditionellen Suchmaschinen, wie z.B. Google, nicht zu formulieren sind und somit einen inhaltlich präziseren Zugang zu internetständigen Daten. Die Gandr-Ontologie könnte als *semantic-web*-Beschreibungssprache genutzt werden, indem sie in RDF oder OWL konvertiert und Wörter in Texten oder HTML-Seiten z.B. mit den Konzepten entsprechenden Tags und den Slots entsprechenden Attributen versehen bzw. annotiert werden. Solche *semantic web*-Annotationen könnten beispielsweise mit GATE erstellt werden. Im *semantic web*-Ansatz mit Gandr-KR-Ideomen annotierte Texte und Internetseiten könnten so unter Nutzung der ontologischen Semantik effizienter ausgewertet werden.

4.9.3 Diskriminanzanalysen, Gruppierungsverfahren und maschinelles Lernen

Eine gängige Anwendung von Taxonomien und Ontologien stellen statistische Gruppierungsanalysen dar [108, 109, 110]. Dabei wird die Verteilung der Instanzen auf die Annotations-Konzepte statistisch untersucht. Ontologische Konzepte können auch als Grundlage für maschinelles überwachtes oder unüberwachtes Lernen verwendet werden. Überwachtes Lernen wurde bereits für die Vorhersage funktioneller Genklassen aus Microarray-Expressionsdaten eingesetzt [111], wobei Gene Ontology-Konzepte die zu erlernenden funktionellen Klassen definierten. Das Vorhersage-Modell lernte aufgrund von Microarray-Expressionsprofilen und GO-Genfunktionen bekannter Gene (Lerndatensatz), auf die Funktion unbekannter Gene (Validierungsdatsatz) zu schließen, zu denen nur die Expressionsdaten

vorlagen. Die in diesem überwachten Lernmodell erlernten Beziehungen zwischen Expressionsprofil und Genfunktion können also zur Vorhersage, d.h. letztendlich zu einer statistischen Art automatischer Genannotation mit Genfunktionen, genutzt werden. Für derartige Ansätze eignet sich die GandrKB in besonderem Maße, da die bereitgestellten *classifier* ontologisch besonders fundiert und konsistent sind.

4.10 Zusammenfassung

Die zunehmende Anwendung automatisierter Datenerfassungsmethoden in der molekularen Medizin, insbesondere der Microarray-Technologie, läßt eine effiziente Auswertung des resultierenden Datenmassivs nur noch mit Unterstützung durch automatische Verfahren wie "Datenbergbau" (*data mining*) und Wissensaufschluß (*knowledge discovery*) zu. Die Microarray-Auswertung beginnt oft mit *information retrieval*-Ansätzen, welche die Datenmassen auf eine im Hinblick auf eine bestimmte Fragestellung besonders interessante und überschaubare Menge von Genen bzw. probe set IDs reduzieren sollen. Voraussetzung für eine effiziente Suche im Datenbestand ist jedoch eine Standardisierung und Semantisierung bzw. Formalisierung der Daten über entsprechende Datenformate. Hier wird eine Ontologie als standardisiertes und semantisch definiertes Repräsentationskonstrukt vorgestellt, welches die Formalisierung von Fachwissen in einem interaktiven Wissensmodell erlaubt, das umfassend abgefragt, konsistent interpretiert und gegebenenfalls automatisiert Weiterverarbeitet werden kann. Anhand einer molekularbiologischen Ontologie aus 1179 hierarchisch strukturierten Begriffen und am Beispiel des Toll-Like Receptor-Signalwegs wird aufgezeigt, wie ein objektorientiertes Beschreibungsvokabular zur Annotierung und Modellierung von Genen auf Affymetrix[®]-Microarrays genutzt werden kann. Hierbei kommt eine lediglich leicht modifizierte Version des *open-source* Wissensbank-Editors Protégé-2000 zum Einsatz. Die Annotationsbegriffe der Sachdomäne "Immunbiologie dendritischer Zellen" werden über ontologische Konzepte, deren Eigenschaften und deren semantische Verbindungen (relationale Slots) modelliert. Annotation bedeutet hier nicht mehr ein Gen mit einem unformalen Beschreibungstext zu versehen, sondern es formal in einen definierten funktionalen Kontext einzubetten. In der Anwendung der Wissensbank entspricht eine Annotation einem "*drag and drop*" von Genen in ontologische, die Funktion dieser Gene beschreibende, Konzepte. Die weitergehende kontextuale Annotation erfolgt über eine Vernetzung der Gene zu anderen Konzepten oder Genen. Durchgeführte Annotationen werden über zugrundeliegende ontologische *constraints* überprüft, was deren Konsistenz bzw. Qualität verbessert. Das so erstellte vernetzte Wissensmodell (die *knowledgebase*) ermöglicht ein inhaltsbasiertes, assoziatives und kontextgeleitetes "Wissens-Browsing". Ontologisch annotierte Gendaten erlauben auch die Anwendung automatischer datengetriebener Visualisierungsstrategien, wie am Beispiel semantischer Netze gezeigt wird. Eine ontologische Anfrageschnittstelle erlaubt auch semantisch komplexe Anfragen an den Datenbestand bei erhöhter Trefferquote und Präzision. Über die Konzept-Hierarchie, bzw. die hierüber ermöglichte Subsumption impliziter Annotationen, gewährleistet die ontologische

Anfrageschnittstelle einen besseren Anfrage-*recall* als üblicherweise hierfür genutzte Tabellenkalkulationsprogramme. Komplexe Anfragen der Form "Zeige mir alle zweifach hochregulierten Gene, die Proteinkinasen kodieren und deren Genprodukte sich in der Kernmembran befinden" können graphisch unter Auswahl entsprechender ontologischer Begriffe erstellt werden, ohne daß der Nutzer eine komplizierte Datenbankabfragesprache wie SQL erlernen muß. Zudem werden dem Nutzer während der Anfrageformulierung Hilfestellungen gegeben und seine Eingaben auch hier auf ontologische Konsistenz überprüft. Das System ermöglicht neben dem Aufbau einer Anfragenbibliothek zudem die Beantwortung verschachtelter Anfragen. Hierbei dienen Ergebnisse vorhandener Anfragen als Bestandteil neuer Anfragen.

Ein weiterer Vorteil der Wissensbank ist die framebasierte Darstellung, die nicht nur intuitiver und übersichtlicher ist als bisher verwandte Tabellen, sondern auch eine adäquate Darstellung verschiedenster Datentypen erlaubt; die Integration externer Websites über *deeplinks* und den Versatz von Datenelementen mit Multimedia-Inhalten wie Graphiken, Film und Ton zum Beispiel.

Die im Rahmen dieser Arbeit vorgestellte molekularbiologische Annotationsplattform und Wissensbank soll den semantischen Erfordernissen einer zunehmend an systemischer und holistischer Betrachtung orientierten funktionalen Genomik gerecht werden und nicht zuletzt einen Weg zu einer integrativeren Bioinformatik aufzeigen.

Literatur

- [1] Baxevanis, A. D. (2003): The Molecular Biology Database Collection: 2003 update, *Nucleic Acids Res* (Band 31), Nr. 1, Seite 1-12. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12519937
- [2] Barnes, J. C. (2002): Conceptual biology: a semantic issue and more, *Nature* (Band 417), Nr. 6889, Seite 587-8. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12050632
- [3] Naisbitt, J., Aburdene, P. (1991): *Megatrends 2000. Zehn Perspektiven für den Weg ins nächste Jahrhundert. Vorhersagen für unsere Zukunft.*, ECON Taschenbuch Verlag, Düsseldorf, Wien.
- [4] Schober, D. (2002): Microarrays, Genexpressionsanalyse und Bioinformatik, *BioSpektrum*, Nr. 3, Seite 307-310.
- [5] Brown O., Botstein D. (1999): Exploring the new world of the genome with DNA microarrays, *Nature Genetics* (Band 21), Seite 33.
- [6] DeRisi J., Iyer V., Brown P. (1997): Exploring the metabolic and genetic control of gene expression on a genomic scale, *Science* (Band Vol 278), Seite 680-686.
- [7] Spalding, K. J. (1931): *Talks on philosophy*, B. Blackwell, Oxford.
- [8] Heidegger, M. (1958): *Einführung in die Metaphysik*, 2. Auflage, M. Niemeyer, Tübingen.
- [9] Gram, S. (1968): *Kant, ontology & the a priori*, Northwestern University Press, Evanston Ill.
- [10] Munitz, Milton Karl (1973): *Logic and ontology*, New York University. Dept. of Philosophy, New York University Press, New York, ISBN: 0814753639.
- [11] Gomez-Perez, Asuncion; Fernandez-Lopez, Mariano und Corcho, Oskar (2004): *Ontological Engineering*, 1. Auflage, Wu, Xindong, Springer, London, ISBN: 1-85233-551-3.
- [12] Uschold, Mike und Grüninger, Michael (1996): *Ontologies: Principles, Methods and Applications*, *Knowledge Engineering Review* (Band 11) Seite 93-155, o.A., München. URL: <http://citeseer.ist.psu.edu/uschold96ontology.html>
- [13] Gruber, T. R. (1993): A translation approach to portable ontologies, *Knowledge Acquisition* (Band 2), Nr. 5, Seite 199-220.
- [14] Guarino, N. und Giaretta, P. (1995): *Ontologies and knowledge bases - Towards a terminological clarification*, Mars, N. J. I., *Towards Very Large Knowledge Base*, IOS Press, Amsterdam.
- [15] Sowa, John F. (1995): Top-level ontological Categories, *International Journal of Human-Computer Studies* (Band 43), Nr. 5/6, Seite 669-686.
- [16] Swartout, B.; Patil, R.; Knight, K. und Russ, T. (1997): *Toward Distributed Use of Large-Scale Ontologies*, *Ontological Engineering*, AAI-97 Spring Symposium Series.
- [17] Vetter, Max (1990): *Strategien der Anwendungsentwicklung*, Teubner, Stuttgart, ISBN: 3-519-12489-0.
- [18] Giarratano, J. und Riley, G. (1994): *Expert Systems: Principles and Programming*, PWS Publication, ISBN: 0-534-93744-6.
- [19] Chaudhri Vinay K., Farquhar Adam, Fikes Richard, Karp Peter D., Rice James P. (1998): *OKBC: A Programmatic Foundation for Knowledge Base Interoperability*, Proc. AAI'98 Conference, Madison, USA.
- [20] Lambrix, P.; Habbouche, M. und Perez, M. (2003): Evaluation of ontology development tools for bioinformatics, *Bioinformatics* (Band 19), Nr. 12, Seite 1564-71. URL:

- http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12912838
- [21] Grosso, W. E.; Eriksson, H.; Tu, S. und Ferguson, R. W. (1999): Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000), Proc. of the 12th International Workshop on Knowledge Acquisition, Modeling and Management (KAW'99), Banff, Canada.
- [22] Baker, P. G.; Goble, C. A.; Bechhofer, S.; Paton, N. W.; Stevens, R. und Brass, A. (1999): An ontology for bioinformatics applications, *Bioinformatics* (Band 15), Nr. 6, Seite 510-20. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=10383475
- [23] Smith, C. L.; Goldsmith, C. A. und Eppig, J. T. (2005): The Mammalian Phenotype Ontology as a tool for annotating, analyzing and comparing phenotypic information, *Genome Biol* (Band 6), Nr. 1, Seite 7. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15642099
- [24] Schulze-Kremer, S. (1997): Adding semantics to genome databases: towards an ontology for molecular biology, *Proc Int Conf Intell Syst Mol Biol* (Band 5), Seite 272-5. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=9322049
- [25] Barsalou, T. (1989): An object-based architecture for biomedical expert database systems, *Comput Methods Programs Biomed* (Band 30), Nr. 2-3, Seite 157-68. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=2582749
- [26] Harris, M. A.; Clark, J.; Ireland, A.; Lomax, J. und Ashburner, M. (2004): The Gene Ontology (GO) database and informatics resource, *Nucleic Acids Res* (Band 32 Database issue), Seite D258-61. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=14681407
- [27] Ingenerf, J.; Reiner, J. und Seik, B. (2001): Standardized terminological services enabling semantic interoperability between distributed and heterogeneous systems, *Int J Med Inf* (Band 64), Nr. 2-3, Seite 223-40. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11734388
- [28] Stevens, Robert; Baker, Patricia; Bechhofer, Sean; Ng, Gary; Jacoby, Alex; Paton, Norman W.; Goble, Carole A. und Brass, Andy (2000): TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources, *Bioinformatics* (Band 16), Nr. 2, Seite 184-186. URL: <http://bioinformatics.oupjournals.org/cgi/content/abstract/16/2/184>
- [29] Hvidsten, T. R.; Komorowski, J.; Sandvik, A. K. und Laegreid, A. (2001): Predicting gene function from gene expressions and ontologies, *Pac Symp Biocomput*, Seite 299-310. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11262949
- [30] Banchereau, J.; Briere, F.; Caux, C.; Davoust, J.; Lebecque, S.; Liu, Y. J.; Pulendran, B. und Palucka, K. (2000): Immunobiology of dendritic cells, *Annu Rev Immunol* (Band 18), Seite 767-811. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=10837075
- [31] Banchereau, J. und Steinman, R. M. (1998): Dendritic cells and the control of immunity, *Nature* (Band 392), Nr. 6673, Seite 245-52. URL:

- http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=9521319
- [32] Lanzavecchia, A. und Sallusto, F. (2001): Regulation of T cell immunity by dendritic cells, *Cell* (Band 106), Nr. 3, Seite 263-6. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11509174
- [33] Reis e Sousa, C. (2001): Dendritic cells as sensors of infection, *Immunity* (Band 14), Nr. 5, Seite 495-8. URL: <http://www.ncbi.nlm.nih.gov/htbin-post/Entrez/query?db=m&form=6&dopt=r&uid=11371351>
- [34] Pulendran, B.; Palucka, K. und Banchereau, J. (2001): Sensing pathogens and tuning immune responses, *Science* (Band 293), Nr. 5528, Seite 253-6. URL:
<http://www.ncbi.nlm.nih.gov/htbin-post/Entrez/query?db=m&form=6&dopt=r&uid=11452116>
- [35] Mellman, I. und Steinman, R. M. (2001): Dendritic cells: specialized and regulated antigen processing machines, *Cell* (Band 106), Nr. 3, Seite 255-8. URL:
<http://www.ncbi.nlm.nih.gov/htbin-post/Entrez/query?db=m&form=6&dopt=r&uid=11509172>
- [36] Ardavin, C.; Martinez del Hoyo, G.; Martin, P.; Anjuere, F.; Arias, C. F.; Marin, A. R.; Ruiz, S.; Parrillas, V. und Hernandez, H. (2001): Origin and differentiation of dendritic cells, *Trends Immunol* (Band 22), Nr. 12, Seite 691-700. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11739000
- [37] Liu, Y. J.; Kanzler, H.; Soumelis, V. und Gilliet, M. (2001): Dendritic cell lineage, plasticity and cross-regulation, *Nat Immunol* (Band 2), Nr. 7, Seite 585-9. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11429541
- [38] Shortman, K. und Liu, Y. J. (2002): Mouse and human dendritic cell subtypes, *Nat Rev Immunol* (Band 2), Nr. 3, Seite 151-61. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11913066
- [39] Ju, X. S.; Hacker, C.; Scherer, B.; Redecke, V.; Berger, T.; Schuler, G.; Wagner, H.; Lipford, G. B. und Zenke, M. (2004): Immunoglobulin-like transcripts ILT2, ILT3 and ILT7 are expressed by human dendritic cells and down-regulated following activation, *Gene* (Band 331), Seite 159-64. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15094202
- [40] Ju, X. S. und Zenke, M. (2004): Gene expression profiling of dendritic cells by DNA microarrays, *Immunobiology* (Band 209), Nr. 1-2, Seite 155-61. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15481149
- [41] Hacker, C.; Kirsch, R. D.; Ju, X. S.; Hieronymus, T.; Gust, T. C.; Kuhl, C.; Jorgas, T.; Kurz, S. M.; Rose-John, S.; Yokota, Y. und Zenke, M. (2003): Transcriptional profiling identifies Id2 function in dendritic cell development, *Nat Immunol* (Band 4), Nr. 4, Seite 380-6. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12598895
- [42] Ju, X. S. und Zenke, M. (2003): Differentiation of human antigen-presenting dendritic cells from CD34+ hematopoietic stem cells in vitro, *Methods Mol Biol* (Band 215), Seite 399-407. URL:

- http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12512315
- [43] Felzmann, T.; Gardner, H. und Holter, W. (2002): Dendritic cells as adjuvants in antitumor immune therapy, *Onkologie* (Band 25), Nr. 5, Seite 456-464.
- [44] Lipshutz R., Fodor S., Gingeras T., Lockhart D. (1999): High density synthetic oligonucleotide arrays, *Nature Genetics* (Band 21), Seite S 20.
- [45] Gruber, T.R. (1995): Towards Principles for the Design of Ontologies used for Knowledge Sharing, *International Journal of Human-Computer Studies* (Band 43), Seite 907-928.
- [46] Liu, Y. J. (2001): Dendritic cell subsets and lineages, and their functions in innate and adaptive immunity, *Cell* (Band 106), Nr. 3, Seite 259-62. URL: <http://www.ncbi.nlm.nih.gov/htbin-post/Entrez/query?db=m&form=6&dopt=r&uid=11509173>
- [47] Ashburner, M.; Ball, C. A.; Blake, J. A.; Botstein, D.; Butler, H.; Cherry, J. M.; Davis, A. P.; Dolinski, K.; Dwight, S. S.; Eppig, J. T.; Harris, M. A.; Hill, D. P.; Issel-Tarver, L.; Kasarskis, A.; Lewis, S.; Matese, J. C.; Richardson, J. E.; Ringwald, M.; Rubin, G. M. und Sherlock, G. (2000): Gene ontology: tool for the unification of biology. The Gene Ontology Consortium, *Nat Genet* (Band 25), Nr. 1, Seite 25-9. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=10802651
- [48] Fukuda, Ken-ichiro und Takagi, Toshihisa (2001): Knowledge representation of signal transduction pathways, *Bioinformatics* (Band 17), Nr. 9, Seite 829-837. URL: <http://bioinformatics.oupjournals.org/cgi/content/abstract/17/9/829>
- [49] Perez-Iratxeta, C.; Bork, P. und Andrade, M. A. (2001): XplorMed: a tool for exploring MEDLINE abstracts, *Trends Biochem Sci* (Band 26), Nr. 9, Seite 573-5. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11551795
- [50] Ashburner, M. und Lewis, S. (2002): On ontologies for biologists: the Gene Ontology--untangling the web, *Novartis Found Symp* (Band 247), Seite 66-80; discussion 80-3, 84-90, 244-52. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12539950
- [51] Kelso, Janet; Visagie, Johann; Theiler, Gregory; Christoffels, Alan; Bardien-Kruger, Soraya; Smedley, Damian; McCarthy, Mark; Hide, Tania und Hide, Winston (2003): eVOC: A Controlled Vocabulary for Gene Expression Data, *Genome Research* (Band 13), Seite 1222-1230.
- [52] Liu, G.; Loraine, A. E.; Shigeta, R.; Cline, M.; Cheng, J.; Valmeekam, V.; Sun, S.; Kulp, D. und Siani-Rose, M. A. (2003): NetAffx: Affymetrix probesets and annotations, *Nucleic Acids Res* (Band 31), Nr. 1, Seite 82-6. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12519953
- [53] Eisen, M. B.; Spellman, P. T.; Brown, P. O. und Botstein, D. (1998): Cluster analysis and display of genome-wide expression patterns, *Proc Natl Acad Sci U S A* (Band 95), Nr. 25, Seite 14863-8. URL: <http://www.ncbi.nlm.nih.gov/htbin-post/Entrez/query?db=m&form=6&dopt=r&uid=9843981>
- [54] Gansner, E. R. und North, S. C. (1999): An open graph visualization system and its applications to software engineering, *Software - Practice and Experience* (Band 00), Nr. S1, Seite 1-5.

- [55] Alani, H. (2003): TGVizTab: An Ontology Visualisation Extension for Protégé, Proceedings of Knowledge Capture (K-Cap'03), Workshop on Visualization Information in Knowledge Engineering (Band 3).
- [56] Lambrix, P. und Edberg, A. (2003): Evaluation of ontology merging tools in bioinformatics, Pac Symp Biocomput, Seite 589-600. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12603060
- [57] Yeh, I.; Karp, P. D.; Noy, N. F. und Altman, R. B. (2003): Knowledge acquisition, consistency checking and concurrency control for Gene Ontology (GO), Bioinformatics (Band 19), Nr. 2, Seite 241-248. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12538245
- [58] von Foerster, H. (1998): Wahrheit ist die Erfindung eines Lügners, Carl-Auer-Systeme Verlag, München.
- [59] Jones, D.M., Paton, R.C. (1999): Toward Principles for the Representation of Hierarchical Knowledge in Formal Ontologies, Data and Knowledge Engineering (Band 31), Nr. 2, Seite 99-113.
- [60] Collins, R. und Quillian, M.R. (1969): Retrieval time from semantic memory, Journal of Verbal Learning and Verbal Behavior, Nr. 8, Seite 240-248.
- [61] Collins, R. und Loftus, E.F. (1975): A spreading activation theory of semantic processing, Psychological Review, Nr. 82, Seite 407-428.
- [62] Bartlett, F. C. (1932): Remembering, Cambridge University Press, Cambridge.
- [63] Hebb, D. O. (1961): Distinctive features of learning in the higher animal, Delafresnaye, J. F., Brain Mechanisms and Learning, Blackwell, London.
- [64] Gangemi, Aldo (2003): Some tools and methodologies for domain ontology building, Comp Funct Genom, Nr. 4, Seite 104-110.
- [65] Hoekstra, R.J. Errors in Modeling & Representation, MSc Thesis in Artificial Intelligence, Department of Computer Science and Law, University of Amsterdam, Amsterdam. URL: <http://www.lri.jur.uva.nl/~rinke/thesis/thesis.pdf>
- [66] Strawson, P.F. (1959): Individuals: An Essay in Descriptive Metaphysics, Routledge, London.
- [67] Korpilahti, T. (2004): Architecture for distributed development of an ontology library, Masters Thesis, Department of Computer Science and Engineering, Helsinki University of Technology, Espoo.
- [68] Guarino, N.; Caburet, S.; Carrara, M. und Garetta, P. (1994): An Ontology of Meta-Level Categories, Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference, San Mateo, CA.
- [69] Breuker, J.; Muntjewerff, A. und Bredewej, B. (1999): Ontological modelling for designing educational systems, Proceedings of the AI-ED 99 Workshop on Ontologies for Educational Systems, Le Mans, France.
- [70] Lenat, Douglas B. und Guha, R. V. (1989): Building large knowledge-based systems : representation and inference in the Cyc project, Addison-Wesley, Reading, Mass., ISBN: 0-201-51752-3.
- [71] Fensel, D (2003): Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce (Band 12/2003), second edition. Auflage, Fensel, D, Springer Verlag, Heidelberg, ISBN: 3540416021.
- [72] Pease, Adam; Niles, Ian und Li, John (2002): The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. URL: <http://ontology.teknowledge.com>

- [73] Biolchini, J. und Patel, V. L. (2004): From thesauri to ontology: knowledge acquisition and organization, *Medinfo* (Band 2004), Seite 1525. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15360354
- [74] McEntire, R.; Karp, P.; Abernethy, N.; Benton, D.; Helt, G.; DeJongh, M.; Kent, R.; Kosky, A.; Lewis, S.; Hodnett, D.; Neumann, E.; Olken, F.; Pathak, D.; Tarczy-Hornoch, P.; Toldo, L. und Topaloglou, T. (2000): An evaluation of ontology exchange languages for bioinformatics, *Proc Int Conf Intell Syst Mol Biol* (Band 8), Seite 239-50. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=10977085
- [75] Rector, A. L. (1999): Terminology and concept representation languages: where are we?, *Artif Intell Med* (Band 15), Nr. 1, Seite 1-4. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=9930613
- [76] Mike Uschold, Robert Jasper (1999): A Framework for Understanding and Classifying Ontology Applications, *Proceedings of the IJCA-99 workshop on Ontologies and Problem Solving Methods (KRR5)*, Stockholm.
- [77] Jacobson, I., Jonsson, P., Christerson, M., Overgaard, G. (1992): *Object-Oriented Software Engineering - A Use Case Driven Approach*, ACM Press Series, Addison Wesley Longman, Upper Saddle River, N.J.
- [78] Cooper, W. S. (1997): On Selecting a Measure of Retrieval Effectiveness, Jones, K. S., Willett, P., *Readings in Information Retrieval*, Morgan Kaufmann.
- [79] Baeza-Yates, R., Ribeiro-Neto, R. (1999): *Modern Information Retrieval*, Pearson Education Limited, London.
- [80] Baddeley, A. D. (1990): *Human Memory: Theory and Practice*, Lawrence Erlbaum Associates, London.
- [81] Broadbent, D. E. (1958): *Perception and Communication*, Pergamon, London.
- [82] Kahneman, D., Tversky, A. (1973): On the psychology of prediction, *Psychology Review* (Band 80), Seite 237-251.
- [83] Wu, Wei und Noble, William S. (2004): Genomic data visualization on the Web, *Bioinformatics* (Band 20), Nr. 11, Seite 1804-1805. URL: <http://bioinformatics.oupjournals.org/cgi/content/abstract/20/11/1804>
- [84] Zhou, M. und Cui, Y. (2004): GeneInfoViz: Constructing and visualizing gene relation networks, *In Silico Biol* (Band 4), Nr. 2, Seite 0026. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15107032
- [85] Takai-Igarashi, T. und Mizoguchi, R. (2004): Cell signaling networks ontology, *In Silico Biol* (Band 4), Nr. 1, Seite 81-7. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15089755
- [86] Demir, E.; Babur, O.; Dogrusoz, U.; Gursoy, A.; Nisanci, G.; Cetin-Atalay, R. und Ozturk, M. (2002): PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways, *Bioinformatics* (Band 18), Nr. 7, Seite 996-1003. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12117798
- [87] Karp, P. D. (2001): Pathway databases: a case study in computational symbolic theories, *Science* (Band 293), Nr. 5537, Seite 2040-4. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11557880

- [88] Kohn, K. W. (1999): Molecular Interaction Map of the Mammalian Cell Cycle Control and DNA Repair Systems, *Molec. Biol. Cell* (Band 10), Nr. 8, Seite 2703-2734.
- [89] Kolpakov, F. A. (2002): BIOUML - Framework for visual modeling and simulation of biological systems, *Proc. Int. Conf. Bioinf. of Genome Regulation and Structure (BGRS'2002)*.
- [90] Wolstencroft, K. J.; Stevens, R.; Taberner, L. und Brass, A. (2005): PhosphaBase: an ontology-driven database resource for protein phosphatases, *Proteins* (Band 58), Nr. 2, Seite 290-4. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15558746
- [91] Midford, Peter (2004): Ontologies for behavior, *Bioinformatics*, Seite 433. URL:
<http://bioinformatics.oupjournals.org/cgi/content/abstract/bth433v1>
- [92] Fong, C. T.; Rosse, C.; Clark, J. I.; Shapiro, L. und Brinkley, J. (2004): An Ontology-based Image Repository for a Biomedical Research Lab, *Medinfo* (Band 2004), Nr. CD, Seite 1598. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15360427
- [93] Demir, E.; Babur, O.; Dogrusoz, U.; Gursoy, A.; Ayaz, A.; Gulesir, G.; Nisanci, G. und Cetin-Atalay, R. (2004): An ontology for collaborative construction and analysis of cellular pathways, *Bioinformatics* (Band 20), Nr. 3, Seite 349-356. URL:
<http://bioinformatics.oupjournals.org/cgi/content/abstract/20/3/349>
- [94] Bodenreider, O.; Mitchell, J. A. und McCray, A. T. (2002): Evaluation of the UMLS as a terminology and knowledge resource for biomedical informatics, *Proc AMIA Symp*, Seite 61-5. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12463787
- [95] Schulze-Kremer, S. (2002): Ontologies for molecular biology and bioinformatics, *In Silico Biol* (Band 2), Nr. 3, Seite 179-93. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12542404
- [96] Stevens, R.; Goble, C. A. und Bechhofer, S. (2000): Ontology-based knowledge representation for bioinformatics, *Brief Bioinform* (Band 1), Nr. 4, Seite 398-414. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11465057
- [97] Sklyar, Nataliya (2001): Survey of existing Bio-ontologies, Technical Report, Dept. of Comp. Science, Universität Leipzig, Leipzig. URL: <http://dol.uni-leipzig.de/pub/2001-30>
- [98] Wroe, C. J.; Stevens, R.; Goble, C. A. und Ashburner, M. (2003): A methodology to migrate the gene ontology to a description logic environment using DAML+OIL, *Pac Symp Biocomput*, Seite 624-35. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12603063
- [99] Cheng, J.; Sun, S.; Tracy, A.; Hubbell, E.; Morris, J.; Valmeekam, V.; Kimbrough, A.; Cline, M. S.; Liu, G.; Shigeta, R.; Kulp, D. und Siani-Rose, M. A. (2004): NetAffx Gene Ontology Mining Tool: a visual approach for microarray data analysis, *Bioinformatics* (Band 20), Nr. 9, Seite 1462-3. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=14962933
- [100] Lindberg, D. A.; Humphreys, B. L. und McCray, A. T. (1993): The Unified Medical Language System, *Methods Inf Med* (Band 32), Nr. 4, Seite 281-91. URL:

- http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=8412823
- [101] Brazma, A.; Hingamp, P.; Quackenbush, J.; Sherlock, G.; Spellman, P.; Stoeckert, C.; Aach, J.; Ansorge, W.; Ball, C. A.; Causton, H. C.; Gaasterland, T.; Glenisson, P.; Holstege, F. C.; Kim, I. F.; Markowitz, V.; Matese, J. C.; Parkinson, H.; Robinson, A.; Sarkans, U.; Schulze-Kremer, S.; Stewart, J.; Taylor, R.; Vilo, J. und Vingron, M. (2001): Minimum information about a microarray experiment (MIAME)-toward standards for microarray data, *Nat Genet* (Band 29), Nr. 4, Seite 365-71. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11726920
- [102] Spellman, P. T.; Miller, M.; Stewart, J.; Troup, C.; Sarkans, U.; Chervitz, S.; Bernhart, D.; Sherlock, G.; Ball, C.; Lepage, M.; Swiatek, M.; Marks, W. L.; Goncalves, J.; Markel, S.; Jordan, D.; Shojatalab, M.; Pizarro, A.; White, J.; Hubley, R.; Deutsch, E.; Senger, M.; Aronow, B. J.; Robinson, A.; Bassett, D.; Stoeckert, C. J., Jr. und Brazma, A. (2002): Design and implementation of microarray gene expression markup language (MAGE-ML), *Genome Biol* (Band 3), Nr. 9, Seite RESEARCH0046. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12225585
- [103] Sherlock, G.; Hernandez-Boussard, T.; Kasarskis, A.; Binkley, G.; Matese, J. C.; Dwight, S. S.; Kaloper, M.; Weng, S.; Jin, H.; Ball, C. A.; Eisen, M. B.; Spellman, P. T.; Brown, P. O.; Botstein, D. und Cherry, J. M. (2001): The Stanford Microarray Database, *Nucleic Acids Res* (Band 29), Nr. 1, Seite 152-5. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11125075
- [104] Rocca-Serra, P.; Brazma, A.; Parkinson, H.; Sarkans, U.; Shojatalab, M.; Contrino, S.; Vilo, J.; Abeygunawardena, N.; Mukherjee, G.; Holloway, E.; Kapushesky, M.; Kemmeren, P.; Lara, G. G.; Oezcimen, A. und Sansone, S. A. (2003): ArrayExpress: a public database of gene expression data at EBI, *C R Biol* (Band 326), Nr. 10-11, Seite 1075-8. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=14744115
- [105] McAdams, H. H. und Shapiro, L. (1995): Circuit simulation of genetic networks, *Science* (Band 269), Nr. 5224, Seite 650-6. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=7624793
- [106] Baxevanis, A. D. und Landsman, D. (1995): The Internet biologist, *Faseb J* (Band 9), Nr. 11, Seite 994. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=7649414
- [107] Cunningham, H. (2000): Software Architecture for Language Engineering, Unpublished PhD thesis, Sheffield Natural Language Processing Group, University of Sheffield. URL: <http://gate.ac.uk/sale/thesis/>
- [108] Cheng, J.; Cline, M.; Martin, J.; Finkelstein, D.; Awad, T.; Kulp, D. und Siani-Rose, M. A. (2004): A knowledge-based clustering algorithm driven by Gene Ontology, *J Biopharm Stat* (Band 14), Nr. 3, Seite 687-700. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15468759
- [109] Adryan, B. und Schuh, R. (2004): Gene-Ontology-based clustering of gene expression data, *Bioinformatics* (Band 20), Nr. 16, Seite 2851-2. URL:

- http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15117760
- [110] Shah, N. H. und Fedoroff, N. V. (2004): CLENCH: a program for calculating Cluster ENriCHment using the Gene Ontology, *Bioinformatics* (Band 20), Nr. 7, Seite 1196-7. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=14764555
- [111] Hvidsten, T. R.; Laegreid, A. und Komorowski, J. (2003): Learning rule-based models of biological process from gene expression time profiles using gene ontology, *Bioinformatics* (Band 19), Nr. 9, Seite 1116-23. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12801872
- [112] H. Alani, S. Kim, D. Millard, M. Weal, und W. Hall, P. Lewis, and N. Shadbolt (2003): Automatic ontologybased knowledge extraction from web documents, *IEEE Intelligent Systems* (Band 18), Nr. 1, Seite 14-21.
- [113] Majoros, W. H.; Subramanian, G. M. und Yandell, M. D. (2003): Identification of key concepts in biomedical literature using a modified Markov heuristic, *Bioinformatics* (Band 19), Nr. 3, Seite 402-7. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12584127
- [114] Maedche, Alexander (2002): *Ontology learning for the semantic Web*, The Kluwer international series in engineering and computer science ; SECS 665, Kluwer Academic Publishers, Boston, ISBN: 0792376560 (alk. paper).
- [115] Le Moigno, S.; Charlet, J.; Bourigault, D.; Degoulet, P. und Jaulent, M. C. (2002): Terminology extraction from text to build an ontology in surgical intensive care, *Proc AMIA Symp*, Seite 430-4. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=12463860
- [116] Li, Q.; Shilane, P.; Noy, N. F. und Musen, M. A. (2000): Ontology acquisition from on-line knowledge sources, *Proc AMIA Symp*, Seite 497-501. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11079933
- [117] Do Amaral, M. B.; Roberts, A. und Rector, A. L. (2000): NLP techniques associated with the OpenGALEN ontology for semi-automatic textual extraction of medical knowledge: abstracting and mapping equivalent linguistic and logical constructs, *Proc AMIA Symp*, Seite 76-80. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=11079848
- [118] Craven, Mark (1998): *Learning to extract symbolic knowledge from the World Wide Web*, School of Computer Science Carnegie Mellon University, Pittsburgh, Pa.
- [119] Buitelaar, Paul; Olejnik, Daniel und Sintek, Michael (2004): A Protégé Plug-In for Ontology Extraction from Text Based on Linguistic Analysis, *Proceedings of the 1st European Semantic Web Symposium (ESWS)*, Heraklion, Greece.
- [120] Brutlag, D. L.; Galper, A. R. und Millis, D. H. (1991): Knowledge-based simulation of DNA metabolism: prediction of enzyme action, *Comput Appl Biosci* (Band 7), Nr. 1, Seite 9-19. URL:
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=2004281

- [121] Peleg, Mor; Yeh, Iwei und Altman, Russ B. (2002): Modelling biological processes using workflow and Petri Net models, *Bioinformatics* (Band 18), Nr. 6, Seite 825-837. URL: <http://bioinformatics.oupjournals.org/cgi/content/abstract/18/6/825>
- [122] Fisher, M. J.; Paton, R. C. und Matsuno, K. (1999): Intracellular signalling proteins as smart' agents in parallel distributed processes, *Biosystems* (Band 50), Nr. 3, Seite 159-71. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=10400267
- [123] M. Gomez, C. Abasolo, E. Plaza (2001): Domain-independent ontologies for cooperative information agents, *Lecture Notes in Artificial Intelligence*, Nr. 2128, Seite 118-129.

Abkürzungen

Abkürzung Text

CAD	Computer aided design
CD	Cluster of differentiation
CLIPS	C-language integrated production system
DAG	Directed acyclic graph
DB	Datenbank
DC	Dendritic cell
DMT	Data mining tool
ER	Entity relationship
FO	Foundational ontology
GUI	Graphical user interface
HUGO	Human genome organisation
I.E.	Information extraction
I.R.	Information retrieval
ICD	International classification of diseases
IFN	Interferon
IL	Interleukin
JVC	Java function calls
KB	Knowledge base
KI	Künstliche Intelligenz
KIF	Knowledge interchange format
KM	Knowledge management
KMS	Knowledge management system
KR	Knowledge representation
LHS	Left hand side
LIMS	Laboratory information management system
MAGE	Microarray and gene expression
MeSH	Medical subject headings
MGED	Microarray gene expression database
MIAME	Minimum information about microarray experiments
NLM	National library of medicine

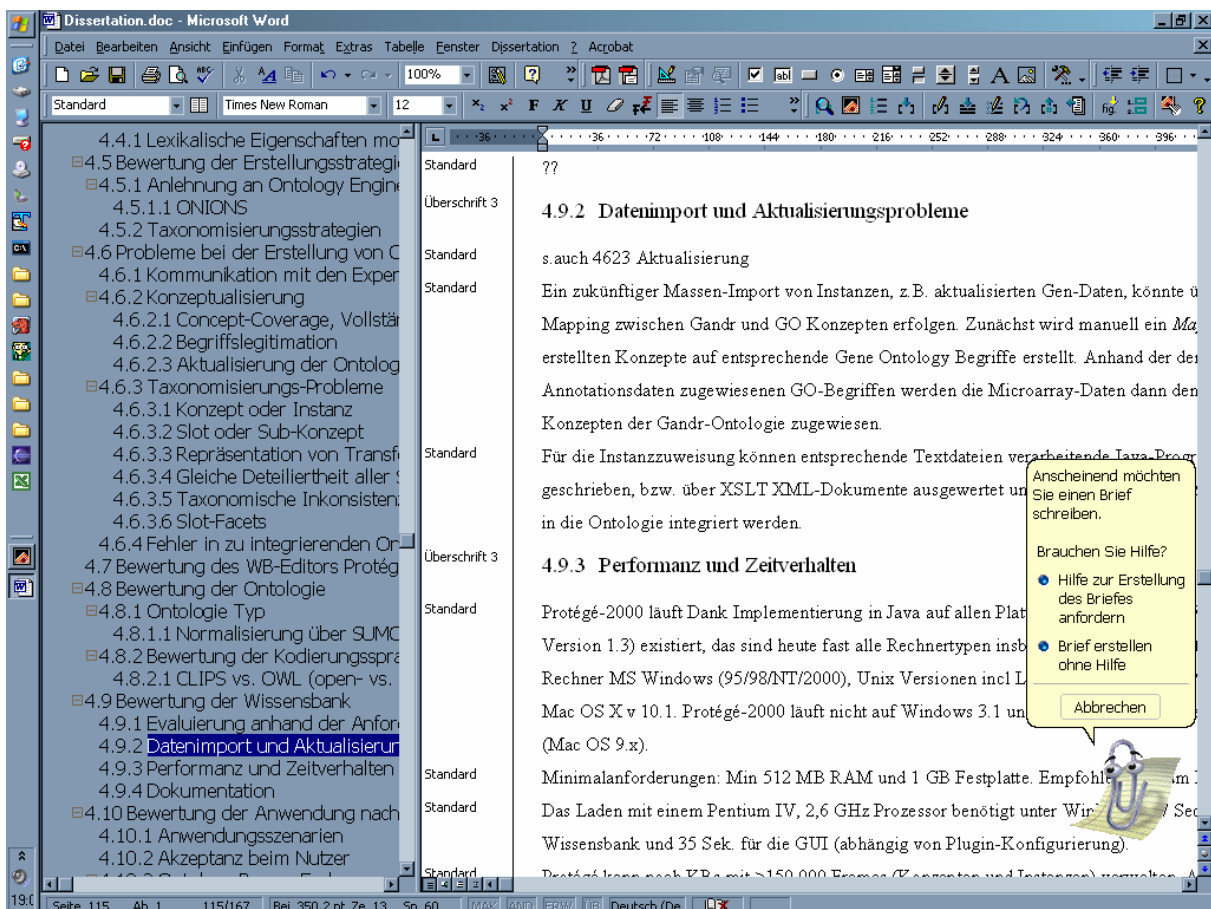
NLP	Natural language processing
OE	Ontology engineering
OIL	Ontology inference layer
OMB	ontology for molecular biology
OMG	Object management group
ONION	Ontological integration of naive sources
OOD	Object-oriented design
OOM	Object-oriented modeling
OOP	Object-oriented programming
OWL	Ontology web language
PAL	Protégé axiom language
PAMP	Pathogen associated molekular pattern
PROMPT	Protégé ontology management and processing tool
Prow	Proteins on the web
PRR	Pattern-recognition receptor
RAR	Retinolsäure Rezeptor
RDMS	Relational database management system
RHS	Right hand side
RNA	Ribonucleic acid
SQL	Structured query language
ST	Signaltransduktion
SUMO	Suggested upper merged ontology
TF	Transcription factor
TLR	Toll-like receptor
TNF	Tumor necrosis factor
TR	Thyroid hormone receptor
UMLS	Unified medical language system
VBA	Visual Basic
WB	Wissensbank
WR	Wissensrepräsentation
XML	Extended markup language

Abbildungsverzeichnis

Abb. 1: Eine Ontologiedefinition nach Sowa.	20
Abb. 2: Beispiel für die Vererbung von Konzept-Eigenschaften in einer Konzept-Taxonomie ..	25
Abb. 3: Die zur Annotation möglichen Datentypen und ihre angepaßte graphische Darstellung im Frame über <i>widgets</i>	26
Abb. 4: Das Protégé CLIPS-Metamodell als UML-Klassendiagramm	27
Abb. 5: Die offene Architektur der Protégé-API	29
Abb. 6: Zugang des textualen Kontexts von potentiellen Konzepten entsprechenden Wörtern im Ausgangstextkorpus über eine Text-Konkordanz.....	42
Abb. 7: UML-Klassendiagramm der wichtigsten Gandr- <i>top-level</i> -Konzepte und ihrer Slots.. ...	49
Abb. 8: Das Cellular_Compound-Konzept und die Konzepthierarchie der Ontologie (Taxonomie der Annotationsbegriffe) im Classes Tab.....	51
Abb. 9: Darstellung des <i>st_successor</i> -Slot als Frame im Slot-Tab. Die rechte Seite zeigt die Slot- <i>facets</i> und ihre entsprechenden Füller / Werte.....	52
Abb. 10: Darstellung einer mit dem Konzept "Protein" annotierten Instanz 31967_at im Instances Tab. Links ist die Hierarchie der annotierenden Konzepte zu sehen.....	53
Abb. 11: Der Import der Primärdaten aus einer relationalen Access [®] -Datenbank.....	58
Abb. 12: Die Funktionalitäten bzw. Anwendungsfälle des GandrKB-Systems dargestellt als UML-Anwendungsdiagramm (<i>use case</i>).....	61
Abb. 13: Die kontextbasierte Darstellung von annotierten Genen über das <i>Instance Tree Tab</i> ...	63
Abb. 14: Die ontologische Anfrageschnittstelle des Gandr-Systems	66
Abb. 15: Mit OntoViz erstellter Graph des TLR-Receptor Pathways, wie er über die Gandr-Annotation in der Wissensbank repräsentiert ist.....	68
Abb. 16: Netzwerk-basierte (<i>force field spring layout</i>)-Visualisierung einer kontextual eingebetteten TLR-Instanz im TGViz-Tab	69
Abb. 17: Die GandrKB als Internet-Anwendung (Tomcat-Server <i>webapplication</i>)	77
Abb. 18: Import von UMLS-Konzepten mit dem UMLS Tab Plugin	104

Danksagungen

Mein herzlicher Dank geht an Herrn Professor Reich, der es mit viel Einfühlungsvermögen verstand mich in Augenblicken des Zweifels aufzubauen, der mir bei der Themenfindung vollkommen freie Hand ließ und mich nie in irgendeine Richtung zu drängen versucht hat. Außerdem danke ich Ihm dafür, daß er mir die Zeit gab das Projekt erfolgreich zu beenden. Professor Ulf Leser danke ich für die immer klaren und präzisen Hinweise und Anregungen zum informatischen Teil des Projekts und für das aufopferungsvolle Begutachten. Herrn Zenke und Frau Hacker danke ich für die Bereitstellung der Daten und der Primärannotationen und Robert Stevens für ein *warm welcome* in Manchester. Frederik Holst danke ich für den Ausgleich auf dem anderen Gebiet und für die Matratze in der Hafenstadt, Simon Brandt für ähnliches in Manchester. Jan Holst sei für das chinesische Zimmer gedankt. Weiter danke ich meinen Eltern für die finanzielle Unterstützung während des Studiums und dafür, daß ich immer noch einen "Koffer" auf dem Lande habe. Zum Schluß möchte ich noch der allwissenden Microsoft®-Büroklammer für ihre stets sinnvollen Ratschläge wie dem Folgenden danken:



... sowie der Microsoft Rechtschreibprüfung, die meine Orthographie so eloquent vom "knowledge engineering" zum "engen Hering" zu Korrigieren versuchte.

Curriculum Vitae

Name: Daniel Schober

Geburtsdatum: 20.10.1970

Geburtsort: Hamburg

Nationalität: Deutscher

Adresse, dienstlich: Max-Delbrück-Zentrum für molekulare Medizin (MDC)
Robert-Rössele-Straße 10
13092 Berlin-Buch
Tel.: 030/9406-3125
Fax: 030/9406-2834
E-Post: schober(bei)mdc-berlin.de

Adresse, privat: Erich Weinert Straße 6
10439 Berlin

Internetseite: www.bioinf.mdc-berlin.de/~schober

Schulbildung:
1977 - 1982 Grundschule Hoisdorf, Schleswig-Holstein
1982 - 21.5.1991 Abitur, Erich-Kästner-Gymnasium, Hamburg

Zivildienst:
1.11.1991 - 31.1.1993 Im Hause der Arbeiterwohlfahrt, Ahrensburg

Auslandsaufenthalte:
2.06.1991 - 18.10.1991 *Au pair*-Aufenthalt in New York, USA
4.03.2003 - 28.03.2003 Forschungsaufenthalt in der *Information Management Group* von Professor Robert Stevens am *Department of Computer Science* an der Universität Manchester, England

Universitäre Projektstudien: Zoologisches Institut Hamburg:
Klonierung eines neuen putativen Octopamin-Rezeptorgens aus einer Baculovirus -Spodoptera frugiperda 9 Zelllinie
Institut für Zellbiochemie und klinische Neurobiologie:
Sequenzhomologie-Screening einer Hühner-ZNS Lambda-cDNA-Bibliothek nach cDNAs für Synapsen-assoziierte Proteine

Hochschulstudium:

2.2.1993 - 17.11.1999

Biologiestudium an der Universität Hamburg
Hauptfach: Zoologie, Schwerpunkt Neurophysiologie
1. Nebenfach: Genetik, 2. Nebenfach: Biochemie
Alle Fächer mit Abschlußnote: „Sehr gut“

Diplomarbeit:

Untersuchung der Expression Synapsen-assoziiierter Proteine im Gehirn von Rattus norvegicus durch radioaktive in situ Hybridisierung

Universitäre Aktivitäten:

Kursbetreuung im Praktikum „Neurophysiologie für Biochemiker“ an der Biologischen Fakultät der Universität Hamburg (1995-1997)
Hilfswissenschaftler am Institut für Zellbiochemie und klinische Neurobiologie, Universitätsklinikum Eppendorf, Hamburg (1999-2000)

Andere Aktivitäten:

Erstellung von Expertisen für die "AG Gentechnologiebericht" der Berlin-Brandenburgischen Akademie der Wissenschaften

Promotion:

Als wissenschaftlicher Angestellter am Max-Delbrück Zentrum für molekulare Medizin, Berlin-Buch in der AG Bioinformatik bei Prof. Jens Reich, Titel: „Eine Wissensrepräsentation zur standardisierten Beschreibung und wissensbasierten Modellierung von Expressionsdaten“

Interessengebiete:

Ontologiebasiertes *Wissensmanagement*, Objektorientierte Datenmodelle, Vernetzte interaktive Visualisierungen, Künstliche Intelligenz (*Description Logic/OWL, Reasoning*), *Semantic Web*, *Natural Language Processing*, Datenbanken

Fremdsprachen:

Verhandlungssicheres Englisch in Wort und Schrift,
Grundkenntnisse in Französisch

Publikationsliste

Schober, D.; Leser, U.; Zenke, M. und Reich, J., GandrKB-ontological microarray annotation and visualization, *Bioinformatics* (Band 21), 2005, Nr. 11, Seite 2785-6. URL: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=15802288

Schober, D., Microarrays, Genexpressionsanalyse und Bioinformatik, *BIOspektrum* 3/2002, S.307-310, Spektrum Verlag Heidelberg

Schober, D., Arten und Funktionen von Datenbanken, publiziert in: F. Hucho, K. Köchi, Materialien für einen Gentechnologiebericht, Spektrum Verlag, ISBN 3-8274-1524-1 Heidelberg 2003

Schober, D., Ontologien in den Biowissenschaften, Expertise für die Berlin-Brandenburgische Akademie der Wissenschaften, veröffentlicht im Internet unter <http://www.bioinf.mdc-berlin.de/~schober/bio-ontologien.htm>.

Schober, D.; Reich, J., Ontology-derived Multiagent-based Signaltransduction-Simulation, (Poster), Workshop on Ontology for Biology, The Studio (Villa Bosch), Heidelberg, November 7-8, 2002

Schober, D.; Reich, J.; Leser, U. A Knowledgebase for domain dependent Microarray Annotation & Analysis, (Poster), 7th International Protégé Conference, Bethesda, Washington, USA, July 4-8, 2004

Erklärung an Eides Statt

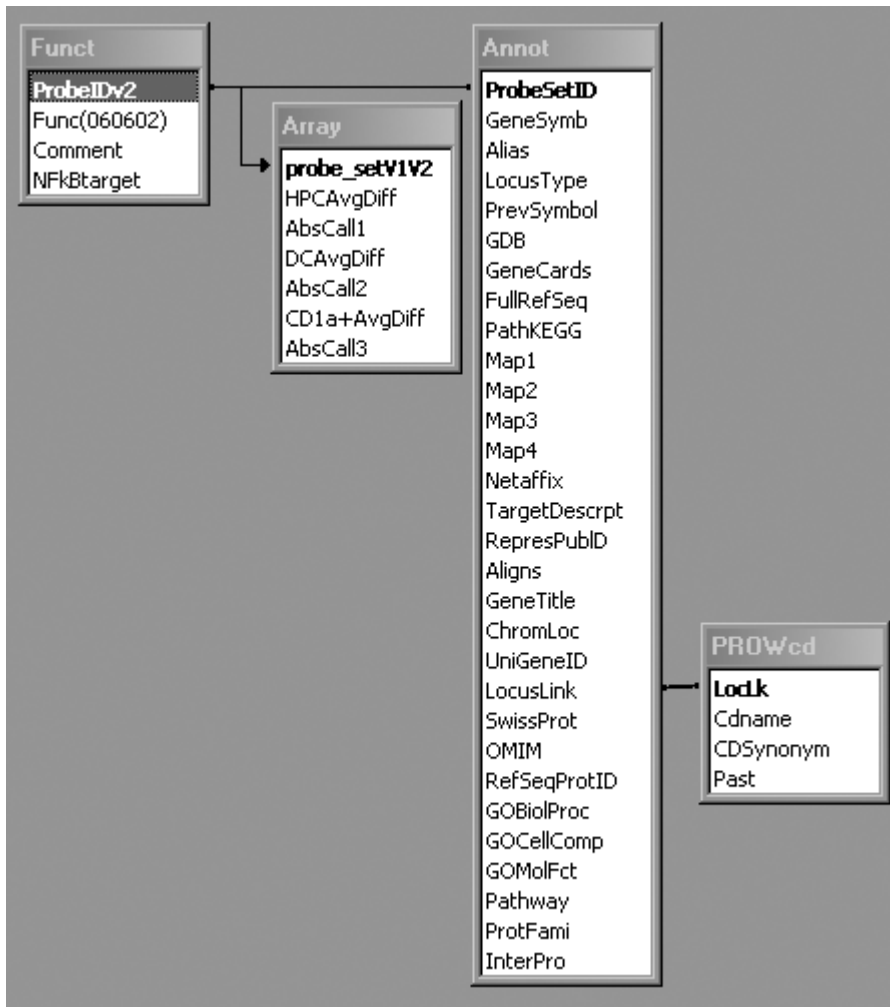
Hiermit erkläre ich, daß ich die vorliegende Arbeit mit dem Titel "*Ein Repräsentationsformat zur standardisierten Beschreibung und wissensbasierten Modellierung genomischer Expressionsdaten*" zur Erlangung des akademischen Grades *Doctor rerum medicarum* selbständig und ohne die unzulässige Hilfe Dritter verfaßt habe. Die Arbeit stellt, auch in Teilen, keine Kopie anderer Arbeiten dar; die für die Arbeit benutzten Literaturstellen, Quellen und Hilfsmittel sind vollständig angegeben.

Daniel Schober

3. September 2005

Anhang

A. ER-Diagramm Access-Datenimport



B. Kompetenzfragen

- Find Kinase annotated Probe Set IDs (finds also ATPases-->shows ontological generalisation)
- Find a certain Probe Set ID
- Find GO Process Apoptosis annotated Probe Set IDs
- Find Probe Set IDs for Surface Proteins annotated with GO Process Apoptosis (nested complex query)
- Find Experiment IDs with AbsCall1=P and CD1+AvgDiff >5000
- Find all Receptors in Hierarchy with expressionvalues like stated in the previous query (nested query)
- Find NFkB targets (simple query for slot only)
- Find ProbeSetIDs localized on Chromosome 21
- Find Transcription Factors located on Chromosome 21

- Find ProbeSetIDs with signaling description=Antigen and which are situated in the plasma membrane
- Find mature not neoplastic blood cell with identifying marker CD41
- Search Probe Set IDs annotated with OMIM number 604655
- Find annotation concepts which contain the string "like" in their name
- Find notes in KB
- Find insecure Concepts (contain "???" in class documentation)
- Find Biocarta maps on immunology
- Find Probe Set IDs found on Biocarta map TLR Pathway which have a Function-Slotvalue starting with "transcription" and which are expressed highly in Megakaryoblasts and are annotated with OMIM number 604655

Formale Kompetenzfragen in CLIPS-Syntax

```
((Gandr_ProjectKB_Instance_623] of String
```

```
(name "SearchTab_Query")

(string_value "Query Name: Find Kinase annotated Probe Set IDs (finds also ATPases-->shows
ontological generalisation)\nMatch All:true\nLength: 1\nKinase          Other: null
\n\nQuery Name: Find a certain Probe Set ID\nMatch All:true\nLength: 1\n Probe Set IDv2
contains          null \n\nQuery Name: Find GO Process Apoptosis annotated Probe Set
IDs\nMatch All:true\nLength: 1\n GOBiolProc          contains          Other:6915 // apoptosis 1
\n\nQuery Name: Find Probe Set IDs for Surface_Proteins annotated with GO Process
Apoptosis (nested complex query)\nMatch All:true\nLength: 2\nSurface_Protein Annot          Ptrs
contains          Query:Find GO Process Apoptosis annotated Probe Set IDs|Find GO Process
Apoptosis annotated Probe Set IDs 1          \n          NFkBtarget          is          Other:NFKB 1
\n\nQuery Name: Find Experiment IDs with AbsCall1=P and CD1+AvgDiff >5000\nMatch
All:true\nLength: 2\nArrayExperiment          AbsCall1          is          Other:P1
\nArrayExperiment          CD1a+AvgDiff          is greater than          Other:5000 1
\n\nQuery Name: Find all Receptors in Hierarchy with expressionValues like stated in the
previous query (nested query)\nMatch All:true\nLength: 1\nReceptor          Array Ptrs          contains
Query:Find Experiment IDs with AbsCall1=P and CD1+AvgDiff >5000|Find Experiment IDs with
AbsCall1=P and CD1+AvgDiff >5000 1          \n\nQuery Name: Find NFkB targets (simple query for
slot only)\nMatch All:true\nLength: 1\n NFkBtarget          is          Other:NFKB 1
\n\nQuery Name: Find Probe Set IDs localized on Chromosome 21\nMatch All:true\nLength:
1\nContextAnnotation          ChromLoc          begins with          Other:21 1          \n\nQuery Name:
Transcription          Factors          located          on          Chromosome          21\nMatch
All:true\nLength:
1\nTranscription_Factor          Annot Ptrs          contains          Query:Find Probe Set IDs localized on
Chromosome 21|Find Probe Set IDs localized on Chromosome 21 1          \n\nQuery Name: Find
Probe Set IDs with signaling description=Antigen and which are situated in the plasma
membrane\nMatch All:true\nLength: 2\nCellular_Compound          signaling_description          contains
Cls:Antigen 1          \nCellular_Compound          present_in          contains
Cls:Plasma_Membrane 1          \n\nQuery Name: Find mature not neoplastic blood cell with
identifying marker CD41\nMatch All:true\nLength: 3\nBlood_Cell          maturity          is
Other:mature 1          \nBlood_Cell          plasticity          is not          Other:neoplastic 1
\nBlood_Cell          identifying_markers          contains          Instance:Funct_Instance_10745|surface
protein, CD041=40643_at 1          \n\nQuery Name: Search Probe Set IDs annotated with OMIM
```

```

number 604655\nMatch All:true\nLength: 1\n      OMIM is      Other:604655 1
\n\nQuery Name: Find annotation concepts which contain the string \"like\" in their
name\nMatch All:true\nLength: 1\n      :NAME contains      Other:like 1
\n\nQuery Name: Find notes in KB\nMatch All:true\nLength: 1\n      :ANNOTATION-TEXT
is not Other:xyx 1 \n\nQuery Name: Find insecure Concepts (contain \"???\" in
class documentation)\nMatch All:true\nLength: 1\n      :DOCUMENTATION contains
Other:??? 1 \n\nQuery Name: Find Biocarta maps on immunology\nMatch
All:true\nLength: 1\nMap_Immunology      Other: null \n\nQuery Name: Find Probe Set
IDs found on Biocarta map TLR Pathway which have a Function-Slotvalue starting with
\"transcription\" and which are expressed highly in Megakaryoblasts and are annotated with
OMIM number 604655\nMatch All:true\nLength: 4\nCellular_Compound      biocarta_map contains
Instance:Immunology_Instance_70|Toll-Like Receptor Pathway 1 \nKinase      Function
begins with      Other:transcription 1 \n      expressed_highly_in contains
Cls:Megakaryoblast 1 \nCellular_Compound      Annot Ptrs      contains
Query:Search Probe Set IDs annotated with OMIM number 604655|Search ProbeIDs annotated
with OMIM number 604655 1 \n\n"))

```

C. Weitere Anwendungen

1. Konkordanz-Plugin zur Einbindung von Literaturdaten

Es wurde an der Einbindung einer Konkordanz-API als Protégé-Plugin gearbeitet, die in einem wählbaren Textkorpus zu Suchbegriffen oder den KR-Ideomen Konzept und Slot korrespondierende Begriffe inklusive ihres unmittelbaren Wortkontexts (Konkordanz) auffindet und darstellt. Eine derartige Integration textualen Wissens zu den KR-Ideomen würde ein schnelles Erforschen des semantischen Kontexts einer Annotation und ihre Verifizierung erlauben. Die Kopplung der Ontologie mit Volltext-Suchen in Textkorpora ermöglicht eine umfassendere Datensuche im textualen Literaturbestand, da sie Synonyme und Subkonzepte als Suchattribute beinhalten würde. Das ontologische Wissen könnte so IR-Systemen zugänglich gemacht werden. Als zu durchsuchender Textkorpus können XML-basierte Formate wie die *endnote[®]-reference-library*, *medline-abstracts* oder Text-Dateien (*full-text-paper*) und Internetseiten durchsucht werden.

2. *Ontology-induction* mit OntoTL

In Bezug auf die Erstellung und Erweiterung der Ontologie gibt es automatisierte Ansätze, ontologische Konzepte und Eigenschaften über NLP aus frei-textlichen Literaturdatenbanken wie Medline zu erzeugen [112]. Auch in der Molekularbiologie, einem mittlerweile sehr beliebten NLP-Anwendungsgebiet, werden derartige **ontology-induction** und **-learning** Ansätze zunehmend häufiger eingesetzt [113, 114, 115, 116, 117, 118]. Das Protégé-Plugin OntoTL [119] erlaubt die regelbasierte Extraktion von CLIPS-Ontologien aus natürlichsprachlichen

Texten. Zuvor müssen die Texte jedoch über NLP-Werkzeuge linguistisch annotiert werden. Diese Annotation kann automatisch erfolgen und umfaßt *POS-tags*, morphologische Annotierung (*inflection* und *decomposition*), *phrases (head-modifier analysis)*, Phrasen- und *predicate-argument*-Strukturen. Die Abbildung eines textualen Begriffes auf ein KR-Ideom wird dann über XPath -Regel-Bedingungen (*pre-conditions*) definiert. Derartige *ontology-induction*-Ansätze sind jedoch bisher relativ unpräzise und erfordern eine genaue nachträgliche Überprüfung durch Domänenexperten. Sie sind daher eher für die *de novo* Ontologie-Erstellung einsetzbar.

3. MDA und automatische Quellcodegenerierung

Von Seiten der Softwareentwicklung wurden in den letzten Jahren Forderungen nach abstrakten Entwicklungsplattformen zur Erstellung von Implementierungsplattform-unabhängigen Modellen laut, die dann in verschiedenste Programmiersprachen transformiert werden könnten. Die OMG stellt mit dem *model driven architecture*, MDA-Ansatz, eine entsprechende Spezifikation vor (<http://www.omg.org/mda/>). Das Gandr-System kann in diesem Sinne für eine plattformunabhängige Quellcodegenerierung genutzt werden. Ein auf die Ontologie aufbauendes objektorientiertes Software-System wird dabei vor der Implementierung im MDA-Ansatz als abstraktes *platform-independent model* (PIM) erstellt und über die Sprache *XML metadata interchange* (XMI) gespeichert. Das PIM dient dann als Gerüst und Konstruktionsplan für die Implementierung des Softwaresystems in einer konkreten Computersprache. Es kann über Abbildungen und entsprechende Compiler automatisch in *platform-dependent models* (PDM) für verschiedenste Systeme und Computersprachen übersetzt werden. Unter Verwendung der Gandr-Ontologie beim Aufbau eines MDA-PIM könnten individuenbasierte biologische Simulations-Modelle generiert werden. Aus entsprechend erweiterten Gandr-Konzepten könnten z.B. grundlegende strukturelle Komponenten für objektorientierte Simulationen automatisch generiert werden. Über das XMI-Zwischenformat kann aus den am Stoffwechsellnetz beteiligten Komponenten eine Java Klassen-Hierarchie mit entsprechenden Attributen generiert werden. Da hier nur Kernmodelle erstellbar sind, die keine PIM-Prozess-Daten enthalten, müssen entsprechende Java-Methoden nachträglich manuell integriert werden. Über eine entsprechende Erweiterung der CLIPS-Metaklassen-Architektur, die KR-Ideome zur formalen Prozeßbeschreibung erfaßt, könnten auch Java-Methoden per MDA generiert werden. Zur Erstellung eines Gandr-MDA-PIM und anschließende Transformation in JAVA wird das Protégé Modell in XMI gespeichert und über UML bearbeitet. Dabei wird ein Gandr-Konzept unter

Erhalt der Mehrfachvererbung zu einer UML-Klasse desselben Namens und Rolle (*abstract / concrete*), wobei Instanzen und Metaklassen z.Zt. nicht exportiert werden. Slots werden zu UML-Attributen entsprechenden Datentyps. Relationale Slots werden zu UML-Assoziationen, inverse Slots zu bidirektionalen Assoziationen. Slot-*facets* wie Multiplizität und Kardinalität werden übernommen. Slots vom Datentyp *instance* werden gegenwärtig nicht transformiert. Hier ein Beispiel für die automatische Java-Quellcode-Generierung über UML im MDA-Ansatz:

Als Beispiel wird hier nur ein Gandr-Konzept betrachtet und gezeigt, wie es in XMI zwischengespeichert, in dem UML-Werkzeug Poseidon CE[®] weiterbearbeitet und hierüber in entsprechenden Java Quellcode übersetzt wird (siehe Abb. 19).

Das Protégé "Cellular_Compound"-Konzept in XMI:

```
<XMI.content>

  <Protégé.StandardClass xmi.id = 'Cellular_Compound' name = 'Cellular_Compound'

    documentation = 'The most important module which contains the concepts used for gene
annotation. Cellular_Compound is an abstract concept, that means it can't be instantiated
itself to enforce further characterization of the ProbeSetIDs Function described through the
Annotation concept.'

    abstract = 'true'>

  <Protégé.StandardInstance.directType>

    <Protégé.ExternalClass xmi.idref = ':STANDARD-CLASS' />

  </Protégé.StandardInstance.directType>

  <Protégé.StandardClass.directSuperClasses>

    <Protégé.StandardClass xmi.idref = 'ProbeSetContainer' />

  </Protégé.StandardClass.directSuperClasses>

  <Protégé.StandardClass.templateSlots>

    <Protégé.StandardSlot xmi.idref = 'signaling_description' />...

  </Protégé.StandardClass.templateSlots>

</Protégé.StandardClass>

</XMI.content>
```

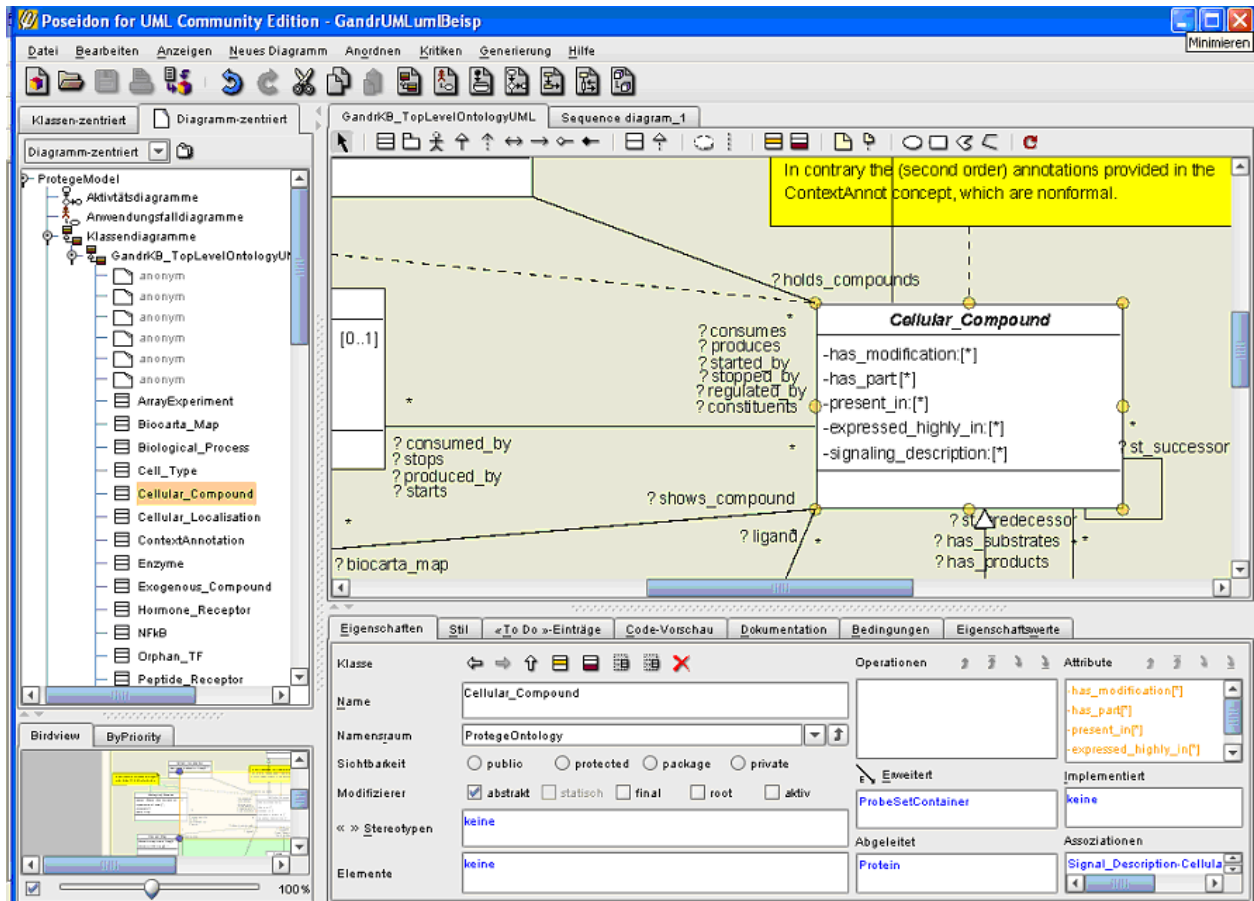


Abb. 19: Bearbeitung der Gandr-Ontologie im UML-Werkzeug Poseidon CE®. Aus den aufgelisteten Konzepten (links) wurde per *drag and drop* ein UML-Diagramm mit entsprechenden Klassen, Attributen und Relationen erstellt (siehe Abb. 7). Das "Cellular_Compound"-Konzept ist markiert und unten als Java-Klassen-"Frame" repräsentiert.

Automatisch generierter JMI-Java Quellcode des "Cellular_Compound"-Konzepts:

```

/** Java class "Cellular_Compound.java" generated from Poseidon for UML.
 * Poseidon for UML is developed by <A HREF="http://www.gentleware.com">Gentleware</A>.
 * Generated with <A HREF="http://jakarta.apache.org/velocity/">velocity</A> template engine. */
package ProtégéOntology;

import java.util.*;

abstract class Cellular_Compound extends ProbeSetContainer {

    // attributes

    private Collection expressed_highly_in; // of type

/** * Represents ... */

    private Collection signaling_description; // of type

} // end Cellular_Compound

```

4. Weitere *constraints* in der *protégé axiom language*

Über die am *knowledge interchange format* (KIF) angelehnte *protégé axiom language* (PAL) können axiomatische *constraints* für Gandr-KR-Ideome formuliert und über deren Überprüfung die Konsistenz der Wissensbank weiter erhöht werden (*PAL facet constraints tab-plugin*, http://protege.stanford.edu/plugins/facet_constraints_tab/).

5. Individuen-basierte Simulationsansätze

Brutlag et al. Zeigen, wie über regelbasierte Veränderungen einer Wissensbank im *forward chaining*-Ansatz DNA-Metabolismus-Simulationen erstellt werden können [120]. Einen Petri-Modell Simulationsansatz, der auf eine Protégé-Wissensbank zugreift, stellen Peleg et al. vor [121]. Über das Jess Tab- und das JFC-Plugin könnten auch in der Gandr-Wissensbank definierte Entitäten in Simulationsansätze eingebunden werden. Der Vorteil regelbasierter Modelle und individuenbasierter Simulationen ist, daß sie leichter von Biologen verstanden werden. Das liegt u.a. daran, daß sie in einer für Biologen intuitiven Sprache formuliert werden können. Kinetische bzw. differentialgleichungsbasierte Modelle dagegen erfordern tiefe Kenntnis einer abstrakten mathematischen Terminologie, deren Begriffe nur schwer auf in der Biologie übliche gedankliche Modelle abgebildet werden können. Zur regelbasierten Simulation benötigte Daten sind ferner leichter zu finden als kinetische, da das meiste und detaillierteste Wissen über biologische Systeme in Textdatenbanken wie Medline vorliegt. Die diskrete und qualitative Simulation größerer Stoffwechsel-Systeme über individuenbasierte, d.h. parallel und dezentralisiert arbeitende Modelle, ist für Nicht-Mathematiker aufgrund ihrer Strukturtreue für Biologen besonders verständlich [122]. Ontologien und Ontologiesprachen können dabei als Faktenrepräsentation und Kommunikationsgrundlage zwischen den Agenten eingesetzt werden [123]. Komplexe offene Systeme, wie Stoffwechsel- und Signaltransduktionssysteme, bei denen sich die Struktur dynamisch ändert und deren Komponenten kontextabhängig reagieren, können mit Multiagentenbasierten Systemen (MAS) simuliert werden [122]. Agenten sind nach dem PDP-Prinzip verteilte autonome Systeme bzw. Software-Entitäten, die mit einer Software-Umwelt und miteinander interagieren. Ihr Verhalten ist abhängig von der Umwelt-Faktenlage, die z.B. Kontext- bzw. Orts-Abhängigkeiten umfaßt. Die Ontologie-basierte Wissensbank ist eine interne symbolische Repräsentation (Faktenbank) der Welt, in der die Agenten handeln und die sie gegebenenfalls über Regeln und Schlußfolgerungen verändern. Die Verhaltenssteuerung der Agenten erfolgt über eine Regelbasis in der bestimmten Umweltsituationen entsprechende Agenten-Aktionen zugeordnet sind. Die Bedingung wird über einen Konzept-Ausdruck der

Ontologie in OKBC-CLIPS formalisiert. Die bei Feststellen dieses Fakts auszuführende Aktion entspricht dann einer Manipulationen der Umwelt-Fakten in der GandrKB Wissensbank. Das JessTab könnte zur Erstellung von Multiagenten-Systemen genutzt werden, wobei jedes Jess-Expertensystem mit einer unabhängigen *rule engine* einem Agenten entspricht. Dabei würden dann alle Agenten auf die Wissensbank als gemeinsames *blackboard* zugreifen.

Der Agenten-Ansatz würde den in Signal-Netzen immanenten parallelverarbeitenden Verrechnungsstrategien und der Kontextabhängigkeit des "Verhaltens" beteiligter Genprodukte Rechnung tragen [122].