

Wissensbasierte Verwaltung von Forschungsergebnissen in den
Prozesswissenschaften auf der Grundlage von Ontologien mit dem
Schwerpunkt auf Modellentwicklung

vorgelegt von
Diplom-Ingenieur
Moritz Weiten

Von der Fakultät III - Prozesswissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
- Dr.-Ing. -

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr.-Ing. Matthias Kraume

Berichter: Prof. Dr.-Ing. Günter Wozny

Berichter: Prof. Dr.-Ing. Wolfgang Marquardt

Tag der wissenschaftlichen Aussprache: 25.11.2008

Berlin 2009

D 83

Vorwort und Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Fachgebiet Dynamik und Betrieb technischer Anlagen des Instituts für Prozess- und Verfahrenstechnik der Technischen Universität Berlin.

Mein ganz besonderer Dank gilt Herrn Prof. Dr.-Ing. Wozny, der Doktorvater dieser Arbeit. Seine fachliche Unterstützung, sein großes Interesse an der Arbeit, seine konstruktiv kritischen Rückfragen und schließlich seine enorme Geduld haben wesentlich zum Gelingen beigetragen. Mein Dank gilt auch Herrn Prof. Dr.-Ing. Marquardt, dem Zweitgutachter für diese Arbeit.

Ebenfalls sehr viel Geduld bewiesen hat meine Familie, in erster Linie jedoch meine Frau Susanne, die auch Korrektur gelesen hat. Für Ihre enorme Unterstützung möchte ich mich an dieser Stelle ausdrücklich bedanken. Korrektur gelesen hat auch meine Mutter Gisela Weiten, bei der ich mich ebenfalls bedanken möchte.

Weiterhin bedanken möchte ich mich bei meinen Diplomarbeitern Riad Hechame und Holger Steinmeyer. Letzterer hat echte Pionierarbeit geleistet und den Grundstein für ein zentrales Kapitel der Arbeit gelegt.

Fruchtbar waren auch die vielen Gespräche und Diskussionen mit den damaligen Kollegen, allen voran Bernd Goers und Thomas Schipolowski. Die anderen netten Kollegen und Kolleginnen und die sehr angenehme und konstruktive Atmosphäre am Institut haben mir eine sehr schöne Zeit am Fachgebiet beschert. Als hilfsbereit hat sich auch stets unser Systemadministrator Daniel Weißmann gezeigt, bei dem ich mich hier gleichfalls bedanken will.

Der Deutschen Forschungsgemeinschaft gilt mein Dank für die Teilfinanzierung der Arbeit in der ersten Hälfte ihrer Entstehung.

Abstract

Die Verfügbarkeit von Ergebnissen aus Forschung und Entwicklung stellt große Anforderungen an die betreffenden Institutionen, sowohl in organisatorischer als auch in technischer Hinsicht. Die Vielzahl wissenschaftlicher Untersuchungen, der teilweise hohe Grad an Spezialisierung und die Tatsache, dass gerade grundlagenorientierter Arbeiten über einen längeren Zeitraum relevant sein können, führen zu der Notwendigkeit einer effektiven Verwaltung anfallender Ergebnisse.

Im Rahmen dieser Arbeit wurde ein Ansatz entwickelt, der die nachhaltige Verwaltung von Forschungsergebnissen aus den Prozesswissenschaften (Verfahrenstechnik, Chemieingenieurwesen und verwandte Gebiete) mit Hilfe von Wissensmodellen, sog. Ontologien, ermöglicht.

Den zentralen Ansatz für die formale Beschreibung von Aussagen über wissenschaftliche Ergebnisse bildet die Betrachtung mathematischer Modelle. Letztere sind Ziel, Mittel und/oder Ausgangspunkt vieler wissenschaftlicher Untersuchungen gerade im Bereich der Prozesswissenschaften.

Neben der fachlich-inhaltlichen Beschreibung von relevanten Konzepten wie mathematischen Modellen, Messdaten, Dokumenten, etc. steht das Erfassen von Bezügen zwischen unterschiedlichen Ansätzen im Vordergrund. Zu diesem Zweck wurde der sog. SchoolOnto-Ansatz (vgl. Buckingham et al., 2000) zur formalen Repräsentation von Aussagen im Sinne eines wissenschaftlichen Diskurses übernommen und auf die gegebene Problemstellung übertragen.

Als softwaretechnische Umsetzung dieses Ansatzes wurde das System SCIFORG entworfen. Es wurde auf der Grundlage des Ontologie-Editors Protégé implementiert. SCIFORG bietet Anwendern verschiedene Schnittstellen für das Editieren von Daten und die Suche in der Wissensbasis sowie eine Reihe von fachspezifischen Funktionen wie etwa die Einheiten-Verwaltung. Das System unterstützt die Verwaltung großer Datenmengen.

Das SCIFORG System in Verbindung mit dem zugrunde liegenden Ansatz erlaubt es Anwendern, wissenschaftliche Ergebnisse durch semantische Informationen anzureichern. Damit lassen sich Zusammenhänge nachvollziehen und Forschungsergebnisse gezielt recherchieren.

The availability of resources resulting from scientific work and development activities poses a great challenge to the respective institutions. This concerns technical as well as organizational aspects. The high number of research activities, the high degree of specialism as well as the fact that results can be relevant for a long period of time demand for the efficient management of scientific resources.

This thesis presents an approach for the sustainable management of results in the field of process sciences (chemical engineering and related areas). The approach is based on knowledge models, so called ontologies, whereas the focus in respect of content is on mathematical models. The latter form the goal, the starting point and/or the means of many scientific analyses – especially in the area of process sciences.

Apart from the domain-specific description of relevant concepts such as mathematical models, experimental data sets, documents, etc. this work focuses on formal statements about how scientific approaches are related. To achieve this, the so called SchoolOnto-approach (vgl. Buckingham et al., 2000) was adopted and modified.

The approach was implemented as the SCIFORG software system, which is based on the ontology-editor Protégé. SCIFORG provides a couple of interface to edit data and search the knowledge base. The system offers several domain-specific interfaces such as for the management of physical units. The system is suited for large amounts of data.

SCIFORG enables users to enrich scientific results by semantic information. Thus relations between different scientific results can be retraced. The results themselves can be retrieved with high precision.

Inhalt

1	Einleitung.....	1
2	Grundlagen.....	4
2.1	Problembeschreibung	4
2.1.1	Problem der proprietären Formate	6
2.1.2	Problem der Heterogenität wissenschaftlicher Ressourcen.....	7
2.1.3	Problem der Metainformationen	8
2.2	Semantische Technologien.....	10
2.2.1	Der Ontologiebegriff.....	10
2.2.1.1	Die Idee des „Semantic Web“	13
2.2.1.2	Sprachen.....	14
2.2.1.3	Werkzeuge	18
2.2.2	Ontologien in den Natur- und Ingenieurwissenschaften	19
2.2.2.1	Konzeptuelle Modelle nach Söderman	20
2.2.2.2	Batres et al.: Ontologien für den Austausch von Modellen	21
2.2.2.3	OntoCAPE	22
2.2.3	Probleme und Grenzen der semantischen Technologien.....	24
2.2.3.1	Massendaten	24
2.2.3.2	Sicherheit/Qualitätssicherung in der Datenhaltung	25
2.2.3.3	Standardkonformität	26
3	Mathematische Modellentwicklung	27
3.1.1	Die Begriffe „System“ und „Modell“	30
3.1.2	Abstraktionsebenen nach Top und Akkermans.....	31
3.2	Eigenschaften von Modellen und Systemen.....	33
3.2.1	Semantik von Modellen.....	33
3.2.2	Klassifizierung von Modellen.....	34
3.2.2.1	Betrachtung des Zeitverhaltens	34
3.2.2.2	Betrachtung des Raumverhaltens	35
3.2.2.3	Mathematische Modellattribute	36
3.3	Modellbildung	37
3.3.1	Deduktive Modellbildung	38
3.3.2	Induktive Modellbildung.....	39
3.3.3	Allgemeine Vorgehensweise.....	39
3.3.4	Konzeptionelle Modellbildung.....	42

3.3.5	Ansätze zur Modellbildung in den Prozesswissenschaften	44
3.3.6	Mathematische Modellbildung/Simulation.....	46
3.3.7	Verifikation und Validierung.....	49
3.3.8	Ansätze zur Unterstützung des Modellierungsablaufs	51
3.3.9	Modellbibliotheken.....	51
3.4	Modellentwicklung und Datenhaltung.....	52
3.4.1	Modellentwicklung und Messdatenverwaltung	52
3.4.2	Modellentwicklung und Dokumentenverwaltung.....	57
3.5	Vergleich der mathematischen und der semantischen Modellentwicklung	58
3.6	Industrielle Feldstudie.....	60
3.6.1	Problemstellung	60
3.6.2	Konzeptionelle Modellbildung	60
3.6.3	Validierung.....	61
3.6.4	Dokumentation und Modellwiederverwendung	61
3.6.5	Fazit	62
4	Konzept	63
4.1	Ziele und Anforderungen.....	63
4.2	Plattformunabhängige Archivierung.....	67
4.2.1	XML als Metadatenformat für wissenschaftliche Daten und Dokumente	67
4.2.1.1	Mathematical Markup Language und Chemical Markup Language.....	68
4.2.1.2	XML als Datenaustausch- und Archivierungsformat	69
4.2.2	Autoren-Unterstützung: Vorlagen für Textdokumente	69
4.2.2.1	Literaturverweise und Literaturverzeichnis	70
4.2.2.2	Anlegen von Tabellen, Abbildungen und Gleichungen	70
4.3	Der SchoolOnto Ansatz	71
4.3.1	Vorteile und Nutzen des SchoolOnto-Ansatzes.....	75
4.3.2	Übertragung des SchoolOnto-Ansatzes auf den Bereich mathematischer Modelle.....	77
4.3.2.1	Bezüge zwischen Modellen.....	78
4.3.2.2	Bezüge zwischen Modellen und Theorien.....	81
4.3.2.3	Bezüge zwischen Modellen und Messungen bzw. Messdaten.....	81
4.4	Ontologiebasierte Aufbereitung und Verwaltung von Forschungsergebnissen	82
4.4.1	Ontologien für Modellierungsansätze.....	83
4.4.1.1	Mathematische Modelle.....	84

4.4.1.2	Ressourcen.....	86
4.4.1.3	SchoolOnto	87
4.4.1.4	Weitere Domänen-unabhängige Konzepte	88
4.5	Werkzeug-Unterstützung von Fachanwendern	88
4.5.1	Daten- und Metadaten-Ebene.....	88
4.5.2	Mehrwert der Ontologie-basierten Datenverwaltung als Anreiz	91
4.5.3	Anwenderschnittstellen/Oberflächen.....	92
4.5.3.1	Formular-basierter Ansatz.....	92
4.5.3.2	Graphen-basierter Ansatz.....	92
5	Implementierung	93
5.1	Architektur	94
5.2	Verwendete Applikationen/Bibliotheken	95
5.2.1	Protégé 2000	95
5.2.1.1	Protégé als hybrides Werkzeug.....	95
5.2.1.2	Protégé Datenmodell	97
5.2.1.3	Konfigurierbare Formular-basierte Editoren	104
5.2.1.4	Plugin Architektur	105
5.2.1.5	Programmierschnittstelle (API)	105
5.2.1.6	Schwachstellen des Systems	106
5.2.2	SimpleORM	109
5.2.3	Algernon.....	110
5.2.4	JGraph	110
5.2.5	Weitere Bibliotheken/Programme	111
5.3	Aufbau von SCIFORG mit Protégé 2000 als Kernplattform	112
5.3.1	Skalierbare objektrelationale Speicherschicht	113
5.3.1.1	Prinzip der Objekt-relationalen Abbildung	115
5.3.1.2	Schema der Objekt-relationalen Abbildung	117
5.3.1.3	Limitierungen der gewählten Abbildung.....	122
5.3.2	Anwender-Schnittstellen für die Prozesswissenschaften	124
5.3.2.1	Mathematische Formeln	124
5.3.2.2	Chemische Formeln	126
5.3.3	Einheiten-Verwaltung	127
5.3.4	Werkzeuge für Regeln, Suchanfragen und zur Navigation.....	129
5.3.4.1	Graphische Schnittstelle	129
5.3.4.2	Formulare für Suchanfragen	137

5.3.5	Rohdatenverwaltung und Datenvisualisierung.....	138
6	Anwendungsszenarien	145
6.1	Erfassen und Abrufen von Modell-Informationen.....	147
6.2	Erfassen und Abrufen von Simulations- und Messdaten (sowie Metadaten)	148
6.3	Konsistenzprüfungen für Metadaten.....	148
7	Fazit und Ausblick	151
	Anhang.....	155
	Glossar	173
	Abbildungsverzeichnis	175
	Tabellenverzeichnis.....	178
	Literaturverzeichnis.....	179

1 Einleitung

Die Verfügbarkeit von Ergebnissen aus Forschung und Entwicklung stellt große Anforderungen an die betreffenden Institutionen, sowohl in organisatorischer als auch in technischer Hinsicht. Die Vielzahl wissenschaftlicher Untersuchungen, der z.T. hohe Grad an Spezialisierung und die Tatsache, dass gerade grundlagenorientierte Arbeiten über einen längeren Zeitraum relevant sein können, führen zu der Notwendigkeit einer effektiven Verwaltung anfallender Ergebnisse. Die Verfügbarkeit und Recherchierbarkeit wissenschaftlicher Ergebnisse ist entscheidend für die nachfolgende Wertschöpfung. Dies gilt umso mehr, wenn Ergebnisse nicht unmittelbar aufgegriffen werden und ein Verwertungsprozess initiiert wird – etwa durch die schrittweise Übertragung in einen Produktionsprozess. Wenn Forschungsergebnisse in erster Linie als Baustein für einen längeren Entwicklungsprozess fungieren, dann steht die Verwendung dieser Ergebnisse im Rahmen von aufbauenden Arbeiten im Vordergrund.

Insbesondere für die Forschung und Entwicklung, sowohl im akademischen als auch im kommerziellen Umfeld, ist die Recherchierbarkeit wichtig, um

- § redundante bzw. obsolete Ansätze zu vermeiden,
- § den Bedarf für Forschungsaktivitäten zu ermitteln
- § und Grundlagen für eigene Arbeiten aufzubauen.

Gerade für Forschungsarbeiten im akademischen Bereich ist es entscheidend, die Einmaligkeit des eigenen Ansatzes sicher zu stellen, da er eine entscheidende Grundlage für den Forschungscharakter darstellt. Im kommerziellen Umfeld müssen redundante Arbeiten vor allem aus Kostengründen vermieden werden. Es kann aber auch für die Positionierung gegenüber Wettbewerbern entscheidend sein, den Stand der Forschung möglichst exakt und vollständig bestimmen zu können. Dazu kommen rechtliche Aspekte, die jedoch vor allem für die Patentrecherche eine Rolle spielen, die zu behandeln wiederum den Rahmen dieser Arbeit sprengen würde.

Am Fachgebiet für Dynamik und Betrieb technischer Anlagen wurde im Rahmen eines DFG-geförderten Projektes das Rahmenwerk „SCIFORG“¹ zur Verwaltung wissenschaftlicher Informationsressourcen entwickelt. Das Ziel hierbei war es, die langfristige und effiziente Verfügbarkeit von Forschungsergebnissen im Bereich der Erforschung physikalisch-chemischer Prozesse zu sichern. Der Schwerpunkt wurde auf die technischen und konzeptionellen Aspekte bei der Verwaltung heterogener Informationsquellen aus interdisziplinären Projekten gelegt. Das Konzept ist in Weiten et al. (2002) sowie in Weiten und Wozny (2004) beschrieben.

Das SCIFORG-System erlaubt es, Quellen in Form von Rohdaten, Modelldaten und Dokumenten zusammen mit Zusatzinformationen zu logischen Zusammenhängen zu verwalten. Als konzeptionelles Gerüst dienen offene, zentral gepflegte Datenmodelle („Ontologien“). Auf ihrer

¹ Scientific Information Organizer

Grundlage erfolgt die hierarchische Strukturierung einiger Grundkonzepte von Forschungsprojekten (wie z.B. „Modell“, „Datensatz“, „Veröffentlichung“, etc.). Dabei wird auf den SchoolOnto-Ansatz zurückgegriffen, der eine sehr generische Repräsentation für die Bezüge zwischen wissenschaftlichen Ansätzen bzw. Ergebnissen ermöglicht. Der SchoolOnto-Ansatz stellt ein sehr „leichtgewichtiges“ Datenmodell dar, das für Anwender verhältnismäßig einfach zu adaptieren ist. Dabei können die definierten Bezüge verfeinert werden, um Zusammenhänge domänenspezifischer ausdrücken zu können.

Unabhängig vom Ursprungsformat der beschriebenen Ressourcen können übergeordnete Informationen (Metainformationen) erfasst und beim Retrieval ausgenutzt werden. Dadurch wird es Anwendern auf der Suche nach Informationen möglich, sich über eine einheitliche Schnittstelle einen Überblick über vorhandene wissenschaftliche Informationsressourcen zu verschaffen, ohne auf verschiedene Werkzeuge zurückgreifen zu müssen. Das Ursprungsformat der Daten ist nur insofern von Bedeutung, als SCIFORG über eine begrenzte Anzahl von Import-Schnittstellen verfügt. Damit wurden zunächst die gängigsten Datenformate abgedeckt. Die Architektur von SCIFORG bietet jedoch grundsätzlich die Möglichkeit, zukünftig weitere Datenformate zu integrieren. Der Kern des Systems ist dabei immer Werkzeug-unabhängig.

Das System soll dabei nicht vorhandene Werkzeuge ersetzen, auch wenn es hinsichtlich der Funktionalität kleinere Überlappungen gibt. Diese ergeben sich aus der Notwendigkeit, eine sinnvolle Visualisierung von Informationen bereit zu stellen. Die einfachen Möglichkeiten zur Datenanalyse beispielsweise stehen nicht in Konkurrenz spezialisierter Werkzeuge im Bereich der Tabellenkalkulation oder von Statistik-Software. Sie erlauben es dem Anwender jedoch, eine Selektion aus einer großen Menge von Daten vorzunehmen ohne dabei auf die übliche Form der Darstellung von Daten zu verzichten. D.h. mit Hilfe von SCIFORG ist es möglich, einen Satz von Messdaten zu suchen, die bestimmte übergeordnete Kriterien erfüllen (die z.B. zur Validierung eines bestimmten Prozessmodells herangezogen wurden). Einfache Filter und die Diagrammdarstellung helfen dabei, diese Daten einzuschätzen. Für eine tiefer gehende Analyse muss auf den Export-Filter und die anschließende Verwendung eines geeigneten Werkzeuges zurückgegriffen werden.

SCIFORG wurde in erster Linie als Plattform für Ontologie-basierte Metainformationen entwickelt, lässt sich jedoch auch zur Datenerfassung im Sinne herkömmlicher Datenbanksysteme (z.B. zur Literaturverwaltung) nutzen. Dies gilt insbesondere für den Bereich der Messdatenverwaltung. Bei vielen Forschungsprojekten im universitären Bereich steht kein Datenbanksystem zur Messdatenverwaltung zur Verfügung. SCIFORG kann für diesen Zweck genutzt werden. Das System arbeitet zeilenorientiert, d.h. ein Satz zusammenhängender Messdaten entspricht einer Zeile in einer Datenbanktabelle. Messdaten, die in einer bestimmten Versuchsanordnung zuzuordnen sind (typischer Weise einer Labor- oder Technikumsanlage), werden in einer eigenen Tabelle verwaltet. Auf diese Weise kann auch ein großes Datenaufkommen effizient verwaltet werden. Umfangreiche Datenmengen (z.B. aus Prozessleitsystemen, die Daten in relativ kleinen Zeitintervallen aufnehmen) stellen somit kein Problem dar. Der Abruf und der Import von Daten

wurden für hohe Datenvolumen optimiert, um lange Wartezeiten und besondere Hardwareanforderungen zu vermeiden.

Verknüpfungen zwischen Konzepten, z.B. in der Form „Veröffentlichung ... basiert auf Rohdaten ...“, können in SCIFORG graphisch in einer Art Fließschema dargestellt werden. Suchanfragen („welche Veröffentlichungen basieren auf den Rohdaten ...“) sind ebenfalls in graphischer Form möglich. Darüber hinaus können Axiome bzw. Regeln zu assoziativen Verknüpfungen graphisch erfasst werden (z.B. die Definition transitiver Beziehungen: „wenn Modell A Modell B erweitert und Modell B Modell C erweitert, dann erweitert Modell A auch Modell C“). Die Grundlage für die Beschreibung dieser Art von Verknüpfungen bietet der so genannte School-Onto-Ansatz. SchoolOnto ist eine von Buckingham et al. (2000) entwickelte Ontologie, die es erlaubt, den wissenschaftlichen Diskurs durch eine formale Beschreibung von Ansätzen und grundsätzlicher Bezüge, wie etwa „verwendet“, „stützt“ oder „steht im Widerspruch zu“.

Der Kern des SCIFORG-Systems besteht aus der an der Stanford-Universität entwickelten Applikation Protégé 2000. Eine Reihe weiterer Applikationen bzw. Bibliotheken wurden eingebunden und z.T. erweitert um Informationsressourcen in verschiedenen Formaten langfristig lesbar zu verwalten und mit Zusatzinformationen anzureichern. Dies beinhaltet u.a. die Speicherung von Word-Dokumenten in einem standardisierten Format, das Erfassen chemischer Strukturformeln und mathematischer Gleichungen sowie das Einbinden relationaler Datenbanken in die verwalteten Datenmodelle. Anwender haben eine zentrale Sicht auf die verschiedenen Informationsquellen und die vorhandene Problemstellung.

2 Grundlagen

2.1 Problembeschreibung

Wie in der Einleitung beschrieben, stellt die Quantität der Daten aus wissenschaftlichen Ergebnissen und der hohe Spezialisierungsgrad von Forschungsergebnissen das Informationsmanagement vor große Herausforderungen. Die langfristige Verfügbarkeit und die Nachvollziehbarkeit der Ergebnisse stellen hierbei zentrale Aspekte dar.

Bestimmung des Standes der Forschung

Innerhalb der Prozesswissenschaften sind Forschungsaktivitäten auch im Grundlagenbereich relativ bedarfsorientiert und nur eingeschränkt am generischen Erkenntnisgewinn ausgerichtet. Das bedingt eine verstärkte Spezialisierung von Forschungsansätzen und dementsprechend ausdifferenzierte Bedarfsanalyse. Im kommerziellen Bereich ergibt sich der Bedarf für Forschung und Entwicklung durch Marktanalysen, Produktionsengpässe, geänderte Richtlinien, veränderte Marktbedingungen etc. Im akademischen Umfeld ist ein zentraler Faktor der Forschungsbedarf für die mittel- und langfristige schrittweise Entwicklung neuer Methodiken und Verfahren. Die unmittelbare Verwertbarkeit von Forschungsergebnissen im industriellen Maßstab ist für den Erfolg einzelner Projekte nicht unbedingt entscheidend. Für einen gesamten Forschungszweig jedoch ist entscheidend, dass Grundlagen-Ergebnisse schrittweise hin zur praktischen Verwertung bzw. Umsetzung ergänzt und erweitert werden. Wie bereits erwähnt, erfordert dies eine genaue Bestimmung der „fehlenden Bausteine“ innerhalb einer Entwicklungslinie.

Die zu füllenden Lücken von Grundlagen bis hin zur Anwendung im industriellen Maßstab ausfindig zu machen erfordert eine präzise Bestimmung des Standes der Forschung. Um diesen zu ermitteln, sind Recherche-Möglichkeiten wichtig, die sowohl möglichst effizient sind als auch möglichst vollständig.

Datenflut

WissenschaftlerInnen sind im Rahmen ihrer Forschungsaktivitäten oftmals mit einer enormen Menge von Daten konfrontiert. Ressourcen wissenschaftlicher Arbeiten wie Veröffentlichungen, Berichte und Simulationsmodelle sind das Ergebnis eines längeren, komplexen und kreativen Prozesses. Um dieses Ergebnis korrekt zu interpretieren und als Grundlage für aufbauende Arbeiten verwenden zu können, sind oftmals Kenntnisse über den Prozess selbst erforderlich.

Für wissenschaftliche Arbeiten mit einer gegebenen Zielstellung sind in Bezug auf bereits vorhandene Forschungsergebnisse im Wesentlichen zwei Aspekte wichtig:

- § Die effektive Möglichkeit, das Vorhandensein von Ergebnissen zu recherchieren („gibt es Ergebnisse im Bereich von ...“):

Die mögliche Redundanz von wissenschaftlichen Untersuchungen stellt ein großes Risiko dar. Für wissenschaftliche Untersuchungen, insbesondere im akademischen Umfeld, ist die Einmaligkeit des Ansatzes Teil der Rechtfertigung für die Notwendigkeit der Durchführung. Im Bereich der industriellen Forschung oder allgemein der Auftragsforschung sind redundante Untersuchungen aus Kosten- und Zeitgründen nicht erwünscht. Daraus ergibt sich die Anforderung, dass sich das Vorhandensein von Forschungsergebnissen in einem gegebenen Forschungsfeld möglichst effizient ermitteln lässt (sofern diese nicht vollständig der Geheimhaltung unterliegen). Den Anspruch auf ein Forschungsergebnis zu erheben ist nur eine Funktion einer Veröffentlichung. Für den Urheber ist auch die effektive Recherchierbarkeit in der Regel von großem Interesse, um Ergebnisse einem möglichst breiten Fachpublikum zugänglich zu machen.

- § Die Nachvollziehbarkeit von Ergebnissen („wie ist das Ergebnis erzielt worden...?“):

Allein die Verfügbarkeit der Information über das Vorhandensein von Ergebnissen ist im Sinne eines effizienten wissenschaftlichen Vorgehens nicht ausreichend. Im Extremfall zieht die mangelnde Nachvollziehbarkeit zwangsweise redundante Untersuchungen nach sich.

Für die Nachvollziehbarkeit und die weitere Verwertung von Forschungsergebnissen ist ein Mindestmaß an Verfügbarkeit unabdingbar. „Mindestmaß“ bedeutet in diesem Zusammenhang, dass nicht nur Endergebnisse in Form von Veröffentlichungen, sondern auch die zugrunde liegenden Daten von Bedeutung sind. Dies betrifft sowohl die eigentlichen Daten (das „Versuchsergebnis“ in Form von Rohdaten) als auch Metadaten (s.u.).

Daraus lässt sich eine Reihe von Anforderungen ableiten, die in der Praxis oft nicht erfüllt sind. Die Möglichkeit der effektiven Recherche beispielsweise ist nur gegeben, wenn eine entsprechende Kategorisierung oder Beschreibung in einer geeigneten Form vorliegt. Eine solche Kategorisierung ist aber abhängig von der Fachdisziplin und gegebenenfalls sogar von der Institution, in der eine Arbeit erstellt wird. Die Bezeichnung von Kategorien kann dabei bereits sehr unterschiedlich ausfallen. Bei interdisziplinär besetzten Forschungsgebieten und interdisziplinären Forschungsansätzen ist dies besonders problematisch. Unterschiedliche Bezeichnungen für gleiche oder sich überlappende Kategorien erschweren die Recherche anhand dieser Kategorien.

Dass diesen Anforderungen in der Praxis oft nicht genügt wird, hat verschiedene Ursachen:

- § Formatprobleme (mangelnde Nachhaltigkeit und mangelnde Metadatenunterstützung proprietärer Formate),
- § Fehlende Infrastruktur,
- § Eingeschränktes Erfassen von Metadaten im Publikationsprozess,
- § Mangelndes Erfassen übergeordneter Metadaten
- § und Verlust von Semantik durch ungeeignete Repräsentation (Excel).

Beide oben genannte Anforderungen bedingen, dass Daten vollständig und nachhaltig gespeichert werden müssen. Dies betrifft sowohl sog. „Rohdaten“, als auch die daraus abgeleiteten Daten.

Definition Rohdaten:

Als Rohdaten werden in dieser Arbeit Daten definiert, die das direkte Ergebnis einer Untersuchung darstellen, wie etwa Messdaten oder Simulationsdaten. Sobald diese Daten einer Nachbereitung unterzogen wurden (z.B. Filtern), wird die Formulierung „abgeleitete Daten“ verwendet.

2.1.1 Problem der proprietären Formate

Die langfristige Verfügbarkeit und Verwertbarkeit von Forschungsergebnissen muss sowohl organisatorisch als auch technisch gewährleistet sein. Der Schwerpunkt in dieser Arbeit liegt auf der technischen Seite, auch wenn der organisatorische Aspekt (z.B. die Autorenunterstützung) ebenso wichtig ist.

Ein Problem bei der Verwaltung von Ressourcen aus wissenschaftlicher Arbeit stellen proprietäre Formate bei Dokumenten dar. Weit verbreitete Formate für Textdokumente und Tabellenkalkulation sind Programm- und Versions-gebunden, liegen binär vor und sind nicht standardisiert. Für die langfristige Verfügbarkeit (in digitaler Form) bedeutet dies, dass die entsprechende Programmversion mit verwaltet werden muss oder diese Dokumente in ein anderes, für die Verwaltung geeigneteres Format gewandelt werden. Dabei sind grundsätzlich zwei Anforderungen denkbar:

- § Das entsprechende Dokument soll so gesichert werden, dass das Layout exakt erhalten bleibt und der Ausdruck in der ursprünglichen Form möglich bleibt.
- § Das Dokument soll so gesichert werden, dass sowohl Suche als auch Zugriff mit der maximalen Funktionalität unterstützt werden.

2.1.2 Problem der Heterogenität wissenschaftlicher Ressourcen

Die Ergebnisse wissenschaftlicher Untersuchungen liegen oftmals in sehr heterogener Form vor, sowohl was die verwendeten Formate bzw. Applikationen betrifft, als auch hinsichtlich der zugrunde liegenden Strukturen bzw. Datenmodelle.

Definition Datenmodell:

Der Begriff Datenmodell beschreibt ein formales Modell der Struktur von Daten. Datenmodelle werden für das Speichern und den Austausch von Daten definiert. Der Begriff Datenmodell schließt im Unterschied zum später definierten Ontologie-Begriff rein syntaktisch definierte Datenstrukturen, wie etwa Dokumententypen-Definitionen für das XML-Format ein.

Diese sind zu unterscheiden in:

- § Datenmodell auf der Applikationsebene („internes Datenmodell“, je nach Sichtweise auch Metamodell)
- § Datenmodell auf der Anwendersebene („externes Datenmodell“)

Ein solches Datenmodell entspricht der „internen“ Repräsentation von Daten innerhalb einer Applikation. Ein CAE-System beispielsweise, welches das Ablegen von Planungsdaten in einer zu definierenden Struktur ermöglicht, gibt den Rahmen für diese Struktur vor. Die Möglichkeiten, die das System für Datenmodelle vorsieht (z.B. „ist-Teil-von“-Strukturen), also das Metamodell, werden hier als internes Datenmodell bezeichnet. Das Datenmodell, welches anwenderseitig zur Organisation der Daten eingesetzt wird (z.B. das in der Norm DIN 6779 beschriebene Kraftwerks-Kennzeichnungs-System), wird hier als externes Datenmodell bezeichnet.

Die Heterogenität von Datenmodellen erschwert das Auffinden von Daten sowie den Zugriff darauf, sofern die Datenquelle (z.B. eine Datei) nicht als „Blackbox“ aufgefasst wird.

2.1.3 Problem der Metainformationen

Definition Metadaten:

Als Metadaten werden in dieser Arbeit Daten aufgefasst, die andere Daten beschreiben („Daten über Daten“). Metadaten stehen somit nicht für sich selbst, wie etwa ein gespeicherter Messwert, sondern sie verweisen auf weitere Daten. Metadaten lassen sich unterscheiden in Metadaten auf Applikationsebene und Metadaten auf inhaltlicher bzw. fachlicher Ebene. Metadaten auf Applikationsebene definieren Datenmodelle, wie etwa Datenbankschemata. Metadaten auf fachlicher Ebene stellen zusätzliche domänen-spezifische Angaben über Daten bereit (beispielsweise die Angabe von durchschnittlichen Toleranzen).

Metadaten sind ein zentraler Aspekt beim Zugriff und bei der Auswertung von Daten. Sie sind erforderlich, um Daten korrekt interpretieren zu können. Dabei gibt es verschiedene Abstufungen, was die Abhängigkeit der Daten von ihren Metadaten betrifft. Ohne Angabe der physikalischen Einheit ist beispielsweise ein Datensatz, der Werte für Messgrößen enthält, nicht verwendbar. Eine weniger enge Kopplung besteht, wenn Metadaten die Beschreibung eines Teils der Versuchsbedingungen enthalten, die jedoch für die Interpretation der Daten nicht unmittelbar erforderlich sind (wie Angaben über die durchführenden Personen, Kalibrierungsdaten etc.). Es ist offensichtlich, dass für diese Art Metadaten kein Vollständigkeitsanspruch postuliert werden kann, da diese nicht die direkte Verarbeitung von Messdaten sondern die Beurteilung oder Einschätzung derselben betreffen. Eine scharfe Abgrenzung ist daher nicht möglich. Inwiefern etwa bei experimentellen Untersuchungen Informationen über die durchführende Person für die Archivierung der Daten eine Rolle spielen, ist nicht allgemein entscheidbar. Für Untersuchungen, bei denen Messverfahren zum Einsatz kommen, die eine qualitative Einschätzung auf der Grundlage manuellen Ablesens erfordern (etwa Verfärbungen) ist die Relevanz sicherlich höher einzuschätzen als für automatisch aufgenommene Messdaten.

Metadaten und Standard-Datenformate

Bezüglich der Relevanz von Metainformationen und der Struktur der Daten ergeben sich in verschiedenen Fachdisziplinen unterschiedliche Anforderungen. Diese werden durch die methodischen Schwerpunkte (grundlagenorientiert oder anwendungsorientiert) und inhaltliche Schwerpunkte bestimmt. Ein wesentlicher Unterschied liegt in den unterschiedlichen Modellansätzen, die den jeweiligen Untersuchungen zugrunde liegen. Werden beispielsweise Materialeigenschaften und deren Einfluss auf Prozesse (wie etwa Transportprozesse) für ein bestimmtes Problem eher makroskopisch oder mesoskopisch betrachtet, kommen Summenparameter zum Einsatz. Ein einzelner Parameter für die „Rauhigkeit“ etwa ist für gängige Druckverlust-Berechnungsmodelle ausreichend. Je weiter sich Berechnungsmodelle in Richtung mikroskopischer Betrachtungen entwickeln, desto weniger sind solche Parameter jedoch hinreichend.

Dies bedeutet für die Beschreibung der Metainformation über Untersuchungen in Verbindung mit dem entsprechenden Modellansatz, dass diese flexibel sein sollte und unterschiedliche Sichten auf dieselbe Problemstellung ermöglichen. Da Modellansätze weiter entwickelt werden, muss die

Beschreibung von Metainformationen an diese Entwicklung anpassbar sein. Auf der anderen Seite sollten die Metainformationen der unterschiedlichen Ansätze aufeinander abbildbar sein. Später wird deutlich, dass eine syntaktische Standardisierung (s.u.) an dieser Stelle große Nachteile hat. Bei interdisziplinären Forschungsarbeiten wird dies besonders deutlich. So lässt sich schwerlich ein Standard-Datenformat übergreifend für eine Reihe von Fachdisziplinen definieren. Werden unterschiedliche Formate für die verschiedenen Disziplinen entwickelt, die aufgrund von Überlappungen gegeneinander abgeglichen werden müssen, wird die Interpretation der Formate wichtig.

Definition Syntaktische Standardisierung:

Unter syntaktischer Standardisierung wird in dieser Arbeit die Festlegung eines Standarddatenformates auf der Grundlage einer Schema-Definition (etwa auf der Grundlage von XML-Schema) verstanden. Sie kann daher auch als Implementierung eines Datenmodells (siehe auch Abschnitt 2.1.2 auf Seite 7) verstanden werden. Diese erfolgt mit Hilfe einer geeigneten Repräsentationsform wie etwa XML. Die Repräsentationsform selbst muss nicht notwendiger Weise über eine formale Semantik (vgl. Abschnitt 2.2 auf Seite 10) verfügen.

Wie eine Instanz eines so definierten Standards zu interpretieren ist, geht nicht aus dem definierten Schema hervor. Es kann lediglich überprüft werden, ob eine solche Instanz (wie z.B. ein XML-Dokument) syntaktisch gültig ist.

Implizite Bezüge

Innerhalb eines Forschungsprojektes entstehen in der Regel eine Reihe unterschiedlicher Ressourcen, die zueinander in Bezug stehen. Diese Bezüge sind oftmals nicht explizit verfügbar. Dieser Umstand erschwert das Verständnis und die Auswertung von Ressourcen, da die erforderlichen Informationen personengebunden sind und nicht Software-gestützt recherchiert werden können. Zwar lassen sich bei textbasierten Dokumenten bestimmte Informationen auswerten, diese sind aber nur sehr bedingt geeignet, komplexere Abhängigkeiten darzustellen. Diese lassen sich über Verfahren der Sprachverarbeitung Literaturlisten auswerten, um zu ermitteln, wie Veröffentlichungen aufgrund von Zitaten bzw. Referenzen voneinander abhängen. Allerdings beschränkt sich der Ansatz auf statistische Auswertungen (welche Veröffentlichung wurde besonders oft zitiert, etc.). Für Fragestellungen der Form „wo wurde der Ansatz aus Veröffentlichung ... aufgegriffen und erweitert...“ sind auf diesem Wege keine präzisen und sicheren Aussagen möglich.

In bestimmten Fällen kann die Information über Abhängigkeiten oder Bezüge explizit gegeben sein, aber dennoch nicht Software-gestützt auswertbar vorliegen. Dies ist z.B. der Fall, wenn die entsprechenden Angaben nicht in standardisierter Form vorliegen und die Semantik damit nicht eindeutig ist.

Bezüge können auf sehr unterschiedliche Weise gegeben sein. Im Folgenden sind einige typische Beispiele beschrieben.

§ Datengrundlage zu Veröffentlichung oder Bericht

Ein Bericht oder eine Veröffentlichung beschreibt ein Prozessmodell, dass mit Hilfe von Messdaten validiert wurde. Die Messdaten liegen als Rohdaten in Dateien oder Messdatenbanken vor. Zusätzlich existieren Dateien (z.B. im Format einer Tabellenkalkulation), die aufbereitete Messdaten enthalten. Dies können etwa gefilterte Daten (Entfernen von Messrauschen, Isolieren stationärer Werte, Isolieren von Sprungantworten, etc.) oder aus Berechnungen abgeleitete Daten sein (z.B. Mengebilanzen, Modellparameter).

§ Metainformationen über Messdaten

So können z.B. Zellen aus Tabellenblättern in Tabellenkalkulationsprogrammen für Metainformationen verwendet werden.

§ Aufbauende Veröffentlichungen/Bericht

Eine Veröffentlichung baut auf den Ergebnissen vorhandener Veröffentlichungen auf. Dabei handelt es sich nicht um ein reines Zitieren (wie z.B. in Übersichts-Veröffentlichungen), sondern vielmehr um die Verwendung von vorhandenen Ergebnissen um darauf neue Untersuchungen aufzubauen.

2.2 Semantische Technologien

2.2.1 Der Ontologiebegriff

Historisch bezeichnet der Begriff Ontologie eine Teildisziplin der Philosophie, die sich mit der Erforschung dessen, „was ist“, auseinandersetzt. Die Idee einer Ontologie geht bereits auf Aristoteles zurück, der von der „Wissenschaft vom Sein, von den Wesensbestimmungen des Seienden“ (Meyers Taschenlexikon Bibliographisches Institut, Mannheim 1985) spricht, wohingegen die Bezeichnung Ontologie erstmals Mitte des 17. Jahrhunderts bei dem Philosophen Ch. Wolff vorkommt. In der Informatik verbindet man eine weitere Vorstellung mit dem Begriff Ontologie. Die am häufigsten zitierte Definition des Begriffs Ontologie in diesem Bereich stammt von Gruber (1993):

„An ontology is an explicit specification of a conceptualization. The term is borrowed from philosophy, where an Ontology is a systematic account of Existence. For AI systems, what 'exists' is that which can be represented.“

Ein Kernpunkt der obigen Definition ist der Begriff der „conceptualization“. Dieser wiederum ist folgendermaßen definiert (vgl. Gruber, 1993):

„A body of formally represented knowledge is based on a conceptualization: the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them [...]. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly.“

Eine Ontologie ist also eine explizite (und damit syntaktisch repräsentierbare) Spezifikation einer Konzeptualisierung, d.h. einer abstrakten, gedanklichen Vorstellung (der Begriffe) eines Weltausschnitts. Der Aspekt der expliziten Repräsentation wird in Welty und Ide (1999) wie folgt definiert:

„A formal ontology is a complete symbolic description of what things in some domain can be. This description is specified in a formal notation, such as a logic or calculus, in which all the possibilities are either enumerated or stated inductively. In other words, the ontology is the definition of the classes, relations, constraints, and rules of inference a knowledge-based system will use.“

Erst diese Definition betont das Vorhandensein einer streng formalen, logikbasierten Theorie als Grundlage der expliziten Repräsentation und demzufolge als wichtigen Baustein einer Ontologie.

Daconta et al. (2003) spezifiziert dies noch etwas genauer. Der Autor betont, dass Ontologien nicht nur grundlegende Begriffe und Relationen eines Themengebietes beinhalten, sondern auch Regeln zum Kombinieren von Begriffen und Relationen, um Erweiterungen der Wissensbasis definieren zu können. Noch etwas genauer: Ontologien beinhalten Objekte, Beziehungen zwischen den Objekten, Attribute und Funktionen für diese Objekte und Einschränkungen und regelhafte Aussagen auf Basis dieser Objekte. Wie zu sehen ist, schränken Daconta und Welty die Definition von Gruber sehr stark ein, da sie einen logikbasierten, regelbehafteten Ansatz bei der Definition propagieren, der bei Gruber völlig fehlt. Weiterhin beschränkt Welty den Einsatz einer Ontologie auf wissensbasierte Systeme. Bei einer genaueren Betrachtung der Definition von Gruber lässt sich hingegen feststellen, dass sie sich beispielsweise auch problemlos auf die Datenbankmodellierung und auf die objektorientierte Modellierung anwenden lässt. Streng genommen wäre demnach auch eine Softwarespezifikation in UML eine Ontologie. Ein zentraler Unterschied zwischen einer Ontologie und einer Softwarespezifikation ist jedoch, dass die Ontologie auch während der Laufzeit als explizites Wissensmodell jederzeit zugreifbar bleibt, während das UML- Schema nur zur Designzeit der Software existiert und dann in der Software aufgeht.

Dies führt dazu, dass Änderungen in der jeweiligen Domäne problemlos in der Ontologie abgebildet werden können, z.B. durch das Hinzufügen eines neuen Konzeptes oder einer neuen Regel. Bei einer Software ist dieser Anpassungsprozess ungleich komplexer und schwieriger, da unter Umständen eine Vielzahl von Komponenten geändert werden muss.

Es existiert also eine Unschärfe im Ontologiebegriff, die im Folgenden auch in der Übersicht der Anwendungsbereiche deutlich wird.

Ein weiterer wichtiger Aspekt von Ontologien ist die Konsensbildung zwischen zukünftigen Benutzern. Der iterative Prozess der Konsensbildung dient der Festlegung der Konzeptualisierung der Ontologie. Borst (1997) modifiziert daher Grubers Definition wie folgt:

„Ontologies are defined as a formal specification of a shared conceptualization.“

Nur wenn die Benutzer der Ontologie sich auf eine gemeinsame Konzeptualisierung einigen, kann ein korrekter Wissensaustausch („knowledge sharing“) auf einer konzeptuellen Ebene stattfinden. In der Literatur wird dieser Aspekt sehr treffend formuliert:

„People can't share knowledge if they don't speak a common language.“
Davenport und Prusak (1998)

Diese „gemeinsame Sprache“ ist gerade auch im Rahmen der Datenintegration sehr wichtig, um einen reibungslosen Datenaustausch zwischen den Fachbegriffsmodellen verschiedener Organisationseinheiten und zwischen den expliziten Datenrepräsentationen der verteilten Datenbanken zu ermöglichen.

Zusammenfassend kann gesagt werden, dass Ontologien komplexes Wissen in strukturierter und computerverständlicher Form beschreiben. Dies führte dazu, dass Ontologien in einer Vielfalt von Bereichen für verschiedene Aufgaben angewendet werden. Innerhalb verschiedener Anwendungsgebiete werden unter dem Begriff Ontologie teilweise recht unterschiedliche Wissensstrukturen verstanden, wie z.B. Thesauri bzw. Glossare, Taxonomien, objektorientierte Modelle, aber auch vollständig formalisierte, logische Theorien. Der Ontologiebegriff dieser Arbeit folgt diesbezüglich den obigen Ausführungen von Gruber (1993) und soll im Folgenden noch einmal kurz formal dargestellt werden.

Eine Ontologie O definiert sich als Tupel

$$O = (C, A, \text{sub}, \text{AttTyp}, R),$$

Welches aus den folgenden fünf Komponenten besteht:

C ist die Menge der Konzepte der Domäne,

A ist die Menge der Attribute und Relationen der Konzepte aus C ,

sub bezeichnet die Subklassenbeziehung zwischen zwei Konzepten aus C , wobei gilt:

$\text{sub}(C_1, C_2)$, wenn C_1 eine Unterklasse von C_2 ist

AttTyp legt den Wertebereich des Attributs a aus A des Konzeptes C_1 aus C fest. $\text{AttTyp}(C_1, a, C_2)$ gilt genau dann, wenn das Konzept C_1 ein Attribut a besitzt, dessen Wertebereich auf Elemente des Konzeptes C_2 eingeschränkt ist.

Die Menge R besteht aus den Teilmengen R_{fix} und R_{dom} und kann beliebige prädikatenlogische Formeln enthalten, wobei R_{fix} Regeln über die vordefinierten Prädikate (sub , AttTyp) und R_{dom} beliebige sonstige Regeln aus der Domäne enthält.

2.2.1.1 Die Idee des „Semantic Web“

Auf der Grundlage von Semantischen Technologien und des Einsatzes von formalen Methoden zur strukturierten Beschreibung von Informationen entstand die Idee des „Semantic Web“ als Erweiterung des herkömmlichen Internets (vgl. Berners-Lee et al., 2001).

Ganz im Sinne des Prinzips der Ontologien (vgl. vorhergehender Abschnitt) steht hier die Idee der Maschinenlesbarkeit von Informationen im Vordergrund. Bei Webseiten kann diese durch Metadaten erreicht werden, die in einem standardisierten Format mit expliziter Semantik abgelegt werden. Dies geschieht üblicher Weise durch die Annotation von Webseiten.

Die Auswertung dieser Metadaten bedingt das Vorhandensein entsprechender Formate sowie einer Infrastruktur, die diese Formate unterstützt, d.h. auch die Interpretation der Daten und das Beantworten von Anfragen erlaubt. Diese Formate liegen in Form der Sprachen RDF(S) und OWL vor und werden im nachfolgenden Abschnitt vorgestellt.

Wenn Webseiten diese Art Metadaten enthalten, ist es im Prinzip möglich Aussagen über diese Webseiten und ihren Kontext abzuleiten. Während herkömmliche Verweise zwischen Ressourcen (im Allg. Webseiten) bereits eine Netzstruktur herstellen, so haben diese Verweise jedoch keine Semantik.

Bei der Umsetzung der Idee des Semantic Web ist nicht allein die formale Semantik der erzeugten Metadaten entscheidend, sondern auch die Entstehung gemeinsam genutzter Ontologien in Nutzergemeinschaften. Eine hohe Zahl unterschiedlicher Modelle über gleiche oder überlappende Sachgebiete würde es erschweren, einen Mehrwert durch semantische Metadaten zu erzielen. Eine Suchmaschine etwa, die in der Lage ist semantische Metadaten auszuwerten, wird nur dann einen Nutzen gegenüber der Stichwort-basierten Suche bringen, wenn für bestimmte Modelle von einem hohen Verbreitungsgrad ausgegangen werden kann.

Tatsächlich ist das Semantic Web derzeit in erster Linie als Vision zu sehen, auch wenn sowohl die notwendigen Repräsentationssprachen als auch die Werkzeuge zu deren Interpretation und zur Annotation von HTML-Seiten oder anderen Textquellen verfügbar sind. Daher ist derzeit nicht klar, ob und in welcher Form sich diese Vision tatsächlich im Sinne einer nächsten Generation von Internet manifestiert.

Als reale Umsetzung kommen sog. „semantische Wikis“ („semantic Wikis“) dieser Vision relativ nahe. Wikis stellen kollaborative, webbasierte Umgebungen zur Verwaltung von vernetzten Inhalten dar (vgl. Ebersbach et al., 2005). Als Erweiterung „einfacher“ Wikis erlauben semantische Wikis die Annotation von Inhalten mit semantischen Metadaten, wie z.B. in der auf dem weit verbreiteten Mediawiki basierten Umsetzung (vgl. Völkel et al., 2006). Diese Metadaten können in strukturierten Suchanfragen innerhalb des Wikis ausgewertet werden. Darüber hinaus kann die durch die Annotationen implizit entstehende Ontologie als Ganzes in einem Standardformat (OWL oder RDF) exportiert werden. Zwar existiert außerhalb der Wikis noch keine zentrale Auswertung der Metadaten in Form einer zentralen Suchmaschine, dennoch kommt das Konzept durch seinen kollaborativen, webbasierten Charakter der Vision eines semantischen Internets relativ nahe.

2.2.1.2 Sprachen

Im Umfeld des Ontology Engineering existiert eine ganze Reihe von Repräsentationssprachen mit unterschiedlicher Herkunft und unterschiedlicher Ausdrucksmächtigkeit. Die Sprachen OWL und RDF(S) (s.u.) sind eng mit der Vision des Semantic Web verbunden und über das W3C Konsortium standardisiert. Allen Sprachen gemein sind der formale, deklarative Charakter und die Maschinen-Interpretierbarkeit.

Für oder gegen die Verwendung einer Sprache sprechen neben Ausdrucksmächtigkeit, Verbreitungsgrad und Standardisierung auch die aktuelle Werkzeugunterstützung und die Eignung für eine bestimmte Domäne. Letzteres hängt vom typischen Charakter und dem Anwendungszweck von Domänenontologien ab. Dies betrifft z.B. folgende Fragestellungen:

- § welche Art von Fragen soll die Ontologie beantworten helfen und sollen dabei Informationen abgeleitet werden; geht es z.B. um einfache Suchprozesse, um typische Klassifizierungsprobleme;

- § sollen mit Hilfe der Ontologien bestehende Datenstrukturen abgebildet werden; ist die formale Interpretation der Sprache innerhalb der Domäne vermittelbar.

Im Folgenden werden einige der verbreiteten Repräsentationssprachen vorgestellt. Dabei wird in erster Linie der derzeitige Status der Sprachen dargestellt, nicht deren Historie. Letztere ist u.a. in Gómez-Perez et al. (2004) beschrieben. Dementsprechend wird an dieser Stelle auch nur ein Ausschnitt der insgesamt entwickelten Sprachen repräsentiert. Die Frame-basierte Sprache des im Rahmen dieser Arbeit verwendeten Werkzeuges Protégé wird in Abschnitt 5.2.1.2 vorgestellt.

RDF und RDF Schema

RDF steht für „Resource Description Framework“. Es handelt sich um eine formale Sprache zur Beschreibung von Ressourcen im Internet (typischer Weise Webseiten). RDF wird im Sinne eines Grundgerüsts zur Bereitstellung von Inhalten auch als semantische Schicht oberhalb von XML (siehe auch Abschnitt 4.2.1 auf Seite 67) und HTML begriffen (vgl. Decker et al., 2000).

Dabei ist festzustellen, dass XML nur die Basis für eine mögliche Syntax für RDF darstellt. RDF baut auf folgenden Grundelementen auf (vgl. Gómez-Perez et al., 2004):

- § Ressourcen, die jedwede Art von Daten (wie z.B. Webseiten) oder Dinge aus der realen Welt (wie z.B. Personen) repräsentieren,
- § „Uniform Resource Identifier“ (URI), die eine Ressource eindeutig bezeichnen,
- § „Properties“ (Eigenschaften), die Ressourcen beschreiben; Properties können hierarchisch über sog. „Subproperties“ organisiert werden,
- § Statements (Aussagen), die Ressourcen Werte für eine bestimmte Eigenschaft zuweisen; möglich sind auch Aussagen über Aussagen (das Prinzip der „Reification“).

RDF liegt damit ein Modell von Subjekt-Prädikat-Objekt Tripeln zugrunde. Jedes Tripel ist als eine Aussage über eine Ressource zu sehen. Die Reihenfolge der Aussagen ist für deren Interpretation nicht von Belang.

Die Ressource, welche durch die Aussage beschrieben wird, fungiert als Subjekt. Dies können, müssen aber nicht zwangsläufig Webseiten sein. Alles, was durch einen Bezeichner benannt werden kann, kommt als Subjekt (sowie auch als Objekt) in Frage. Als Bezeichner werden typischer Weise URIs verwendet. Das Prädikat in einer Aussage steht für eine bestimmte Eigenschaft der Ressource. So kann beispielsweise ein Prädikat „hat Autor“ dazu dienen, den Autor oder die Autorin einer Webseite auf maschinen-interpretierbare Weise zu benennen. Prädikate können auf einfache Werte oder auf eine andere Ressource verweisen. Durch letzteres ist es möglich, Bezüge zwischen Ressourcen herzustellen und damit Graphstrukturen aufzubauen.

RDF selbst ermöglicht keine Aussagen über Bezüge zwischen Ressourcen und Prädikaten. Dies wiederum ist Aufgabe von RDF Schema, abgekürzt RDFS. RDFS setzt einige der objektorien-

tierten Prinzipien wie etwa Klassenhierarchien und Vererbung um und weist große Parallelen zu dem noch vorzustellenden Frame-basierten Datenmodell (siehe auch Abschnitt 5.2.1.2 „Protégé Datenmodell“ auf Seite 97). Allerdings existieren auch wesentliche Unterschiede sowohl zu objektorientierten Datenmodellen als auch zu anderen Frame-Modellen. Eine Besonderheit von RDFS im Vergleich zu anderen Datenmodellen mit z.T. ähnlichen Prinzipien ist die Tatsache, dass nicht vorgeschrieben ist, wie Implementierungen mit Einschränkungen wie etwa Kardinalitätsbedingungen umgehen sollen. Hier besteht auch ein großer Unterschied zum relationalen Datenmodell, bei welchen Integritätsbedingungen erzwungen werden. Anders als bei Datenbanken können aus Bedingungen Informationen abgeleitet werden. Wenn beispielsweise für eine Eigenschaft „basiert auf“ festgelegt wird, dass der Wertebereich die Klassen „Theorie“ und „Ansatz“ umfasst, so kann für jedes RDF Statement in der Art² „A ,basiert auf‘ B“ abgeleitet werden, dass B sowohl eine Theorie als auch ein Ansatz sein kann (siehe auch Abschnitt 4.3 „Der SchoolOnto Ansatz“ auf Seite 71).

RDFS stellt eine Reihe von Modellierungsprimitiven bereit, die es ermöglichen, Schemata für RDF-Statements zu definieren. Dazu gehören eine Reihe vordefinierter Klassen, die sich alle aus der Klassendefinition für Ressourcen ableiten. Weiterhin enthält RDFS eine Reihe vordefinierter Eigenschaften (Properties), die es z.B. erlauben hierarchische Beziehungen zwischen Klassen oder die Klassenzugehörigkeit von Ressourcen auszudrücken.

OWL

OWL bezeichnet die „Web Ontology Language“, die vom W3C-Konsortium standardisiert wurde. OWL geht historisch betrachtet auf DAML+OIL zurück (vgl. Gómez-Perez et al., 2004). OWL weist Überlappungen mit RDF(S) auf und unterstützt die RDF(S) Syntax. OWL geht bzgl. der Ausdrucksmächtigkeit und der Möglichkeit per Inferenz Informationen abzuleiten über RDF(S) hinaus. OWL verwendet dabei sog. Beschreibungslogiken (vgl. Puppe et al., 2000), die sich v.a. für die indirekte Klassifizierung von Entitäten eignen.

OWL basiert auf einem Schichtenmodell, das die Sprache unterteilt:

- § OWL-Lite bietet die geringste Ausdrucksmächtigkeit und ist in erster Linie für einfache Taxonomien geeignet, die nur wenige Axiome benötigen. OWL-Lite hat große Überlappungen mit RDF(S), ist jedoch gegenüber letzterem eingeschränkt. OWL-Lite ist eine echte Untermenge von OWL DL (s.u.), was wiederum einige der Möglichkeiten von RDF(S) ausschließt.
- § OWL DL basiert auf einer Beschreibungslogik und stellt zusätzliche Modellierungsprimitive bereit. Die zugrunde liegende Beschreibungslogik stellt eine entscheidbare Untermenge der Prädikatenlogik erster Stufe dar. OWL DL bietet Möglichkeiten, die deutlich über RDF(S) hinausgehen. So können beispielsweise Klassenhierarchien über Inferenzmaschinen abgeleitet werden. Auf der anderen Seite verbietet OWL DL be-

² Zur Veranschaulichung wurde hier kein syntaktisch korrektes RDF-Statement verwendet.

stimmte Modellierungskonstrukte von RDF(S). So kann z.B. eine Klasse nicht Instanz einer anderen Klasse sein. Modelle in OWL DL können von verfügbaren Inferenzmaschinen bzgl. ihrer formalen Semantik vollständig ausgewertet werden.

- § OWL Full enthält die Modellierungsprimitive von OWL DL, verzichtet aber auf verschiedene Einschränkungen zugunsten der durch RDF(S) gegebenen Möglichkeiten sowie zu Lasten der vollständigen Auswertbarkeit durch Inferenzmaschinen. Damit können über OWL Full (wie auch schon für RDF(S)) Modellstrukturen erzeugt werden, für die keine Unterstützung durch Inferenzmaschinen im Sinne einer Interpretation auf der Basis der Prädikatenlogik erster Stufe existiert.

Jede Schicht ist in der darüber liegenden Schicht vollständig enthalten, d.h. jede OWL-Lite Ontologie ist eine gültige OWL DL Ontologie und jede OWL DL Ontologie ist eine gültige OWL Full Ontologie.

Ähnlich wie schon bei RDF(S) ist aus praktischer Sicht die Feststellung wichtig, dass sich die Semantik von OWL DL insbesondere bzgl. der Einschränkungen („Constraints“) wie z.B. Kardinalitätsbedingungen deutlich von dem unterscheidet, was Anwender mit einem (anwendungsorientierten) Vorwissen im Bereich Datenbanken mit hoher Wahrscheinlichkeit erwarten. Constraints dienen in OWL dazu, Informationen abzuleiten, nicht dazu, sie (analog zu Integritätsbedingungen bei relationalen Datenbanken) zu erzwingen. Existiert beispielsweise eine Kardinalitätsbedingung für die Eigenschaft einer Klasse, die einen Maximalwert von eins festlegt, so kann aus dieser Bedingung ggf. die Gleichheit von Individuen (Instanzen der Klasse) abgeleitet werden. Hierzu ein Beispiel: Für eine Klasse „Mathematisches Modell“ wird eine Eigenschaft „erweitert Modell“ mit dem Wertebereich „Mathematisches Modell“ und einem maximalen Auftreten von eins definiert. Weiterhin werden die Mathematischen Modelle „Modell A“, „Modell B“ und „Modell C“ beschrieben. Es wird mit Hilfe der erwähnten Eigenschaft definiert, dass „Modell A“ „Modell B“ erweitert. Auf die gleiche Art wird die Beziehung zwischen „Modell A“ und „Modell C“ beschrieben. Aus dieser Beschreibung kann folgen, dass „Modell B“ und „Modell C“ identisch sind.

An dieser Stelle wird noch ein anderes Charakteristikum von OWL deutlich: es existieren keine eindeutigen Bezeichner, d.h. zwei Element einer Ontologie sind nicht als unterschiedlich anzusehen allein aufgrund der Tatsache, dass sich ihre Bezeichner unterscheiden.

OWL stellt (in der derzeitigen Version) eine Sprache mit einer wohl definierten Grundlage dar, was sowohl das theoretische Fundament als auch die Unterstützung durch Implementierungen betrifft. Als besonders kritisch ist aus Anwendersicht zu sehen, dass die Sprache oberhalb der OWL-Lite Schicht relativ komplex ist und deren Semantik sich Anwendern ohne Hintergrundwissen im Bereich der formalen Logik und insbesondere der Beschreibungslogiken nur schwer erschließt. Darüber hinaus ist die Auswertung von OWL-Ontologien je nach Charakteristik der entwickelten Ontologie und nach Art der Anfrage relativ rechenaufwendig. Dies gilt v.a. für den Bereich, in dem die Sprache eine ihrer wesentlichen Stärken aufweist: bei der Verarbeitung unter-spezifizierter Information (vgl. Möller et al., 2006). Es existieren zwar Ansätze, die unter be-

stimmten Einschränkungen skalierbare und effiziente Auswertungsstrategien ermöglichen, wie z.B. von Möller et al. (2006) oder Horrocks et al. (2004) beschrieben. Für den gesamten Sprachumfang gilt dies jedoch derzeit nicht.

F-Logic

F-Logic (vgl. Kifer und Lausen, 1997) steht für „Frame Logic“ und stellt einen Ansatz dar, der Prinzipien aus objektorientierten bzw. frame-basierten Sprachen auf eine Untermenge der Prädikatenlogik erster Stufe abbildet. F-Logic unterstützt komplexe Objekte sowie das Prinzip der Vererbung. Damit beinhaltet F-Logik einige der Möglichkeiten von OWL und RDF(S), wenngleich z.T. mit unterschiedlicher Semantik. F-Logic bietet darüber hinaus die Unterstützung von Regeln und Anfragen. Die zugrunde liegende Syntax ist einheitlich. Eine Reihe von Modellierungsprimitiven wie z.B. Listen für Wertebereiche wird in F-Logic nicht unterstützt. Auch werden Schema-Informationen (wie z.B. Kardinalitätsbedingungen) in F-Logic nicht so ausgewertet, dass ohne zusätzliche (vom Anwender erstellte) Regeln Informationen abgeleitet werden.

Regeln und Anfragen können auf allen Ebenen formuliert werden. So kann über Regeln z.B. die Klassenzugehörigkeit definiert werden. Im Unterschied zu OWL wird jedoch keine Disjunktion unterstützt, d.h. eine Klassenzugehörigkeit kann beispielsweise nicht exklusiv definiert werden.

Im Unterschied zu RDF(S) stellen in F-Logic Eigenschaften (Relationen und Attribute) keine eigenen Objekte dar, die als solche unabhängig von Klassen definiert und referenziert werden können. Objekte haben in F-Logic eindeutige Bezeichner, d.h. es gilt die sog. „unique name assumption“.

Die Stärke von F-Logic liegt in der Flexibilität und Mächtigkeit als Regelsprache sowie der Verfügbarkeit effizienter Algorithmen zur Auswertung. Abgesehen von nicht standardisierten Erweiterungen fehlen Regeln im Sprachumfang von RDF(S) und OWL.

2.2.1.3 Werkzeuge

Inzwischen existieren eine Reihe von Werkzeugen zur Erstellung und Verarbeitung von Ontologien. Diese lassen sich grob klassifizieren:

- § Editoren bzw. Entwicklungsumgebungen zur Erstellung von Ontologien (einschließlich Axiomen und Regeln), zur Annotation von Dokumenten, etc.
- § Laufzeitumgebungen zur Interpretation von Ontologien, zur Unterstützung von Anfragen, zur Realisierung Server-basierter Middleware, etc.

Tatsächlich verläuft zwischen diesen Kategorien keine scharfe Trennlinie. So kann die Auswertung von Axiomen oder Regeln bereits zum Zeitpunkt des Editierens für den Anwender notwendig sein. Dies gilt z.B. für Konsistenzprüfungen, die Ableitung von Daten auf der Schema-Ebene oder der Fehlerbehebung.

Typischer Weise unterstützen Werkzeuge zur Erstellung und Verarbeitung von Ontologien in erster Linie eine Sprache. Weitere Sprachen werden z.B. durch Import- und Exportfunktionalitäten unterstützt.

Derzeit existieren nur wenige kommerzielle Anwendungen, besonders in der zweiten der oben genannten Kategorien. Eine Ausnahme bilden u.a. die Produkte RacerPro (Racer Systems GmbH & Co. KG) und OntoBroker[®] sowie OntoBroker OWL[®] (ontoprise GmbH).

Editoren unterscheiden sich abgesehen von den unterstützten Ontologie-Sprachen durch die Art der Anwender-Schnittstelle (Web-basiert oder als sog. „Rich Client“ bzw. Desktop-Anwendung), die Art der Anbindung an Inferenzmaschinen und die dadurch unterstützten Funktionalitäten, sowie durch die Erweiterbarkeit. Eine Übersicht von Editoren wird u.a. von Gómez-Perez et al. (2004) gegeben.

Insbesondere die Erweiterbarkeit, die Popularität und die Verfügbarkeit von verständlichen Schnittstellen haben im Rahmen dieser Arbeit zur Verwendung des Werkzeuges Protégé geführt, welches in Abschnitt 5.2.1 auf Seite 95 beschrieben wird.

2.2.2 Ontologien in den Natur- und Ingenieurwissenschaften

Im natur- und ingenieurwissenschaftlichen Bereich existieren eine Reihe von Ontologie-basierten Ansätzen für unterschiedliche Problemstellungen. Dies betrifft z.B. Problemlösungskomponenten bzw. Expertensysteme (z.B. Ceccaroni et al., 2001) oder die Unterstützung von Recherche-Prozessen im Internet für den verfahrenstechnischen Bereich. (vgl. Kokossis und Bañares-Alcántara, 2003).

Auf Gruber und Olsen (1994) geht die Entwicklung der EngMath-Ontologie zurück, welche grundsätzliche Konzepte von Anwendungen im Ingenieurs-Bereich beschreibt. EngMath bietet u.a. eine Basis für die formale Beschreibung physikalischer Größen sowie eine Ontologie-basierte Definition physikalischer Einheiten. Unter Verwendung von EngMath für den mathematischen Bereich wurde von Borst (1997) die Ontologie PhysSys zur Beschreibung generischer physikalischer Systeme entwickelt. PhysSis ist die Grundlage einer Modellbaustein-Bibliothek für physikalische Systeme.

Tudorache (2006) beschreibt einen Ansatz zur Unterstützung kollaborativer Produktentwicklungsprozesse im Maschinenbau. Entwickelt wurde ein Rahmenwerk, das die Konsistenzprüfung von Entwurfsmodellen ermöglicht und Änderungen zwischen verschiedenen Modellen propagiert. Grundlage hierfür sind Ontologien, die u.a. dazu dienen semantische Metadaten zu Entwurfsmodellen zu erfassen und auszuwerten.

Alberts (1994) hat eine Ontologie entwickelt, die der Beschreibung von ingenieurwissenschaftlichem Entwurfswissen auf der Grundlage von Systemtheorie dient.

2.2.2.1 Konzeptuelle Modelle nach Söderman

Söderman (1995) beschreibt einen allgemeinen Ansatz für die konzeptuelle Modellierung physikalischer Systeme. Der Autor greift dabei auch, aber nicht ausschließlich die Ansätze aus dem Bereich der qualitativen Modellierung auf.

Söderman (1995) betrachtet den Modellierungsprozess als Folge unterschiedlicher Phasen:

1. die physikalische Abstraktion,
2. die mathematische Abstraktion
3. und die Zusammenstellung der zu verwendenden Gleichungen (vgl. auch Abschnitt 3.4.3 „Allgemeine Vorgehensweise“ auf Seite 39).

Der Autor hebt die erste Phase als entscheidend hervor und weist auf die Vielzahl der möglichen Ansätze zur Verwendung von Ontologien an dieser Stelle hin. Zwei im Bereich der Ingenieurwissenschaften gängige Ansätze werden von Söderman (1995) beschrieben: die „distributed parameter ontology“ (Ontologie für örtlich verteilte Parameter) und die „lumped parameter ontology“ (Ontologie für Summenparameter). Die Ontologien werden als Konzept beschrieben, im Unterschied zu Ansätzen wie OntoCAPE (s.u.) jedoch nicht formal spezifiziert.

Nach Söderman (1995) zeichnet sich der Ansatz örtlich verteilter Parameter dadurch aus, dass die räumlichen Koordinaten eine Rolle für Ursache-Wirkung-Prinzipien spielen. Ontologien für örtlich verteilte Parameter führen bei der Umsetzung in mathematischen Modellen üblicherweise auf partielle Differentialgleichungen. Ontologien für Summenparameter können nach Söderman (1995) als Abstraktion des Prinzips verteilter Parameter angesehen werden. Ein Modell auf der Grundlage dieser Ontologie basiert auf miteinander verbundenen Teilmodellen, in denen jeweils keine örtliche Verteilung berücksichtigt wird. Die Teilmodelle entsprechen nicht den physikalischen Teilen des Systems, sondern den konzeptuellen Teilen, wie etwa der Systemenergie und der Systemmasse. Als Ergebnis der mathematischen Modellierung entstehen hier gewöhnliche Differentialgleichungen. Söderman (1995) hebt den Unterschied zum Ansatz örtlich verteilter Parameter bzgl. der Kausalität bei Änderungen von Systemparametern hervor: Wird die örtliche Verteilung nicht modelliert, gibt es keine ausgezeichnete Kausalität.

Söderman (1995) beschreibt als zentrales Element für eine Ontologie den „physical mechanism“. Letzteren definiert der Autor als „nexus of a set of physical quantities“, also die Vernetzung bzw. die Verknüpfung physikalischer Größen. Mechanismen werden von Söderman (1995) als sehr allgemeines Beschreibungselement definiert: Jede wahrnehmbare kausale Beziehung zwischen physikalischen Größen kann potentiell als Mechanismus gesehen werden. Dabei handelt es sich um ein abstraktes Konzept, das nicht eins zu eins abgebildet werden kann auf Systemkomponenten, sondern sich auf die zwischen den Komponenten auftretenden Phänomene bezieht. Mechanismen interagieren über die physikalischen Größen, die sie gemein haben. Darüber hinaus können Mechanismen modular betrachtet werden, d.h. die Summe an „feingranularen“ Mechanismen wird als ein komplexer Mechanismus angesehen.

Söderman (1995) beschreibt eine Klassifizierung von Mechanismen, die sich vor allem für den ingenieurwissenschaftlichen Bereich bewährt hat (aber dennoch auch in diesem Bereich Einschränkungen unterliegt, wie der Autor am Beispiel des Förderbandes beschreibt). Zunächst werden physikalische Größen kategorisiert. Dies kann zum einen nach der Art geschehen, wie sie wahrgenommen oder gemessen werden. „Across-quantities“ werden zwischen (zwei) örtlich verschiedenen Punkten gemessen (z.B. Fluss), „through-quantities“ dagegen an einem Punkt gemessen (z.B. Druck). Eine andere Kategorisierung ist die nach „effort“ (z.B. Kraft) und „flow“ (z.B. Geschwindigkeit).

Die grundlegendste Kategorie ist der sog. „Multiport-Mechanismus“. Dieser Mechanismus wird analog zu einem energetischen Prozess definiert. Eine Komponente interagiert mit ihrer Umgebung über den Austausch (bzw. die Übertragung) von Energie (oder einem im Sinne der Abstraktion analogen Prozess). Mit jedem Port werden Größen assoziiert. Unterkategorien dieses Mechanismus sind wiederum „Element-Mechanismen“ und „Interaktions-Mechanismen“. Aus diesen Kategorien leiten sich fünf Basis-Mechanismen ab: der „Quellen-Mechanismus“, Dissipation, Speicherung, Umwandlung und Transport bzw. Verteilung („distribution“). Letzterer ist dabei der einzige Interaktionsmechanismus.

Söderman (1995) illustriert seine Beschreibung mit einer Reihe von einfachen Beispielen.

Teile des Ansatzes wurden aufgrund ihrer Eignung für den ingenieurwissenschaftlichen Bereich übernommen (siehe Abschnitt 4.4 „Ontologiebasierte Aufbereitung und Verwaltung von Forschungsergebnissen“ auf Seite 82). Allerdings ist das Ziel im Rahmen dieser Arbeit nicht die möglichst detaillierte formale Beschreibung von Modellen, sondern die Unterstützung der Recherche und der Kategorisierung. Aufgrund der sich daraus ergebenden großen Zielgruppe wurden alle Teilontologien (für mathematische Modelle, für Ressourcen, etc.) in ihrem Detaillierungsgrad eingeschränkt.

2.2.2.2 Batres et al.: Ontologien für den Austausch von Modellen

Batres und Naka (2000) haben Ontologien für den Daten-Austausch zwischen Softwarewerkzeugen im Bereich der Verfahrenstechnik entwickelt. In Batres et al. (2001) ist die Anwendung auf die Verwaltung von Modellbibliotheken beschrieben. In Kraines et al. (2005) ist ein Ansatz für die kollaborative Entwicklung und Verwendung von Modellen beschrieben. Die Autoren unterscheiden zwischen unterschiedlichen Arten von Modellen (vgl. Abschnitt 3.1):

- § Simulationsmodelle zur numerischen Simulation, z.B. CFD-Modelle.
- § „Analysemodelle“, die das Verhalten von Systemen durch eine im Vergleich zu Simulationsmodellen stärkere Vereinfachung von inneren Vorgängen beschreiben. Als Beispiel werden etwa neuronale Netze oder einfache mathematische Modelle (etwa zur Kosten-schätzung) genannt. Laut Kraines et al. (2005) ist der Zweck dieser Art von Modell die Analyse von Beobachtungen des realen Systems.

- § „Feedback models“; welche ausschließlich für dynamische Systeme verwendet werden und qualitativer Natur sind.
- § Koeffizienten-Modelle („Coefficient Models“), die als empirische Modelle nach Kraines et al. (2005) kein Domänenwissen innerhalb des Modells (der Gleichungen) selbst enthalten.
- § „Database Query Models“; bei denen der Eingang einer Anfrage an eine Datenbank entspricht und der Ausgang dem Ergebnis der Anfrage entspricht.

Die Aufteilung ist insofern willkürlich, als die Trennlinien zwischen den Kategorien eher unscharf sind. So lassen sich etwa Simulationsmodelle oder Koeffizienten-Modelle unter bestimmten Voraussetzungen in „Database Query Models“ überführen, wobei z.B. diskrete Modellparameter in der Anfrage verwendet werden können.

Die Autoren beschreiben einen Ansatz, der mehrere Schritte zur verbesserten Auffindbarkeit und Wiederverwertbarkeit von Modellen umfasst:

- § die Modularisierung von Modellen und Verwendung von Modellkomponenten unter Verwendung eines geeigneten Maßes an Granularität und die Entwicklung entsprechender Schnittstellen für die Komponenten;
- § die Entwicklung von Ontologien zur Beschreibung von Modellen (und verwandten Konzepten, wie etwa physikalischen Phänomenen).

Die Autoren verwenden eine generische sog. „Upper-Level-Ontology“ zur Beschreibung allgemeiner, domänenunabhängiger Konzepte. In domänen-spezifischen Erweiterungen werden Konzepte wie z.B. Material definiert.

Die Zielbereiche sind hier wesentlich breiter angelegt als in der vorliegenden Arbeit (etwa durch Einbeziehung von Kostenmodellen). Die vorgenommene Klassifizierung erscheint aus den o.g. Gründen nicht geeignet, um als Grundlage für Metadaten zu wissenschaftlichen Ergebnissen zu dienen (wie in Abschnitt 4.4 „Ontologiebasierte Aufbereitung und Verwaltung von Forschungsergebnissen“ auf Seite 82 beschrieben).

2.2.2.3 OntoCAPE

Sehr umfangreiche Arbeiten zur Unterstützung verfahrenstechnischer Entwicklungsprozesse sind im Umfeld der Entwicklung und Anwendung der Ontologie OntoCAPE zu finden. CAPE steht hier für „Computer-Aided Process Engineering“. OntoCAPE wurde im Rahmen des Sonderforschungsbereiches 476 (IMPROVE) entwickelt und unterstützt die integrierte Beschreibung von Dokumenten, Arbeitsprozessen und Produktdaten in der Verfahrenstechnik (vgl. Morbach et al., 2007). Ein weiteres Einsatzgebiet von OntoCAPE ist die mathematische Modellierung (vgl. Yang et al., 2004).

OntoCAPE geht auf das Datenmodell CLiP zurück (vgl. Bayer und Marquardt, 2004). CLiP erlaubt die Beschreibung von Produkten im verfahrenstechnischen Bereich sowie auch von Arbeitsprozessen im Lebenszyklus verfahrenstechnischer Anlagen.

OntoCAPE ist durch mehrere Schichten, Module und Partialmodelle gekennzeichnet. Die oberste Schicht ist ein Metamodell, das den Sprachumfang definiert. Die derzeit vorliegende Version von OntoCAPE ist auf der Grundlage von OWL (siehe Abschnitt 2.2) modelliert. Da OWL keine Metamodellierung (Instanziierung von Klassen über mehrere Ebenen) unterstützt, konnte diese Schicht nicht konsequent umgesetzt werden (vgl. Morbach et al., 2007). Weitere Schichten beinhalten einen sog. „Upper Layer“, in dem generische, domänenunabhängige Konzepte beschrieben sind, eine konzeptionelle, domänenspezifische Schicht, eine anwendungs-orientierte Schicht und schließlich eine anwendungs-spezifische Schicht.

Die Strukturierung durch Module wirkt quasi orthogonal zu den Schichten. Module sind relativ unabhängige Teilmodelle innerhalb der gesamten Ontologie. Partialmodelle wiederum fassen Module für bestimmte, in sich relativ geschlossene Bereiche zusammen.

Die oberste Schicht von OntoCAPE enthält drei Module. Das Modul „System“ beschreibt grundlegende Elemente wie „system“, „system property“ oder „quantity“. Ein System kann durch seine Subsysteme beschrieben werden, die wiederum Aspekte wie den strukturellen Aufbau oder das Verhalten des Systems beschreiben. Weitere Module sind „tensor quantity“ zur Beschreibung von Tensoren sowie „coordinate system“.

OntoCAPE enthält als Teil der konzeptionellen Schicht sowie der anwendungs-orientierten Schicht die Partialmodelle „chemical process system“ für physikalisch-chemische Prozesse und den entsprechenden operativen Betrieb in verfahrenstechnischen Anlagen, „material“ und „model“. Letzteres Partialmodell betrifft Modelle und die Modellentwicklung.

Der Aufbau zeigt, dass der Fokus der Ontologie auf Prozessen und verfahrenstechnischen Anlagen im industriellen Umfeld liegt, weniger auf Versuchsaufbauten im Bereich von Grundlagenuntersuchungen. So sind beispielsweise verfahrenstechnische Standard-Operationen für industrielle Prozesse formal beschrieben. Dies schließt die Beschreibung gängiger mathematischer Modelle für diese Operationen ein.

Wie bereits erwähnt, existieren für OntoCAPE eine Reihe von Einsatzgebieten, darunter auch die Computer-gestützte Prozess-Modellierung (vgl. Yang et al., 2004). Nach einem modularen Prinzip werden im von Yang et al. (2004) beschriebenen System Teilmodelle für physikalisch-chemische Phänomene bzw. die Berechnung von Stoffdaten ausgewählt und später angepasst. Grundlage hierzu ist die Ontologie. Die Teilmodelle wiederum stammen aus einer vordefinierten Bibliothek. In einem konzeptuellen Modell werden Systeme und deren Eigenschaften sowie die damit verbundenen physikalisch-chemischen Phänomene formal beschrieben. Ein Werkzeug zur automatischen Modellgenerierung („Model Generation Engine – MGE“) ermittelt die zu den Konzepten des durch die Ontologie beschriebenen Modells dazugehörigen mathematischen Modelle aus der Bibliothek. Schließlich wird auf der Ebene konkreter Modellinstanzen die Überfüh-

rung des konzeptuellen Modells in das mathematische Modell vorgenommen, was aufgrund der vordefinierten Abbildung als reiner Ersetzungsprozess möglich ist. Das bedeutet, dass Modellvariablen im mathematischen Modell durch konzeptuell beschriebene Systemeigenschaften belegt werden.

OntoCAPE geht bzgl. des Umfangs und des Detaillierungsgrades deutlich über den in dieser Arbeit beschriebenen Zweck (siehe Abschnitt 4.4 „Ontologiebasierte Aufbereitung und Verwaltung von Forschungsergebnissen“ auf Seite 82) hinaus. Allerdings gibt es auf einer abstrakteren Ebene Überlappungen. Die für OntoCAPE beschriebenen Einsatzgebiete lassen sich dementsprechend mit den für die vorliegende Arbeit entwickelten Ontologien nicht abdecken (was auch nicht dem Ziel der Arbeit entspricht).

2.2.3 Probleme und Grenzen der semantischen Technologien

Um die Grenzen semantischer Technologien verstehen zu können, müssen die Konsequenzen des deklarativen Ansatzes verdeutlicht werden. Eine Anfrage an ein System dieser Art löst einen Suchprozess aus. In Abhängigkeit von der formalen Repräsentation der Ontologien und dem zugrunde liegenden Paradigma sind verschiedene Evaluierungsstrategien möglich. Für bestimmte Problemstellungen ist dieser Ansatz sehr effizient. In Bereichen jedoch, wo prozedurale Ansätze sehr geeignet sind und gleichzeitig Performanz eine Rolle spielt, kann sich der Ontologie-basierte Ansatz als nicht praktikabel erweisen. In diesem Falle kann, sofern die zugrunde liegende Repräsentationssprache das hinreichend unterstützt, ein gemischter Ansatz verfolgt werden.

Als Beispiel sei hier das Problem der Einheitenkonvertierung angeführt. Unterschiedliche Einheitensysteme sind ein grundsätzliches Problem bei der Interoperabilität physikalischer Datenquellen (wie etwa Messdatensysteme). Dieses Problem wird gerne als Beispielanwendung für den Einsatz von Ontologien verwendet (vgl. Gruber und Olsen, 1994). Mit Hilfe von Ontologien lassen sich Einheitensysteme deklarativ und erweiterbar beschreiben. Auf diese Weise können auf der Grundlage eines Einheitenmodells beliebige neue Einheiten definiert werden. Der Bezug der unterschiedlichen Einheiten zueinander ist formal festgelegt. Jedes System, welches die Ontologien entsprechend auswerten kann, ist damit in der Lage Einheiten auf Konsistenz zu prüfen (etwa, ob zwei Einheiten sich auf dieselbe Größe beziehen) und Umrechnungen vorzunehmen.

2.2.3.1 Massendaten

Als Massendaten werden an dieser Stelle in erster Linie Simulations- und Rohdaten (vgl. Abschnitt 2.1 „Problembeschreibung“ auf Seite 4) bezeichnet, die in großem Umfang vorliegen und aus diesem Grund etwa für die Verwaltung und Bearbeitung in herkömmlichen Tabellenkalkulationsprogrammen ungeeignet sind.

Als problematisch kann sich der Ontologie-basierte Ansatz bei großen Datenmengen erweisen, sofern nicht allein Metadaten semantisch beschrieben werden. In den Prozesswissenschaften können große Datenmengen im Bereich von Simulationen (z.B. CFD-Simulationen) und automatisch erfassten Messdaten auftreten. Beispielsweise ergeben sich bereits bei einer Anlage im

Technikums-Maßstab mit 30 Messstellen und Messungen im Sekundentakt über 100.000 Messwerte in einer Betriebsstunde. Daraus folgt ein relativ hohes Datenaufkommen, auch wenn möglicherweise nur ein Teil der im Betrieb aufgenommenen Daten für spätere Untersuchungen relevant ist.

Für die Verwaltung der Ergebnisse von wissenschaftlichen Untersuchungen ist im Übrigen ein wichtiger Aspekt, dass ein möglichst großer Teil der Daten verfügbar bleibt, auch wenn er nicht vollständig verarbeitet wurde (im Rahmen von Analysen, Modellvalidierungen, etc.). Wenn nun Messdaten inklusive der Einheiten mit Hilfe von Ontologien beschrieben und verarbeitet werden, so bedeutet dies, dass die jedem Messpunkt anhängige physikalische Einheit als „Objektstruktur“ dargestellt und ausgelesen wird. Der Auswertungsprozess erfordert eine entsprechende Rechenleistung und Speicher.

Werden im obigen Beispiel die Werte von 5 Messstellen im Zeitraum von 6 Stunden angefragt, so würde im Extremfall (ohne Optimierungen) das Auslesen und Interpretieren der Einheiten als Ontologien und gegebenenfalls Umrechnungen über 100.000-mal erfolgen müssen. Dies könnte z.B. passieren, wenn eine Anwendung zur Datenanalyse oder auch nur zur reinen Datenvisualisierung auf ein Quellsystem zurückgreift, das diese Daten Ontologie-gestützt vorhält. An dieser Stelle wird deutlich, dass bei Massendaten Kompromisse erforderlich sind.

Ein möglicher Kompromiss besteht darin, die semantische Datenbeschreibung allein auf die Metadatenebene zu beschränken. Die Möglichkeiten semantischer Technologien können in diesem Fall nur zur Beschreibung und Selektion ganzer Datensätze oder Datenquellen genutzt werden.

2.2.3.2 Sicherheit/Qualitätssicherung in der Datenhaltung

Bei der Verwaltung von Daten aus wissenschaftlichen Untersuchungen spielt die Sicherheit eine große Rolle. Datenbankmanagementsysteme bieten an dieser Stelle ausgereifte Mechanismen, um die Zugriffsrechte auf Quellsysteme zu verwalten. Nutzerrechte (Lese-, Schreib- und Administrationsrechte in verschiedenen Formen) können feingranular verwaltet werden. Spezialisierte Systeme, wie CAE-Systeme und LIMS bieten ihre eigenen Rechteverwaltungen.

Bei semantischen Technologien sind bei den derzeit verfügbaren Systemen zum Management von Ontologien (und für damit verbundene Aufgaben) keine vergleichbaren Möglichkeiten gegeben. Das ist jedoch in erster Linie durch den Stand der Technik bedingt. Grundsätzlich bestünde sogar die Möglichkeit, Zugriffsrechte selbst mit Hilfe von Ontologien zu beschreiben.

Das Management von Zugriffsrechten und eine entsprechende Nutzerverwaltung lässt sich prinzipiell koppeln mit der Qualitätssicherung der Daten. Diese wird in diesem Fall mit dem Arbeitsablauf („Workflow“) bei der Durchführung von Untersuchungen abgeglichen. So kann beispielsweise sichergestellt werden, dass einzelne Schritte beim Erfassen von Informationen über experimentelle Abläufe autorisiert werden. Die damit verbundene Qualitätssicherung betrifft somit nicht allein die Verwaltung der Daten. Eine solche Funktionalität ist nur für spezialisierte Systeme (wie etwa LIMS) verfügbar. Mögliche Ansätze für zukünftige Entwicklungen wären

- § die Integration semantischer Technologien in herkömmliche Systeme mit der beschriebenen Funktionalität;
- § die Entwicklung spezialisierter Ontologie-basierter Systeme, welche die derzeit verfügbaren Systeme um die entsprechende Funktionalität erweitert.

2.2.3.3 Standardkonformität

Der Ontologie-Ansatz ist in erster Linie als Methodik begründet und nicht fest an eine Repräsentationsform gebunden. Es existieren unterschiedliche Repräsentationsformate/-sprachen für Ontologien, die auch konzeptionell voneinander abweichen (siehe Abschnitt 2.2 „Semantische Technologien“ auf Seite 10). Zwei der verfügbaren Sprachen haben den Status offizieller Standards des W3Cs (World Wide Web Consortium).

In Bezug auf die Interoperabilität von Systemen, die Austauschbarkeit von Informationen und die Planungssicherheit sind Standards für Datenformate wichtig.

2.2.4 Fazit

Semantische Technologien bieten eine Grundlage für die Anreicherung von wissenschaftlichen Ressourcen zur Unterstützung von Rechercheprozessen sowie zur Erhöhung der Nachvollziehbarkeit von Zusammenhängen und damit der Nutzung von Forschungsergebnissen. Zur Beschreibung mathematischer Modelle mit Hilfe von Ontologien und zur Erfassung des Modellierungsprozesses existieren einige Vorarbeiten.

Trotz der Standardisierung von Ontologie-Repräsentationssprachen ist der Reifegrad verfügbarer Systeme nicht vergleichbar etwa mit dem der Datenbanktechnologie. Dennoch existieren Anwendungen und Komponenten, die Lösungen für viele der Probleme bzw. Einschränkungen „klassischer“ Technologien ermöglichen. Das in Abschnitt 4 diskutierte Konzept und die in Abschnitt 5 beschriebene Implementierung zeigen, wie auf der Grundlage vorhandener Arbeiten (wie etwa dem SchoolOnto-Ansatz entsprechend Abschnitt 4.3 und vorhandenen Ansätzen zur Beschreibung von Modellen) Lösungen für wissenschaftliche Ressourcen entwickelt werden können.

3 Mathematische Modellentwicklung

Die Modellentwicklung ist ein zentrales Element der Prozesswissenschaften. Forschungsergebnisse und damit verbundene Daten sind nur interpretier- und verwertbar, wenn der Modellierungsprozess transparent ist. In den Prozesswissenschaften, aber auch in anderen Bereichen stehen bei der Modellentwicklung zwei Aspekte im Vordergrund:

- § Die Modellentwicklung als Werkzeug, um durch praktische Probleme motivierte konkrete Problem- oder Fragestellungen angehen zu können;
- § Der Erkenntnisgewinn durch a) Modelle und b) den Prozess der Modellentwicklung.

Der erste Punkt ergibt sich allgemein aus dem in Abbildung 1 gezeigten Zyklus. Die Auslegung einer neuen oder Modifizierung einer bestehenden Anlage etwa kann eine Motivation für die Entwicklung eines Modells sein. Die Abstraktion des gegebenen Systems, in diesem Falle der Anlage, ermöglicht über mehrere Schritte die konkrete, ggf. quantitative Beantwortung von Fragestellungen und schließlich die Rückkopplung auf das reale System. Letztere kann z.B. in Form einer modell-gestützten Regelung umgesetzt werden.

Nach Lunze (1995) folgt die Abstraktion des gegebenen technischen Systems einer in der betreffenden Fachdisziplin etablierten Ontologie. Diese beschreibt die Art der verwendeten Elemente und die Kompositionsregeln, durch welche ein System durch miteinander verkoppelte Teile beschrieben werden kann. Diese Ontologie (bzw. das „ontologische System“) gibt demnach die Prinzipien für die Modellbildung vor. Als Beispiel für die Verfahrenstechnik nennt Lunze (1995) Phasen und Phasengrenzen, die auf Bilanzräume führen, für welche Bilanzgleichungen aufgestellt werden.

Ontologien als Grundlage für Modellierungsprinzipien bestimmen also, wie ein zu beschreibendes System strukturiert und formalisiert wird. Nach Lunze (1995) legt eine solche Ontologie drei Modelleigenschaften fest:

- § Modellprimitive;
- § Regeln für die Kopplung der Modellprimitive zum Gesamtmodell;
- § Regeln für die Ableitung des Verhaltens aus den Modellgleichungen.

An dieser Stelle wird deutlich, dass die Art der Modellentwicklung viel über die Sichtweise realer Zusammenhänge innerhalb einer Fachdisziplin aussagt bzw. eng damit verkoppelt ist. Dies spricht wiederum für die Verwendung von Modellen bzw. Modellbildungen als Ausgangspunkt der formalisierten Beschreibung von Forschungsergebnissen.

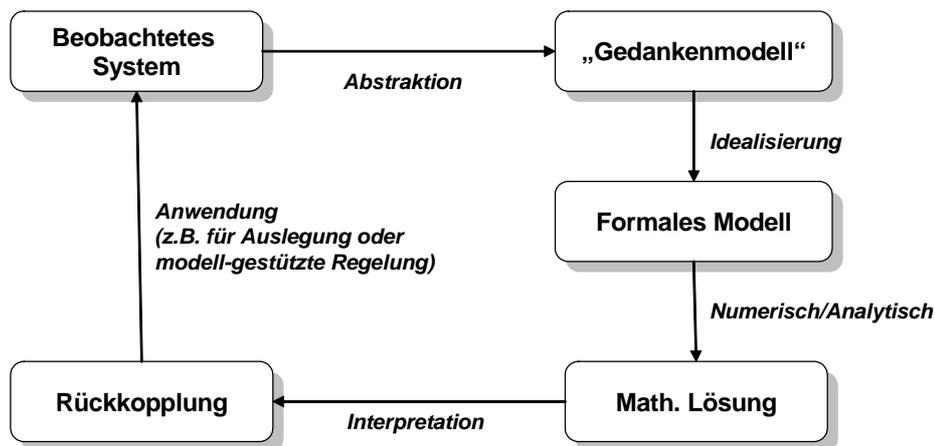


Abb. 1: Prozess der Modellentwicklung

Der Erkenntnisgewinn bei der Modellentwicklung ist eine Konsequenz aus der iterativen Vorgehensweise (siehe Abbildung 1). So erlaubt ein Modell etwa Rückschlüsse auf Systemteile bzw. – Komponenten, die sich der direkten Beobachtung oder Messung entziehen (vgl. Tabelle 1). Top und Akkermans (1994) etwa betonen den Lernprozess, der die Modellierung begleitet.

Ein Modell ist eine vereinfachte Darstellung eines interessierenden Systems. Dabei wird in den folgenden Abschnitten deutlich, dass die Abgrenzung zwischen „System“ und „Modell“ nicht absolut ist und z.T. gar nicht vorgenommen wird (siehe Abschnitt 3.1). Der Abstraktionsprozess erlaubt es, Lösungen für bestimmte Fragestellungen mit Hilfe formaler Methoden zu finden. Im Bereich der Prozesswissenschaften handelt es meist um mathematische und numerische Methoden. Der Abstraktionsprozess muss somit auf eine Modellbeschreibung hinführen, für die entsprechende Methoden zur Verfügung stehen. Ein System von Differentialgleichungen etwa stellt eine solche formale Beschreibung dar.

Die in dieser Arbeit relevanten mathematischen Modelle sind eine Abstraktion von technischen Systemen, die mit Hilfe der Mathematik beschrieben werden. Zur Beschreibung werden Gleichungen bzw. Gleichungssysteme verwendet.

Mathematische Modelle werden in vielfältiger Weise angewendet. Die Vorteile ihrer Verwendung sind in Tabelle 1 aufgelistet.

Tab. 1: Gründe für den Einsatz mathematischer Modelle

Einsatzgrund	Beispiel
Am realen System werden Eingriffe vermieden, so dass empfindliche Systeme nicht gestört oder zerstört werden.	<ul style="list-style-type: none"> § ökologische Systeme § gesellschaftliche oder soziale Strukturen § militärische Szenarien § Abläufe in der Medizin
Mathematische Modelle erlauben die Untersuchung nicht-existenter bzw. nicht-zugänglicher Systeme.	<ul style="list-style-type: none"> § Auslegung von neuen Anlagen § Machbarkeitsstudien § Hypothesen zu neuen Gesetzmäßigkeiten
Die Kosten für die Modellerstellung und Simulation sind in der Regel wesentlich geringer als die Kosten für Untersuchungen am realen System oder die Verwendung von physikalischen Modellen.	<ul style="list-style-type: none"> § komplexe Fertigungsprozesse § komplexe technische Systeme
Bei dynamischen Systemen kann der zeitliche Ablauf verkürzt oder auch gedehnt werden, was genauere Beobachtungen ermöglicht.	<ul style="list-style-type: none"> § geologische Vorgänge § langfristige Wetterentwicklung § Prozesse der Kernphysik

Hartmann und Bailer-Jones (1999) führen zusätzlich u.a. folgende Zwecke für den Einsatz von Modellen an:

- § Anwendung und Test von Theorien;
- § Modelle als Hilfsmittel zur Konstruktion von Theorien;
- § Fördern des Verständnisses abstrakter Theorien und Formalismen;
- § Modelle als Bestandteil wissenschaftlicher Erklärungen;
- § Modelle als Hilfsmittel zur Konstruktion und Auswertung von Experimenten.

In diesem Abschnitt werden grundlegende Betrachtungen zu Modellen und Systemen gemacht.

Zunächst werden die Begriffe definiert. Anschließend werden Eigenschaften von Modellen und Systemen vorgestellt. Des Weiteren werden der Modellierungsvorgang beschrieben und Rechner-

gestützte Systeme vorgestellt, die den Modellbildungsablauf unterstützen. Anschließend wird der Bezug der Modellentwicklung zu Ontologie-basierten Ansätzen erörtert.

3.1 Die Begriffe „System“ und „Modell“

Nach Hartmann und Bailer-Jones (1999) leitet sich der Begriff ‚Modell‘ vom Lateinischen ‚modulus‘ (das Maß) ab und wird in den Naturwissenschaften erstmals im 19. Jh. verwendet. Die Autoren beschreiben ein Modell kurz als „idealisierte Nachbildung eines konkreten Objektes oder Systems“. Sie unterscheiden zwischen Skalenmodellen, Analogmodellen und theoretischen Modellen. Unter letztere fallen die in dieser Arbeit behandelten mathematischen Modelle.

Ein System ist eine räumlich abgeschlossene, logisch zusammengehörende und zeitlich begrenzte Einheit, die voneinander abhängende Komponenten umfasst. Systeme sind im allgemeinen komplexe Gebilde, die nicht vollständig beschrieben werden können. Systeme besitzen Merkmale und Funktionalitäten. Unter Merkmalen sind Komponenten eines Systems zu verstehen, die verschiedene Ausprägungen annehmen können (vgl. Kowalk, 1996). Nach dieser Definition sind Systeme eine Abstraktion der Realität.

Zur Beschreibung von Systemen werden Modelle verwendet. Eine allgemeine Modelldefinition findet sich bei Minsky (1968):

Ein Objekt A ist ein Modell eines Objekts B für einen Beobachter C, wenn der Beobachter das Objekt A nutzen kann, um Fragen über B, die ihn interessieren, zu beantworten Minsky (1968)

Ein Modell ist ein vereinfachtes Abbild oder eine Abstraktion eines Gegenstandsbereiches bzw. Systems (vgl. Abbildung 2). Dabei werden nur die als wichtig erachteten Merkmale und Funktionalitäten des Gegenstandsbereiches als Attribut bzw. als Methode abgebildet. Diese Aspekte sind abhängig von den Zielen des Modellierers. Das bedeutet, dass ein Modell eines Systems nicht eindeutig ist (vgl. Marquardt, 1995).

Aus der von Hartmann und Bailer-Jones (1999) gegebenen Beschreibung naturwissenschaftlicher Modelle greifen diese „die für eine Problemstellung als wesentlich ereigneten Charakteristika (Eigenschaften, Beziehungen, etc.) eines Untersuchungsgegenstandes heraus und machen diesen so einem Verständnis bzw. einer weiterführenden Untersuchung zugänglich“. Für wissenschaftliche Modelle unterscheiden die Autoren zwischen Modellen einer Theorie und phänomenologischen Modellen. Erstere ergeben sich aus der Anwendung allgemeiner Theorien (Beispiel Newtonsche Mechanik) auf ein spezifisches Modellobjekt. Letzteres kann im Sinne der obigen Definition auch als System gesehen werden.

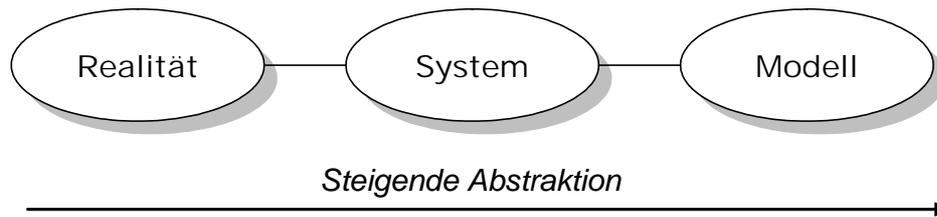


Abb. 2: Zusammenhang Realität - System – Modell

Zu einem System können viele verschiedene Modelle gebildet werden. Auf der anderen Seite können aus einem Modell weitere Modelle abgeleitet oder gebildet werden (vgl. Schwarze, 1990).

Hartmann und Bailer-Jones (1999) unterscheiden auf einer allgemeinen Ebene Skalenmodelle, Analogmodelle und theoretische Modelle. Zu letzteren gehören auch die hier behandelten mathematischen Modelle. Grundsätzlich lassen sich Modelle für den technischen Bereich nach Lohmann und Marquardt (1996) in vier Kategorien einteilen:

Als materielle Modelle werden z.B. Laboranlagen bezeichnet. Bei verbalen Modellen wird der betrachtete Gegenstandsbereich in natürlichsprachlicher Form wiedergegeben. Textuelle Beschreibungen können dabei die Voraussetzungen und Annahmen verbal beschreiben. Die semi-formalen Modelle beschreiben den Gegenstandsbereich durch strukturierte Darstellungsformen wie z.B. Graphen oder Tabellen. Die Beschreibung ist jedoch nicht eindeutig, da die einzelnen Elemente nur grob definiert sind. Eine eindeutige Beschreibung des Gegenstandsbereiches liefern formale Modelle. In diese Kategorie gehören die für diese Arbeit relevanten mathematischen Modelle.

Im Bereich der Systemtheorie wird der Begriff des Systems und des Modells häufig nicht getrennt (vgl. Unbehauen, 1980).

3.2 Abstraktionsebenen nach Top und Akkermans

Top und Akkermans (1994) beschreiben für die Ingenieursdisziplinen die verschiedenen Abstraktionsebenen im Modellierungsprozess. Analog zu Lunze (1995) legen sie Ontologien als Strukturierungsprinzipien für die Modellbildung zugrunde. Die Autoren beschreiben sowohl Informationen, die ein Modell selbst enthält, als auch Informationen über das Modell. Dabei unterscheiden sie drei Abstraktionsebenen, mit denen drei verschiedene Ontologien verbunden sind:

- § funktionale Komponenten;
- § physikalische Prozesse;
- § mathematische Bedingungen.

Die erste Abstraktionsebene ergibt sich durch einen in den Ingenieursdisziplinen verbreiteten Komponenten-basierten Ansatz. Die Autoren beschreiben das Aufstellen funktionaler Komponenten als Teilsysteme des zu modellierenden Gesamtsystems als Abstraktionsschritt. Für jedes Teilsystem erfolgt die Abgrenzung von der Umgebung und eine erste Kategorisierung. Letztere bezieht sich auf eine reale Komponente, d.h. für eine Pumpe, ein Ventil, etc. Teilsysteme und deren Verbindungen werden üblicher Weise als Knoten und Kanten in einem Graphen dargestellt.

In einem weiteren Abstraktionsschritt wird der das Verhalten jedes Teilsystems bestimmende Prozess oder Mechanismus behandelt (anhand der sog. Bond Graph-Methode). An dieser Stelle greifen oft übergeordnete Annahmen. Nach Top und Akkermans (1994) legt bereits die Anwendung der Bond Graph-Methode einen makroskopischen Ansatz zugrunde. Der Bond Graph beschreibt Eigenschaften wie etwa Offenheit/Geschlossenheit in Bezug auf Energieflüsse oder Arten von Rückwirkungen.

Die dritte Abstraktionsebene schließlich führt auf die mathematische Struktur eines Modells, was typischer Weise der Form des Differential-Algebra-Systems entspricht. Wie schon bei den vorangegangenen Abstraktionsebenen so kann auch diese als Graph aufgefasst werden, wobei Variablen den Kanten entsprechen und Knoten Beziehungen zwischen Variablen. Die Autoren unterscheiden an dieser Stelle explizit zwischen der mathematischen und der physikalischen Modellierung. Sie weisen darauf hin, dass diese in der Praxis oft als Einheit behandelt werden, was wiederum problematisch ist. Beide stehen für unterschiedliche Arten des Wissens. Die mathematische Repräsentation erfordert Know-How über Teilgebiete der Mathematik, während physikalische Modelle das prinzipielle Erfassen von Wirkzusammenhängen erfordern. An dieser Stelle lässt sich ergänzen, dass gerade unerfahrene Modellierer z.T. mathematische Zusammenhänge oder numerische Effekte mit physikalischen Effekten gleichsetzen. Als einfaches Beispiel wäre eine numerische Instabilität zu nennen, die physikalisch interpretiert wird. Aber auch bei geschlossenen mathematischen Lösungen sind (zumindest im Bereich der Lehre) solche Missverständnisse möglich. Dies ist etwa der Fall, wenn Modellparameter so gewählt werden, dass sie im Widerspruch zu den Annahmen stehen. Top und Akkermans (1994) heben hier die unterschiedliche Sichtweise auf Annahmen hervor. Eine Annahme ist zunächst ein notwendiger Schritt beim Abstrahieren des realen Systems. Sie kann gleichzeitig eine Anforderung an die Charakteristik des mathematischen Systems bedeuten. Die Autoren nennen als Beispiel Linearität. Durch sie ist die Form des mathematischen Systems eingeschränkt auf lineare Systeme (Anforderung). Gleichzeitig ist damit vorausgesetzt, dass dieser Ansatz für das betreffende reale System geeignet ist (Annahme). Die Annahme schränkt den Gültigkeitsbereich des Modells ein. Das Abbilden von mathematisch berechneten Ergebnissen auf physikalische Zusammenhänge ist außerhalb dieses Bereiches nicht mehr zulässig.

Hier besteht ein enger Bezug zur Semantik von Modellen, die im nächsten Kapitel diskutiert wird. Abschließend ist noch anzumerken, dass die von Top und Akkermans (1994) beschriebene Vorgehensweise stark durch die Ingenieurwissenschaften und eine komponenten-orientierten

Vorgehensweise geprägt ist. Die Überlegungen lassen sich jedoch größtenteils auf ein generischeres Verständnis der Modellentwicklung übertragen.

3.3 Eigenschaften von Modellen und Systemen

3.3.1 Semantik von Modellen

Die Semantik eines Modells gibt vor, wie ein Modell zu interpretieren ist. Grundsätzlich ist zwischen der formalen Semantik und der durch Anwender gegebenen Semantik zu unterscheiden. Letztere ist nicht eindeutig bestimmbar. Sie ergibt sich aus der Tatsache, dass Modelle sich nicht selbst erklären. So lassen sich Modellgleichungen zunächst als rein syntaktische Elemente ansehen (vgl. Hartmann, 2003). Tatsächlich lässt sich aus einem Satz von Differentialgleichungen ohne zusätzliche Kontextinformation wenig über das modellierte System aussagen. Allein aus der etablierten Nomenklatur und den mathematischen Eigenschaften sind zwar Rückschlüsse möglich (z.B. die Klassifizierung als statisches oder dynamisches Modell), dennoch ist das Gleichungssystem vor dem Hintergrund der Nachvollziehbarkeit nicht selbst-beschreibend sondern nur im formalen Sinne. Lunze (1995) sieht hingegen bei quantitativen Modellen die Semantik als eindeutig festgelegt. Der Autor argumentiert vor dem Hintergrund der dem Modellierungsprozess zugrunde liegenden Ontologie, welche die Prinzipien der Modellierung für eine Fachdisziplin definiert (s.o.).

Dieser scheinbare Widerspruch ergibt sich nicht bei einer Trennung der Semantiken. Die formale Semantik eines (quantitativen) mathematischen Modells ist durch die Mathematik eindeutig definiert. Ein System mathematischer Gleichungen lässt sich nur im Rahmen der Mathematik (etwa der differentiellen Algebra) formal auswerten. Der Rahmen für die Interpretation, vor allem durch automatische (rechner-gestützte) Verfahren ist damit festgelegt.

Dabei ist aus rein pragmatischen Erwägungen zu beachten, dass die Interpretation im Sinne eines deklarativen Verständnisses von mathematischen Modellen nur unter der Voraussetzung als festgelegt angesehen werden kann, dass entsprechende Algorithmen anwendbar sind. Wenn beispielsweise die Lösung eines gegebenen Differential-Algebra-Systems zu ermitteln ist, so kann die Forderung nach Vollständigkeit und Korrektheit des angewandten Algorithmus auf der Grundlage der differentiellen Algebra erhoben werden, sofern die Problematik von geschlossenen und Näherungslösungen an dieser Stelle außer Acht gelassen wird. Das Kriterium der Vollständigkeit würde hier jedoch eine sehr harte Anforderung darstellen, da dies beispielsweise für Optimierungsprobleme bedeutete, dass grundsätzlich das globale Optimum bestimmt werden muss.

Die intendierte Semantik eines mathematischen Modells wird nur im Kontext des Modellierungsprozesses deutlich. Mathematische Größen stehen im Zusammenhang mit Systemparametern. Die Art des aufgestellten Systems von Gleichungen ist abhängig vom Abstraktionsprozess, vom gewählten Koordinatensystem, von Annahmen, etc.. Die Interpretation des berechneten Systemverhaltens durch einen menschlichen Experten ist daher nur bei Kenntnis der Abstraktionsschritte, Annahmen, etc. möglich.

In Borst et al. (1997) wird einleitend anhand eines einfachen Beispiels auf die Semantik von Modellen Bezug genommen. Dabei machen die Autoren deutlich, dass es sich bei Modellgleichungen um Objekte handelt, bei denen zunächst nur die mathematische Bedeutung klar ist. Um unterschiedliche Interpretationen zu ermöglichen, muss z.B. das Konzept der physikalischen Dimension von Variablen bekannt sein. Und auch dann ist eine Interpretation ohne Hintergrundwissen zur modellierten Domäne nicht möglich.

Zusammenhang von Semantik und experimenteller Bestimmung von Modellparametern

Zu den beschriebenen Ausführungen zur Semantik mathematischer Modelle lässt sich ergänzen, dass vor dem Hintergrund der Experiment-gestützten Modellentwicklung und der Verwaltung von Forschungsergebnissen allein die Information über physikalische Wirkzusammenhänge nicht ausreichend ist. So ist etwa die praktische Bestimmung von Modellparametern oftmals entscheidend. Im Idealfall lassen sich Modellparameter unabhängig vom Modell selbst vorausbestimmen. In diesem Fall ist die Abbildung eines mathematischen Parameters auf eine physikalische Größe (wie in Borst et al., 1997) auch durch die separate Bestimmung quasi fixiert. Die Validierung des Modells ist damit ein Vergleich der mathematischen Kausalität mit der beobachteten Kausalität. Problematisch ist ein solcher Vergleich, wenn sich die Bestimmung des Parameters aus den Modellgleichungen selbst ergibt (mit Hilfe experimenteller Daten).

Zusammengefasst lässt sich festhalten, dass für die Verwendung und Weiterentwicklung die Semantik von Modellparametern möglichst transparent sein muss. Angaben zu deren Ermittlung bzw. zu grundlegenden Korrelationen sind an dieser Stelle hilfreich.

3.3.2 Klassifizierung von Modellen

In den folgenden Abschnitten wird eine grundlegende Klassifizierung von Modellen anhand verschiedener Eigenschaften vorgenommen.

3.3.2.1 Betrachtung des Zeitverhaltens

Bezüglich des Zeitverhaltens lassen sich Modelle in dynamische, statische und stationäre Modelle einteilen.

Statische Modelle erfahren keine Änderungen über die Zeit. Die Größe Zeit tritt nicht als Variable auf.

Stationäre Modelle erreichen nach einem Übergangsvorgang einen Gleichgewichtszustand, den stationären Zustand (vgl. Schwarze, 1990). In der Praxis wird der Begriff stationär häufig anstelle des Attributs statisch verwendet.

Die zu beschreibenden Systeme sind genau genommen immer dynamisch, auch wenn sie statisch erscheinen. Ein leerer Behälter lässt sich beispielsweise als ein statisches System interpretieren. Über eine längere Zeit gesehen wirken auf ihn jedoch Alterserscheinungen, die bei gewissen Un-

tersuchungen eine Rolle spielen können. Im Folgenden sei von dynamischen Systemen die Rede, wenn sich der Zustand des Systems innerhalb eines interessierenden Zeitraums verändert (vgl. Bossel, 1994). Im Rahmen dieser Arbeit werden Abwasserreinigungssysteme betrachtet, die in der Regel dynamische Systeme darstellen.

Wird das zeitliche Verhalten der Größen betrachtet, welche das Realsystem beschreiben, so lassen sich dynamische Modelle in zeitkontinuierliche und zeitdiskrete Modelle einteilen. Verändern sich die Größen über die Zeit hinweg kontinuierlich, so ist auch das Modell kontinuierlich. Diskrete Modelle dagegen sind gekennzeichnet durch diskontinuierliche Zustandsübergänge der Größen. Oft treten in Modellen sowohl zeitkontinuierliche als auch zeitdiskrete Zustandsübergänge auf (vgl. Liebl, 1992). Diese Modelle werden als hybrid bezeichnet.

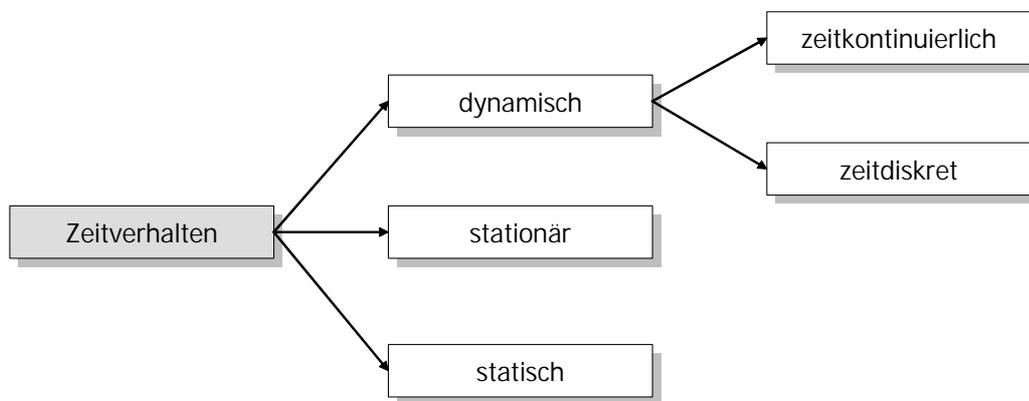


Abb. 3: Zeitverhalten von Modellen

3.3.2.2 Betrachtung des Raumverhaltens

Systeme haben in der Regel eine räumliche Ausdehnung. Wird ein System durch ein Modell abstrahiert, so kann die räumliche Verteilung für den Modellzweck irrelevant sein. Beispielsweise ist der Druck in einem geschlossenen Gasbehälter an jedem Ort gleich. Das Modell ist in diesem Fall ortsunabhängig. In vielen Modellen ist die räumliche Betrachtung jedoch von essentieller Bedeutung. So macht z.B. das Vernachlässigen der räumlichen Verteilung bei der Modellierung von Luftströmungen an Tragflügeln wenig Sinn. Modelle, in denen Modellgrößen abhängig von der räumlichen Verteilung sind, werden auch als verteilte Modelle bezeichnet.

Die ortsabhängigen Modelle lassen sich weiter beschreiben durch die Anzahl der betrachteten Dimensionen (ein-, zwei- bzw. dreidimensional) und durch die Unterscheidung zwischen raumdiskreten und raumkontinuierlichen Modellen (vgl. Bossel, 1994).

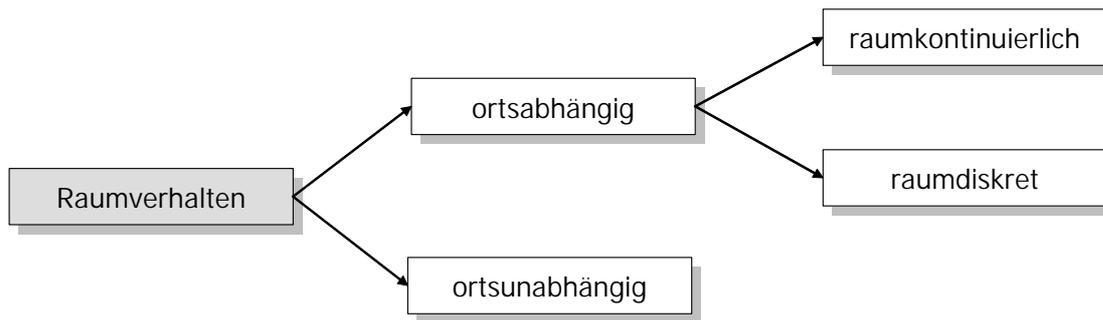


Abb. 4: Raumverhalten von Modellen

Es sei darauf hingewiesen, dass mathematische Modelle, die auf einem Rechner umgesetzt werden, in Bezug auf das Raum- und Zeitverhalten nur diskret berechnet werden können. Bei ausreichend kleiner Diskretisierung wird ein kontinuierliches Modellverhalten allerdings gut nachgebildet (vgl. Bossel, 1994).

Die zu beschreibenden Systeme werden in geschlossene bzw. offene Systeme unterteilt. Offene Systeme stehen im Austausch mit der Umwelt und werden von ihr beeinflusst oder haben selber Einfluss auf die Systemumgebung. Systeme sind nie vollkommen isoliert von ihrer Umgebung (vgl. Bossel, 1994). Im Folgenden sei von geschlossenen Systemen die Rede, wenn vorhandene Einflüsse der Umwelt auf das System bzw. des Systems auf seine Umgebung für die Problemstellung nicht funktionsrelevant sind.

3.3.2.3 Mathematische Modellattribute

Mathematische Modelle beschreiben Systeme mit Hilfe von Gleichungen bzw. Gleichungssystemen. Die verwendeten Gleichungen können theoretisch hergeleitet oder empirisch bzw. halbempirisch ermittelt werden.

Theoretische bzw. deduktive oder auch axiomatische Modelle basieren auf theoretisch hergeleiteten Gleichungen, die auf Wissen bzw. Hypothesen fundieren. Sie sind eine Nachbildung der Systemstruktur. Das Originalsystem muss daher erkannt und verstanden sein. Systeme, die durch theoretische Modelle beschrieben werden, werden in diesem Zusammenhang auch als glass-box bezeichnet (vgl. Bossel, 1994).

Empirische Modelle werden durch Gleichungen beschrieben, die aus Messungen von interessierenden Eingangs- und Ausgangsgrößen bzw. durch auf Beobachtung gründende Erfahrung gewonnen werden. In der Literatur wird anstelle von empirisch auch der Ausdruck experimentell verwendet (vgl. Möller, 1992). Das Systemverhalten wird nachgeahmt, das bedeutet, dass die Wirkungsstruktur des Originalsystems nicht interessiert und als black-box interpretiert werden kann. Das Modell ist streng genommen kein Modell des Systems, sondern ein Modell des Verhaltens (vgl. Bossel, 1994).

Des Weiteren existiert eine Mischform empirischer und theoretischer Modelle. In technischen Anwendungen sind die Wirkungsstrukturen von Systemen und die Parameter oft nur teilweise bekannt. Die zugehörigen Modelle bestehen aus Gesetzmäßigkeiten in Formelgestalt und aus technischen Kenngrößen, die empirisch ermittelt werden. Das Originalsystem lässt sich als grey-box oder als opaque (halbdurchsichtig) interpretieren (vgl. Bossel, 1994). Diese Modelle werden in der Praxis häufig als halbempirisch bezeichnet.

Als halbempirisch werden auch Modelle angesehen, die auf Plausibilität basieren. Die Modellgleichungen können nicht theoretisch hergeleitet werden. Die mathematische Beschreibung erscheint aber plausibel bzw. sinnvoll.

Bei deterministischen Modellen sind zufällige Veränderungen der Parameter ausgeschlossen. Alle interessierenden Parameter sind genau determiniert.

Bei stochastischen Modellen werden zufällige Einflüsse explizit berücksichtigt (vgl. Bossel, 1994). Tritt bei mindestens einem Modellparameter eine zufällige Streuung der Werte auf, so ist das Modell stochastisch. Die auftretenden Werte für zufällige Parameter werden unter Verwendung von Zufallszahlen generiert. Eine Simulationsberechnung liefert daher nur einen Ausschnitt aus der Menge aller möglichen Ergebnisse. Für die Systembewertung muss daher eine ausreichende Anzahl an Simulationsläufen durchgeführt werden.

Bei linearen Modellen bewirkt die Änderung einer Größe eine proportionale Änderung der anderen Modellgrößen. Nicht-lineare Modelle weisen dagegen nicht-proportionale Änderungen auf.

Eine analytische Lösung bedeutet eine geschlossene exakte mathematische Systembeschreibung ohne Näherungen, Linearisierungen, etc. Die analytische Lösung kann vor allem bei linearen Modellen verwendet werden.

Nicht jede Gleichung bzw. jedes Gleichungssystem lässt sich durch eine analytische Methode lösen. Häufig kann nur eine näherungsweise numerische Lösung bestimmt werden. Jede Gleichung lässt sich in der Form $f(x)=0$ darstellen. Die Nullstellen sind die gesuchten Lösungen der Funktion. Geometrisch betrachtet besteht das Problem darin, die Schnittpunkte des Graphen von f mit der Ordinate zu ermitteln. Die numerische Lösung muss in der Regel bei nicht-linearen Modellen verwendet werden.

3.4 Modellbildung

Unter Modellbildung wird eine zielorientierte Vereinfachung der Realität durch Abstraktion verstanden (vgl. Van Gigch, 1991). Die Modellierung erfordert dabei ein möglichst exaktes Wissen über das zu abstrahierende System, um die für die gestellte Aufgabe wesentlichen Faktoren in das Modell zu übernehmen. Dabei sind vorhandene Wissenslücken zu identifizieren.

In diesem Abschnitt werden in der Literatur dokumentierte Vorgehensweisen zur Modellbildung vorgestellt. Zunächst wird der Unterschied zwischen deduktiver und induktiver Modellbildung

erläutert. Danach wird eine allgemeine Vorgehensweise vorgestellt. Der Schwerpunkt bei der Beschreibung liegt neben der konzeptionellen und mathematischen Modellbildung auf der Überprüfung von Modellen mittels Validierung und Verifikation. Die programmiertechnische Umsetzung und die Durchführung von Simulationen werden nicht näher behandelt.

Zu beachten ist dabei, dass diese allgemeine Vorgehensweise nur einen Leitfaden darstellt. Die Abstraktion der Realität ist stets subjektiv. Der Modellierungsvorgang ist ein sehr intuitiver Vorgang und stark geprägt durch die Kenntnisse und Erfahrung des Modellierers (vgl. Marquardt, 1995). In Aris (1991) wird die Modellierung gar als „Kunst“ bezeichnet. Als weitere Randbedingungen sind die zur Verfügung stehenden Ressourcen wie Zeit, Arbeitskräfte, Rechnerleistungen, Geldmittel, früher verwendete und validierte Modelle, begleitende Versuche, u.a. zu nennen (vgl. Lugner und Bub, 1990).

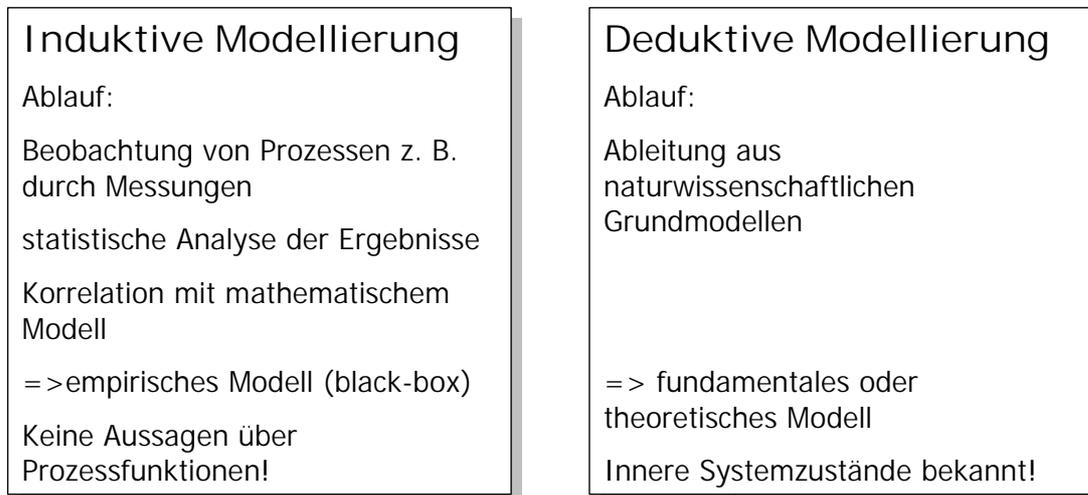


Abb. 5: Induktive und deduktive Modellbildung

3.4.1 Deduktive Modellbildung

Die deduktive Modellbildung (vgl. Abbildung 5) ist dadurch gekennzeichnet, dass mit Hilfe der Grundgesetze der Naturwissenschaften eine mathematische Formulierung der Vorgänge abgeleitet wird, um von einer qualitativen Vorstellung eine quantitative Beschreibung zu finden. Die deduktive Modellbildung wird auch als theoretische bzw. axiomatische Modellbildung bezeichnet (vgl. Möller, 1992).

Die deduktive Methode erfordert eine hinreichend qualitative Kenntnis des Systems. Das zu beschreibende System muss dabei nicht real existieren.

Der Aufwand für die Modellentwicklung ist in komplexen Fällen sehr hoch. Die Übertragbarkeit auf vergleichbare Systeme ist innerhalb der Gültigkeit der Voraussetzungen möglich und erlaubt somit einen wiederholten Modelleinsatz, der den hohen Entwicklungsaufwand rechtfertigen

kann. Ein weiterer Vorteil der deduktiven Modellbildung ist die Vertiefung des Systemverständnisses von Modellen, z.B. in Bezug auf worst-case Bereiche (vgl. Möller, 1992).

3.4.2 Induktive Modellbildung

Bei der induktiven bzw. empirischen oder experimentellen Modellbildung werden zunächst über eine Messung die zugänglichen Ein- und Ausgangsgrößen des Systems beobachtet. Mittels einer statistischen Analyse sind diese Beobachtungen mit einem vorgegebenen mathematischen Modell zu korrelieren, um einen Zusammenhang zwischen den wesentlichen Ein- und Ausgangsgrößen herzustellen und ein empirisches Modell zu erhalten (vgl. Abbildung 6). Durch den Vergleich von beobachteten Messgrößen am realen System mit den durch Simulation erhaltenen Ergebnissen wird unter Berücksichtigung von Fehlerkriterien die Gültigkeit des Modells iterativ überprüft (vgl. Möller, 1992).

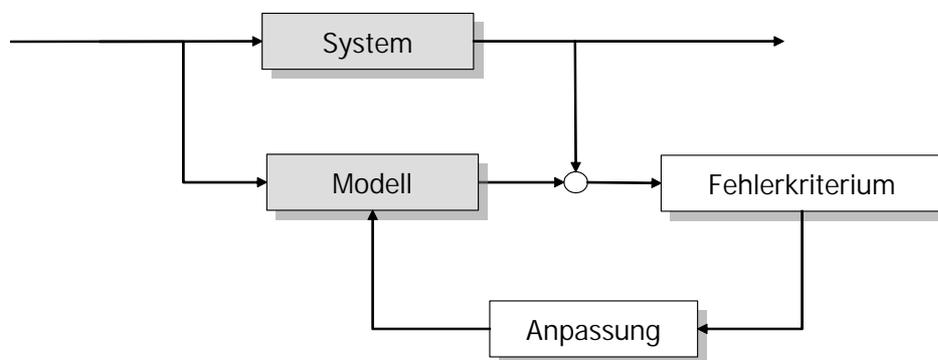


Abb. 6: Induktive Modellbildung nach Möller (1992)

Voraussetzung für eine induktive Modellbildung ist eine genaue experimentelle Untersuchung des Originalsystems. Komplexe Systeme lassen sich durch einfache Modellstrukturen relativ gut beschreiben. Die Übertragbarkeit auf weitere Systeme ist prinzipiell nicht gegeben, so dass jeder Betriebsfall erneut modelliert werden muss. Die Modellparameter haben nur arithmetische Bedeutung, eine elementare Interpretation des Modells ist nicht möglich (vgl. Möller, 1992). Das Modell gibt keine Aussage über die Systemfunktion (vgl. Marquardt, 1995).

3.4.3 Allgemeine Vorgehensweise

Abbildung 7 zeigt die grundsätzliche Vorgehensweise von der Problemstellung bis zur Simulation, wie sie in Lugner und Bub (1990) beschrieben ist.

In der Problemstellung wird der Modellzweck für die Modellentwicklung vorgegeben. Dabei wird bereits ein Fragenbereich definiert, auf den das Modell Antworten geben soll. Je genauer der Modellzweck definiert wird, desto präziser, schärfer und knapper kann das Modell formuliert werden (vgl. Bossel, 1994).

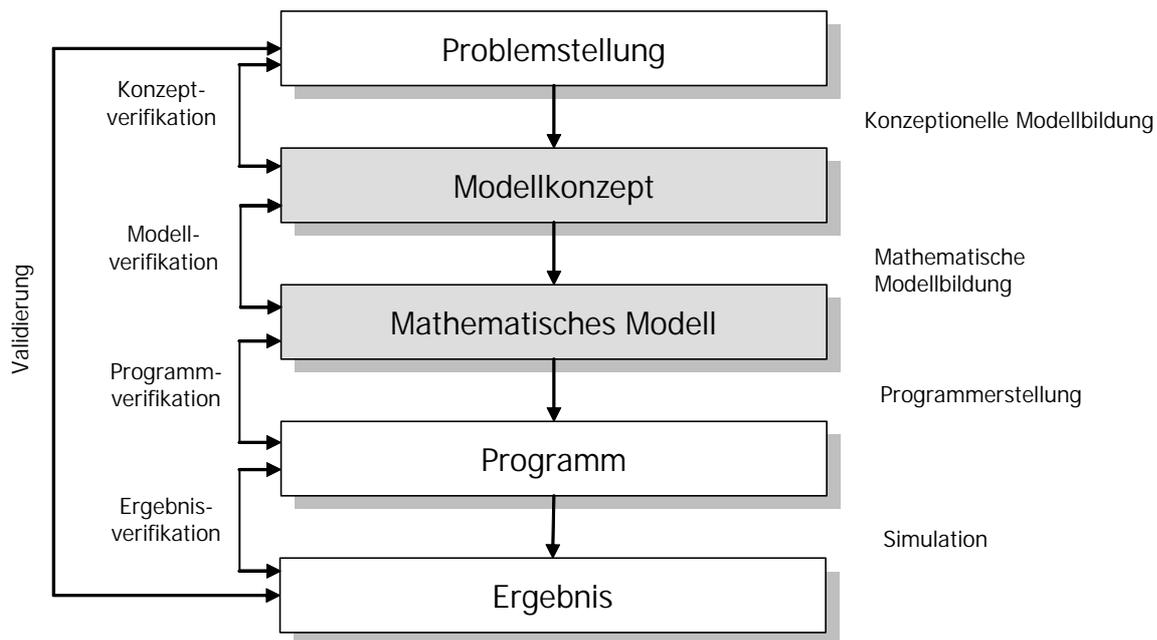


Abb. 7: Vorgehensweise bei der Modellbildung und Simulation nach Lugner und Bub (1990)

Das Modellkonzept besteht aus einer Beschreibung eines modellierten Ersatzsystems. Die konzeptionelle Modellbildung ist hierbei klar getrennt von der mathematischen Beschreibung. Mathematische Gleichungen treten noch nicht auf. Das Konzept wird verbal beschrieben, Diagramme und Bilder ergänzen die Beschreibung. Auf die konzeptionelle Modellbildung wird in Abschnitt 3.4.4 näher eingegangen.

Im Anschluss an das Modellkonzept wird das mathematische Modell mittels mathematischer Gleichungen und Funktionszusammenhänge formuliert. Die mathematische Modellbildung wird in Abschnitt 3 behandelt.

Durch konzeptionelle und mathematische Modellbildung ist der Modellierungsvorgang zunächst abgeschlossen. Das mathematische Modell wird anschließend in Programmcode umgesetzt, mit dessen Hilfe numerische Simulationen durchgeführt werden.

Zwischen den einzelnen Arbeitsschritten dienen Verifikationsschritte als Kontrolle (vgl. Lugner und Bub, 1990). Bei der Validierung wird schließlich die Modellgültigkeit überprüft, wobei eine Modellmodifikation erforderlich sein kann.

Eine alternative Darstellung der Modellbildung ist in Abbildung 8 gegeben (vgl. Lunze, 1995). Sie enthält Schritte der konzeptionellen Modellbildung, die im anschließenden Abschnitt erörtert ist.

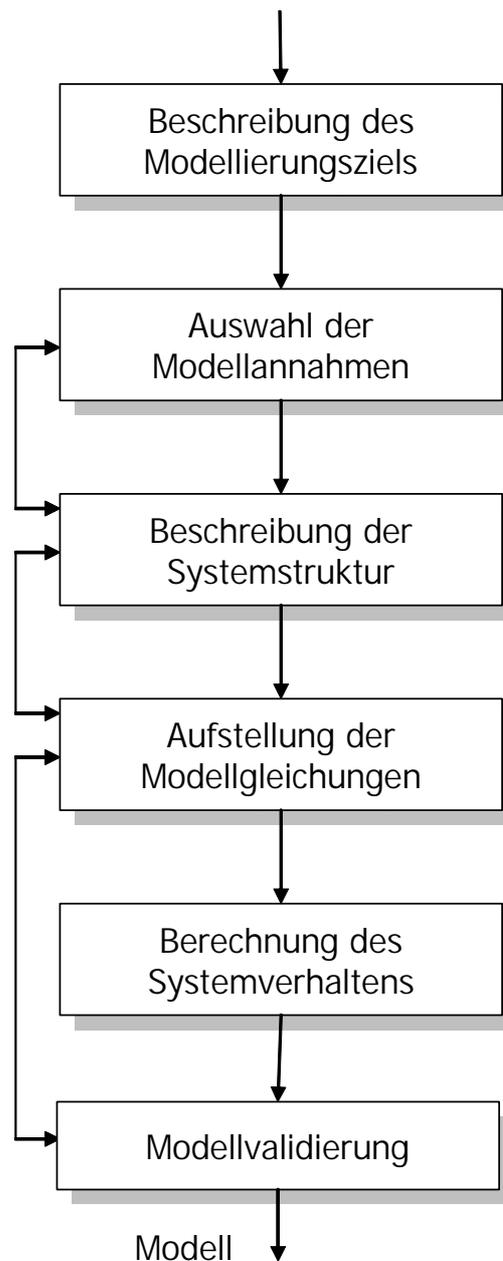


Abb. 8: Modellbildung nach Lunze (1995)

Einen wichtigen Aspekt bei der Modellentwicklung stellt die Tatsache dar, dass es sich um einen Lernprozess handelt. Top und Akkermans (1994) beschreiben die Entwicklung von Modellen als kreative Entwurfstätigkeit. Die Autoren betonen besonders den begleitenden Lernprozess und die Revision der ursprünglichen Modellannahmen. Letztere geben die Rahmenbedingungen vor, unter denen ein Modell validiert kann.

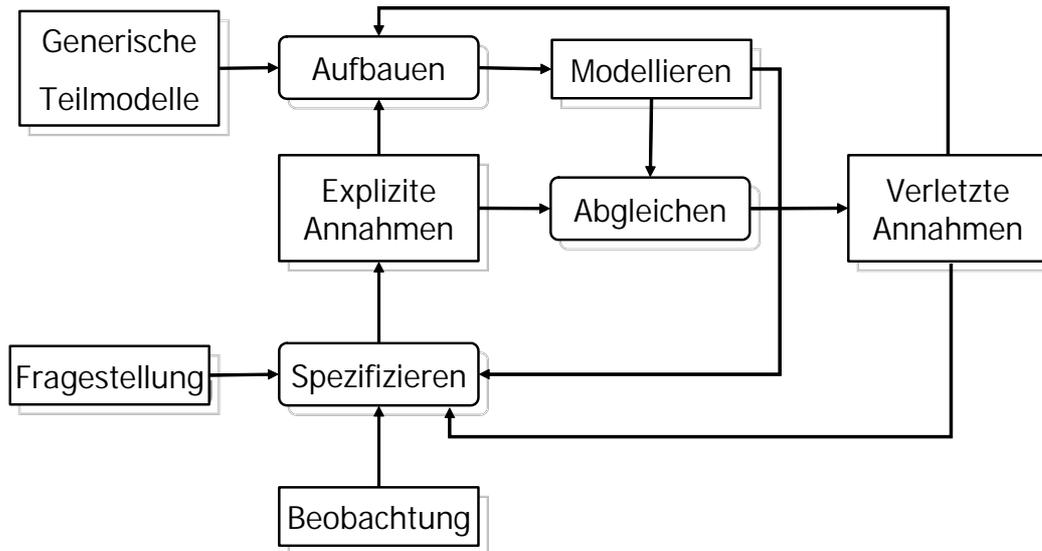


Abb. 9: Modellentwicklung entsprechend Top und Akkermans (1994)

3.4.4 Konzeptionelle Modellbildung

Die Hauptaufgabe der konzeptionellen Modellbildung ist die Festlegung der Systemabgrenzung. Dabei müssen die relevanten Systemelemente bestimmt werden (vgl. Liebl, 1992). Das Modell muss problemorientiert sein, d.h. der Modellzweck muss erfüllt werden. Ein allumfassendes Modell ist nicht problemorientiert und auch nicht realisierbar. Der Detaillierungsgrad unterliegt keinen prinzipiellen Grenzen. Limitierende Faktoren sind die notwendigen Rahmenbedingungen, wie z.B. die vorhandenen Arbeitskräfte, Zeit, Geldmittel und für die Validierung benötigten Daten.

Die intendierte Zweck und die Auswirkungen limitierender Rahmenbedingungen (etwa auf Vereinfachungen) gehören zu den Kontextinformationen, die zum Verständnis und zur Bewertung eines Modells wichtig sind. An dieser Stelle greift die Aufwertung von Metainformation zu Modellen, wie sie durch den in dieser Arbeit beschriebenen Ansatz vorgenommen wird. Metainformationen (d.h. alle Angaben über die Deklaration von Gleichungen und Beschreibung von Algorithmen hinausgehen) werden durch das Konzept der semantischen Metadaten sowie durch die Systemunterstützung bei der Recherche aufgewertet (siehe Abschnitt 4.4 „Ontologiebasierte Aufbereitung und Verwaltung von Forschungsergebnissen“ auf Seite 82).

Abbildung 10 zeigt einen detaillierten Ablauf der konzeptionellen Modellbildung.

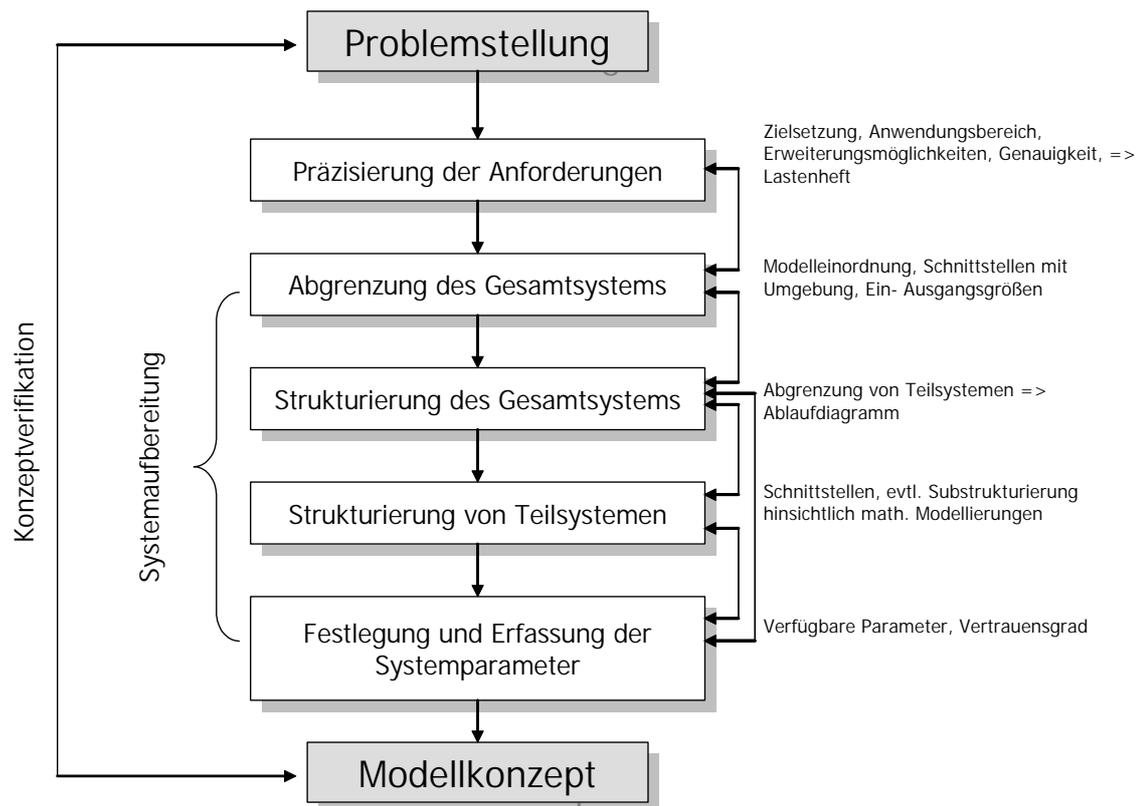


Abb. 10: Strukturierung der konzeptionellen Modellbildung nach Liebl (1992)

Unter Präzisierung der Anforderungen wird die genaue Formulierung der Problemstellung und ihrer Zielsetzung verstanden. Dabei sollen auch Erweiterungsmöglichkeiten, das Einbinden schon bestehender Modelle und die Methodik von Verifikation und Validierung berücksichtigt werden. Zur Dokumentation können Lastenhefte oder Checklisten verwendet werden.

Im nächsten Schritt wird das Gesamtsystem gegenüber der Umgebung abgegrenzt. Schnittstellen mit der Umgebung, Ein- und Ausgangsgrößen werden festgelegt.

Danach wird das Gesamtsystem in sinnvolle Teilsysteme unterteilt und die Wechselwirkungen untereinander werden beschrieben. Zur Beschreibung können Struktur- oder Blockschaltbilder erstellt werden.

Bei der Strukturierung der Teilsysteme können weitere Teilsysteme definiert werden. Die für die mathematische Modellierung erforderlichen Beschreibungen werden vorgenommen. Dabei sollten Überlegungen zur Modellvereinfachung eingebracht werden.

Schließlich werden die notwendigen Parameter festgelegt und erfasst. Es ist vor allem sicher zu stellen, dass alle notwendigen Daten vorhanden sind. Die Parameterbetrachtung wird z.T. schon bei den Strukturierungen der Systeme behandelt. Die Vielzahl an Systemgrößen muss in eine kleine Zahl von Modellgrößen überführt werden, ohne dass wesentliche verhaltensentscheidende Größen herausgelassen werden (vgl. Bossel, 1994).

Der Gesamtprozess der konzeptionellen Modellbildung ist kein linearer Vorgang, sondern stets iterativ, was in Abbildung 10 durch die rückführenden Pfeile angedeutet wird (vgl. Lugner und Bub, 1990).

3.4.5 Ansätze zur Modellbildung in den Prozesswissenschaften

Im Folgenden sollen einige grundlegende Ansätze zur Modellbildung in den Prozesswissenschaften diskutiert werden. Dabei geht es nicht um eine vollständige Beschreibung, sondern vielmehr darum, bestimmte Aspekte zu verdeutlichen, die für die späteren Betrachtungen -insbesondere zur Datenverwaltung- wichtig sind (siehe Abschnitt 3.5.1).

Wie in den vorangegangenen Abschnitten beschrieben, gehen bei der mathematischen Modellentwicklung der Aufstellung des eigentlichen Modells einige Schritte voraus, so etwa die Abstraktion des realen Systems und physikalische Modellentwicklung. In bestimmten Fällen jedoch findet sich die Abbildung physikalischer Wirkzusammenhänge auf mathematische Korrelationen im Modell nicht wieder und es liegt ein reines „Blackbox Modell“ vor (siehe Abbildung 11). Das Modell kann dementsprechend auch nur mathematisch interpretiert werden.



Abb. 11: „Blackbox Modell“

Dieser Fall kann als Extremfall der Modellentwicklung gesehen werden, bei dem die physikalische Modellierung in den Hintergrund tritt.

Abbildung 12 zeigt in verkürzter Darstellung die in den vorangegangenen Abschnitten beschriebenen Schritte zur Modellentwicklung anhand eines Beispiels. Dabei sind zwei typische Formen von Modellen als mögliches Ergebnis dargestellt. Als Beispiel wurde das Problem der Rohrströmung gewählt. Das System wird zunächst abstrahiert und vereinfacht (z.B. durch Reduzierung des dreidimensionalen Raumes auf eine zweidimensionale Struktur). Grundsätzliche Wirkzusammenhänge, wie etwa der Einfluss der Rohroberfläche auf den Impuls der Teilchen, werden betrachtet. Bei der mathematischen Abbildung werden weitere Vereinfachungen getroffen. So wird der Einfluss der Oberflächenbeschaffenheit, die zunächst eine räumlich verteilte Eigenschaft ist, auf einen Parameter reduziert.

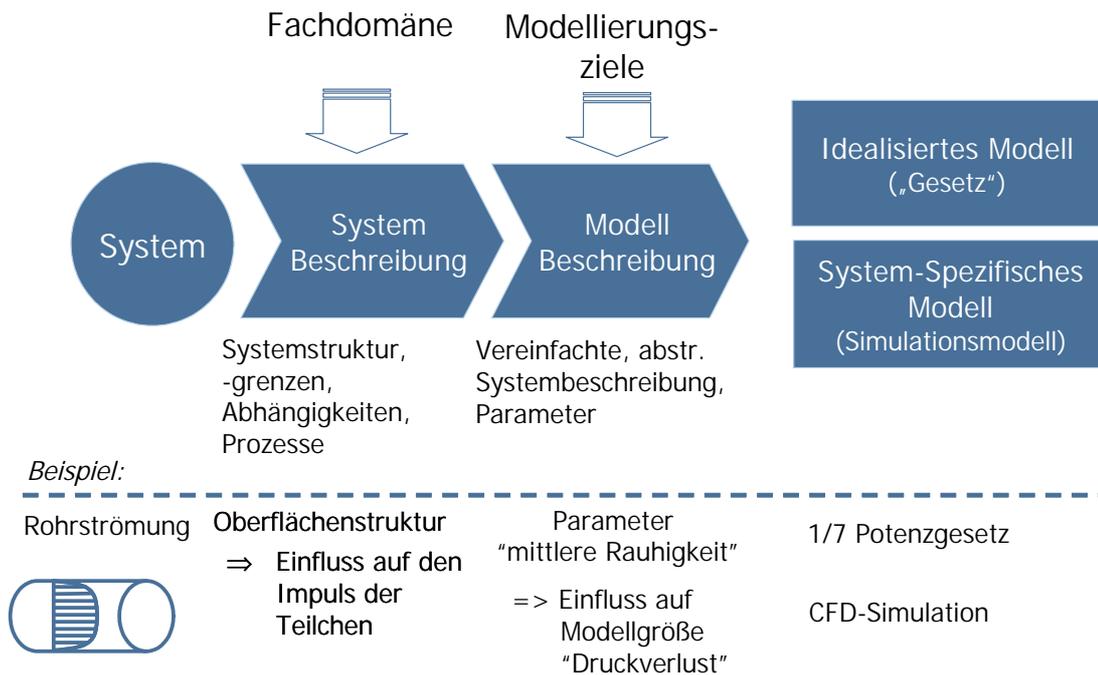


Abb. 12: Modellentwicklung in einer Fachdomäne: Beispiel Rohrströmung

Der eine hier schematisch dargestellte Fall ist das Ableiten eines Modells mit hoher Allgemeingültigkeit (vor allem hinsichtlich der Systembeschreibung und getroffener Annahmen). Wie in den vorangegangenen Abschnitten beschrieben wird die Modellentwicklung durch die zugrunde liegende Fachdomäne und die Ziele der Modellentwicklung beeinflusst.

Im Idealfall kann für ein solches Modell eine geschlossene mathematische Lösung angegeben werden. Das Modell enthält keine oder wenige phänomenologischen Parameter und fußt auf einer geschlossenen Theorie. Ein solches Modell ist sehr flexibel einsetzbar, aber in der Regel in sich nicht flexibel, d.h. die Modifikation von Annahmen ist mitunter schwierig oder nicht möglich (bzw. führt auf ein komplett neues Modell).

Der zweite Fall entspricht einem Modell, das spezifisch für einen bestimmten Einsatzzweck erstellt wird. Beim hier vorgestellten Fall der Rohrströmung kann ein solches Modell etwa einer CFD³-Simulation entsprechen. Ein Vorteil eines solchen Modells ist die mögliche Kopplung mit anderen Modellen, etwa zur Beschreibung des Stofftransportes oder von chemischen Reaktionen. Dadurch, dass keine geschlossene mathematische Lösung existiert, sind bei der Interpretation errechneter Ergebnisse nicht nur die Abbildung des physikalischen Modells und entsprechende mathematische Rahmenbedingungen (etwa die Ordnung des Differentialgleichungssystems), sondern auch eventuelle numerische Effekte zu berücksichtigen.

³ „Computational Fluid Dynamics“ – numerische Strömungssimulation

3.4.6 Mathematische Modellbildung/Simulation

Die mathematische Modellbildung ist die Umsetzung des konzeptionellen Modells in ein mathematisches Modell. Ein mathematisches Modell ist eine ideelle Abbildung bestimmter, ausgewählter Eigenschaften eines Systems unter Verwendung mathematischer Beziehungen (vgl. Blass, 1997). Die Elemente und Attribute des Systems werden im Modell durch mathematische Größen, die Aktivitäten durch mathematische Funktionen beschrieben. Die mathematischen Funktionen sind u.a. Polynome, algebraische Gleichungen, Differenzen-, Differential- und Integralgleichungen Möller (1992).

Nach Shannon (1975) ist Simulation wie folgt definiert:

Simulation is the process of designing a model of a real system and conducting experiments with this model for the purpose either of understanding the behavior of the system and its underlying causes or of evaluating various designs of an artificial system or strategies for the operation of the system.

Shannon (1975).

Darin finden sich die Prinzipien der mathematischen Modellentwicklung, auch wenn Simulationen nach dieser Definition nicht zwangsweise mit mathematischen Mitteln entwickelt werden müssen.

Eine weitere Definition findet sich bei der VDI-Gesellschaft Fördertechnik Materialfluss Logistik (2000):

Simulation ist ein Verfahren zur Nachbildung eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. Im weiteren Sinne wird unter Simulation das Vorbereiten, Durchführen und Auswerten gezielter Experimente mit einem Simulationsmodell verstanden. Mit Hilfe der Simulation kann das zeitliche Ablaufverhalten komplexer Systeme untersucht werden (Simulationsmethode).

VDI-Gesellschaft Fördertechnik Materialfluss Logistik (2000)

Eine im Zusammenhang mit Simulationen häufig angewandte Methodik ist die der komponentenorientierten Modellbildung.

Komponentenorientierte Modellbildung

Bei einem komponenten-orientierten Ansatz steht die Zerlegung des Systems in Teilsysteme im Vordergrund. Dies ermöglicht eine Reduktion der Komplexität des Gesamtsystems, in dem die Teilsysteme einzeln betrachtet werden. Darüber hinaus hat diese Vorgehensweise praktische Vor-

teile, wie etwa das Entwickeln von Modellbibliotheken (siehe Abschnitt 3.4.9) und die Modularisierung von Programmen zur rechnergestützten Modellentwicklung.

Darüber hinaus entspricht speziell bei praktischen Modellen die logische Teilung des Gesamtsystems der physikalischen Unterteilung des realen Systems. Dies ist insbesondere in der Verfahrenstechnik der Fall, wo eine typische Anlage Rohrleitungen, Pumpen, Apparate (Kolonnen, Reaktoren, ...) etc. umfasst, die oft einzeln modelliert werden.

Nach Lunze (1995) ergeben sich für die komponentenorientierte Modellbildung folgende Schritte:

1. Abgrenzung des betrachteten Systems von seiner Umgebung;
2. Zerlegung des Systems in seine Teilsysteme und Bestimmung der Kopplungen unter den Teilsystemen;
3. Aufstellung der Modellgleichungen für die Teilsysteme;
4. Aufstellung der Modellgleichungen für die Koppelbeziehungen der Teilsysteme.

Die komponentenorientierte Modellbildung ist typisch für bestimmte Fachdisziplinen wie etwa die Verfahrenstechnik und ergibt sich nach Lunze (1995) aus dem zugrunde liegenden „ontologischen System“ (s.o.).

Die im System auftretenden Größen werden unterschieden in Parameter, Variablen und Konstanten. Systemparameter sind Größen, die nicht durch Veränderungen im System selbst beeinflusst sind. Sie können konstant sein aber auch von der Zeit abhängen. Systemparameter wirken auf das System ein, werden aber vom System nicht beeinflusst oder verändert (vgl. Bossel, 1994). Hierzu können etwa geometrische Größen von Anlagen oder auch Randbedingungen wie Temperatur und Druck gehören. Systemvariablen sind im Gegensatz zu Parametern innerhalb des Systems veränderlich, beispielsweise ist die Zeit in dynamischen Systemen eine Variable. Konstanten sind absolut feste Größen wie z.B. die Erdbeschleunigung.

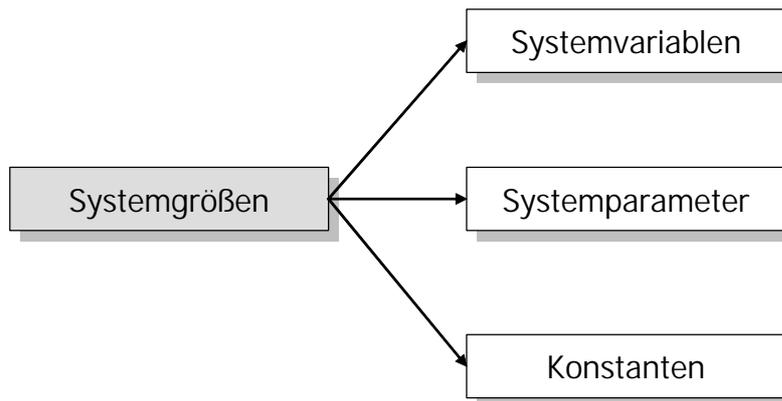


Abb. 13: Beschreibung der Systemgrößen

Merkmale oder Eigenschaften des Systems werden im Modell durch Modellparameter oder Modellvariablen dargestellt. Analog zu den Systemgrößen werden Modellparameter nicht vom Modell bzw. während der Durchführung einer Simulation verändert. Wird beispielsweise die Auslegung einer Anlage modelliert, so sind die geometrischen Größen keine Parameter, sondern veränderliche Modellvariablen.

Modellgrößen lassen sich einteilen in bekannte Eingangs- und zu ermittelnde Ausgangsgrößen. Die Eingangsgrößen können dabei unterschiedlich bestimmt werden (vgl. Abbildung 14). Eingangsgrößen können experimentell ermittelt, der Literatur entnommen, aus anderen Modellen bzw. aus einer Simulation bestimmt oder abgeschätzt werden. Bei der Bestimmung aus Experimenten lässt sich wiederum unterteilen, ob die Größe direkt aus dem Originalsystem gemessen oder aus anderen Experimenten indirekt oder direkt bestimmt wird.

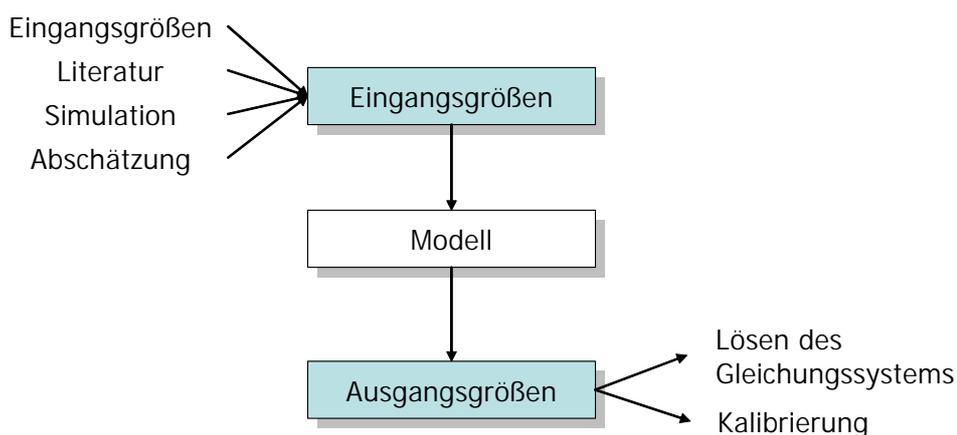


Abb. 14: Beschreibung der Modellgrößen

Die Ausgangsgrößen werden entweder über die Lösung des Gleichungssystems bestimmt oder durch eine Kalibrierung, d.h. die Anpassung der Größe bei der Validierung, ermittelt (siehe nächster Abschnitt).

Mögliche Probleme bei der Interpretation mathematischer Modelle

In Abschnitt 3.3.1 ist die Problematik der Interpretation mathematischer Modelle und einzelner Modellparameter bereits deutlich geworden. Die möglichst detailgetreue Abbildung des physikalischen Modells auf ein mathematisches Modell bzw. das Reduzieren der Anzahl vereinfachender Annahmen ist damit nicht an und für sich ein Vorteil. Dies ist vor allem dann nicht der Fall, wenn es sich schlussendlich um rein mathematische Parameter handelt, da keine Bestimmung außerhalb des Modells möglich ist und auch keine qualitativen Rückschlüsse zu Korrelationen angestellt werden können, die den prinzipiellen Wirkzusammenhang validieren können. Dies ist für die Interpretation der aus mathematischen Modellen errechneten Ergebnisse wichtig.

In Golomb (1970) werden vor diesem Hintergrund eine Reihe von Prinzipien für den kritischen Umgang mit mathematischen Modellen und deren Interpretation angegeben:

- Don't believe in the 33rd order consequences of a 1st order model.
- Don't extrapolate beyond the region of fit.
- Don't apply any model until you understand the simplifying assumptions on which it is based, and can test their ability.
- Don't believe the model is reality.
- Don't distort reality to fit the model.
- Don't limit yourself to a single model: More than one may be useful for understanding different aspects of the same phenomenon.
- Don't retain a discredited model.
- Don't fall in love with your model.
- Don't apply terminology of subject A to the problems of subject B if it is to the enrichment of neither.
- Don't expect by having named a demon you have destroyed him.

3.4.7 Verifikation und Validierung

Verifikation und Validierung dienen der Überprüfung des Modells, um Glaubwürdigkeit und Vertrauen in die Ergebnisse herzustellen (vgl. Lugner und Bub, 1990).

Unter Verifikation wird die formale Überprüfung der Umsetzung der einzelnen Arbeitsschritte von der konzeptionellen Modellbildung bis zur Simulation nach Abbildung 7 bezeichnet. Für jede einzelne Phase der Modellbildung ist eine Verifikation erforderlich, um Fehlerquellen lokalisieren zu können. In diesem Zusammenhang wird auch von einer inneren Korrektheit des Modells gesprochen. Balci (1988) bezeichnet den Vorgang der Verifizierung mit:

„Building the model right.“

Die Verifikation ist die elementare Voraussetzung für die Gültigkeit eines Modells (vgl. Liebl, 1992). Zur Überprüfung der inneren Korrektheit werden modellerzeugte Daten mit einer Referenzdatenbank verglichen, die bereits für die Modellentwicklung verwendet wurde (siehe Abbildung 15). Eine Referenzdatenbank kann beispielsweise in Form von experimentellen Daten zur Verfügung stehen, aus der das Modell abgeleitet wurde. Die Datenbasen sind daher voneinander abhängig (vgl. Lugner und Bub, 1990).

Bei der Validierung bzw. Validation wird die Gültigkeit von Modellen überprüft. Balci (1988) verwendet für die Validierung die Beschreibung:

„Building the right model.“

Das Verhalten des Originalsystems mit den Modellergebnissen wird verglichen. Dadurch wird die äußere Korrektheit des Modells überprüft. Die zu vergleichenden Datenbasen sind voneinander unabhängig, da die Referenzdatenbank bei der Modellentwicklung nicht verwendet wird. Die Validierung setzt voraus, dass das System überhaupt existiert bzw. dass experimentelle Daten vorliegen (vgl. Lugner und Bub, 1990).

Die Abweichungen zwischen Modell- und Referenzdaten lassen sich mit Mitteln der Statistik quantifizieren. So können Kriterien aufgestellt werden, die bei der Verifikation und Validierung erfüllt werden müssen.

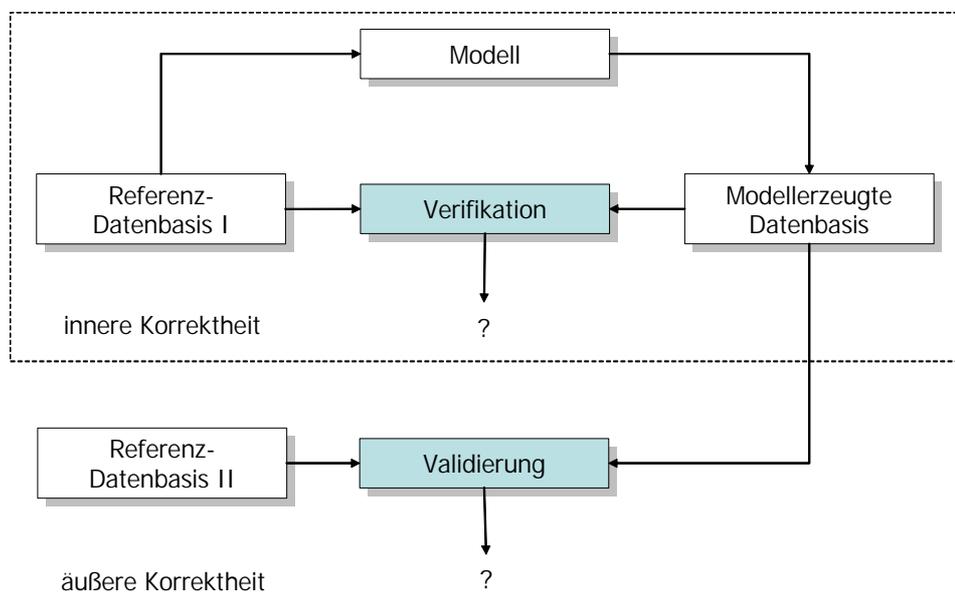


Abb. 15: Verifikation und Validierung entsprechend Lugner und Bub (1990)

In der Praxis stehen häufig nicht genügend Gleichungen zur Verfügung, um die unbekanntes Modellgrößen berechnen zu können. Durch Modellkalibrierung lassen sich Modellparameter bei der Validierung ermitteln. Kalibrierung bedeutet die Variation unbekannter Parameter, so dass Ergebnisse aus der Simulation an experimentelle Ergebnisse angepasst werden und möglichst gut übereinstimmen.

Die Begriffe Validierung und Verifikation haben in der Literatur häufig verschiedene Bedeutungen. In Möller (1992) wird als Modellverifikation die Überprüfung der Abbildungstreue zwischen mathematischem Modell und realem dynamischen System bezeichnet.

Im Rahmen dieser Arbeit wird die erfolgreiche Validierung als Validation bezeichnet, während das Scheitern Falsifikation genannt wird.

Zur Überprüfung von Modellen ist neben der Validierung und Verifikation die Sensitivitätsanalyse ein wichtiges Instrument. Dabei wird die Robustheit und Stabilität von Modellen hinsichtlich der Variation kritischer Parameter geprüft (vgl. Möller, 1992).

3.4.8 Ansätze zur Unterstützung des Modellierungsablaufs

Modellbildung ist mit einem sehr hohen Aufwand verbunden und bedarf qualifizierter Simulationsexperten mit umfassenden Kenntnissen im Bereich Modellierungstechnik, angewandter Mathematik und den zu untersuchenden Fachgebieten. Neben der Verfügbarkeit von Simulationspezialisten führt auch Zeitmangel zum Qualitätsproblem in der Modellbildung, welches sich häufig in der Dokumentation und Systematik niederschlägt. Wartung und Wiederverwendung von Modellen sind somit oft erschwert oder gar unmöglich, was zur Mehrfachbearbeitung verwandter Aufgabenstellungen führt.

Ein Modellierungsengpass kann nur durch Unterstützung hinsichtlich Entwicklung, Analyse, Implementierung, Dokumentation und Wartung von Modellen vermieden werden (vgl. Marquardt, 1995).

3.4.9 Modellbibliotheken

Die Wiederverwendung bereits entwickelter Modelle oder Modellteile nimmt eine Schlüsselrolle rationaler Modellierungskonzepte ein. Bereits fertig codierte Modelle von Teilsystemen aus einer Programmbibliothek erlauben die schnelle Erstellung von Modellen. Entscheidend für den Nutzen einer Modellbibliothek ist die Wartung und Pflege.

Die Aufnahme von Modellen in eine Modellbibliothek erfordert eine Standardisierung bzw. Strukturierung. Eine mögliche Strukturierung ist die Zerlegung von Modellen in abgeschlossene Modellbausteine, um die Komplexität zu reduzieren. Diese Modellbausteine können in anderen Modellen verwendet werden. Im industriellen Bereich finden diese blockorientierten Ansätze als kommerzielle Flowsheeting-Systeme breite Anwendung (vgl. Marquardt, 1995).

3.5 Modellentwicklung und Datenhaltung

3.5.1 Modellentwicklung und Messdatenverwaltung

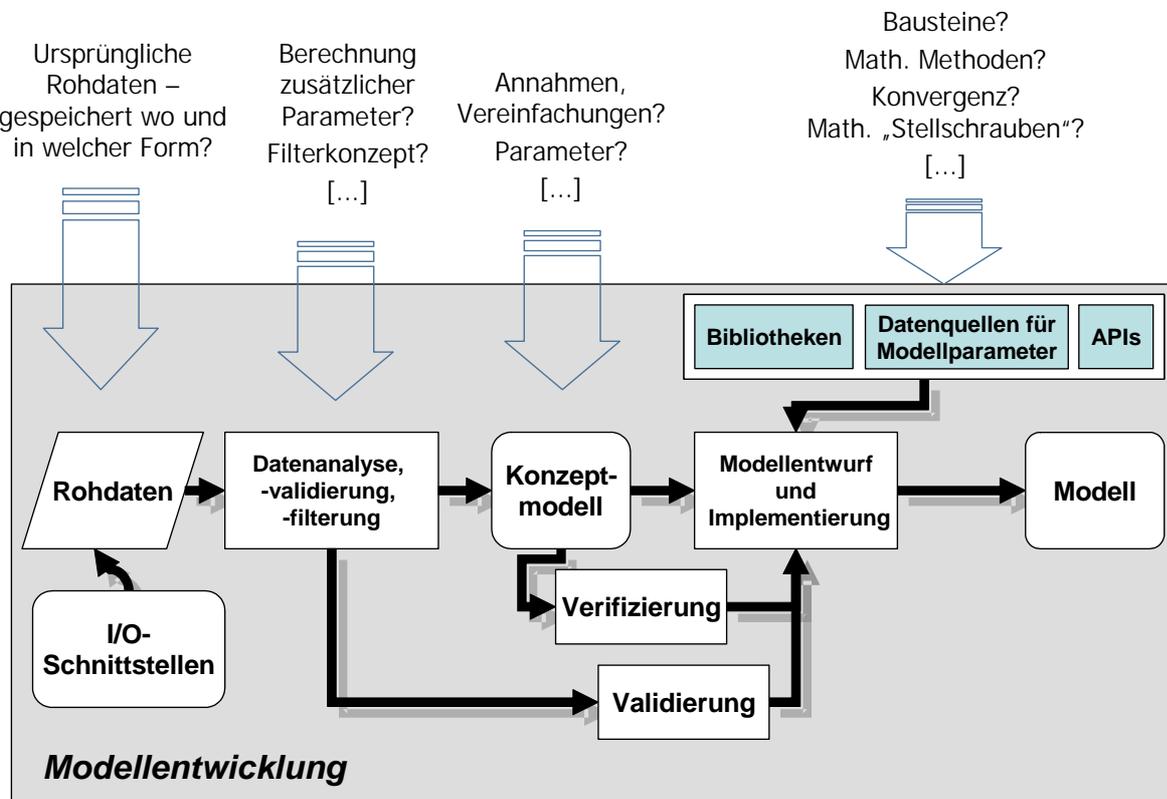


Abb. 16: Metainformationen in der Modellentwicklung und resultierende Fragestellungen

Ein zentraler Ansatzpunkt der vorliegenden Arbeit ist der Zusammenhang zwischen der Modellentwicklung in den Prozesswissenschaften und der Datenhaltung.

Abbildung 16 zeigt den Datenfluss für einen möglichen Weg der Modellentwicklung analog zu den Erörterungen in den vorangegangenen Abschnitten. Entlang des Prozesses sind Fragestellungen dargestellt, die sowohl die Modellentwicklung als auch die damit verbundene Erzeugung und Verwendung von Daten betreffen. Die Darstellung illustriert lediglich den möglichen Verlust von Informationen entlang der Prozesskette und soll nicht über den iterativen Charakter der Modellentwicklung hinwegtäuschen (siehe etwa Abbildung 9 auf Seite 42).

Dabei handelt es sich um Fragestellungen, die bei der Einschätzung von Forschungsergebnissen entstehen, etwa beim Vergleich verschiedener Ansätze oder im Rahmen von Folge-Arbeiten. Oftmals geben Publikationen über diese Fragen keinen hinreichenden Aufschluss. Publikationen werden üblicher Weise dazu genutzt, den grundlegenden Ansatz einer Arbeit zu erörtern und Aspekte wie etwa Versuchsaufbau und –durchführung, konzeptionelle Modellentwicklung, Lösungsverfahren, etc. zu beschreiben. Daten werden auszugsweise und meist in aufbereiteter Form wiedergegeben.

Für die Interpretation von Messdaten etwa sind oft Metadaten entscheidend um etwa folgende Fragestellungen beantworten zu können:

- § Wurde eine Messgröße direkt gemessen oder aus anderen Größen abgeleitet (etwa per Bilanzierung)?
- § Wie wurden die Daten gemessen (Messverfahren)?
- § Wurde eine Kalibrierung durchgeführt?
- § Wurden Daten gefiltert (Glättung, etc.)?
- § Welche genauen Kriterien wurden für charakteristische Messpunkte (etwa Stationarität) zugrunde gelegt?

Dies sind nur einige der möglichen Fragestellungen. Viele der sich dahinter verbergenden Entscheidungen (etwa bzgl. der Wahl des Messverfahrens oder der Datenaufbereitung) obliegen dem Ermessen des Wissenschafters bzw. der Wissenschaftlerin im Rahmen der entsprechenden Forschungsarbeit. Sie hängen stark vom Kontext ab, in dem die Arbeit erstellt wurde: Fachdisziplin, Stand der Forschung, Ziel der Arbeit, etc.

Messdaten werden bei verändertem Stand der Forschung unter Umständen neu beurteilt. Dabei kann es entscheidend sein, die Metadaten zu kennen. Wenngleich einige der Fragestellungen aus Publikationen hervorgehen, so ist es dennoch allein durch das Datenhaltungskonzept einzelner Institutionen festgelegt, wie Rohdaten zu interpretieren sind und wie gut Zusammenhänge zwischen Rohdaten und anderen Daten ersichtlich sind.

Laborinformationsmanagementsysteme (LIMS)

Ein Ansatz zur Datenerfassung im Rahmen experimenteller Untersuchungen sind sogenannte Laborinformationssysteme (LIMS). Sie dienen der Datenerfassung zur Dokumentation und Qualitätssicherung von Arbeitsprozessen in Laboren (vgl. Fransen und Nyrup, 1998 und McDowall, 1999).

Der Ursprung der Entwicklung von LIMS liegt in der Unterstützung und Verwaltung von Laborprozessen. Fortschritte in der instrumentellen Analytik, ganz besonders aber die Entwicklung von Analysenautomaten und die Notwendigkeit des Umgangs mit großen Datenmengen (etwa im Bereich der Chromatographie oder Spektroskopie) führten zur Entwicklung von LIMS (vgl. Schuchardt, 1998).

Zu den Aufgaben eines LIMS gehört u.a.:

- § das Erfassen von Probandaten;
- § das "Tracking" von Proben;

- § die Qualitätssicherung:
 - Unterstützen von vordefinierten Arbeitsabläufen;
 - Abzeichnen von Arbeitsschritten.

Die Möglichkeiten eines LIMS umfassen u.a.

- § das Speichern und Suchen analytischer Daten;
- § das Erstellen von Reports und analytischen Arbeitsblättern;
- § das Erstellen von Rechnungen;
- § die Barcode-Unterstützung.

Nach Stafford (1998) sind LIMS üblicher Weise nicht als Informationssystem konzipiert, sondern im Wesentlichen zur Prozesssteuerung.

Vor dem Hintergrund der oben beschriebenen Herausforderungen von LIMS sind es folgende Aspekte, die LIMS allein nicht geeignet für den Zweck der wissensbasierten Dokumentation von Forschungsergebnissen erscheinen lassen:

§ Festes Datenmodell

Das Datenmodell ist bis auf einige Anpassungsmöglichkeiten fest mit der Applikation verbunden. Darüber hinaus ist das Datenmodell auf die Anwendung und die erwähnten Aufgaben zugeschnitten und nicht auf Domänenwissen. Der Aspekt des Zusammenhangs zwischen Modellentwicklung und experimenteller Datenerfassung lässt sich nicht flexibel abbilden.

§ Datenmodell ist nicht oder nur bedingt explizit sichtbar für Anwender und nicht wiederverwendbar

Abgesehen von der mangelnden Flexibilität handelt es sich um applikationsinterne Datenmodelle, die nur begrenzt explizit für den Anwender verfügbar sind (etwa durch bestimmte Elemente der graphischen Oberfläche). Sie lassen sich nicht in einem syntaktisch und semantisch standardisierten Format serialisieren.

§ Suchmöglichkeiten stark eingeschränkt

Die Abfragemöglichkeiten für Teile des Datenmodells sind grundsätzlich vorgegeben bzw. vorstrukturiert – eng gekoppelt an die graphische Oberfläche. Datenmodelle zur Wissensrepräsentation dagegen wie etwa Ontologien in einer geeigneten Sprache erlauben (die entsprechende Infrastruktur vorausgesetzt) sehr flexible Fragestellungen.

§ Integration von externen Informationsquellen sehr eingeschränkt

Die Integration von externen Datenquellen ist bei LIMS (aufgrund des Spezialisierungsgrades) nicht so flexibel.

§ Vorgabe von Arbeitsprozessen

Ziel von LIMS ist es, bestimmte Arbeitsprozesse zu unterstützen und bis zu einem gewissen Grad -vor dem Hintergrund der Qualitätssicherung- vorzugeben.

Daten und Metadaten

Die nachhaltige und transparente Verwaltung von experimentellen Daten im wissenschaftlichen Kontext erfordert die Berücksichtigung von Metadaten. Dies betrifft Informationen auf unterschiedlichen Ebenen. Auf der Ebene der Messwerte umfasst dies u.a.:

§ die physikalischen Einheiten bei gemessenen Werten;

§ die Art des Messverfahrens (siehe auch Fragestellungen zu Beginn dieses Abschnittes);

Auf der Ebene des Kontextes, in dem Experimente durchgeführt wurden, betrifft dies u.a.:

§ Modelle, die mit Hilfe der Messdaten entwickelt oder validiert werden sollen;

§ Modelle, die bei der Ermittlung einzelner Parameter verwendet wurden;

§ Messdaten, die mit den betreffenden Datensatz in Verbindung stehen.

Kao et al. (1994) greifen das Problem der Repräsentation komplexer wissenschaftlicher Daten auf. Die Autoren führen an, dass Versuche zur Standardisierung und Etablierung von Datenformaten auf niedriger Abstraktionsebene (wie etwa das „Common Data Format“ und das „Hierarchical Data Format“) die Semantik von wissenschaftlichen Daten nicht hinreichend erfassen. Für wissenschaftliche Datenhaltungssysteme fordern die Autoren umfangreiche Analysefunktionen und Metadatenverwaltung. Dabei unterscheiden sie zwischen syntaktischen Metadaten, die das Datenschema vorgeben, sowie semantischen Metadaten, die domänen- und anwendungsspezifisch sind. Semantische Metadaten umfassen nach Kao et al. (1994) drei wesentliche Aspekte:

§ Definieren von charakteristischen Zusammenfassungen für Datensätze

Aquirierte (etwa experimentell erhobene) wissenschaftliche Datensätze enthalten typischer Weise zusätzliche Daten. Diese können entweder in Form von neuen Datenstrukturen auftreten (wie z.B. bei gefilterten Daten) oder durch Anwendung von Berechnungsfunktionen, die bei vorhandenen Datensätzen das Hinzufügen von Parametern ermöglichen (z.B. durch einfache arithmetische Funktionen wie das Aufsummieren über verschiedene Werte).

§ Behandlung von Fehlern über Fehlerfunktionen

Fehler, wie sie etwa bereits bei der Erfassung der Daten auftreten, müssen nach Meinung der Autoren explizit behandelt und über Fehlerfunktionen sowie die Beschreibung der Fehlerpropagierung berücksichtigt werden.

§ Ableitungshistorien

Ableitungshistorien beschreiben nach Kao et al. (1994) Bezüge zwischen Datensätzen und erklären bzw. rechtfertigen Analysen. Sie beantworten bestimmte Fragen, wie etwa durch wen, wann und mit welchen Mitteln Daten erhoben wurden.

Ähnliche Fragestellungen und ein dazu passendes Datenmodell werden beispielsweise von Massey et al. (1997) und Gupta und Baru (1999) beschrieben.

Im Bereich wissenschaftlicher Datenbanken existieren eine ganze Reihe von Ansätzen mit unterschiedlichem Fokus, wie etwa der Metadatenverwaltung (vgl. Buneman et al., 2004), der Herkunft von Daten (vgl. Buneman et al., 2001), dem Wissenstransfer (vgl. Kiricz und Roosendaal, 1996), dem Informations- und Erfahrungsaustausch (vgl. Kaestle et al., 1999), dem Entwickeln von Suchstrategien (vgl. Schaible, 2001) sowie der Modularisierung (vgl. Kiricz, 1998). Für bestimmte Bereiche existieren eine Reihe von Spezialsystemen zur Verwaltung experimenteller Daten und Metadaten (vgl. Lee et al., 1996 sowie Hunter und Little, 2005).

Einige Autoren greifen auf semantische Technologien bzw. Repräsentationsformate zurück, um wissenschaftliche Daten und Metadaten zu beschreiben (vgl. Soldar und D., 2002). Hunter und Little (2005) verwenden Ontologien für die Verwaltung multi-medialer Daten zur Anreicherung wissenschaftlicher Daten. Die Autoren verwenden standardisierte Metadaten und bereichern diese durch die Ontologie-basierte Beschreibung von Datensatz-übergreifenden Informationen an. Die Metadatenschemata sind z.T. domänenspezifisch und spezialisiert für den Bereich Mikroskopie sowie für den Anwendungsbereich Brennstoffzellen-Technik. Über eine übergreifende Ontologie („Top-Level Ontology“) werden die unterschiedlichen Domänenmodelle zusammengeführt. Dieser Ansatz ermöglicht es domänen-spezifische Suchanfragen auf die verwalteten Daten durchzuführen.

Die beschriebenen Ansätze bieten teilweise Lösungen für die eingangs beschriebene Problematik. Auf der konzeptionellen Ebene fehlt den Ansätzen die zentrale Verknüpfung von Modellentwicklung (basierend auf der Semantik mathematischer Modelle) sowie dem entsprechenden wissenschaftlichen Diskurs mit der Verwaltung experimenteller Daten.

Auf der Ebene von Implementierungen wird kein System beschrieben, das:

- § die skalierbare Datenhaltung mit einer flexiblen Metadatenverwaltung mit expliziter Semantik beschreibt (wenn gleich dies von Hunter und Little als Erweiterung in Aussicht gestellt wird);

- § dabei Anwender durch funktionale Metadaten und deren Interpretation (Einheitenumrechnung) unterstützt;
- § flexible, an die Domäne anpassbare Anwender-Schnittstellen bietet;
- § durch die verwendeten Modelle und die Ausprägung der oben genannten Eigenschaften besonders für die Prozesswissenschaften geeignet ist.

Dieser Aspekt wird in Abschnitt 4 wieder aufgegriffen.

3.5.2 Modellentwicklung und Dokumentenverwaltung

Dokumente wie etwa Publikationen, technische Berichte oder Präsentationen stehen oft in engem inhaltlichen Zusammenhang mit Modellen. Sie beschreiben entwickelte Modelle, die physikalische Interpretation mathematischer Zusammenhänge, die Bedeutung einzelner Parameter, etc. Sie sind die Informationsressource, die über einen vergleichsweise langen Zeitraum Forschungsergebnisse verfügbar machen. Dokumentenmanagementsysteme (vgl. Gulbins et al., 1999) erlauben die systematische Verwaltung von Dokumenten und bieten hierzu Funktionen wie etwa Versionierung.

Ein weiterer Ansatz unabhängig von der Unterstützung durch konkrete Systeme ist die Entwicklung und Standardisierung von Metadaten, wie sie etwa das Dublin Core Schema bietet (vgl. Sugimoto et al., 2002). Im Rahmen von digitalen Bibliotheken werden Metadaten systematisch eingesetzt (vgl. Alexa T. McCray, 1999).

Problematisch ist bei (Text-)Dokumenten als Informationsquelle zur Modellentwicklung, dass:

- § sie oft in proprietären Formaten vorliegen;
- § sie nur eingeschränkt recherchierbar sind,
 - da ihre logische Struktur bei Suchen nicht berücksichtigt werden kann;
 - da sie Zusammenhänge natürlichsprachlich und nicht inhaltlich beschreiben.

Die Recherchierbarkeit lässt sich prinzipiell durch Metadaten verbessern, die nicht zwangsweise im Dokument abgelegt werden müssen. Allerdings sind proprietäre Formate insbesondere für die langfristige Verwaltung der Dokumente problematisch, da sie entsprechende Werkzeuge zum Zugriff voraussetzen.

Darüber hinaus ist die Problematik ähnlich gelagert wie schon bei den experimentellen Daten. Der Zusammenhang zwischen Dokument und Modellentwicklung ist meist nicht explizit. Metadaten helfen nur dann weiter, wenn sie auf nachvollziehbare Weise Aussagen über die Dokumente und Aussagen über Modelle verknüpfen und diese Zusammenhänge auch recherchierbar machen.

3.6 Vergleich der mathematischen und der semantischen Modellentwicklung

Die Entwicklung von Ontologien erfordert sowohl fachliches Know-How in der betreffenden Domäne für die fachspezifischen Teile als auch Expertise bezüglich der eingesetzten Technologien. Modellierungswerkzeuge unterstützen den Modellierungsprozess und kapseln z.T. die Komplexität der Repräsentationsformalismen. Auf der anderen Seite reduzieren sie die Mächtigkeit der Sprache und damit die Flexibilität der Werkzeuge. Hier bestehen Parallelen zum Modellierungsprozess bei der Erstellung von Prozessmodellen. Diese sind in Tabelle 2 zusammengefasst.

Tab. 2: Gegenüberstellung von mathematischen Prozessmodellen und semantischen Modellen

	Mathematische Prozessmodelle	Semantische Modelle [Ontologien]
Formale Repräsentation	Auf der Grundlage höherer Mathematik (zum Lösen von Differential-Algebra-Systemen)	Auf der Grundlage einer Repräsentationssprache, mitunter Logik-basiert
Modell-Input	Inputparameter	Konstanten in Anfragen
„Datenerzeugende“ Modellelemente ⁴	Modellgleichungen	Axiome, Regeln
„Datenartige Modellelemente“	Modellkonstanten, die nicht im Rahmen von Rechnungen variiert werden (wie etwa vorgegebene Stoffdaten)	Fakten („Instanzen“, „Individuen“, „Attributwerte“, ...); Schema-Daten („Klassen“, „Konzepte“, „Attribute“, ...)
Laufzeitverhalten	Konvergenzverhalten und Performanz (beides abhängig vom Gleichungslöser)	Sofern Semantik eindeutig definiert, Problem entscheidbar und Implementierung fehlerfrei: Nur Performanz.
Anfrage	Belegen variabler Größen des Gleichungssystems bis der Freiheitsgrad null ist	Eine Möglichkeit: Suchanfrage mit passender Anfragesprache (Abhängig von der verwendeten Repräsentation)

⁴ „Datenerzeugend“ bedeutet in diesem Zusammenhang, dass diese Modellelemente unabdingbar sind, um Daten abzuleiten, die nicht explizit deklariert sind. Dabei handelt es sich mehr um eine anschauliche Definition.

	Mathematische Prozessmodelle	Semantische Modelle [Ontologien]
Verarbeiten des Modells	Interpretation des Systems oder Kompilieren (etwa in höhere Programmiersprache) Verarbeitung i.d.R. iterativ	I.d.R. Interpretation des Systems; z.T. Vorabverarbeitung etwa durch Materialisierung
Lösung	Variablenbelegung nach Lösung des Gleichungssystems	Bei Verwendung entsprechender Anfragesprache: Variablenbelegung
Existenz einer Lösung für eine Anfrage	Existenz einer oder mehrerer mathematischer Lösungen	Bei Sprachen auf Basis formaler Logik (in erster Linie Prädikatenlogik erster Stufe): Existenz einer oder mehrerer Modelle

Gleichungsorientierte Modellierungswerkzeuge im Bereich der Prozesssimulation erfordern vom Anwender ein relativ gutes Verständnis der mathematischen Konzepte (wie etwa dem Prinzip des Freiheitsgrades). Auf der anderen Seite haben Anwender einen direkten Einfluss auf die mathematischen Eigenschaften des zu modellierenden Systems und damit auch auf andere Eigenschaften wie z.B. das Konvergenzverhalten oder die Performanz. Die Eignung solcher Werkzeuge für eine bestimmte Disziplin ergibt sich etwa aus der Charakteristik der zur Verfügung stehenden Gleichungslöser und den in Form von Bibliotheken zur Verfügung stehenden Modellen. Fließbild-basierte Werkzeuge („Flowsheet-Simulatoren“) dagegen abstrahieren von der mathematischen Ebene und präsentieren eine stark an die Fachdomäne angepasste graphische Anwenderschnittstelle. Sie bieten eine verminderte Flexibilität zugunsten eines intuitiveren und im günstigsten Falle auch produktiveren Arbeitens seitens des fachlich geprägten Anwenders. Sie erlauben es Anwendern nicht, die volle Mächtigkeit der darunter liegenden Module (wie etwa dem Gleichungslöser) auszunutzen.

Im Bereich der semantischen Technologien ist hinsichtlich der Software-Werkzeuge eine ähnliche Situation gegeben. Ontologiesprachen wie OWL-DL oder F-Logic können textuell erfasst werden. In diesem Falle steht dem Anwender die volle Ausdrucksmächtigkeit der jeweiligen Sprache zur Verfügung. Graphische Werkzeuge, die z.B. der Anfrage- oder Axiom- bzw. Regelerstellung dienen, abstrahieren von der formalen Wissensrepräsentation und beschränken in der Regel die Ausdrucksmächtigkeit. Graphische Schnittstellen können auf der anderen Seite stärker an in einer Domäne etablierte visuelle Paradigmen angepasst werden. Dies betrifft z.B. die Visualisierung von Partonomien („Teil-von-Beziehungen“) als Bäume.

3.7 Industrielle Feldstudie

Von Foss et al. (1998) wurde eine industrielle Studie zum Thema Modellbildung durchgeführt. Diese basiert auf Interviews, die mit 16 Ingenieuren aus verschiedenen Industriebranchen geführt wurden. Jeder Teilnehmer sollte den Modellierungsablauf eines von ihm entwickelten Modells beschreiben.

Ziel der Studie war die Verbesserung des Verständnisses der Modellbildung. Es sollte ein detaillierter Ablaufplan erarbeitet werden, um ein rechnergestütztes Modellierungswerkzeug zu erstellen. Im Folgenden werden die Ergebnisse hinsichtlich Problemstellung, konzeptioneller Modellbildung und Validierung sowie Modelldokumentation vorgestellt.

3.7.1 Problemstellung

Die Angabe detaillierter Anforderungen an das zu erstellende Modell wird von den Entwicklern als wichtig angesehen. Bei vielen Modellierungsprojekten wird die Problembeschreibung jedoch nicht ausreichend durchgeführt. Informationen über funktionale Eigenschaften werden häufig erst während der Bearbeitungszeit festgelegt. Am Anfang eines Projektes sind Modellspezifikationen oft noch nicht bekannt. Sie werden erst im Laufe des Entwicklungsprozesses mit wachsendem Prozessverständnis identifiziert. Die Modellspezifikationen können sich auch während des Modellierungsprozesses verändern.

Eine zentrale Aufgabe der Problemstellung ist die Lokalisierung von Aufwandsschwerpunkten. Fehlende experimentelle Daten sollen aufgezeigt und angefordert werden.

In der Phase der Problemstellung sollen noch keine Beurteilungen abgegeben werden.

3.7.2 Konzeptionelle Modellbildung

Die konzeptionelle Modellbildung wird von den Entwicklern als ein hoch kreativer und oft intuitiver Vorgang bezeichnet. Der Grad der Formalisierung und die Dokumentation ist sehr unterschiedlich und hängt stark von der Persönlichkeit und Erfahrung des Modellierers ab.

Die Phase ist häufig geprägt durch Literatursuche und grobe Kalkulationen. Viele Interviewpartner verwenden Checklisten mit nützlichen Fragen, die als Leitfaden für die Modellierung dienen.

Die Mehrheit der interviewten Modellierer verwendet in der konzeptionellen Modellbildungsphase bereits mathematische Gleichungen. Andere bevorzugen eine klare Trennung zwischen konzeptioneller und mathematischer Modellbildung. Erfahrene Modellierer verwenden während der konzeptionellen Phase häufig keine Gleichungen. Beim konzeptionellen Modell wird auf eine mathematische Beschreibung verzichtet, wenn es in interdisziplinären Arbeitskreisen erstellt wird. Viele der beteiligten Fachleute sind vertrauter mit graphischen oder phänomenologischen Darstellungen von Modellen als beispielsweise mit partiellen Differentialgleichungen. Die Vorgehensweise ist abhängig vom Wissen bzw. von der Ausbildung der Modellierer. Mathe-

matiker beispielsweise bevorzugen Konzepte, die bereits mathematische Formulierungen enthalten.

3.7.3 Validierung

Die Validierung wird von allen Entwicklern als extrem wichtig angesehen. Häufig wird die Validierungsphase am Anfang eines Projektes unterschätzt.

Die Modellvalidierung ist oft eingeschränkt durch Zeit-, Arbeitskraft- und Geldmangel. Eine weitere Einschränkung ist das Fehlen von Daten-Material aus dem Labor oder aus der Produktion. Oft fehlen Daten zur Validierung aber auch, weil das zu untersuchende System nur als Modell existiert, z.B. wenn Voraussagen mit Hilfe von Modellen getroffen werden sollen. Der Idealfall für eine Validierung liegt vor, wenn jeder wichtige Bereich durch Messwerte aus dem Labor überprüft werden kann.

Eine gängige Vorgehensweise ist der Vergleich von Messwerten, die mit den berechneten Modellen nur annähernd vergleichbar sind. Parameter werden dann angepasst, so dass die Modellergebnisse mit den Messwerten korrelieren. Daten zur Validierung müssen daher stets interpretiert werden, was ein fundiertes Wissen der Prozesse und Instrumente bedarf.

Die Interviewpartner bemängeln, dass im Bereich der Validierung keine systematischen Vorgehensweisen existieren, um eine ausreichende Modellgüte zu garantieren.

3.7.4 Dokumentation und Modellwiederverwendung

Eine detaillierte Modelldokumentation wird als wichtig angesehen, um Modelle wieder verwenden zu können. Die Dokumentation wird häufig vernachlässigt, da Modelle unter hohem Zeitdruck erstellt werden und Vorgesetzte den Dokumentationsaufwand wenig honorieren. Die Unterschiede der Dokumentation in bezug auf Quantität und Qualität sind sehr groß. Häufig verschlechtert sich während der Modellierungsphase die Dokumentation.

Die Dokumentation des Endmodells wird als nicht ausreichend angesehen. Wichtig ist auch die Beschreibung der verschiedenen Modellversionen. Hierzu gehört neben den Vereinfachungen und Annahmen auch eine Modellbewertung und die Dokumentation kritischer Punkte, die bei der Modellierung auftreten.

Einige Interviewpartner versuchten eigene Modellbibliotheken zu erstellen. Durch den hohen Aufwand für Aufbau und Wartung wurden diese Versuche jedoch durchgehend aufgegeben.

Aus der Studie wird deutlich, dass eigene Modelle oder Modellteile vom Modellierer häufig wiederverwendet werden. Der Austausch zwischen anderen Modellierern findet dagegen nur selten statt.

3.7.5 Fazit

Der Prozess der Modellierung ist kein linearer Vorgang und ist geprägt durch zahlreiche iterative Schritte. Daher lässt sich der Modellbildungsablauf nur grob beschreiben. Eine transparente Gliederung der Ablaufschritte des kompletten Modellbildungsprozesses bis ins Detail gelingt nicht. Oft existieren versteckte Arbeitsschritte, über die sich Modellierer häufig nicht bewusst sind. Der Modellierungsprozess wird daher als ein Netzwerk interpretiert, in dem der Modellierer nicht strengen Ablaufplänen folgt, sondern sich sehr unstrukturiert bewegt.

Lediglich gut verstandene Teilbereiche lassen sich sinnvoll durch mehrere alternative Ablaufpläne beschreiben, um so beispielsweise rechnergestützte Werkzeuge zur Unterstützung der Modellierung einzusetzen.

Die Modelldokumentation ist in vielen Fällen unzureichend. Der Aufbau eigener Modellbibliotheken ist aufwendig und wird nicht realisiert. Die Wiederverwendung von Modellen findet zu selten statt.

4 Konzept

In diesem Kapitel wird das entwickelte Konzept für die Verwaltung von Forschungsergebnissen beschrieben, welches als Grundlage für die prototypische Entwicklung des SCIFORG-Systems dient.

4.1 Ziele und Anforderungen

In der Problembeschreibung wurden bereits einige Anforderungen für ein Konzept zur Verwaltung von Forschungsergebnissen erwähnt. Grundsätzlich geht es darum, den Mehrwert von Forschungsergebnissen aus den Prozesswissenschaften zu erhöhen.

Ein grundsätzliches Problem besteht darin, dass Publikationen als textuelle Wissensressource nur eingeschränkt recherchierbar sind und es bei bestimmten Aspekten vergleichsweise aufwendig ist, sich einen Überblick zu verschaffen. Dies betrifft etwa Fragestellungen wie die der Semantik von Modellparametern (siehe Abschnitt 3.3.1), die sich nur durch vollständiges Erarbeiten des Textes erschließen (sofern sie überhaupt vollständig erörtert werden). Bei einer immer größer werdenden Zahl von Publikationen und einem z.T. sehr ausgeprägten Spezialisierungsgrad wissenschaftlicher Arbeiten wird die Recherche zur wachsenden Herausforderung. Das hier vorgestellte Konzept sieht die Möglichkeit vor, wissenschaftliche Ergebnisse so durch Metadaten anzureichern, dass präzise Recherchen auf unterschiedlichen Ebenen möglich sind.

Ein weiteres Problem wurde in Abschnitt 3.5.1 erörtert. Die Verwaltung wissenschaftlicher Daten, insbesondere experimenteller Daten und daraus abgeleiteter Analyseergebnisse wird dem Problem der Nachvollziehbarkeit von Forschungsergebnissen und dem Problem der effektiven Recherche nicht gerecht, wenn Metadaten Hintergründe und Zusammenhänge nicht abbilden. Dazu gehört eine flexible, domänenspezifische Beschreibung der Daten. Ebenso gehört dazu auch der Zusammenhang mit der Modellentwicklung.

Die Modellentwicklung ist für viele Forschungsarbeiten im Bereich der Prozesswissenschaften ein zentraler Aspekt. Zum einen kann die (Weiter-)Entwicklung von Modellen das primäre Ziel von Forschungsaktivitäten sein, motiviert durch die in Tabelle 1 angegebenen Gründe. Zum anderen kommen Modelle als Werkzeug zum Einsatz, etwa im Bereich nicht zugänglicher Messgrößen (siehe auch Tabelle 1 auf Seite Seite 29). In Abschnitt 3 ist deutlich geworden, dass es sich bei der Modellentwicklung um einen kreativen, sehr domänen-spezifischen Prozess handelt.

In diesem Kapitel wird das entwickelte Konzept für die Verwaltung von Forschungsergebnissen beschrieben. Das Konzept dient als Grundlage für die prototypische Entwicklung des SCIFORG-Systems. Im Zentrum des Konzeptes stehen mathematische Modelle und deren Entwicklung. Dies schließt den wissenschaftlichen Diskurs über Modelle und mit Modellen in Zusammenhang stehende Datenquellen aus experimentellen Untersuchungen oder Simulationsergebnissen ein.

Während mathematische Modelle keine zwingende Voraussetzungen für wissenschaftliche Untersuchungen im Bereich chemo-physikalischer Prozesse sind, so sind sie jedoch oft ein wichtiger Bestandteil von Forschungsansätzen. Entsprechend den in Tabelle 1 sowie im darauf folgenden Abschnitt aufgeführten Einsatzgründen für mathematische Modelle unterstützen diese den Erkenntnisgewinn bzgl. der beschriebenen Prozesse. Dies betrifft den unmittelbaren Erkenntnisgewinn, etwa wenn Modellgestützte Messungen für nicht zugängliche Systeme oder Systemteile beschrieben werden.

Die Dokumentation von Modellen und den bei ihrer Entwicklung und Anwendung akquirierten Daten spielt für die Beurteilung von Forschungsergebnissen eine zentrale Rolle. Dies betrifft z.B. den Vergleich unterschiedlicher Ansätze für verwandte bzw. identische Probleme oder die Entwicklung weiterführender Ansätze. Bereits für die Anwendung von mathematischen Modellen etwa im Rahmen der simulationsgestützten Auslegung oder Optimierung von Prozessen ist ein gutes Verständnis des Modells und seiner Einsatzgrenzen wichtig. Dabei versteht sich das hier beschriebene Konzept nicht als Modellkatalog zur Unterstützung des Modellierungsprozesses als reine Anwendung bestehender Modellkomponenten, sondern als Grundlage für die Verwaltung und den Austausch von Forschungsergebnissen im Sinne der Nachvollziehbarkeit und Nachhaltigkeit. Damit bietet der Ansatz eine Basis für die Erfassung und Verwaltung von Forschungsergebnissen, die sich unmittelbar auf die mathematische Modellentwicklung gründen oder damit im Zusammenhang stehen.

Wie im vorangegangenen Kapitel bereits deutlich geworden ist, gibt es an dieser Stelle unterschiedliche Ansatzpunkte, wie etwa die im Forschungsprozess anfallenden Daten und Metadaten oder die Beschreibung der verwendeten bzw. entwickelten Modelle. Ein Aspekt ist hierbei der Austausch bzw. die Weitergabe von Ergebnissen und die dazu verwendeten Formate. Dies schließt etwa das Etablieren von Standardformaten für die Verwaltung und Verarbeitung von Daten ein. In den folgenden Abschnitten wird deutlich, dass die Extensible Markup Language (XML) eine plattformunabhängige und flexible Grundlage hierfür bietet. Nachteilig an diesem Ansatz ist das Fehlen einer expliziten Semantik von XML-Daten (vgl. Erdmann und Studer, 2001). Einen weitergehenden Ansatz stellt die Verwendung von Ontologien zur Beschreibung von Forschungsergebnissen und den im Forschungsprozess verwendeten und entwickelten Konzepten dar. Vereinfacht formuliert setzen Ontologien bei einem höheren Abstraktionsgrad an als XML. Erdmann und Studer (2001) beschreiben, wie XML als Datenformat für Ontologien verwendet werden kann. Ontologien sind demnach komplementär zu XML und anderen Datenmodellen zu sehen, wie im Folgenden noch deutlich wird. Sie können einerseits auf der Ebene von Daten eingesetzt werden, etwa zur strukturierten Beschreibung der Zusammenhänge unterschiedlicher Forschungsansätze. Darüber hinaus können Ontologien auf der Ebene von Metadaten verwendet werden, z.B. um Messergebnisse mit zusätzlichen (expliziten) Informationen anzureichern.

Neben der Austauschbarkeit von Ressourcen als Ergebnis von Forschungsarbeiten gibt es den Aspekt der Systemunterstützung. Anwender müssen in die Lage versetzt werden, ihre Forschungsergebnisse so aufzubereiten, dass Zusatzinformationen auf geeignete Art erfasst, reprä-

sentiert und für Rechercheprozesse zur Verfügung gestellt werden können. Entsprechende Systeme müssen die hierzu notwendigen Funktionalitäten unterstützen und dabei ein Mindestmaß an Flexibilität, Skalierbarkeit und Offenheit bieten. Dabei lassen sich detaillierte Anforderungen nicht allgemein definieren, da diese u.a. von der entsprechenden Institution oder Forschungsgemeinschaft und der Charakteristik der anfallenden Daten abhängen. Auch spielen Arbeitsprozesse und die Veröffentlichungspolitik eine Rolle, woraus sich zusätzliche Anforderungen - etwa an die Sicherheit des Systems - ableiten lassen.

Im Rahmen dieser Arbeit wurde ein Ansatz entwickelt, der sowohl den Aspekt der Weitergabe und des Austausches von Forschungsergebnissen als auch der Systemunterstützung betrachtet. Das entwickelte System unterstützt eine Reihe von konkreten Anwendungs-Szenarien auf der Grundlage eines allgemeinen Lösungsansatzes.

Das entwickelte Konzept legt eine plattformunabhängige Verwaltung von Daten und Metadaten zugrunde. Im Abschnitt 4.2.1 wird erläutert, wie mit Hilfe von XML Metadaten standardisiert und Daten strukturiert werden. Damit werden der Austausch von Daten und der werkzeugunabhängige Zugriff erleichtert. XML bietet zwar keine grundsätzlichen neuen Möglichkeiten in Bezug auf die Datenverwaltung gegenüber alternativen (proprietären) Strukturierungstechniken, hat aber große Vorteile für die Standardisierung von Datenformaten.

XML lässt sich auch für die Strukturierung und Standardisierung von Textdokumenten einsetzen. So lassen sich etwa Layout und Struktur von Dokumenten trennen. Die Struktur wiederum lässt sich standardisieren und durch nicht angezeigte (und nicht strukturelle) Metadaten anreichern. Diese können bei Suchprozessen berücksichtigt werden. XML wird durch eine große Anzahl bestehender Werkzeuge in unterschiedlichen Programmiersprachen, aber auch durch Datenbank-Systeme unterstützt. Um Autoren bei der Erstellung von XML-Dokumenten aus bestehenden Texten zu unterstützen, wurde eine umfangreiche Editierhilfe auf der Basis von Microsoft Word® erstellt, wie in Abschnitt 4.2.2 erläutert wird. Dies betrifft den bereits erwähnten Aspekt der Systemunterstützung.

Der nächste Schritt besteht darin, Daten nicht nur strukturell sondern auch semantisch zu beschreiben. Hierzu werden Ontologien verwendet, wie im Abschnitt 4.4 beschrieben wird. Diese beschreiben direkt Informations-Ressourcen wie Datensätze oder Textdokumente. Zum anderen repräsentieren Ontologien den der betreffenden Forschungsarbeit zugrunde liegenden Ansatz sowie den Bezug zu existierenden Ansätzen im Sinne eines wissenschaftlichen Diskurses – entsprechend dem SchoolOnto-Ansatz (siehe Abschnitt 4.3). Dies wird im Abschnitt 4.3.2 erörtert. Ontologien sind auch gleichzeitig die Grundlage, um zwischen verschiedenen Ressourcen Zusammenhänge darzustellen.

Schließlich wurde ein System entworfen und implementiert, das es Anwendern aus dem Bereich der Prozesswissenschaften ermöglicht, Informations-Ressourcen aus Forschungsarbeiten zu erfassen, zu verwalten und wissensbasiert anzureichern und abzufragen. Die Umsetzung und Anwendung des Systems wird in Kapitel 5 beschrieben.

Das System ist so aufgebaut, dass es nicht allein die Verwaltung ontologie-basierter Metadaten ermöglicht, sondern auch Basisfunktionalitäten für die Verwaltung wissenschaftlicher Daten bereitstellt. Dadurch bietet es einen Mehrwert für die Anwender selbst und nicht erst durch Recherche-Prozesse auf archivierte Daten.

Die Entwicklung des Systems und des zugrunde liegenden Konzeptes hat folgende Ziele:

- § flexible Metadaten
- § explizite Semantik der Daten
- § skalierbare Datenhaltung
- § funktionale Metadaten
- § konfigurierbare, flexible (domänen-spezifische) Benutzeroberfläche

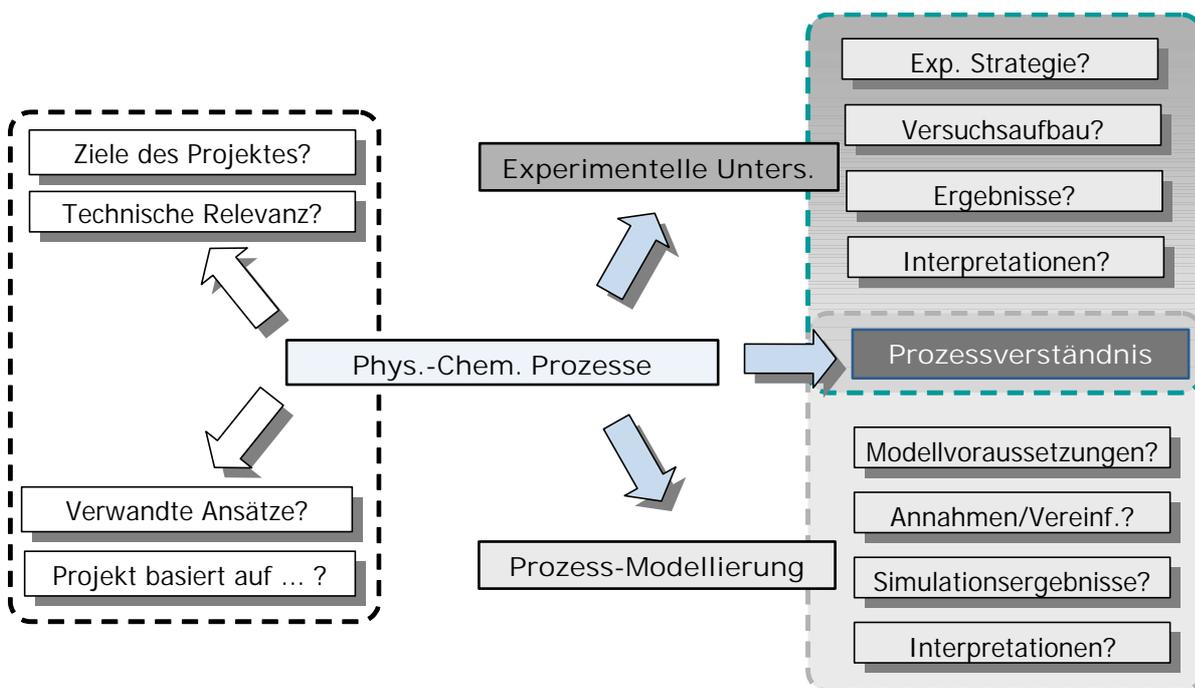


Abb. 17: Erkenntnisgewinn im Rahmen von Forschungsprojekten

Abbildung 17 illustriert das Prozessverständnis physikalisch-chemischer Prozesse als zentrale Motivation für die Entwicklung des hier beschriebenen Konzeptes und damit verbundene Fragestellungen. Diese oder ähnliche Fragestellungen treten im Zusammenhang mit der Einschätzung von Ansätzen oder Ergebnissen, aber auch als Motivation für eine gezielte Recherche auf.

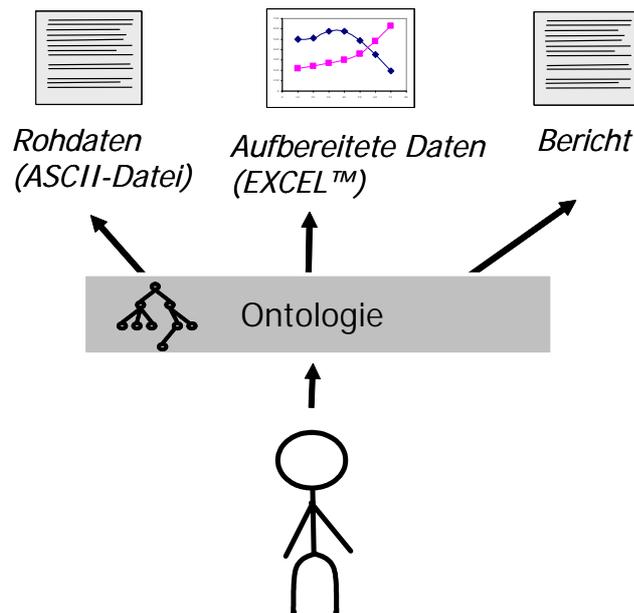


Abb. 18: Ontologie-basierter Zugriff auf heterogene Informationen

Abbildung 18 zeigt, wie Ontologien den Zugriff auf heterogen repräsentierte Informationen bündeln und kapseln können. Anwender müssen nur noch von einem Modell (und dem zugrunde liegenden Datenmodell oder Metamodell) ausgehen. Zugriffe auf Informationen über Suchanfragen erfolgen über dieses Modell. Aussagen über Ressourcen werden auf der Grundlage dieses Modells erfasst.

Metadaten bekommen eine eindeutig definierte Semantik. So werden z.B. die Größen einer Messung formal beschrieben werden. Der Modellierungsprozess selbst wird transparenter, da neben der mathematischen Repräsentation (die eindeutig interpretierbar ist) und deren Interpretation in Form von natürlich-sprachlichem Text (der individuell erschlossen werden muss) eine formale und maschinenlesbare Dokumentation von Ergebnissen dieses Prozesses verfügbar ist.

Daraus ergibt sich ein höheres Maß an Nachvollziehbarkeit und Recherchierbarkeit, wie am Ende von Abschnitt 4.4 sowie in Abschnitt 6 deutlich wird.

4.2 Plattformunabhängige Archivierung

4.2.1 XML als Metadatenformat für wissenschaftliche Daten und Dokumente

Die Extensible Markup Language (XML) ist ein vom World Wide Web Konsortium (W3C) entwickelter Standard, der auf dem SGML-Standard basiert (vgl. Lobin, 2000). Im Gegensatz zum HTML-Standard ist XML keine reine Layout-Sprache. XML ist eine Metadatensprache, in der Strukturelemente nicht wie bei HTML fest vorgegeben, sondern frei definierbar sind. XML sieht Methoden vor, Dokumente zu standardisieren, indem Vorgaben für die Art und die Anordnung der zu verwendenden Strukturelemente gemacht werden. Standards können für beliebige Ver-

wendungszwecke entwickelt werden, z.B. für bestimmte Textdokumente mit einer festen Struktur (Kapitel, Unterkapitel, Überschrift, ...) oder als spezielles Datenformat für Schnittstellen, Grafiken, etc.

Die Vorteile von XML sind zusammengefasst Folgende:

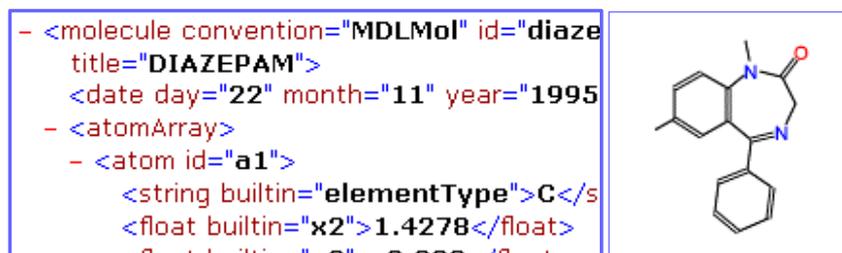
- § XML ist ein internationaler, offener Standard;
- § XML ist plattformunabhängig, d.h. nicht gebunden an bestimmte Software oder ein bestimmtes Betriebssystem;
- § Es gibt bereits eine große Anzahl von Werkzeugen (Programme und Programmbibliotheken), die XML auf verschiedene Arten verarbeiten können;
- § XML ist flexibel einsetzbar, da die Strukturelemente zur Strukturierung von Daten frei definierbar sind;
- § XML ist einfacher handhabbar als der übergeordnete SGML-Standard;
- § XML ermöglicht bei Textdokumenten die Trennung von Layout und Inhalt.

XML eignet sich aufgrund dieser Vorteile grundsätzlich als plattformunabhängiges Format zur Langzeitarchivierung. In den folgenden Abschnitten werden Erweiterungen von XML dargestellt, die besonders für den wissenschaftlich-technischen Bereich von Interesse sind.

4.2.1.1 Mathematical Markup Language und Chemical Markup Language

Auf der Basis von XML können spezielle Auszeichnungssprachen für besondere Zwecke entwickelt werden. Ein Beispiel hierzu ist die vom World Wide Web Consortium (W3C) entwickelte Mathematical Markup Language (MathML), die Strukturelemente für mathematische Ausdrücke festlegt. Software-Werkzeuge wie Maple™ und Mathcad™ bieten in den neueren Versionen bereits eine MathML-Unterstützung in Form von Export-Funktionen an. MathML bietet damit eine Plattform für das standardisierte Speichern und Austauschen mathematischer Ausdrücke, wobei auch hier Inhalt (mathematische Operationen, etc.) und Darstellung (Anordnung von Symbolen, etc.) getrennt werden können. Ein weiteres Beispiel ist die Chemical Markup Language (CML; vgl. Murray-Rust und Rzepa, 2001), mit welcher u.a. Strukturformeln und chemische Reaktionen standardisiert dargestellt werden können. CML ist im Gegensatz zu MathML kein Standard des W3C-Konsortiums.

- Chemical Markup Language (CML)



- Mathematical Markup Language (MathML)

<pre> <math> <mrow></mrow> <mrow> <msub> <mi>J</mi> </pre>	$J_V = \frac{\Delta p - \Delta \Pi}{\eta_p(I, c) \cdot (r_M + r_{Ad})}$
--	---

Abb. 19: XML-Formate für den naturwissenschaftlichen Bereich (jeweils links im Quellcode und rechts beispielhaft in graph. Darstellung mit entsprechenden Komponenten; siehe Abschnitt 5.3.2.1 „Mathematische Formeln“ auf Seite 124 und Abschnitt 5.3.2.2 „Chemische Formeln“ auf Seite 126).

4.2.1.2 XML als Datenaustausch- und Archivierungsformat

Ein Ansatz für die Datenverwaltung (Messdaten, Simulationsdaten, etc.) auf XML-Basis besteht in Form des bei der NASA in der Entwicklung befindlichen XDF-Formates. XDF standardisiert z.B. die Verwendung physikalischer Einheiten und mehrdimensionaler Datenstrukturen. Auf der Grundlage von XDF können speziellere Formate für bestimmte Fachdisziplinen entwickelt werden, die im Kern kompatibel bleiben.

Neben der Repräsentation der Daten selbst (Werte, Einheiten, Strukturen) kann XML als Plattform für zusätzliche Informationen über Eigenschaften und Herkunft der Daten (semantische Metadaten, (vgl. Kao et al., 1994) eingesetzt werden. In Mihaila et al. (2002) wird ein entsprechender Ansatz für die Qualität von Daten aus Internet-Datenbanken bezüglich definierter Kriterien, wie etwa Aktualisierungsfrequenz oder Feinauflösung zeitlich abhängiger Messwerte formuliert. Eine auf XML basierende Auszeichnungssprache zur Beschreibung modellbasierter Experimente am Computer wird von Kaestle et al. (1999) vorgestellt. Der Ansatz stellt keine allgemeine Basis für Experimente im naturwissenschaftlich-technischen Bereich her.

4.2.2 Autoren-Unterstützung: Vorlagen für Textdokumente

Im Rahmen dieser Arbeit wurde eine Textvorlage für MS Word™ erstellt, die es Autoren ermöglicht, Dokumente in standardisierter Form zu erstellen und einem gegebenen XML-Schema bzw. der Dokumententypen-Definition (DTD) entsprechend abzuspeichern. Die DTD entspricht im Wesentlichen dem Docbook-Format (vgl. Walsh und Muellner, 1999).

Die Dokumentenvorlage automatisiert eine Reihe von Arbeitsprozessen, wie sie bei wissenschaftlichen Arbeiten typischer Weise erforderlich sind. Darunter fallen das Anlegen und Verwalten von bibliographischen Daten, Tabellen, Gleichungen und Abbildungen (siehe folgende Abschnitte und Abschnitt A 2 „Verwenden der Dokumentenvorlage“ auf Seite 171).

4.2.2.1 Literaturverweise und Literaturverzeichnis

Die Dokumentenvorlage verfügt über eine Datenbankschnittstelle, über die Referenzen Datenbank-gestützt verwaltet und in das Dokument eingefügt werden können. Dabei werden die automatische Verknüpfung von Zitaten und Literaturweisen auf der Grundlage verschiedener Zitierweisen unterstützt. Die Struktur von Einträgen in der Liste der Referenzen kann flexibel eingestellt werden.

Die Datenbank wiederum kann an das SCIFORG-System angebunden werden (siehe Abschnitt 5.3.1 „Skalierbare objektrelationale Speicherschicht“ auf Seite 113), so dass verwendete Referenzen auch unabhängig vom Dokument verfügbar und mit anderen Informationen verknüpfbar sind.

Literaturverweise werden, wie sämtliche strukturellen Elemente, mit unterschiedlichen Formatvorlagen formatiert. So ist z.B. „Autor“ ein eigenes Zeichenformat, was für jeden Eintrag in der Liste der Referenzen entsprechend verwendet wird. Bei der Umwandlung in XML werden die Formatvorlagen abgebildet auf Strukturelemente des XML-Dokumentes.

4.2.2.2 Anlegen von Tabellen, Abbildungen und Gleichungen

Das Anlegen von Tabellen, Abbildungen und Gleichungen ist bzgl. der Nummerierung und Formatierung von Beschriftungselementen vollständig automatisiert. Auch hier werden für Beschriftung und Absätze vordefinierte Formate verwendet, die bei der Umwandlung auf XML-Elemente abgebildet werden. Die Schemata für die Nummerierungen können zentral eingestellt werden. Darüber hinaus ist eine Reihe von Standardaufgaben automatisiert, wie etwa das Einfügen einer Tabelle im Querformat.

4.3 Der SchoolOnto Ansatz

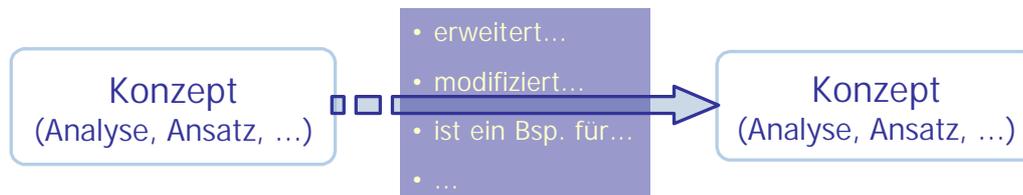


Abb. 20: SchoolOnto-Ansatz nach Buckingham et al. (2000)

Das SchoolOnto-System wurde von Buckingham et al. (2000) entwickelt. Es handelt sich um eine digitale Bibliothek, basierend auf dem Prinzip des „ontology-based document enrichment“ (Ontologie-basierte Anreicherung von Dokumenten). Die zugrunde liegende Ontologie repräsentiert Grundzüge des wissenschaftlichen Diskurses und erlaubt es, wohldefinierte Bezüge zwischen wissenschaftlichen Dokumenten explizit und maschinenlesbar herzustellen. Der Ansatz fußt auf der Beobachtung, dass sich aus dem wissenschaftlichen Diskurs typische Beziehungen zwischen Veröffentlichungen ableiten lassen, für die sich ein übergeordnetes, domänenunabhängiges Modell entwickeln lässt. Dieses wird bestimmt durch typische Kategorien von Fragestellungen, die das Interesse von Wissenschaftlern bestimmen (etwa im Rahmen von Recherchen):

- § Entwicklungslinien von wissenschaftlichen Ansätzen (wo hat eine Idee ihren Ursprung);
- § Der Einfluss von bestimmten Ansätzen (wo wurde Ansatz x aufgegriffen und umgesetzt oder erweitert);
- § Perspektiven oder Sichtweisen (welche „Schulen“ gibt es, die bestimmte Problemlösungen entwickelt haben);
- § Widersprüche/Inkonsistenz (gibt es Arbeiten mit widersprüchlichen Ergebnissen, die auf der selben Theorie basieren);
- § Konvergenz (gibt es verschiedene Gruppen oder Strömungen, die sich auf die gleiche Grundlage wie z.B. die gleiche Datenbasis stützen).

Digitale Bibliotheken bieten gegenwärtig nur sehr begrenzt und auch nur indirekt Möglichkeiten, Recherchen für diese Art von Fragestellung zu unterstützen.

Das Modell bzw. die Ontologie, die SchoolOnto zugrunde liegt, ist verhältnismäßig klein. Das vereinfacht die Kategorisierung von Veröffentlichungen anhand dieses Modells. Darüber hinaus sind die Bezüge (zwischen Veröffentlichungen) so generisch, dass sie domänenübergreifend funktionieren und „zeitlos“ sind. D.h. es lässt sich aufgrund seiner Allgemeingültigkeit auf gegenwärtige, lang zurück liegende und mit sehr großer Wahrscheinlichkeit auch zukünftige Arbeiten anwenden.

Wesentliche Konzepte des Modells sind Aussagen bzw. Behauptungen („Claims“), die sich auf Dokumente oder formale Modelle als Grundlage stützen. „Claims“ wiederum beziehen sich auf bestimmte Grundkonzepte wissenschaftlicher Arbeiten, wie etwa Theorien/Modelle, Analysen und Daten. Diese Grundkonzepte können über definierte Bezüge mit anderen Grundkonzepten oder „Claims“ explizit assoziiert werden, wie etwa „Aussage x bestätigt Aussage y“ oder „Aussage a ist widersprüchlich zu Aussage b“. Damit unternehmen die Autoren nicht den Versuch, ein domänenübergreifendes Modell für die vollständige Formalisierung wissenschaftlicher Ergebnisse zu entwickeln. Vielmehr werden Aussagen über diese Ergebnisse formalisiert, die kein detailliertes Domänenmodell erfordern. Das Modell ist jedoch so offen, dass eine Erweiterung im Sinne einer Spezialisierung für bestimmte Fachdomänen möglich ist.

Der SchoolOnto-Ansatz basiert auf folgenden zwei wesentlichen Prinzipien:

§ „minimal ontological commitment“ (vgl. Gruber, 1995)

Das gemeinsam verwendete Modell einer Domäne (hier der wissenschaftliche Diskurs) sollte nur so viel beschreiben, wie es unbedingt erforderlich ist, um innerhalb der betroffenen Gemeinschaft einen Wissensaustausch zu ermöglichen. Gruber unterscheidet an dieser Stelle zwischen Wissensbasis und Ontologie. Während Wissensbasen hinreichend sind, um bestimmte Problemlösungen zu ermöglichen oder Fragen zu beantworten, dienen Ontologien nach Gruber lediglich der konsistenten Verwendung eines gemeinsamen Vokabulars. Nach dieser Definition können Ontologien also zu Wissensbasen erweitert werden.

§ Inkrementelle Formalisierung (vgl. Shipman und McCall, 1999)

Das Erstellen von Ontologien als Wissensmodelle mit formaler Semantik ist relativ „teuer“ bzgl. des zu investierenden Aufwandes. Aus diesem Grund sollte die Formalisierung gezielt zunächst in den Bereichen erfolgen, wo der größte Nutzen zu erwarten ist. Dabei ist es von Vorteil, wenn die angestrebte Lösung bereits für einen geringen Anteil von semantischen Metadaten (im Verhältnis zu den eigentlichen Quellen) sinnvoll genutzt werden kann. Dies kann durch eine inkrementelle Formalisierung erreicht werden. Dabei werden semantische Metadaten schrittweise aufgebaut und erweitert.

Der SchoolOnto-Ansatz basiert auf einer relativ kleinen Ontologie. Dadurch kann ein „minimal commitment“ und damit eine Akzeptanz durch eine größere Anzahl von Nutzern bzw. Anwendern einfacher erreicht werden. Auch ist es einfacher, wie schon von Buckingham et al. (2000) beschrieben, einen Mehrwert durch zusätzliche semantische Dienste zu erzielen. Letztere können umgesetzt werden, indem Bezüge systematisch ausgewertet werden und damit z.B. Aussagen darüber möglich sind, wie oft eine bestimmte Theorie verwendet oder modifiziert wurde.

Ein grundlegendes Element zum Beschreiben von Aussagen („Claims“) sind Konzepte. SchoolOnto sieht folgende Arten von Konzepten vor:

- § Analysis („Analyse“)
- § Approach („Ansatz“)
- § Data („Daten“)
- § Evidence („Beleg/Beweis“)
- § Language („Sprache“)
- § Methodology („Methodologie“)
- § Opinion („Meinung“)
- § Phenomenon („Phänomen“)
- § Problem
- § Model („Modell“)
- § Definition
- § Theory („Theorie“)
- § Solution („Lösung“)
- § Hypothesis („Hypothese“)
- § Assumption („Annahme“)

Zwischen Konzepten können Bezüge hergestellt werden, wie z.B. „proves“ (beweist). Bezüge haben weitere Eigenschaften. Die Polarität („polarity“) gibt an, ob es sich grundsätzlich um einen positiven oder einen negativen Bezug handelt. Ein positiver Bezug wäre etwa das genannte Beispiel „proves“. Ein negativer Bezug stellt eine Art von Widerspruch dar, wie z.B. „isInconsistentWith“ („ist nicht konsistent zu“).

Weitere Eigenschaften von Bezügen betreffen u.a. deren Klassifizierung in Kategorien. Diese orientieren sich an der rhetorischen Bedeutung der Bezüge. Darüber hinaus sind Bezüge gewichtet, d.h. „ist ein Anhaltspunkt für“ ist ein schwächerer Bezug als „beweist“.

§ General („Allgemein“)

- isAbout („ist im Bereich“)
- uses-applies-isEnabledBy („verwendet“ / „wird ermöglicht durch“)
- improvesOn („verbessert“)
- impairs („beeinträchtigt“)
- otherLink („sonstiger Bezug“)

§ Supports/Challenges („Unterstützt/Ficht an“)

- proves („beweist“)
- refutes („widerlegt“)
- isEvidenceFor („ist ein Anhaltspunkt für“)
- isEvidenceAgainst („ist ein Anhaltspunkt gegen“)
- agreesWith („stimmt überein mit“)
- disagreesWith („stimmt nicht überein mit“)
- isConsistentWith (ist konsistent mit“)
- isInconsistentWith („ist nicht konsistent mit“)

§ Problem-related („Problem-orientiert“)

- addresses („behandelt, befasst sich mit“)
- solves („löst“ [z.b. ein Problem])

§ Taxonomic („Taxonomisch“)

- partOf („Teil von“)
- notPartOf („nicht Teil von“)
- exampleOf („Beispiel für“)
- notExampleOf („kein Beispiel für“)
- subclassOf („Unterklasse von“)
- notSubclassOf („keine Unterklasse von“)

§ Causality („Kausalität“)

- causes („verursacht“)
- isCapableOfCausing („kann verursachen“)
- isPrerequisiteFor („ist Voraussetzung für“)
- isUnlikelyToAffect („hat wahrscheinlich keinen Einfluss auf“)
- prevents („verhindert“)

§ Similarity („Ähnlichkeit“)

- isIdenticalTo („identisch mit“)
- isSimilarTo („ähnlich zu“)
- isDifferentTo („anders als“)
- isTheOppositeOf („ist das Gegenteil von“)
- sharesIssuesWith („hat inhaltliche Überlappung mit“)
- hasNothingToDoWith („hat nichts zu tun mit“)
- isAnalogousTo („ist analog zu“)
- isNotAnalogousTo („ist nicht analog zu“)

4.3.1 Vorteile und Nutzen des SchoolOnto-Ansatzes

Der SchoolOnto-Ansatz hat verschiedene Vorteile. So ist der Grad der Formalisierung von Domänenwissen grundsätzlich offen. Bezüge in der Form „Theorie A steht im Widerspruch zu

Theorie B“ sind sehr generisch und lassen sich domänenübergreifend anwenden. Inwieweit diese generische Ebene zugunsten stärker domänen-spezifischer Ansätze verlassen wird, ist nicht festgelegt. Es ist jedoch offensichtlich, dass es umso schwieriger wird dem Prinzip des „minimal ontological commitment“ gerecht zu werden, je mehr Domänenwissen einbezogen wird. Der SchoolOnto-Ansatz bietet die Flexibilität, den Grad der Formalisierung hinsichtlich des Domänenwissens anzupassen.

Ein weiterer Vorteil liegt darin, dass SchoolOnto relative Aussagen ermöglicht. D.h. anstatt eine Arbeit absolut zu positionieren (etwa über Attribute wie „rigoros“ oder „detailliert“ für Ansätze), werden Aussagen relativ zu bestehenden Arbeiten getroffen. Dadurch wird sowohl dem Problem der Zeitabhängigkeit als auch dem der Perspektivenabhängigkeit entgegen gewirkt. Zeitabhängig sind Aussagen über wissenschaftliche Ansätze, was offensichtlich ist – u.a. aufgrund des sich ändernden Maßstabes in Folge der wissenschaftlichen Entwicklung. Das Problem der Perspektivenabhängigkeit betrifft die Tatsache, dass Aussagen über wissenschaftliche Ansätze subjektiv und geprägt durch eine bestimmte Schule oder fachliche Richtung sein können. Durch relative Aussagen wird diese Problematik zwar nicht grundsätzlich beseitigt, aber zumindest abgemindert, da sie geeignet sind, das Selbstverständnis der entsprechenden Arbeit zu repräsentieren. Auf der Grundlage eines solchen Ansatzes ist es dann unter bestimmten Voraussetzungen auch möglich, Widersprüchlichkeiten abzuleiten. Zirkelbezüge in der Art „Ansatz [...] ist eine Erweiterung von Ansatz [...]“ sind beispielsweise problematisch.

Gerade bei der Verwendung relativer Aussagen ermöglicht der SchoolOnto-Ansatz das Navigieren zwischen Dokumenten auf Grundlage der Metadaten. So können kontext-sensitive Anfragen in der Form „finde Dokumente, die eine Erweiterung des beschriebenen Ansatzes dokumentieren“ unterstützt werden. Diese Art der Dokumenten-Suche kann zusätzlich zu herkömmlichen Suchverfahren über Volltext-Indizes oder Thesauri eingesetzt werden.

SchoolOnto ermöglicht sowohl „harte“ Aussagen (z.B. „Daten [...] widerlegen Theorie [...]“) als auch unsichere oder „weiche“ Aussagen (z.B. „Phänomen [...] hat wahrscheinlich keinen Einfluss auf [...]). Durch die unterschiedliche Art und den generischen Charakter der Aussagen wird es Anwendern der Ontologie einfacher gemacht, diese zur Charakterisierung der eigenen Arbeit zu verwenden bzw. adäquate Bezüge zu finden.

Die Polarität und die Gewichtung machen es möglich, sich bei der Suche nach Informationen über ein Konzept nicht auf einen konkreten Bezug festzulegen, aber die Suche dennoch einzuschränken (z.B. auf positive Bezüge).

4.3.2 Übertragung des SchoolOnto-Ansatzes auf den Bereich mathematischer Modelle

Aufgrund seines allgemeingültigen Ansatzes lässt sich der SchoolOnto-Ansatz auf den Bereich der mathematischen Modelle übertragen. Zunächst kann der generische Ansatz für Bezüge zwischen Theorien, Experiment etc. direkt übernommen werden. Die generischen Bezüge lassen sich dann durch spezifischere Konzepte erweitern. Abbildung 21 zeigt schematisch, wie dabei unterschiedliche Ontologien kombiniert werden.

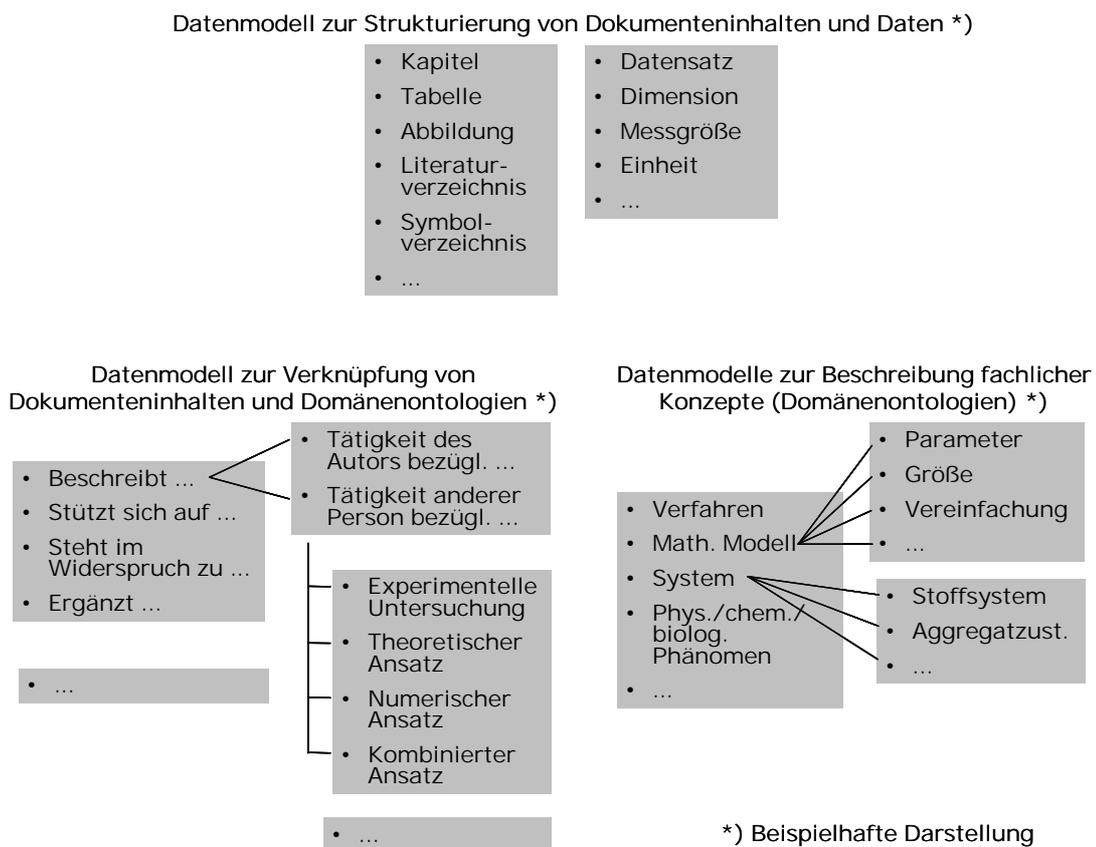


Abb. 21: SchoolOnto-Ansatz für die Modellentwicklung

Für mathematische Modelle lassen sich eine Reihe von grundsätzlichen, generischen Bezügen angeben. Ein nahe liegendes Beispiel ist die partonomische Beziehung zwischen Modellen und den sie konstituierenden Teilmodellen. Sie lässt sich z.B. durch ein „verwendet Teilmodell“-Attribut formal repräsentieren. Einige dieser Bezüge sind im Folgenden beschrieben. Es handelt sich ganz im Sinne des SchoolOnto-Ansatzes um eher generische Bezüge. Allerdings erfordert der Transfer in der Form, wie er hier beschrieben ist, teilweise domänen-spezifischere Informationen und damit Kompromisse bzgl. des „ontological commitment“.

4.3.2.1 Bezüge zwischen Modellen

Die Verwendung von Teilmodellen

Wie im bereits angeführten Beispiel werden Modelle oft im Sinne eines modularen Ansatzes kombiniert. Dies ist z.B. bei der Verwendung von Modellbibliotheken der Fall (siehe Abschnitt 3.4.9 „Modellbibliotheken“ auf Seite 51). Bei dieser Art der Wiederverwendung werden nicht nur die mathematischen Konstrukte, sondern auch die zugrunde liegenden Annahmen, Theorien, etc. integriert bzw. übernommen. Dementsprechend wichtig ist diese Art von Bezug für die Nachvollziehbarkeit von Modellierungsansätzen. Für die kontext-basierte Recherche im Bereich mathematischer Modelle kann dieser Bezug dabei in beide Richtungen nützlich sein: einerseits die Angabe von (Teil-)Modellen, die von einem bestimmten Modell verwendet werden und andererseits die Angabe von Modellen, die ihrerseits ein bestimmtes Modell als Teilmodell integrieren. Letztere Variante erlaubt es beispielsweise, Einsatzmöglichkeiten bzw. die Wiederverwendbarkeit von Modellen nachzuvollziehen.

Wie bereits beschrieben ist eine Möglichkeit, die Modell-Teilmodell Beziehung formal zu beschreiben, eine „verwendet Teilmodell“ oder „ist Teilmodell von“ Eigenschaft für eine Klasse „Modell“. Dieser Ansatz bedingt, dass Teilmodelle bei der Wissensakquisition explizit angegeben werden. Alternativ könnte versucht werden, eine Reihe von Eigenschaften zu definieren, aus der sich die Verwendung von Teilmodellen ergibt. Von der Schwierigkeit, solche Eigenschaften zu finden abgesehen, reflektiert die erste Variante die bewusste Auswahl bestimmter Modelle zur Wiederverwendung durch Modellierende.

Die formale Repräsentation von Teilmodellen in Verbindung mit weiteren Informationen ermöglicht neben der Recherchierbarkeit auch Konsistenzprüfungen. So können beispielsweise widersprüchliche Annahmen und Bedingungen für verschiedene verwendete Modelle ermittelt werden. Dies ist möglich, wenn diese Angaben so formal beschrieben sind, dass Widersprüche ebenfalls formal beschreibbar werden. Für Anwender entsteht hier neben der verbesserten Recherchierbarkeit ein weiterer potentieller Mehrwert. Grundsätzlich wäre es denkbar, eine solche Konsistenzprüfung bereits als unterstützende Funktion für den Modellierungsprozess zur Verfügung zu stellen. Im Rahmen dieser Arbeit wird allerdings nicht der Aufbau von Modellierungsumgebungen oder Unterstützungsfunktionen für dieselben, sondern die Dokumentation und Nachvollziehbarkeit von Modellentwicklungen behandelt. Indirekt besteht damit auch eine Nutzbarkeit im Rahmen des Modellierungsprozesses. Der Ansatz dieser Arbeit zielt also nicht allein auf die reine Anwendung, sondern mehr auf die (konzeptionelle) Weiterentwicklung von Modellen ab.

Konkurrierende/alternative Modelle für das gleiche Problem bzw. das gleiche zu modellierende System

Da die Eignung von Modellansätzen bzw. Modellen für eine bestimmte Problemstellung von vielen Parametern des Modellierungsziels abhängt, sind diese Modelle nicht nach absoluten Kriterien vergleichbar. Randbedingungen für die Verwendung eines Modells sind z.B. der Aufwand zur Ermittlung von Modellparametern. Dieser kann dazu führen, dass pragmatischeren Lösungen mit mehr empirischen Modellanteilen der Vorzug gegeben wird. Eine weitere Randbedingung ist

die erforderliche Rechenleistung zur Berechnung von Lösungen. So können kurze Rechenzeiten entscheidend sein, weil viele Berechnungen notwendig sind (etwa im Rahmen von Optimierungen) oder weil kurze Antwortzeiten gefordert sind (z.B. für Beobachtermodelle). Dementsprechend werden verschiedene alternative Lösungen entwickelt.

Um diesen Zusammenhang formal zu erfassen, könnte eine explizite Referenz auf die Alternativen verwendet werden („alternativer Ansatz“ oder „konkurrierender Ansatz“). Nachteilig hierbei ist, dass die Angabe dieser Referenz problematisch ist in Bezug auf Vollständigkeit und Dynamik. Obwohl im wissenschaftlichen Umfeld eine Analyse des Standes der Forschung üblich ist, kann sich das Auffinden sämtlicher vergleichbarer Modellansätze extrem aufwendig gestalten, zumal es sich nicht um Alternativen im engeren Sinne (für den angestrebten Einsatzzweck) handelt. Darüber hinaus muss die Angabe alternativer Ansätze stets aktuell gehalten werden. Es wäre zwar denkbar, dass diese Referenz symmetrisch modelliert wird, so dass es genügt, für neu erfasste Modelle den Verweis auf alternative/konkurrierende Ansätze zu erfassen. Für die bereits erfassten Modelle würde dieser Bezug umgekehrt automatisch aktualisiert. Dieser Ansatz würde aber eine „autonome“ Beschreibung einzelner Modelle einschränken und immer noch aufwendige Recherchen bedingen.

Da sich die Vergleichbarkeit von Modellen aus der Abstraktion des modellierten Systems ergibt, liegt es nahe, dieses und weitere Modelleigenschaften heran zu ziehen, um automatisiert auf alternative Modelle zu schließen. Dann ergibt sich der Bezug zu weiteren Modellen allein aus Angaben, die für das betreffende Modell gemacht wurden. Der Nachteil an dieser Stelle ist, dass diese Variante nicht mehr vollständig im Einklang mit der Idee des SchoolOnto-Ansatzes ist, das „ontological commitment“ gering zu halten und möglichst generische Bezüge zu verwenden. Je detaillierter die formale Beschreibung von Modellen ist, desto schwieriger wird es, einen Konsens über die entsprechende Ontologie herzustellen. Darüber hinaus steigt die Komplexität der Anwendung des Systems (in diesem Falle des SCIFORG-Systems) mit der Komplexität der zugrunde liegenden Ontologien. Dies wirkt sich auf die Einarbeitungszeiten, die Akzeptanz, etc. aus. Für Anwendungen die unmittelbar der Lösung spezieller und komplexer Probleme dienen sind umfangreiche und komplexe Ontologien angemessen bzw. unabdingbar. Diese Anwendungen erfordern die Akzeptanz von Ontologien als Wissensmodellen für einen vergleichsweise kleinen Anwenderkreis.

Der SCIFORG-Ansatz zielt auf einen breiten Anwenderkreis. Regeln oder Axiome, die aus unterschiedlichen Eigenschaften einer Modellbeschreibung eine Kategorisierung ermöglichen würden die Komplexität des Ansatzes signifikant erhöhen. Die Vergleichbarkeit von Modellen ist allgemein sehr schwer zu erfassen. Sie kann im Einzelfall trotz grundsätzlicher Abweichungen z.B. in der Geometrie zugrunde liegender Systeme gegeben sein, da die verschiedenen Ansätze auf die jeweils andere Geometrien übertragbar sind. Umgekehrt können abweichende Geometrien die Übertragbarkeit verhindern, etwa wenn die Strömungsverhältnisse grundlegend differieren (etwa durch den Abriss an Kanten) und sich dies auf das Systemverhalten auswirkt. Dementsprechend sind Ansätze für bestimmte Anwendungs-spezifische Lösungen denkbar. Eine allgemeine Lösung erscheint nicht möglich. Darüber hinaus ist die Einschätzung von Modellen kon-

textabhängig bzw. zielabhängig (und damit stark domänenabhängig). Hier bietet der SchoolOnto-Ansatz deutliche Vorteile.

Erweiternde oder verfeinernde Modellierungsansätze

Das Erweitern eines vorhandenen Modellansatzes ist eine weitere Form der Wiederverwendung. Erweiterungen können auf unterschiedliche Art vorgenommen werden. Eine Möglichkeit besteht darin, einen Modellierungsansatz zu übernehmen, aber eine oder mehrere der getroffenen vereinfachenden Annahmen fallen zu lassen, wie z.B. die Erweiterung auf den dynamischen Fall durch Wegfallen der Annahme eines stationären Systems. Eine weitere Möglichkeit ist die Berücksichtigung von zusätzlichen Phänomenen, wie beispielsweise chemischen Reaktionen. Letzterer Fall ist nicht unbedingt abzugrenzen von der Verwendung von Teilmodellen. Das Modellieren einer Rohrströmung mit Reaktion etwa lässt sich unter Umständen als Kombination von Modellen zur Strömungsmechanik und zu chemischen Reaktionen auffassen. Ob dies möglich ist, hängt von der Auffassung des Modularitätsbegriffes für Modelle ab. Eine generische Definition würde auch das konzeptionelle Kombinieren von Modellen als modularen Ansatz einschließen. Alternativ kann dieser Fall als Erweiterung eines reinen Strömungsmodells dargestellt werden.

Ein besonders hervor zu hebender Aspekt für das Verständnis von Modell-Entwicklungen ist der erkenntnistheoretische Hintergrund von Modell-Erweiterungen (siehe auch Abschnitt 3.3.1 „Semantik von Modellen“ auf Seite 33). Dieser kann grundsätzlicher Art sein oder sehr anwendungsspezifisch ausfallen. Ein sehr grundlegender Erkenntnisgewinn ist im (seltenen) Fall der Erweiterung theoretischer Grundlagen gegeben. Davon abgesehen sind verschiedene Abstufungen möglich. Dies wird am Beispiel des „Lebenszyklus“ von vereinfachenden Annahmen deutlich. So kann eine Annahme aus pragmatischen Erwägungen in Verbindung mit Erfahrungswerten oder aus grundsätzlicheren Überlegungen heraus erfolgen. Eine pragmatische Erwägung wäre z.B. die Annahme eines konstanten Diffusionskoeffizienten für ein Stofftransport-Problem. Stellt sich heraus, dass diese Annahme im Widerspruch zu Beobachtungen steht, so handelt es sich oft weniger um einen grundsätzlichen Erkenntnisgewinn, sondern um die Identifikation einer unzulässigen Näherung. Dabei kann es sein, dass die Annahme unter bestimmten Voraussetzungen zulässig ist und eine Erweiterung auf einen veränderlichen Diffusionskoeffizienten nur für bestimmte Randbedingungen erforderlich ist. Eine andere Situation liegt vor, wenn grundsätzlichere Erwägungen hinsichtlich der möglichen Systemzustände heraus erfolgen, wie z.B. die Annahme von maximal monomolekularen Bedeckungen bei Adsorptionsvorgängen. Die formale Beschreibung von Erweiterungen bietet grundsätzlich die Möglichkeit, den Erkenntnisgewinn explizit und dadurch schneller bzw. einfacher nachvollziehbar zu machen. Dabei kann allerdings nicht davon ausgegangen werden, dass sich Modellerweiterungen in allen Aspekten formal beschreiben lassen.

Um den Zusammenhang zwischen Modellen und ihren Erweiterungen formal auszudrücken, wäre grundsätzlich ein einfaches Attribut denkbar (in der Art „erweitert Modell [...]“). Darin ließe sich allerdings nicht die Art der Erweiterung ausdrücken. Ein Weg diesen einfachen und für Recherchen bzw. Anfragen leicht zu verwendenden Bezug bereit zu stellen und dennoch komplexere Aussagen über Modellerweiterungen zu ermöglichen besteht darin, Axiome bzw. Regeln

zu verwenden. Diese müssen so aufgebaut sein, dass sie aus einer komplexeren Beschreibung von Modellerweiterungen die einfache Eigenschaft ableiten.

Transfer von Modellierungsansätzen

Dieser Bezug kann als Grenzfall zur Erweiterung von Modellen angesehen werden. Ein Modellierungsansatz wird auf veränderte Systemeigenschaften angewendet, ohne dass das Modell konzeptionell erweitert wird. Dies ist z.B. möglich, wenn eine veränderte Geometrie vorliegt, die aber keinen grundlegend neuen Ansatz erfordert bzw. nach sich zieht. Dennoch kann der Transfer Probleme bzw. Herausforderungen mit sich bringen, etwa durch zusätzliche Schwierigkeiten bei der Berechnung von Lösungen.

4.3.2.2 Bezüge zwischen Modellen und Theorien

Oft wird zwischen empirischen, semi-empirischen und theoretischen Modellen unterschieden. Diese Kategorien verdeutlichen, dass die mathematische Modellentwicklung sowohl auf der Grundlage von allgemeinen Theorien als auch allein auf der Basis von Beobachtungen bzw. Messungen oder Plausibilität stattfinden kann. Mögliche Bezüge zwischen Modellen und Theorien sind z.B. „verwendet“ (für theoretische oder semi-empirische Modelle) oder „ist konsistent/inkonsistent zu“. Theoretische Modelle sind vollständig aus einer Theorie abgeleitet und haben damit den stärksten Bezug zu allgemeinen Theorien. Für den wissenschaftlichen Diskurs ist das Zurückführen auf eine solche Theorie insbesondere für die Weiterentwicklung von Modellen interessant. So kann eine Erweiterung durch Fallenlassen von Vereinfachungen bzgl. der ursprünglichen Theorie zustande kommen.

Im Rahmen der Recherche können Modelle aus unterschiedlichen Anwendungsbereichen allein aufgrund des Zurückführens auf eine Theorie interessant sein, wie etwa im Bereich der Strömungslehre.

4.3.2.3 Bezüge zwischen Modellen und Messungen bzw. Messdaten

Verwenden von Messdaten zur Ermittlung von Modellparametern

Modellparameter, die nicht vertafelt sind oder auf andere Art im Vorhinein verfügbar sind, müssen direkt vermessen oder indirekt aus dem Abgleich von Messdaten mit berechneten Daten angepasst werden. In beiden Fällen besteht ein Bezug zwischen den Datensätzen aus Messungen und den quantifizierten Modellparametern. Wird dieser Bezug explizit gemacht, so lässt sich die Parameterbestimmung besser nachvollziehen. Noch nicht erfasst sind mit diesem Ansatz allerdings Metadaten zu den Messdaten selbst, also z.B. Informationen über die Messungen, eine eventuelle Datenaufbereitung, etc. (siehe auch Abschnitt 3.5.1 „Modellentwicklung und Messdatenverwaltung“ auf Seite 52).

Verwenden von Messdaten zur Validierung

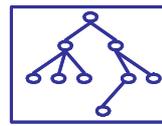
Oft sind gesonderte Messdaten zur Validierung von Modellen erforderlich. Anhand dieser Daten werden die grundsätzliche Gültigkeit und Anforderungen hinsichtlich der geforderten Präzision an das Modell abgeglichen. Wenn ein Abgleich vor diesem Hintergrund nicht möglich ist, ergeben sich negative Bezüge, ansonsten positive. Der Charakter der Bezüge kann durchaus unterschiedlich sein. Im Bereich der negativen Bezüge ist es z.B. möglich, dass Messdaten nicht konsistent sind mit Modellannahmen oder dass sie sogar eine Theorie widerlegen.

Messdaten als Ausgangspunkt der Modellentwicklung

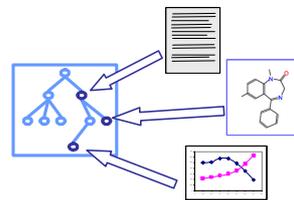
Wie im Abschnitt 3.4 „Modellbildung“ ab Seite 37 beschrieben, kann die Modellentwicklung sowohl induktiv als auch deduktiv erfolgen. Dementsprechend kann der Bezug zu Messdaten auch unterschiedliche Ausprägungen annehmen.

4.4 Ontologiebasierte Aufbereitung und Verwaltung von Forschungsergebnissen

- 1) „Sortieren“: Hierarchisches Strukturieren von Wissensgebieten (Erstellen der Ontologie);
„Es gibt Modelle, Daten und Dokumente; Modelle lassen sich unterteilen in...“



- 2) „Einordnen“: Zuordnen von Informationsquellen zur geschaffenen Wissensstruktur (bzw. Datenmodell oder Ontologie)



- 3) „Anreichern“: Erfassen von zusätzlichen Informationen („Metainformationen“) und assoziativen Zusammenhängen => implizite Verknüpfungen explizit machen

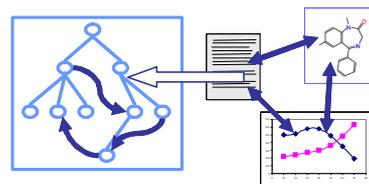


Abb. 22: Anschauliche Darstellung der wesentlichen Schritte zur Informationsverwaltung auf Ontologie-Basis

Für die Aufbereitung und Verwaltung wurde im Rahmen dieser Arbeit ein Ontologie-basierter Ansatz gewählt. Dieser Ansatz ermöglicht es, Ressourcen mit Metadaten anzureichern, die eine formale Semantik besitzen und damit maschinen-interpretierbar sind. Die Schritte zur ontologie-basierten Informationsverwaltung sind in Abbildung 22 veranschaulicht. In Verbindung mit den weiteren charakteristischen Eigenschaften ontologie-basierter Ansätze ergibt sich eine Reihe von Vorteilen:

- § Metadaten können strukturiert werden, so dass (potentiell komplexe) Aussagen über Datensätze, Simulationsmodelle und Veröffentlichungen formal und recherchierbar ausgedrückt werden können;
- § Der SchoolOnto-Ansatz ermöglicht es, für die Forschung sowie für die Entwicklung und Anwendung von Modellen relevante Aussagen mit relativ geringem Aufwand formal zu erfassen.
- § Ontologie-basierte Metadaten lassen sich dynamisch erweitern und an den Bedarf (z.B. bzgl. des Detaillierungsgrades) für Recherchen anpassen.
- § Ein im Wesentlichen auf Metadaten beschränkter Ansatz erlaubt es, bestehende Datenhaltungskonzepte (Formate, etc.) beizubehalten.
- § Ontologien eignen sich aufgrund der Ausdrucksmächtigkeit der zur Verfügung stehenden Sprachen, um sehr unterschiedliche Bereiche zu beschreiben.

4.4.1 Ontologien für Modellierungsansätze

Die hier beschriebenen Ontologien verstehen sich als Ausgangsbasis für die Umsetzung des beschriebenen Ansatzes und keinesfalls als endgültige Repräsentation der entsprechenden Domänen. Einer der Vorteile des Ontologie-Ansatzes liegt darin begründet, dass ein dynamischer Ansatz für die Repräsentation von Domänenwissen von vornherein vorgesehen ist. Die eingesetzten Systeme machen keine Annahmen über die Ontologie selbst – abgesehen von der Art der Wissensrepräsentation, d.h. des verwendeten Ontologie-Modells (siehe auch Abschnitt 5.2.1.2 „Protégé Datenmodell“ auf Seite 97). Eine Ausnahme bildet die Repräsentation von physikalischen Einheiten (siehe Abschnitt 5.3.3 „Einheiten-Verwaltung“ auf Seite 127). Diese ist aus Gründen der effizienten Verwendung auch für große Datenmengen starr auf eine bestimmte Repräsentation und eine bestimmte Interpretation abgebildet. Davon abgesehen können die vorhandenen Modelle erweitert oder bei Bedarf sogar ausgetauscht werden.

Die Ontologien werden an dieser Stelle nicht in jedem Detail beschrieben. Vielmehr soll dieser Abschnitt einen Überblick über die wichtigsten Teile bzw. Eigenschaften gegeben.

Der SCIFORG-Ansatz sieht in der hier beschriebenen Form drei Teilmodelle vor:

- § Eine Ontologie zur Beschreibung mathematischer Modelle, die ein wesentliches Konzept der Prozesswissenschaften beschreibt;
- § Eine Ontologie zur Beschreibung von Ressourcen, wie z.B. Dokumenten oder Datensätzen;
- § Eine Ontologie zur Beschreibung von Bezügen zwischen wissenschaftlichen Ergebnissen in der Form von Ansätzen, Theorien, etc. auf der Basis des SchoolOnto-Ansatzes.

4.4.1.1 Mathematische Modelle

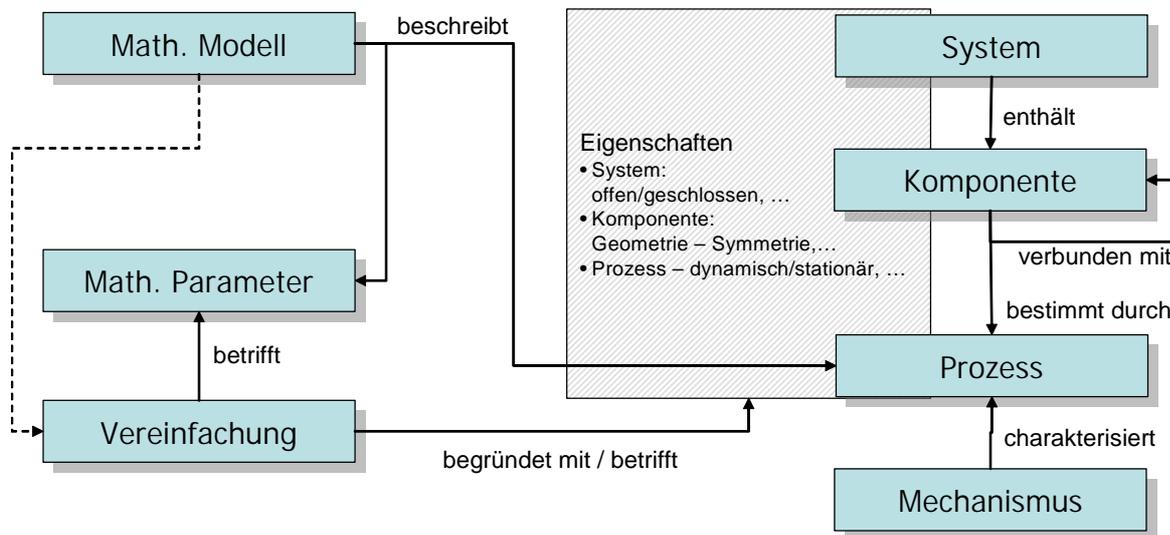


Abb. 23: Übersicht der wichtigsten Elemente zur Beschreibung mathematischer Modelle

Die Ontologie über mathematische Modelle orientiert sich an vorhandenen Ansätzen, wie sie in Abschnitt 2.2 ab Seite 10 beschrieben sind. Jedoch wird auch hier eine möglichst „leichtgewichtige“ Modellierung angestrebt.

Modelle beschreiben physikalische Mechanismen (analog zu Söderman, 1995; siehe auch Abschnitt 2.2 „Semantische Technologien“ auf Seite 10), die in Systemen auftreten. Mechanismen repräsentieren wiederum physikalische Phänomene.

Die Systemsicht leitet sich z.T. aus dem Ziel der Modellierung ab. Systeme haben Eigenschaften, die sie z.B. als „offen“ oder „geschlossen“ charakterisieren. Die Trennung zwischen Modellen und Systemen erlaubt es, die unterschiedlichen Abstraktionsstufen im Modellierungsprozess abzubilden. Dabei ist bereits die Systembeschreibung ein Modell, wenn auch kein mathematisches. Systemsichten und Modellansätze sind immer domänenabhängig. Aus diesem Grund wird davon ausgegangen, dass die Ontologien nur innerhalb einer Domäne oder Nutzergemeinschaft verwendet werden.

Die Systemsicht enthält die mereologischen und topologischen Eigenschaften des Systems (analog zu (vgl. Borst et al., 1997 und Morbach et al., 2007)

Systeme und Mechanismen werden wie im Ansatz von Borst et al. (1997) auf eine mathematische Sicht abgebildet. Anders als in der EngMath-Ontologie werden mathematische Konzepte nicht ontologisch repräsentiert (s.u.). Es ist jedoch grundsätzlich möglich, die in SCIFORG vorhandenen Ontologien entsprechend zu erweitern bzw. zu ersetzen (sofern das Frames-Datenmodell von Protégé für das jeweilige Modell hinreichend ist). Dies könnte z.B. erfolgen, um Konsistenzprüfungen auch für mathematisch motivierte Fragestellungen zu ermöglichen (siehe auch Abschnitt 6.3 „Konsistenzprüfungen für Metadaten“ auf Seite 148). Dabei besteht das Problem,

dass die Angabe von mathematischen Eigenschaften redundant zur mathematischen Repräsentation selbst ist.

Ein weiterer Unterschied zwischen dem SCIFORG-Ansatz und bestehenden Modellen wie EngMath ist die Repräsentation von physikalischen Einheiten. In SCIFORG steht die Unterstützung von Funktionalitäten im Vordergrund. Einheiten werden, wie in Abschnitt 5.3.3 Einheiten-Verwaltung auf Seite 127 beschrieben, auf der Metaebene für bestimmte Attribute („Slots“) definiert. Die Interpretation von Einheiten ist in das System unabhängig von der sonstigen Interpretation von Ontologien eingebaut. In EngMath sind Einheiten als Teilontologie beschrieben.

Die Systemsicht ermöglicht die Angabe von verschiedenen Informationen, die insbesondere aus Anwendersicht zur Beurteilung des Modellierungsprozesses und zu Recherchezwecken relevant sind. Dies betrifft z.B. die Unterscheidung von Stoffsystemen und die Anwendung von unterschiedlichen Modellansätzen auf unterschiedliche Stoffsysteme.

Mathematische Modellgrößen sind assoziiert mit physikalischen Größen in den Mechanismen bzw. mit vereinfachenden Annahmen. Letztere geben Aufschluss über den Modellierungsprozess. Mathematische Modellgrößen tauchen in Gleichungen auf, die das Modell konstituieren. Die Modellgrößen sind die einzige Verbindung zwischen der mathematischen Repräsentation (die für sich steht und für sich interpretierbar ist) und der Ontologie-basierten Repräsentation. Eine physikalische Größe kann durch eine mathematische Größe repräsentiert sein oder aufgrund des Abstraktionsprozesses und getroffener vereinfachender Annahmen kein direktes Pendant haben. Annahmen beziehen sich daher auf Größen. Das ermöglicht wiederum die gezielte Suche nach Modellen, in denen bestimmte Teile der Prozessbetrachtung vereinfacht wurden.

Mathematische Modelle können mit Hilfe der Ontologie mit ihren Eigenschaften, wie sie in Abschnitt 3 beschrieben wurden, repräsentiert werden. So können Modelle beispielsweise als „theoretisch“, „empirisch“ oder „semiempirisch“ klassifiziert werden. Diese Art der Klassifizierung bietet die Möglichkeit, bei der Recherche nach Modellen relativ allgemeine Kriterien anzuwenden. Spezifischere Kriterien können z.B. über die Systembeschreibung erreicht werden, wie das oben genannte Beispiel der Stoffsysteme zeigt.

Die mathematische Repräsentation selbst ist als Teil des Modells hinterlegt (in einer definierten Syntax und über entsprechende Schnittstellen - siehe Abschnitt 5.3.2.1 „Mathematische Formeln“ auf Seite 124). Allerdings wird keine „Ontologie der Mathematik“ zugrunde gelegt, die die Gleichungen und verwendeten Operatoren, etc. mit Hilfe von Frames ausdrückt. Die einzige „Brücke“ zwischen der mathematischen Repräsentation und den verwendeten Ontologien besteht in den innerhalb der Gleichungen auftretenden Modellparametern.

Hier wird davon ausgegangen, dass mathematische Modelle durch Eigenschaften von Systemen und Prozessen begründet werden bzw. dass sie diese in der Konsequenz ihrer Anwendung betreffen. Beispiel: Die Oberfläche einer Komponente wird über einen einzelnen skalaren Parameter „mittlere Rauigkeit“ beschrieben. Dieser taucht als mathematischer Parameter auf. Die Vereinfachung betrifft die Oberflächenbeschaffenheit. Die Begründung für diese Vereinfachung

ist, wenn sie formal repräsentiert werden soll, ggf. relativ komplex. Aus diesem Grund kann an dieser Stelle auf eine natürlichsprachliche Beschreibung zurückgegriffen werden.

Dieser Ansatz ist stark ingenieurwissenschaftlich geprägt und passt zu einer makroskopischen Perspektive. Dies betrifft vor allem die zu EngMath (vgl. Borst et al., 1997) analogen Teile, wie die System- und Prozessbeschreibung. Dennoch ist der Ansatz, ggf. mit Modifikationen, auch für Grundlagen-orientiertere Modelle mit entsprechendem Mehrwert anwendbar, da die Abstraktion und damit einhergehende Vereinfachungen sehr abstrakt beschrieben werden.

4.4.1.2 Ressourcen

Diese Ontologie beschreibt Ressourcen, wie z.B. Dateien oder Datenbanken. Tatsächlich handelt es sich hier um semantisch beschriebene Metadaten. Generell werden in den unterschiedlichsten Anwendungen bereits Metadaten zu Ressourcen, v.a. Dateien, verwaltet. So bieten z.B. Textverarbeitungen die Möglichkeit, nicht sichtbare Daten über den Autor eines Dokumentes und die assoziierte Institution zu erfassen. Diese sind jedoch nicht in einem formal spezifizierten, anwendungs-übergreifenden Format erfasst.

Downes (2004) diskutiert die Beschreibung von Ressourcen. Der Autor geht dabei davon aus, dass beschreibende Metadaten Nutzer von Ressourcen dabei unterstützen, diese zu erzeugen, zu speichern und aufzufinden sowie darauf zuzugreifen. Er unterscheidet verschiedene Arten von Ressourcen-Beschreibungen, deren Eignung für Anwender von der Art der Ressource und dem vorrangigen Zweck der Beschreibung abhängt. Dies betrifft z.B.:

- § bibliographische Metadaten: Autor; Institution, etc.;
- § Metadaten zur Klassifizierung: Stichworte, Themenkategorien;
- § Metadaten zur Rechteverwaltung;
- § Technische Metadaten (z.B. Angaben zum Format).

Für die unterschiedlichen Zwecke existieren bereits Lösungen, z.B. auch Ontologie-basiert. Bibliographische Metadaten etwa sind im sog. Dublin Core Standard beschrieben (vgl. Sugimoto et al., 2002).

Da nicht jede Art von Beschreibung für jede Art von Ressource sinnvoll ist, bietet es sich an, pro Ressource mehrere Beschreibungen aus unterschiedlichen Klassifizierungen zuzulassen. D.h. wenn eine der Kategorien nicht relevant ist, kann diese Kategorie für die entsprechende Ressource weggelassen werden.

Eine besondere Art der Beschreibung betrifft relationale Metadaten („relational Metadata“; vgl. Downes, 2004). Diese beschreiben Bezüge zwischen Ressourcen und anderen Ressourcen oder anderen Entitäten. An dieser Stelle werden Bezüge nicht in Ressourcenbeschreibungen modelliert, sondern über den erweiterten SchoolOnto-Ansatz beschrieben.

Ressourcen spielen vor dem Hintergrund der Dokumentation von Forschungsergebnissen unterschiedliche Rollen. Sie können als verwendete Daten auftreten (z.B. in Form von vorhandenen Stoffdaten) oder selbst Ergebnis der Forschungsarbeit sein. Darüber hinaus können zwischen Ressourcen Bezüge bestehen, die das Resultat bestimmter Arbeitsprozesse sind. Hierzu ein Beispiel: Aus einem Rohdatensatz, der experimentell ermittelte, zeitabhängige Systemgrößen enthält, wird ein Teil extrahiert. Dieser Teil enthält Messpunkte, die als stationär identifiziert wurden. Anschließend werden über ein Bilanzmodell weitere Größen berechnet und den Daten hinzugefügt. Der resultierende Datensatz wird wiederum gespeichert, um in einem anderen Kontext (Vergleich mit anderen Daten, Modellvalidierung) verwendet werden zu können. Der zweite Datensatz ist somit sowohl durch eine Reduktion der Daten als auch durch eine Ergänzung aus dem ersten hervorgegangen. Mit dem hier beschriebenen Ansatz können solche Verbindungen nachvollzogen werden. Ressourcen sind mit Ressourcen über Bezüge verbunden, die wiederum auf Prozesse, wie etwa das Filtern von Daten, verweisen.

4.4.1.3 SchoolOnto

Der SchoolOnto-Ansatz und die SchoolOnto-Ontologie sind in Abschnitt 4.3 auf Seite 71 beschrieben. Er bietet eine Möglichkeit, sehr allgemeine Aussagen für Bezüge zwischen Modellen, Ressourcen und anderen Entitäten auszudrücken.

In dieser Arbeit wurde die SchoolOnto-Ontologie in abgewandelter Form eingesetzt. Dies hat folgenden Hintergrund: In der SchoolOnto-Ontologie sind Bezüge als „Properties“ definiert. Dies schränkt die Erweiterung im Sinne einer Spezialisierung stark ein. Hierzu ein Beispiel: Der Bezug „improves on“ (siehe Abschnitt 4.3 „Der SchoolOnto Ansatz“ ab Seite 71) soll so ergänzt werden, dass deutlich wird, dass es sich um die Erweiterung eines Ansatzes handelt, die auf weniger vereinfachenden Annahmen beruht. Diese Zusatzinformation ist Teil des Bezuges.

Das Problem an dieser Stelle besteht darin, dass die Möglichkeiten zur Erweiterung auf der Seite der Modellierung bzw. der Repräsentation eingeschränkt sind. Für Properties bestehen grundsätzlich zwei Möglichkeiten: das Hinzufügen von Parametern und die Hierarchisierung. Dabei können weder in RDF(S) noch im Frames-Modell von Protégé Parameter für die Instanziierung von Properties (in Protégé „Slots“) verwendet werden. Die Hierarchisierung, d.h. das Einführen von sog. Subproperties, auf der anderen Seite wird der Problematik nicht gerecht. Subproperties ermöglichen zwar eine Spezialisierung (etwa „erweitert“ als Spezialisierung von „verwendet“), erlauben es aber nicht, diese mit weiteren Daten zu versehen.

Die hier verwendete Lösung besteht in der Verwendung von Klassen anstelle von Properties bzw. Slots. Die Instanz einer solchen Klasse repräsentiert den entsprechenden Bezug, d.h. es wird eine Klasse „Erweiterung“ anstelle eines Properties/Slots „erweitert“ verwendet. Diese Klasse kann durch Unterklassen spezialisiert werden, die wiederum beliebige Eigenschaften haben können.

4.4.1.4 Weitere Domänen-unabhängige Konzepte

Zusätzlich zu den beschriebenen Elementen bestehen noch eine Reihe weiterer, Domänen-unabhängige Klassen und Slots. Dazu gehört u.a. die Beschreibung von Institutionen und Personen.

4.5 Werkzeug-Unterstützung von Fachanwendern

Die hier beschriebene Ontologie-basierte Erfassung und Verwaltung wissenschaftlicher Ressourcen und Metainformationen (zur Herkunft der Daten, etc.) geht davon aus, dass die fachspezifische Repräsentation von Daten (i) nicht vollständig in der Ontologie-Darstellung aufgeht und (ii) domänenspezifische Werkzeuge erfordert, die über eine generische Unterstützung des Erstellens und Anwendens von Wissensmodellen hinausgeht. Das beinhaltet, dass das System der Charakteristik domänenspezifischer Daten hinsichtlich Darstellung, Editierung, Performanz, etc. gerecht wird. Dies gilt z.B. für die typischen Metadaten wissenschaftlicher Ressourcen, wie etwa die physikalischen Einheiten für Datensätze aus Simulation oder Experiment.

4.5.1 Daten- und Metadaten-Ebene

Die Verwaltung wissenschaftlicher Daten kann auf der Daten-Ebene und auf der Metadaten-Ebene unterstützt werden. So ist ein rein Metadaten-orientierter Ansatz denkbar, bei dem Dokumente und numerische Daten lediglich referenziert werden. Metadaten beschreiben Dokumente und andere Quellen im Sinne von zusätzlichen Informationen oder sie ermöglichen bestimmte „Sicht“ auf die Daten selbst. Letzteres wird v.a. durch Integrationsansätze umgesetzt. Dieser Ansatz setzt für Datenquellen voraus, dass entsprechende Schnittstellen verfügbar sind. Darüber hinaus wird davon ausgegangen, dass eine bestehende Infrastruktur besteht, in der die benötigten Funktionalitäten zur Verwaltung und zum Abruf von Daten vorhanden sind.

Domänenspezifische Daten können auf unterschiedliche Arten durch Ontologien angereichert werden. Die Konvertierung bestehender Daten in ein Format mit expliziter Semantik stellt dabei einen Extremfall dar. Umzusetzen ist ein solcher Ansatz nur für voll strukturierte Daten, etwa aus Datenbanken oder XML-Dateien. Sämtliche Möglichkeiten ontologie-basierter Systeme, was die Erweiterung, das Zusammenführen mit anderen Ontologien („Merging“) oder die Suche betrifft, können direkt auf die Daten angewendet werden. Für Systeme, welche die entsprechende Wissensrepräsentation nicht unterstützen, sind die Daten dann nicht mehr nutzbar. Damit stehen die Funktionalitäten dieser Systeme nicht mehr zur Verfügung. Darüber hinaus ist dieser Weg für sehr große Datenmengen nicht praktikabel.

Ein weiterer Ansatz besteht in der bereits erwähnten Integration der Daten. Dabei wird die verwendete Infrastruktur bzw. der verwendete Container der Daten (die Datei, die Datenbank oder die Applikation) durch den Ontologie-basierten Datenzugriff gekapselt. Auf diese Art können mehrere Quellen zusammengeführt und durch semantische Beschreibung angereichert werden. So ist es etwa möglich, Daten zu konsolidieren, Konflikte auf semantischer Ebene aufzulösen

oder über Axiome bzw. Regeln zusätzliche Informationen abzuleiten. Die vorhandene Infrastruktur zur Datenhaltung bleibt hierbei unberührt, d.h. die Daten des Systems werden durch die Ontologie(n) interpretiert. Der Mehrwert kommt durch die Verwendung semantischer Metadaten sowie durch entsprechende Schnittstellen zustande. Bestehende Systeme können damit parallel genutzt werden.

Das in Abschnitt 5 näher beschriebene SCIFORG-System basiert auf einem gemischten Ansatz. Es ermöglicht das Erstellen von Domänenontologien zur Beschreibung wissenschaftlicher Ressourcen auf Metadaten-Ebene und bietet „Repository“-Funktionalitäten. Letzteres beinhaltet eine Infrastruktur, in der nicht nur Metadaten, sondern auch die eigentlichen Daten, d.h. semistrukturierte Dokumente, Messdaten oder berechnete Daten verwaltet werden können.

Für Forschungsinstitutionen im Bereich der Prozesswissenschaften sind zum Zeitpunkt dieser Arbeit keine Systeme zur Datenhaltung wissenschaftlicher Ergebnisse verfügbar, die als Quasi-Standard gelten können. Die Archivierung erfolgt oft Datei-basiert, wobei keine für Anwender effizienten Zugriffsstrukturen auf Daten sowie keine semantische, explizite und eindeutig zu interpretierende Repräsentation und Verwaltung von Metadaten gegeben ist. Auf dieser Grundlage ist es relativ aufwändig, Daten gezielt anhand bestimmter Kriterien herauszusuchen, zu filtern und zu visualisieren. Individuelle Formen der Datenorganisation (wie z.B. ein spezielles Ablagesystem, das Daten Datei-basiert in Verzeichnissen verwaltet) haben den Nachteil, dass sie stark an persönliches Wissen geknüpft sind. Die bereits beschriebenen Probleme bzgl. der Nachhaltigkeit werden dadurch begünstigt.

Datenverwaltung im hier beschriebenen Ansatz

Für kleinere, aber stark verschachtelte bzw. durchstrukturierte Daten (etwa XML-Dateien die keinem sehr einfachen Schema genügen, oder Daten aus Tabellenkalkulationen, die durch komplexere Tabellenstrukturen repräsentiert sind) bietet sich die direkte Überführung in ein ontologie-basiertes Format an.

Für größere Datenmengen, die z.B. aus automatisierten Messungen stammen (etwa aus Prozessleitsystemen) bietet das SCIFORG-System die Möglichkeit der Verwaltung in einem klassischen (relationalen) Datenbank-Schema, wobei der Zugriff auf die Daten und die Beschreibung der Metadaten frame-basiert erfolgt. Dies geschieht über die in Abschnitt 5.3.1 beschriebene objektrelationale Abbildung. Anwender können damit auf Daten (wie Messwerte) und Metadaten (wie Einheiten) bzw. übergeordnete Informationen (Datenherkunft, Kontext) auf der Grundlage eines Datenmodells zugreifen. Darüber hinaus bietet der Ansatz Anwendern die Möglichkeit, ein Daten-Repository aufzubauen, das in Bezug auf die Skalierbarkeit, die Möglichkeiten des Datenzugriffs, etc. wesentlich besser für die Archivierung großer Datensätze geeignet ist als die verbreiteten Tabellenkalkulationen.

Bei der Unterstützung domänenspezifischer Datenstrukturen und Metadaten stellt der hier vorgestellte Ansatz einen Kompromiss zwischen der Ontologie-basierten Beschreibung von Infor-

mationen und dem Einsatz pragmatischer Methoden zur Datenverwaltung dar. Letzteres betrifft etwa chemische und mathematische Formeln, sowie physikalische Einheiten.

Chemische und mathematische Formeln

Chemische und mathematische Formeln werden hier als Eigenschaften von Elementen (Instanzen) modelliert, für die eine besondere Interpretation vereinbart wird. Ein komplexer mathematischer Ausdruck beispielsweise wird also nicht Ontologie-basiert „aufgelöst“, sondern als eine einfache Zeichenkette repräsentiert. Alternativ wäre denkbar, eine „Mathematik-Ontologie“ zu erstellen, die auf der Grundlage einer Ontologie-Sprache mit Hilfe von Klassen und deren Merkmalen ein Schema für mathematische Ausdrücke definiert. Ein konkreter mathematischer Ausdruck wäre eine Instanziierung dieses Schemas. Dabei ist jedoch zu berücksichtigen, dass

- § mathematische Ausdrücke bereits über eine definierte Semantik verfügen, die durch den jeweiligen Teilbereich der Mathematik (differentielle Algebra, lineare Algebra, etc.) bestimmt ist;
- § Ontologie-Sprachen diese Interpretation nicht unterstützen, die mathematische Interpretation aber Aufgabe domänenspezifischer Anwendungen wie etwa von Simulationswerkzeugen oder symbolischen Interpretern ist;
- § der wissensbasierte Ansatz nicht darauf abzielt die Ausdrücke mathematisch zu interpretieren, sondern Aussagen über sie zu machen („Gleichung X beschreibt Zusammenhang Y“), bzw. sie zu verwenden um Aussagen über andere Elemente einer Ontologie zu treffen.

Für mathematische Formeln existieren bereits Formate mit definierter Semantik, die dem Austausch von Daten dienen, z.B. der MathML-Standard sowie das OpenMath-Format (vgl. Abschnitt 4.2.1.1).

Eine ähnliche Situation ist für chemische Formeln gegeben. Neben XML-basierten Formaten wie der in Abschnitt 4.2.1.1 erwähnten Chemical Markup Language existieren eine Reihe von Formaten mit unterschiedlichen Möglichkeiten zur Beschreibung chemischer Konzepte, wie z.B. SMILES (vgl. Weininger, 1988).

Das in Kapitel 5 beschriebene SCIFORG-System erlaubt es, mathematische und chemische Formeln eindeutig zu beschreiben, ohne dass hierzu gesonderte Ontologien verwendet werden. Die Grundlage sind einfache Attribute (Zeichenketten). Die dazu verwendeten Komponenten unterstützen XML-basierte Formate/Standards, wie CML und MathML.

Physikalische Einheiten

Das SCIFORG-System unterstützt direkt die Verwendung und Auswertung physikalischer Einheiten, wie in Abschnitt 5.3.3 „Einheiten-Verwaltung“ auf Seite 127 erörtert. Einheiten werden beim Datenimport, dem Export und bei der Datenvisualisierung unterstützt. Dabei ist die Inter-

pretation von Einheiten in das System integriert. Die Einheiten selbst werden jedoch nicht Ontologie-basiert repräsentiert und ausgewertet. Es existieren keine Klassen, Slots oder Instanzen, die die Struktur von Einheiten als mathematische Ausdrücke explizit machen. Sie werden vielmehr als Zeichenketten dargestellt, die besonders interpretiert werden – ähnlich wie bei mathematischen und chemischen Formeln.

Einheiten werden innerhalb des Datenmodells als Metadaten zu bestimmten Eigenschaften („Slots“ – siehe Abschnitt 5.2.1.2 „Protégé Datenmodell“ auf Seite 97) angegeben. Eine auf diese Weise annotierte Eigenschaft bekommt dadurch die Bedeutung einer physikalischen Größe. System-intern wird der als Einheit angegebene Ausdruck interpretiert und bei Einheiten-Umrechnungen bzw. dem Abgleich von Einheiten verwendet. Selbst für große Datenmengen sind Einheiten-Umrechnungen auf sehr effiziente Weise möglich, da die Interpretation der Einheit etwa bei der Datenvisualisierung nicht für jeden Datensatz einzeln durchgeführt wird.

Dieser Ansatz hat große Vorteile bzgl. der Effizienz der Datenverwaltung und –auswertung. Der Nachteil liegt darin, dass dieser Ansatz nicht mit Ontologie-Standards wie RDF(S) kompatibel ist. Vor dem Hintergrund der Austauschbarkeit der verwalteten Wissensmodelle wäre es daher erforderlich, die intern verwendete Darstellung von Einheiten auf eine externe Repräsentation abzubilden, die im Sinne einer Ontologie interpretierbar ist. Allerdings ist dabei zu beachten, dass der hier beschriebene Ansatz auf dem Prinzip der Reifizierung („Aussagen über Aussagen“) beruht. Es wird definiert, dass bestimmte Elemente (z.B. „Datensatz aus Messung“) eine bestimmte Eigenschaft aufweisen (z.B. „Volumen“). Zusätzlich wird festgelegt, dass diese Aussage impliziert, dass die entsprechende Eigenschaft auf eine bestimmte Art zu interpretieren ist (z.B. als „Wert in Litern“). Diese Art der Modellierung wird von einigen Ontologie-Sprachen nicht unterstützt. Alternativ ließe sich dieser Zusammenhang über ein zusätzliches Objekt darstellen. Dabei ist „Wert in kg“ kein einfacher Zahlenwert, sondern ein Objekt mit verschiedenen Eigenschaften. Die physikalische Einheit könnte in diesem Schema ebenfalls als eigenes Objekt (und nicht als besonders zu interpretierende Zeichenkette) modelliert werden. Eine ontologie-basierte Sicht auf Einheiten ist z.B. bei Gruber (1995) beschrieben.

Im hier vorgestellten Ansatz wurde auf die durchgängig Ontologie-basierte Repräsentation zugunsten von effizienten (und effizient zu implementierenden) Auswertungs- und Zugriffsmethoden verzichtet. Allerdings ist es möglich, das entwickelte System durch einen entsprechenden Export zu ergänzen. Ein solcher Export würde allerdings die verwendeten Ontologien zwangsläufig verändern.

4.5.2 Mehrwert der Ontologie-basierten Datenverwaltung als Anreiz

Da das im Rahmen dieser Arbeit entwickelte System nicht allein auf die Akquisition von Information zum Zwecke der Nachhaltigkeit und besseren Recherchierbarkeit von Forschungsergebnissen abzielt, sondern auch Vorteile für Anwender beim Zugriff auf die eigenen Daten bietet, werden Anreize zur Verwendung des Systems geliefert. Dabei besteht keineswegs der Zwang, die Datenverwaltung vollständig auf das System zu verlagern, da vorhandene Anwendungen ergänzt,

aber nicht notwendigerweise ersetzt werden. Die Verwaltung von semantischen Metadaten ermöglicht bereits einen Mehrwert, ohne dass etwa Messdaten direkt in SCIFORG verwaltet werden müssen – auch wenn das System die Möglichkeit dazu bietet.

In Kapitel 5 wird deutlich, dass die Implementierung der Ontologie-basierten Verwaltung wissenschaftlicher Ergebnisse einen gezielten und schnellen Zugriff auf bestimmte Daten ermöglicht.

4.5.3 Anwenderschnittstellen/Oberflächen

Ein Schwerpunkt liegt beim entwickelten System auf domänen-spezifischen Anwenderschnittstellen, die sowohl für das Erfassen als auch für das Abrufen von Daten zur Anwendung kommen. Charakteristische Konzepte werden entsprechend visualisiert. Dabei werden parallel zwei verschiedene Ansätze unterstützt: formularbasierte und Graphen-basierte Oberflächen.

4.5.3.1 Formular-basierter Ansatz

Das zugrunde liegende Werkzeug Protégé bietet bereits einen Formular-Generator und –Editor für die Wissensakquisition. Dieser generiert Eingabeformulare basierend auf dem zugrunde liegenden Klassenschema. Der Formular-Editor wurde in dieser Arbeit durch domänenspezifische Komponenten erweitert (siehe Abschnitt 5.3.2 „Anwender-Schnittstellen für die Prozesswissenschaften“ auf Seite 124). Darüber hinaus wurde eine Anwender-Schnittstelle geschaffen, die auch Suchanfragen über Formulare ermöglicht (siehe Abschnitt 5.3.4 „Werkzeuge für Regeln, Suchanfragen und zur Navigation“ auf Seite 129). Dieser Ansatz eignet sich vor allem für einfache Suchanfragen ohne Mehrfachverknüpfungen über mehrere Klassen und Slots. Vergleichsoperatoren wie „größer“, „kleiner“, etc. können direkt im Formular ausgewählt werden.

Der Formular-basierte Ansatz kommt Anwendern entgegen, die den Umgang mit Eingabemasken gewohnt sind. Mit gewissen Einschränkungen lassen sich die Formulare darüber hinaus anpassen.

4.5.3.2 Graphen-basierter Ansatz

Suchanfragen sowie regelhafte Zusammenhänge lassen sich auch über eine Graphen-basierte Schnittstelle erstellen. Dabei können komplexere Anfragen graphisch formuliert werden. Domänenspezifische Symbole für bestimmte Klassen erhöhen die Lesbarkeit von graphisch formulierten Suchanfragen und Regeln sowie den Wiedererkennungswert für die entsprechenden Objekte.

Der Graphen-basierte Ansatz ist mit dem Formular-basierten Ansatz kombiniert, d.h. Eingabeformulare für einfache Suchkriterien stehen auch in dieser Lösung zur Verfügung.

5 Implementierung

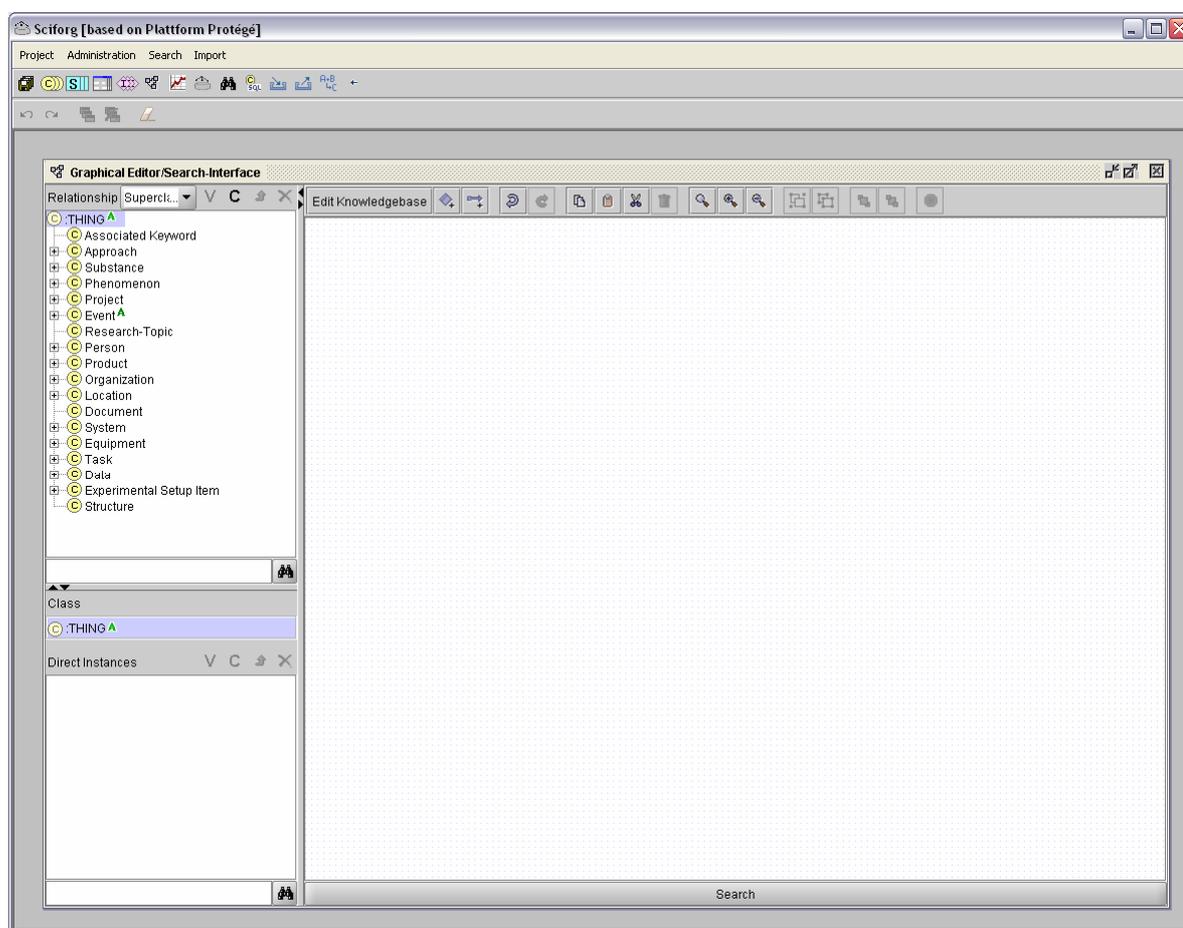


Abb. 24: SCIFORG Haupt-Ansicht

Das SCIFORG-System ist als Referenzimplementierung für den in dieser Arbeit beschriebenen Ansatz zu sehen. Es zeigt, wie auf der Grundlage bestehender Technologien aus unterschiedlichen Bereichen eine Plattform für die wissensbasierte Verwaltung von Forschungsergebnissen umgesetzt werden kann. Auch wenn Teile des Systems deutlich über eine prototypische Umsetzung hinausgehen, so ist das Ziel an dieser Stelle nicht die Umsetzung einer Produktivlösung für unterschiedlichste Institutionen und Anwenderprofile gewesen.

SCIFORG ist als integrierte Anwendung entwickelt worden, die Daten aus externen Anwendungen aufnehmen kann. Das System ist so aufgebaut, dass es vor allem, aber nicht ausschließlich für die Erfassung und Verwaltung semantischer Metadaten verwendet werden kann. Für semantische Metadaten stehen eine Reihe von Funktionalitäten und Schnittstellen zur Verfügung, wie z.B. die domänenspezifischen Formulare. Darüber hinaus kann das System auch für die Verwaltung der Daten selbst, d.h. für Messdatensätze und Simulationsdaten, oder für Dokumente eingesetzt werden. Hierzu stehen Import-Schnittstellen und eine Einheiten-Verwaltung sowie die Datenvisualisierung zur Verfügung.

5.1 Architektur

Abbildung 25 zeigt die Architektur des Systems. Im Kern wird der Ontologie-Editor Protégé verwendet. Protégé wird in Abschnitt 5.2.1 zusammen mit den für SCIFORG entwickelten Erweiterungen vorgestellt. Die Wissensrepräsentation von SCIFORG beruht vollständig auf dem Frame-Datenmodell von Protégé (siehe Abschnitt 5.2.1.2). Dabei wurde das Persistenzmodell bzw. die Datenspeicherung von Protégé um eine Datenbank-basierte Lösung erweitert, die auf dem Prinzip der objekt-relationalen Abbildung beruht, und ist in Abschnitt 5.3.1 erklärt. Diese unterstützt effiziente Suchanfragen für große Datenmengen.

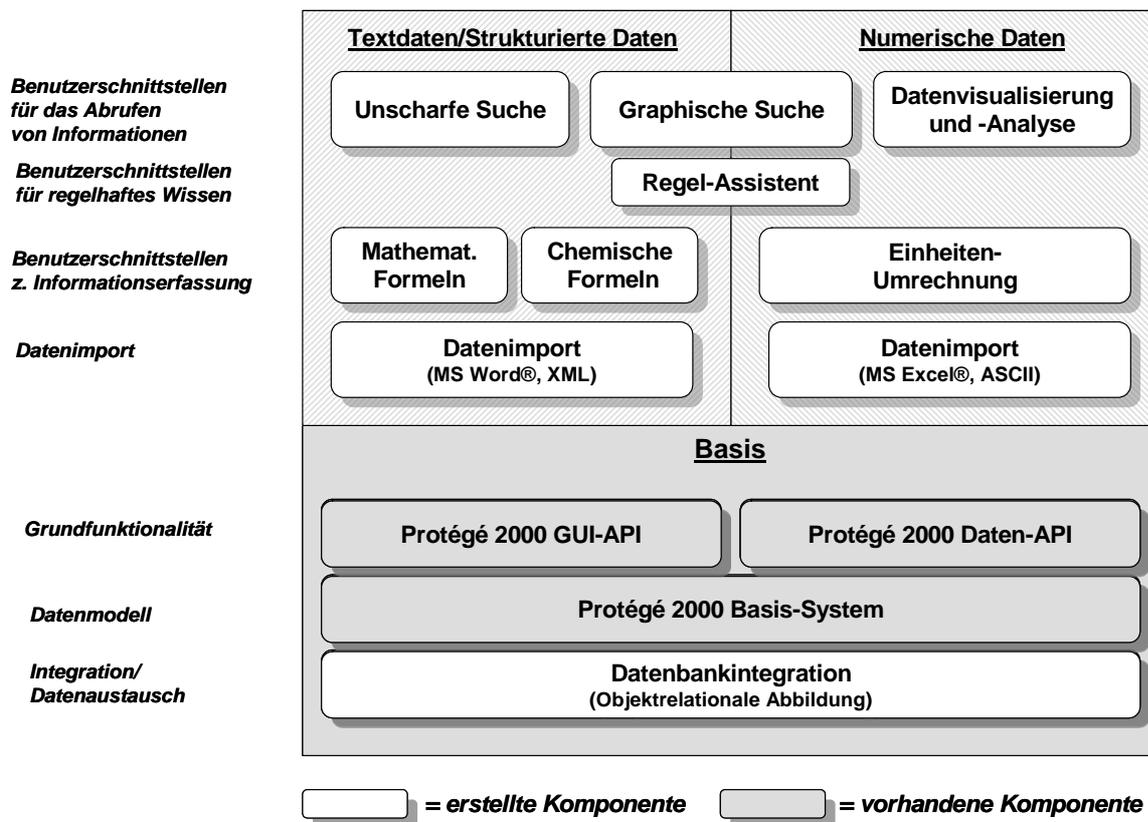


Abb. 25: Grundlegende SCIFORG-Komponenten

SCIFORG wurde als integriertes System entworfen und entwickelt. Es bietet eine Reihe von Grundfunktionalitäten speziell für den Bereich der Prozesswissenschaften und kapselt dabei Details der Wissensrepräsentation. Darüber hinaus stehen, wie in Abschnitt 4.5.3 beschrieben, graphische Komponenten für domänenspezifische Konzepte und für Suchanfragen bzw. Regeln zur Verfügung.

5.2 Verwendete Applikationen/Bibliotheken

Das SCIFORG-System ist aus einer Reihe von bestehenden Anwendungen und Bibliotheken aufgebaut. Das entspricht dem Selbstverständnis des in dieser Arbeit beschriebenen Ansatzes als eine Transferleistung.

In diesem Abschnitt werden die wichtigsten der verwendeten Anwendungen und Bibliotheken kurz vorgestellt. Das Hauptaugenmerk liegt dabei auf dem Ontologie-Editor Protégé 2000, welches den Kern des SCIFORG-Systems bildet.

5.2.1 Protégé 2000

Protégé ist ein von der Stanford Medical Informatics (SMI) Gruppe entwickelter Ontologie Editor. Es handelt sich um ein Java-basiertes Programm mit konfigurierbarer graphischer Oberfläche, das in einer langen Reihe von Entwicklungsschritten aus einem ursprünglich 1989 entwickelten Werkzeug zur Wissensakquisition für Expertensysteme hervorgegangen ist (vgl. Gómez-Perez et al., 2004). Die in den letzten Jahren erschienenen Versionen von Protégé bieten umfassende Funktionen zum Erstellen und Verwalten von Ontologien, wobei aktuelle Standards wie OWL und RDF(S) unterstützt werden. Das zentrale Format, an welchem sich auch die interne Datenspeicherung orientiert, basiert auf Frames (im Folgenden mit „Protégé Frames“ bezeichnet) und hat große Überlappungen mit dem CLIPS-Format (vgl. Giarratano und Riley, 1994).

Protégé zeichnet sich insbesondere durch folgende Eigenschaften aus:

- § Unterstützung verschiedener Formate: Protégé unterstützt eine Reihe von Formaten durch die Editiermöglichkeiten (OWL, RDF(S) und Protégé Frames) und über Export- und Import-Filter (u.a. OKBC, F-Logic).
- § Offene Architektur: Protégé bietet ein Plugin-Konzept, welches verschiedene Formen von Erweiterungen erlaubt.
- § Flexible, formularbasierte Anwenderschnittstellen: Die graphische Oberfläche kann auf verschiedene Arten konfiguriert und angepasst werden.
- § Austauschbare Datenspeicherung: Die standardmäßig eingestellte Speicherung basiert auf Dateien, während durch entsprechende Konfiguration auch eine relationale Datenbank eingesetzt werden kann.

5.2.1.1 Protégé als hybrides Werkzeug

Wie bereits erwähnt unterstützt Protégé verschiedene Formate. Dies betrifft nicht nur den Import und Export von Dateien sondern auch die Möglichkeiten zur Erstellung und Bearbeitung von Ontologien über formularbasierte Editoren, die zusätzlichen Funktionalitäten wie etwa Konsistenzprüfungen oder strukturierte Suchen und nicht zuletzt die internen Schnittstellen. Protégé

unterstützt Frames auf der einen und OWL/RDF(S) auf der anderen Seite durch unterschiedliche Modi mit unterschiedlichen Eingabefeldern.

Dabei bleibt der grundsätzliche Aufbau des Programms (Position von Menüleiste, Navigationsansicht für das Klassen-Schema, Position von Formularen) identisch. Der Detailaufbau der Komponenten und die über Editoren, Menüeinträge, etc. verfügbaren Funktionen sind jedoch auf die unterschiedlichen Sprachen abgestimmt und unterscheiden sich daher z.T. erheblich. Aufgrund dieser Tatsache, aber auch durch den internen Aufbau der Unterstützung verschiedener Formate kann Protégé als hybrides Programm gesehen werden, das (zwei) verschiedene Werkzeuge vereint und dabei bis zu einem gewissen Grad ein einheitliches Erscheinungsbild und eine geteilte Infrastruktur (Persistenzmechanismus, Aufbau der Oberfläche, standardisierte „Widgets“).

Entwicklungshistorisch sowie vom inneren Aufbau betrachtet bildet der Frames-basierte Teil von Protégé den Kern des Programms. Die Persistenzschicht ist über so genannte „FrameStores“ realisiert, die das Speichern und Laden von Ontologien übernehmen. Dabei kann es mehrere FrameStores geben, um unterschiedliche Speichermechanismen zu erlauben. Für den Anwender bzw. Programmierer steht eine Frame-basierte Schnittstelle zur Verfügung, die auf die FrameStores gründet. Auch die Schnittstelle für OWL und RDF(S)-Modelle nutzt dieses Kernmodell als Persistenzschicht.

Für die vorliegende Arbeit wurde der Frame-basierte Teil des Programms und damit der „Kern“ des Systems als Grundlage verwendet. Damit wurde zugleich die Ausdrucksmächtigkeit von Protégé-Frames für die verwendeten Ontologien und deren mögliche Erweiterung zugrunde gelegt. Die Gründe hierfür werden noch erörtert. Im Folgenden wird mit der Bezeichnung Protégé allein auf den Frame-basierten Teil Bezug genommen, ohne dass dies jedes Mal explizit gemacht wird.

5.2.1.2 Protégé Datenmodell

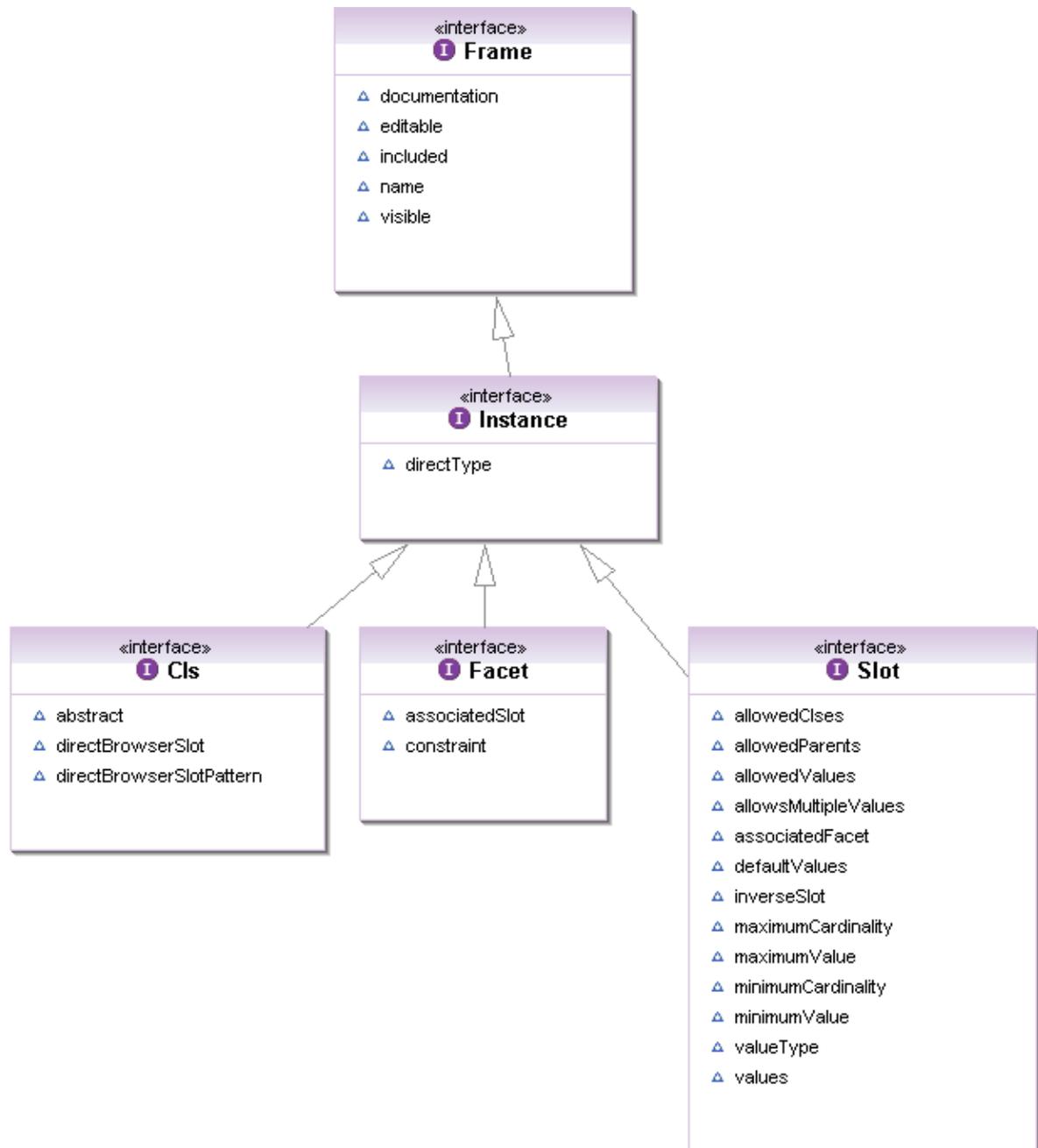


Abb. 26: Interfaces der Protégé Frames API (Ausschnitt ; ohne Methoden)

Das Datenmodell von Protégé ist in Noy et al. (2000) beschrieben. Es basiert auf Frames und ist kompatibel mit dem „Open Knowledge Base Connectivity“ Protokoll (OKBC; vgl. Chaudhri et al., 1998). Frame-basierte Datenmodelle sind u.a. im Bereich so genannter Expertensysteme gängig. Das Modell von Protégé hat seinen Ursprung bei der CLIPS („C Library Integrated Production System“), einem Format für Produktionsregelsysteme (vgl. Giarratano und Riley, 1994). Das Datenmodell ist in Form von Java-Interfaces zugänglich (siehe Abbildung 26).

Wenngleich das Datenmodell von Protégé sich von dem W3C-Standards RDF(S) unterscheidet, so ist es dennoch möglich, die Frame-basierte Darstellung von Protégé auf RDF(S) abzubilden und mit Hilfe von Protégé RDF(S) Wissensbasen aufzubauen (vgl. Noy et al., 2001).

Grundelemente des Datenmodells

Die grundlegenden Elemente des Datenmodells von Protégé umfassen Klassen („Classes“), „Slots“, „Facets“ und Instanzen. Das Verhältnis von Klassen und Instanzen ist ähnlich dem von objektorientierten Sprachen im Bereich der Programmierung (vgl. Balzert, 1999), ebenso das Prinzip der Vererbung innerhalb von Klassenhierarchien.

Klassen werden verwendet, um hierarchische Strukturen im Sinne einer expliziten Klassifizierung aufzubauen. So kann etwa eine Klasse „Kolbenpumpe“ als Unterklasse einer Klasse „Pumpe“ definiert werden. „Kolbenpumpe“ hat damit alle Eigenschaften von „Pumpe“ und die speziellen Eigenschaften von Kolbenpumpen. Eine Klasse kann als Kategorie aufgefasst werden, für die konkrete Ausprägungen, die Instanzen, existieren können. Eine Instanz wäre im o.g. Beispiel eine bestimmte Pumpe, die ggf. eine real existierende Pumpe repräsentiert.

Eigenschaften bzw. Attribute und Merkmale werden über Slots beschrieben. Es werden verschiedene Arten von Slots unterstützt, die unterschiedliche Wertebereiche zulassen. Dazu gehören Attribute mit einfachen Datentypen wie Zeichenketten, Zahlenwerte oder boolesche Werte („wahr“/„falsch“) und Relationen auf andere Klassen oder Instanzen. Ein typischer Slot für Instanzen einer Klasse „Pumpe“ könnte etwa ein einfaches Attribut „Bezeichnung“ sein. Relationen stellen Bezüge zwischen Instanzen her. Ein Slot „hat Hersteller“ mit Instanzen einer Klasse „Hersteller“ als Wertebereich ist ein Beispiel für eine solche Relation.

Wie auch bei anderen objektorientierten Datenmodellen können für Slots Kardinalitäten definiert werden, die den Wertebereich bezüglich der Anzahl der Werte einschränken. So kann festgelegt werden, dass eine Pumpe genau eine Bezeichnung hat. Die Eigenschaften von Slots werden über so genannte Facets festgelegt. Dies geschieht über die Verwendung von „Template-Slots“, die noch erklärt werden.

Slots existieren in unterschiedlichen Ausprägungen, je nach Modellierungsebene. Für eine Klasse können so genannte „Template-Slots“ definiert werden. Diese definieren, welche Eigenschaften Instanzen einer Klasse grundsätzlich haben, z.B. dass eine Pumpe einen Hersteller hat. Für Instanzen der Klasse werden Template-Slots zu Slots, die die konkreten Werte für die Eigenschaften enthalten. Ein Slot definiert also Eigenschaften der Klasse selbst (z.B. eine Dokumentation oder Beschreibung der Klasse), während ein Template-Slot Eigenschaften der Instanzen dieser Klasse definiert.

Eine Besonderheit und gleichzeitig ein wichtiger Unterschied etwa zu objektorientierten Programmiersprachen besteht darin, dass Slots klassenübergreifend definiert werden. Wenn für die Klasse „Pumpe“ ein Template-Slot „Bezeichnung“ definiert werden soll und andere Klassen (die keine Unterklassen von „Pumpe“ sind) ebenfalls einen Template-Slot „Bezeichnung“ erhalten

sollen, so muss ein identischer Template-Slot „Bezeichnung“ für alle entsprechenden Klassen verwendet werden. Dies hängt mit der Rolle von Bezeichnern im Datenmodell zusammen.

Jedes Grundelement im Datenmodell hat einen eindeutigen Bezeichner. Dieser darf unabhängig von der Art des Elements insgesamt nur einmal verwendet werden. D.h. es ist nicht zulässig, eine Klasse „Hersteller“ und gleichzeitig einen Slot „Hersteller“ zu definieren. Der Bezeichner allein ist hinreichend, um ein Element zu identifizieren.

Ähnlich wie bei objektorientierten Programmiersprachen existiert das Prinzip der abstrakten Klasse. Abstrakte Klassen haben keine direkten Instanzen. Wird z.B. die Klasse „Pumpe“ als abstrakte Klasse definiert, so hat das zur Konsequenz, dass jede konkrete Pumpe einer spezielleren Kategorie wie etwa „Kreiselpumpe“ entsprechen muss (vorausgesetzt diese Unterklasse ist nicht abstrakt definiert). Damit kann keine konkrete Pumpe repräsentiert werden, für die eine speziellere Kategorie nicht bekannt ist.

Protégé unterstützt das Prinzip der Mehrfachvererbung. Klassen können als Unterklassen mehrerer, über die Klassenhierarchie nicht verbundener Klassen definiert sein. Sie erben in diesem Fall alle in den unterschiedlichen Oberklassen definierten Eigenschaften.

Die Ebene des Klassenschemas und die der Instanzen sind nicht streng getrennt. So kann der Wertebereich von Slots auch Klassen enthalten. In der Programm-internen Umsetzung des Datenmodells wird deutlich, dass jede Klasse automatisch auch eine Instanz ist.

Tabelle 3 fasst die Grundelemente des Datenmodells von Protégé zusammen. Abbildung 26 zeigt die Java-Schnittstellen (Interfaces) des Datenmodells ohne Methoden.

Tab. 3: Elemente des Datemodells von Protégé Frames

	Beschreibung	Beispiel
Klasse („Class“)	Klassen dienen dem Aufbau einer hierarchischen Klassenstruktur zur expliziten Klassifizierung von Domänenkonzepten.	Klasse „Pumpe“ Unterklasse „Kreispumpe“
Slot	Slots dienen der Beschreibung von Eigenschaften, Merkmalen, Attributen von Objekten. Slots werden an Unterklassen vererbt. Welche Eigenschaften eine Instanz einer Klasse prinzipiell haben kann, ist über Template-Slots definiert.	Slot „Bezeichnung“ Slot „Hersteller“
Facet	Facets definieren Eigenschaften von Slots, wie etwa die Kardinalität oder den Wertebereich.	Wertebereich Zeichenkette und Kardinalität „eins“ für den Slot „Bezeichnung“
Instanz	Instanzen sind Objekte, die eine bestimmte Klassenzugehörigkeit haben.	Eine konkrete Pumpe oder ein konkreter Hersteller

Metaklassen und Metamodell

Das Datenmodell von Protégé unterstützt die Metamodellierung. Das bedeutet, dass Klassen ihrerseits Instanzen von Metaklassen sind. Die Klasse-Instanz Beziehung existiert damit auf zwei Ebenen. Auf der für den Anwender in erster Linie sichtbaren Ebene werden Klassen mit deren Eigenschaften definiert und Instanzen zu diesem Schema erfasst. Auf einer Metaebene sind Klassen, die Anwender verwenden oder definieren, Instanzen von Metaklassen. Auf diese Weise können die Eigenschaften einer Klasse (und nicht nur die der Instanzen) bestimmt werden. Dadurch ist es möglich, das Metamodell für die Entwicklung von Klassen-Schemata zu verändern. So kann beispielsweise definiert werden, dass für bestimmte Klassen Beschreibungen in verschiedenen Sprachen angegeben werden können, indem eine Metaklasse definiert wird, die Template-Slots für jede unterstützte Sprache enthält.

Mit Ausnahme von Instanzen existieren für jede der beschriebenen Elemente des Frame-Datenmodells Metaklassen. Während Klassen ein wesentliches Element für die Erstellung von Domänenmodellen darstellen, erlauben es Metaklassen Metadaten zur Modellierung selbst vorzugeben. Im Rahmen dieser Arbeit wurde der Mechanismus genutzt, um unterschiedliche Speichertechniken für Instanzen von Klassen zu unterstützen. So wurden spezielle Klassen eingeführt, die auf Tabellen von relationalen Datenbanken abgebildet werden.

Metaklasse Standard-Class

Metaklassen für die Klassen geben vor, welche Eigenschaften Klassen bei der Modellierung bekommen können, wie z.B. die Rolle der Klasse (abstrakt/konkret). Im Rahmen der vorliegenden Arbeit wurden eine Metaklasse (als Unterklasse von „Standard-Class“) eingeführt, die es erlaubt, Instanzdaten zu bestimmten Klassen in dafür vorgesehene Tabellen einer relationalen Datenbank zu speichern.

Folgende Eigenschaften sind für jede Klasse (als Template-Slot in der Metaklasse) fest vorgegeben und können nicht modifiziert oder gelöscht werden:

- § „Direct-Instances“ – enthält alle direkten Instanzen der Klasse;
- § „Direct Subclasses“ – enthält alle direkten Unterklassen der Klasse (dieser Slot wird für Klassen automatisch belegt);
- § „Direct Superclasses“ – enthält die direkten Oberklassen (wird durch den Anwender festgelegt);
- § „Direct Template-Slots“ – enthält die Template-Slots einer Klasse, die die Slots von Instanzen der Klasse definieren;
- § „Direct-Type“ – enthält den Typen, also die Metaklasse;
- § „Documentation“ – enthält einen oder mehrere freie Texte als Dokumentation;
- § „Name“ – enthält den Namen einer Klasse;
- § „Role“ – legt die Rolle/Art der Klasse fest (abstrakt oder konkret);
- § „Slot-Constraints“ – erlaubt die axiomatische Definition von Einschränkungen („constraints“).

Metaklasse Standard-Slot

Metaklassen für Slots geben Eigenschaften von (Template-)Slots vor, wie etwa den Grundbereich und den Wertebereich („domain“ und „range“) sowie die Kardinalität. Im Rahmen dieser Arbeit wurde eine Metaklasse für Slots erstellt, die zu den oben beschriebenen Datenbank-Metaklassen gehören. Diese erlaubt es, Metadaten für das Speichern von Werten in Tabellenspalten vorzugeben.

Folgende Eigenschaften sind für jeden Slot vorgegeben und können nicht modifiziert oder gelöscht werden:

- § „Associated-Facet“ – gibt das Facet vor, mit dem der Slot verbunden ist, wenn der Slot dazu dienen soll, Eigenschaften für andere Slots zu definieren (d.h. wenn dieser Slot die Vorlage für ein Facet ist⁵); diese Eigenschaft wird nicht durch Anwender festgelegt;
- § „Direct-Domain“ – enthält den Grundbereich des Slots, also die Klassen, die den Slot „besitzen“ können;
- § „Direct-Subslots“ – enthält spezialisierte bzw. dem betreffenden Slot abgeleitete Slots⁶;
- § „Direct-Superslots“ – analog zu „Direct-Subslots“;
- § „Direct-Type“ – Klasse des Slots (entspricht also i.d.R. der Klasse „Standard-Slot“);
- § „Documentation“ – Beschreibung des Slots;
- § „Name“ – Name des Slots;
- § „Slot-Constraints“ – hier können axiomatisch Einschränkungen definiert werden, die über einfache Einschränkungen wie Kardinalitäten hinausgehen;
- § „Slot-Defaults“ – hier können Standardwerte eingegeben werden; bevor ein Anwender Werte explizit einträgt, ist der Slot mit den Standardwerten belegt;
- § „Slot-Inverse“ – es kann ein inverser Slot angegeben werden, dessen Wert automatisch belegt ist (z.B. „hat Bauteil“ als inverser Slot zu „gehört zu Ausrüstungsgegenstand“);
- § „Slot-Maximum-Cardinality“ – hier wird die Obergrenze für die Kardinalität festgelegt, d.h. die maximale Anzahl von Belegungen für den Slot pro Instanz;
- § „Slot-Minimum-Cardinality“ – die Untergrenze entsprechend zu „Slot-Maximum-Cardinality“;
- § „Slot-Numeric-Maximum“ – dies betrifft das numerische Maximum für Slots mit einem numerischen Datentyp („Integer“ oder „Float“);
- § „Slot-Numeric-Minimum“ – entsprechend zu „Slot-Numeric-Maximum“;
- § „Slot-Value-Type“ – betrifft den Datentyp des Slots (Zeichenkette, ganze Zahl, etc.);
- § „Slot-Values“ – hier werden die eigentlichen Werte zugeordnet.

⁵ Jedes Facet ist mit einem Slot assoziiert, welcher einen Teil der Eigenschaften des Facets bestimmt.

⁶ Slots lassen sich hierarchisch aufbauen, d.h. ein generischer Slot „ist Teil von“ (für den Grundbereich und den Wertebereich „Gegenstand“) ließe sich spezialisieren durch einen Slot „ist Bauteil von“ (für den Grundbereich und den Wertebereich „Ausrüstungsgegenstand“). Immer wenn gilt „A ist Bauteil von B“, gilt auch „A ist Teil von B“.

Metaklasse Standard-Facet

Metaklassen für Facets bestimmen die Eigenschaften von Facets. Folgende Eigenschaften sind fest für jedes Facet definiert:

- § „Associated-Slot“ – hier wird der Slot angegeben, der zu diesem Facet gehört (siehe „Standard-Slot“);
- § „Direct-Type“ – Klasse des Facets (entspricht also i.d.R. „Standard-Facet“);
- § „Documentation“ – Beschreibung des Facets;
- § „Name“ – Name des Facets.

Protégé enthält die oben aufgelisteten Standard-Metaklassen für Klassen, Slots und Facets. Diese werden in der vordefinierten Konfiguration des Werkzeuges verwendet. Sie können durch eigene Metaklassen erweitert werden. Im Rahmen dieser Arbeit wurden die Metaklassen „Standard-Slot“ und „Standard-Class“ erweitert, wie oben beschrieben.

Das Datenmodell von Protégé ist quasi „selbstreferentiell“, da die Werkzeuge zum Erstellen von Modellen (Ontologien) durch Modelle angepasst werden können. Dadurch sind verschiedene Rollen für die Nutzung und Wartung bzw. Administration des Werkzeuges möglich. Für die in dieser Arbeit entwickelten Erweiterungen ist die Rolle des Administrators der Metadaten von Modellelementen wichtig. In Tabelle 4 sind die unterschiedlichen Rollen mit entsprechenden Beispielen beschreiben.

Tab. 4: Anwender-Rollen für Protégé bzw. Protégé-basierte Werkzeuge

	Aufgabe	Verwendete Schnittstellen
Knowledge-Engineer	§ erstellt und wartet Ontologien in Zusammenarbeit mit Domänenexperten;	§ Kernkomponenten von Protégé für die Modellierung: Navigations-Komponenten, Klasseneditoren, etc.
	§ erstellt und verwaltet Klassen, Slots, Instanzen	§ Spezialkomponenten für den Bereich der Prozesswissenschaften
Administrator von Ontologien und Metadaten	§ verwaltet systemspezifische Metadaten zu Modellelementen; § im SCIFORG-System: Metadaten, die die Speicherung von Ontologiedaten betreffen; Datenbankmetadaten	§ Protégé Klasseneditoren für Metaklassen; § Spezialkomponenten für die Verwaltung von Datenbank-Metadaten, die Verwaltung von physikalischen Einheiten, etc.
Endanwender	§ erfasst Daten auf der Grundlage des gegebenen Modells; § nutzt das System zur Verwaltung von Messdaten, Simulationsdaten, semantische Metadaten, etc.; § nutzt das System zur Recherche	§ Protégé Instanzeditoren; § Spezialkomponenten für das Erfassen und Bearbeitung von domänenspezifischen Konzepten, wie mathematischen Formeln; § Graphische Komponenten für das Erfassen und Bearbeiten von regelhaften Zusammenhängen und Suchanfragen

5.2.1.3 Konfigurierbare Formular-basierte Editoren

Eine Kernkomponente von Protégé ist der Formular-Generator für dynamisch konfigurierbare Anwender-Schnittstellen. Dieser erzeugt aus einem Datenmodell ein Eingabeformular für Instanzdaten. Für jede Klasse wird eine Formulardefinition erzeugt, die über eine graphische Schnittstelle verändert werden kann. Dabei wird jedem Slot, d.h. jeder Eigenschaftskategorie, ein so genanntes „Widget“ zugeordnet. Ein Widget kann etwa ein Textfeld sein, in dem einfache Texte als Eigenschaftswerte erfasst werden. Die Standard-Widgets können über Plugins ausgetauscht werden (siehe folgender Abschnitt).

Anwender können für jede Klasse die Art des Widgets für jeden Slot sowie die Anordnung auf dem Formular bestimmen und damit die Anwenderschnittstelle für die Wissensakquisition anpassen. Dabei ist die Auswahl eines Widgets abhängig vom Typ des betreffenden Slots. Für einen Slot, dessen Wertebereich Zahlen sind, können nur entsprechende Widgets verwendet werden. Auch die Kardinalität wird berücksichtigt, d.h. für Slots, für die mehrere Werte erfasst werden können, sind etwa Listen-basierte Widgets vorhanden.

Dadurch, dass das Frame-Modell von Protégé Metaklassen unterstützt, bestimmt der Formular-Generator auch das Aussehen von Eingabefeldern für Klassen. D.h. der Formular-Generator (und –Konfigurator) kann als Werkzeug zur Erzeugung von Wissensakquisitionsschnittstellen für

Anwender auf der Instanzebene verwendet werden, kann aber auch zur Konfiguration der Oberfläche für die Bearbeitung auf der Schemaebene eingesetzt werden.

Das zugrunde liegende Formular-Modell unterliegt allerdings Einschränkungen. So handelt es sich um einfache Formulare, die z.B. nicht über Registerkarten oder „Master-Detail“-Strukturen unterteilt werden können. Bei Klassen mit vielen Slots können dadurch relativ unübersichtliche Formulare entstehen.

5.2.1.4 Plugin Architektur

Protégé unterstützt die Entwicklung von Erweiterungen über sog. „Plugins“. Plugins können zur Anpassung der graphischen Oberfläche oder als Erweiterung der Speicherschicht entwickelt werden. Protégé sieht standardisierte Kategorien für Plugins mit entsprechenden Standard-Schnittstellen (Java-Interfaces) vor. Zu diesen Kategorien gehören:

§ Slot-Widget Plugins

Mit Hilfe von Slot-Widget Plugins kann die Menge der verfügbaren Widgets für Formulare erweitert werden.

§ Tab-Widget Plugins

Tab-Widget Plugins erlauben „flächige“ Erweiterungen der graphischen Oberfläche von Protégé ohne große Einschränkungen.

§ Persistenz („Backend“)-Plugins

Über Backend-Plugins lassen sich alternative bzw. zusätzliche Persistenzmodelle, d.h. Speicherungs-Techniken für Protégé Ontologien entwickeln.

Im Rahmen dieser Arbeit wurden alle genannten Kategorien von Plugins entwickelt. Slot-Widget Plugins wurden entwickelt, um das Erfassen von domänenspezifischen Konzepten wie mathematischen Gleichungen einfacher zu gestalten bzw. zu ermöglichen. Als Tab-Widget Plugin wurde eine Anwenderschnittstelle zum Erstellen von Regeln und Suchanfragen erstellt. Eine Datenbankschnittstelle, die das skalierbare und performante Verwalten datenlastiger Modelle erlaubt, wurde als Backend-Plugin entwickelt. Letztere behebt wesentliche Schwachstellen der Standard-Datenbank-Schnittstelle von Protégé (siehe auch Abschnitt 5.2.1.6).

5.2.1.5 Programmierschnittstelle (API)

Protégé verfügt über ein API, das es Entwicklern von Plugins erlaubt, lesend und schreibend auf im System verwaltete Ontologien zuzugreifen. Die wesentlichen Elemente der API zum Zugriff auf das Datenmodell (z.B. zum Hinzufügen von Klassen) sind in Abbildung 26 dargestellt.

Darüber hinaus existieren jeweils Schnittstellen für die Implementierung von Plugins entsprechend der oben genannten Kategorien. Diese werden bei der Beschreibung der entwickelten

Komponenten entlang der Erweiterungen im Rahmen dieser Arbeit diskutiert, ohne diese Schnittstellen allgemein zu erörtern.

5.2.1.6 Schwachstellen des Systems

Das Kernsystem von Protégé (bis Version 3.2) hat einige Schwachstellen, von denen vor allem die Wesentlichen für den hier vorgesehenen Anwendungsbereich im Folgenden kurz diskutiert werden. Einige der Schwachstellen konnten durch die entwickelten Erweiterungen behoben werden, wie noch deutlich werden wird.

Performanz der existierenden Datenbankschnittstelle

Protégé verfügt über eine Standard-Datenbankschnittstelle, die als Alternative zur Datei-basierten Speicherung und Verwaltung von Ontologien verwendet werden kann. Diese Datenbankschnittstelle fußt auf einem einfachen Schema mit nur einer Tabelle. Die Tabelle speichert im Wesentlichen Tripel in der Form Objekt-Attribut-Wert mit wenigen Metadaten. Sie enthält die folgenden Spalten:

- § „frame“
- § „frame-type“
- § „slot“
- § „facet“
- § „is-template“
- § „value-index“
- § „value-type“
- § „short-value“
- § „long-value“

Die Spalte „frame“ enthält den Bezeichner des Objektes und ist damit immer belegt. Attribut und Wert sind über die Spalten „slot“ und „short-value“ bzw. „long-value“. Letztere Spalte dient der Speicherung langer Zeichenketten. Short-value enthält jede mögliche Art von Wert für einen Slot (Zahlen, Zeichenketten, Bezeichner von anderen Frames).

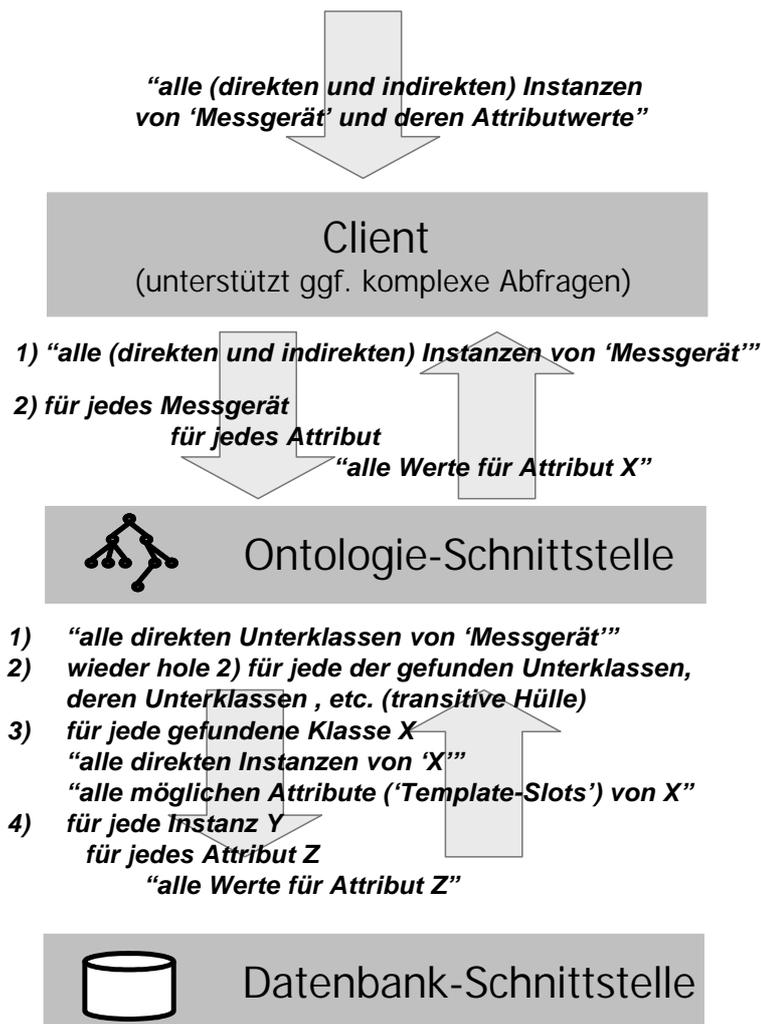


Abb. 27: Umsetzung komplexerer Anfragen an das Datenmodell mit Datenbank-Schnittstelle

Das Persistenzmodell ist generisch und doch sehr flexibel. Der Nachteil liegt in der mangelnden Effizienz für bestimmte Anfragemuster. So resultiert eine Anfrage an die Ontologie-Schnittstelle „liefere alle direkten und indirekten Instanzen von Klasse X mit deren Attributwerten“ in einer großen Anzahl von Datenbank-Abfragen. Dies ist beispielhaft in Abbildung 27 illustriert.

Tatsächlich existiert kein echtes Abfragemodell, das Einzel-Abfragen mit der Mächtigkeit des genannten Beispiels unterstützt. Die vorhandene Schnittstelle des Datenmodells erlaubt nur sehr einfache Anfragemuster. D.h. das anfragende Programm muss komplexere Anfragen selber abbilden, indem es sie auf mehrere Anfragen abbildet und über Zwischenergebnisse iteriert.

Die Performanz des Datenbank-Persistenzmodells liegt dadurch weit unter der des Standard-Persistenzmodells, wo die geladene Ontologie zu jeder Zeit vollständig im Hauptspeicher gehalten wird.

Die eingeschränkte Performanz ist auch darauf zurückzuführen, dass keine Anfrage-Optimierung stattfindet. Komplexere Anfragen auf Ontologie-Ebene werden immer in atomare Anfragen auf Datenbankebene aufgelöst. So werden für die Standard-Schnittstelle für Suchen beispielsweise die Vergleichs-Operatoren auf der Ebene der Objekte verwendet und nicht die Möglichkeiten der darunter liegenden Datenbank ausgereizt. Für sehr große Datenmengen bedeutet dies, dass eine große Menge von Einträgen in der Datenbank erst aus dieser gelesen und anschließend in Objekte des Datenmodells transformiert werden müssen, bevor die Operatoren angewendet werden können.

Gänzlich ungeeignet ist das Modell für sehr große Instanzmengen bei einer großen Zahl von Attributen, da sich die Anzahl der Einträge in der Datenbank in diesem Fall aus dem Produkt von Attributzahl und Instanzzahl ergibt. Abhilfe schafft an dieser Stelle die objektrelationale Abbildung des SCIFORG-Systems.

Numerische „Identifizier“

Jedes Modellelement bzw. jeder Frame in Protégé verfügt über einen eindeutigen Bezeichner. Über diesen Bezeichner kann auf Frames und deren Eigenschaften zugegriffen werden. Der Bezeichner ist eine ganzzahlige Nummer. End-Anwender sehen diesen Bezeichner üblicherweise nicht, sondern interagieren mit dem System auf der Grundlage des Namens eines Frames. Dieser ist ebenfalls eindeutig, d.h. jeder Name darf nur einmal verwendet werden. Der Name wird jedoch intern nicht nur zur Identifizierung oder für Verweise verwendet, sondern auf den Bezeichner abgebildet. Dies hat den Vorteil, dass bestimmte Operationen wie etwa das Umbenennen eines Frames intern relativ einfach umzusetzen sind. So wird in dem genannten Falle nicht der Bezeichner, sondern nur der Name des Frames geändert und Verweise (z.B. durch die Klassenhierarchie oder durch Slot-Belegungen) sind unberührt von der Operation. Die Operation „Umbenennen“ kann also auf eine einzige interne Operation abgebildet werden.

Nachteilig an diesem Konzept ist, dass alternative Persistenzmodelle einen Mechanismus für das Erzeugen und Verwenden von Bezeichnern bereitstellen müssen, der in eine ganze Zahl sämtliche benötigte Informationen kodiert. Dies kann aus verschiedenen Gründen problematisch sein. So sieht Protégé die parallele Verwendung von verschiedenen Persistenzmodellen über sog. „FrameStores“ vor. Ein FrameStore stellt Methoden zum Erzeugen und Verwalten von Frames sowie deren Eigenschaften vor. Wenn mehrere FrameStores gleichzeitig verwendet werden, muss bei einer Anfrage (z.B. „gib alle Oberklassen der Klasse ‚Modell‘ aus“) ermittelbar sein, welcher FrameStore diese Anfrage bedienen kann. Bei einem Bezeichner auf der Basis von Zeichenketten wäre es verhältnismäßig einfach, solche Informationen (wie die „physische Herkunft“ des Frames) zu kodieren.

Im Rahmen dieser Arbeit wurde ein Persistenzmodell entwickelt, das ein dynamisches Datenbankschema verwendet. Die Verwendung ganzzahliger Bezeichner stellte hierfür aus den oben genannten Gründen eine besondere Schwierigkeit dar. Hier wurde ein Ansatz gewählt, der (i) einen bestimmten Bereich von Bezeichnern für solche Frames reserviert, die nicht dem Protégé-Standard-Persistenzmodell entsprechend verwaltet werden und (ii) das Protégé-Standard-

Persistenzmodell für Frames auf der Schema-Ebene verwendet. Dieses Modell ist in Abschnitt 5.3.1 näher erläutert.

Eingeschränkte Inferenz-Unterstützung

Protégé bietet nur eingeschränkt Unterstützung durch Inferenz-Verfahren. Einfache Inferenzen, wie etwa die Unterstützung der Vererbung von Attributen (transitive Hülle über die Klassenhierarchie) werden durch die zentrale Ontologie-Schnittstelle unterstützt. Darüber hinaus können Ontologien entlang von Axiomen, die mit Hilfe der „Protégé Axiom Language“ (PAL) definiert wurden, auf Konsistenz überprüft werden. PAL ist eine eigene Sprache mit eigener Syntax, die für Endanwender relativ schwer anzuwenden ist. Graphische Editoren, die die Komplexität der Sprache kapseln, sind nicht verfügbar. Dadurch stellt PAL einen Bruch innerhalb des Konzeptes von Protégé dar, das dem Anwender sonst ausschließlich graphische Schnittstellen (Formulare, Baumansichten, Graphen) präsentiert.

Was die Mächtigkeit von PAL betrifft, so fehlt insbesondere die Möglichkeit, neue Informationen -über die Verletzung von Axiomen hinausgehend- abzuleiten. Diese Möglichkeit wird u.a. durch regelbasierte Systeme als Erweiterung zum Kernsystem geboten. Problematisch ist dabei, dass (vereinfachend ausgedrückt) diese Erweiterungen außerhalb des Datenmodells von Protégé ansetzen, also von der Verwaltung von Fakten „abgeschirmt“ sind. Damit ist Protégé zusammen mit regelbasierten Erweiterungen kein homogenes System, was sich ohne tief greifende Änderungen insgesamt optimieren lässt.

Eine mögliche (und bei der Erstellung und Bearbeitung von OWL-Ontologien übliche) Strategie besteht darin, eine externe Inferenzmaschine zu verwenden, die über ihre eigene Datenverwaltung verfügt. Ontologien müssen dann teilweise oder komplett in das externe System überführt werden, bevor dieses genutzt werden kann. Auf der Basis dieses Ansatzes lässt sich kein integriertes System anbieten, wie es in dieser Arbeit erstellt wurde.

5.2.2 SimpleORM

SimpleORM⁷ ist eine Schnittstelle für den Zugriff auf relationale Datenbanken. Die Bibliothek wurde verwendet, um das Datenbank-basierte Persistenzmodell umzusetzen. ORM steht für „Object-Relational-Mapping“, ein gängiges Konzept, bei dem objektorientierte Datenmodelle auf relationale Datenmodelle abgebildet werden. Das Prinzip des objekt-relationalen Mappings ist u.a. bei Balzert (1999) beschrieben und wird in Abschnitt 5.3.1 näher beschrieben.

SimpleORM erlaubt es, Java-Klassen auf Tabellen einer Datenbank abzubilden und dabei von Hersteller-spezifischen Merkmalen des Datenbankzugriffs zu abstrahieren. Die Bibliothek trennt zwischen generischen, für jede (unterstützte) Datenbank gültigen Funktionalitäten und spezifischen Implementierungen, die in sog. „Treiber“ ausgelagert sind. Der Wechsel zwischen ver-

⁷ SimpleORM ist unter <http://www.simpleorm.org> verfügbar (Stand April 2007). Die Webseite enthält eine kurze Dokumentation und Verweise auf weitere Webseiten. Publikationen sind nicht verfügbar.

schiedenen Datenbanken wird damit deutlich vereinfacht. Dabei stellt SimpleORM eine sehr „schlanke“ Schicht oberhalb des eigentlichen Datenbankzugriffs dar. Es handelt sich um eine vergleichsweise kleine Bibliothek.

SimpleORM eignet sich als Grundlage für ein Persistenzmodell für Protégé, da es den Datenbankzugriff sowie die Unterstützung verschiedener Systeme vereinfacht und dabei einen sehr geringen Konfigurationsaufwand erfordert. Darüber hinaus enthält es neben den benötigten Funktionalitäten kaum zusätzliche Datenstrukturen.

Im Rahmen dieser Arbeit wurde SimpleORM durch eine Unterstützung von dynamischen Tabellenschemata ergänzt. D.h. das Tabellenschema muss nicht im Vornherein bekannt sein und kann sich zur Laufzeit ändern.

5.2.3 Algernon

Die Inferenzmaschine Algernon geht auf Crawford und Kuipers (1991) zurück. Algernon basiert auf dem Prinzip der „Access Limited Logic“ („zugriffs-beschränkte Logik“), kurz ALL. Dieses Prinzip ermöglicht den Autoren nach nur eine eingeschränkte Vollständigkeit und dafür eine vergleichsweise hohe Effizienz. ALL ist laut Crawford und Kuipers (1991) „socratically complete“, d.h. bestimmte logisch aus der Wissensbasis folgende Fakten lassen sich nur unter bestimmten Voraussetzungen, unter Zuhilfenahme zusätzlicher Suchanfragen, vom System ableiten. So wird zwischen Vorwärts- und Rückwärtsverkettung nicht allein auf der Ebene des Systems (etwa durch Verwendung bestimmter Einstellungen), sondern auf der Ebene der Sprache unterschieden. Die Kombination voneinander abhängiger vorwärts- und rückwärts-verkettender Regeln kann dazu führen, dass nicht alle Anfragen vollständig beantwortet werden können. Dies kann z.B. durch die ausschließliche Verwendung rückwärts-verkettender Regeln vermieden werden.

Damit muss die Eignung von Algernon im Prinzip für jede Domäne neu nachgewiesen werden. In dieser Arbeit wurde auf einen solchen Nachweis verzichtet, da es den Rahmen der Untersuchung gesprengt hätte.

Die aktuelle Version von Algernon ist vollständig kompatibel mit dem Protégé Frames-Datenmodell und bietet eine Lisp-artige Syntax für Regeln und Anfragen. Das System unterstützt, wie bereits beschrieben, vorwärts- sowie rückwärts-verkettende Schlussfolgerungen. Der Bedingungsteil der Regel muss immer gültige Pfadausdrücke enthalten. Ein gültiger Pfadausdruck in Algernon erfordert, dass jede an erster Stelle in einem Prädikat auftretende Variable bereits zuvor in einem Prädikat aufgetreten ist. Prädikate repräsentieren Frames oder Slots (siehe auch Abschnitt 5.2.1.2 „Protégé Datenmodell“ auf Seite 97).

5.2.4 JGraph

JGraph (vgl. Bagga und Heinz, 2002) ist eine Bibliothek zur Entwicklung von Anwenderschnittstellen auf der Basis von Graphen-Darstellungen. Es handelt sich um eine Java-Implementierung,

die es erlaubt, Datenmodelle über eine standardisierte Schnittstelle anzubinden und darüber Visualisierungen oder graphische Editoren zu realisieren.

JGraph wurde eingesetzt, um eine graphische Oberfläche für Anwender zu implementieren, die es ermöglicht, Wissensbasen zu visualisieren, regelhafte Zusammenhänge zu formulieren (basierend auf Algernon; siehe oben) und Suchanfragen zu definieren.

Ergänzt wurde JGraph in dieser Arbeit durch die Möglichkeit, Symbole für Graphendarstellungen dynamisch (zur Laufzeit des Systems) zu definieren und zu verwenden. Diese können mit der Wissensbasis verknüpft werden und deren Inhalte darstellen.

5.2.5 Weitere Bibliotheken/Programme

Das SCIFORG-System verwendet eine Reihe weiterer Bibliotheken, von denen die wichtigsten in diesem Abschnitt vorgestellt werden.

Majix

Majix ist ein Java-basiertes Programm zur Konvertierung von Dateien im Richtext-Format (RTF) in XML-Dokumenten. Majix erlaubt Anwendern, Formatierungselemente auf die Struktur des Zieldokuments abzubilden.

JChemPaint

JChemPaint erlaubt das Erstellen von chemischen Formeln mit Hilfe einer graphischen Oberfläche (vgl. Krause et al., 2000). JChemPaint unterstützt die Chemical Markup Language (siehe Abschnitt 4.2.1.1 auf Seite 68) als Speicherformat. Das Programm bietet einen zweidimensionalen Struktureditor für chemische Formeln. Der Anwender wird durch eine Reihe von Funktionalitäten unterstützt, wie z.B. Muster für aromatische Verbindungen oder dem einfachen editieren von Isotopen.

JOME

JOME (Java OpenMath Editor) ist ein Java-Programm zur Erstellung und Bearbeitung mathematischer Gleichungen im OpenMath-Format bzw. (mit Einschränkungen) im MathML-Format (vgl. Dirat, 2000). JOME wurde verwendet (und ergänzt), um Anwendern das Erfassen von mathematischen Gleichungen zu erlauben.

JFreeChart

JFreeChart⁸ ist eine in Java implementierte Bibliothek, die das Erstellen von Diagrammen in eigenen Applikationen ermöglicht. JFreeChart unterstützt verschiedene Arten von Diagrammen, wie z.B. Balkendiagramme, Liniendiagramme oder Kreisdiagramme.

⁸ JFreeChart ist unter <http://www.jfreechart.org> verfügbar (Stand April 2007).

5.3 Aufbau von SCIFORG mit Protégé 2000 als Kernplattform

Einen Überblick über das SCIFORG-System bietet Abbildung 25. Protégé bildet den Kern des Systems. Mit Hilfe der erwähnten Bibliotheken wurden zusätzliche Komponenten erstellt, die die für den Bereich der Prozesswissenschaften spezialisierten Funktionalitäten und Schnittstellen (Oberflächen) bereitstellen.

Die Erweiterungen betreffen drei verschiedene Ebenen:

- § Die Verwaltung der Ontologie-Daten (Persistenzmodell);
- § Metadaten von Slots und Klassen;
- § Oberflächenkomponenten.

In den folgenden Abschnitten werden die Komponenten näher beschrieben. Anschließend wird der Einsatz des Systems (Lösungsansätze für beschriebene Probleme, Anwendungsszenarien) erörtert.



Abb. 28: Haupt-Symbolleiste

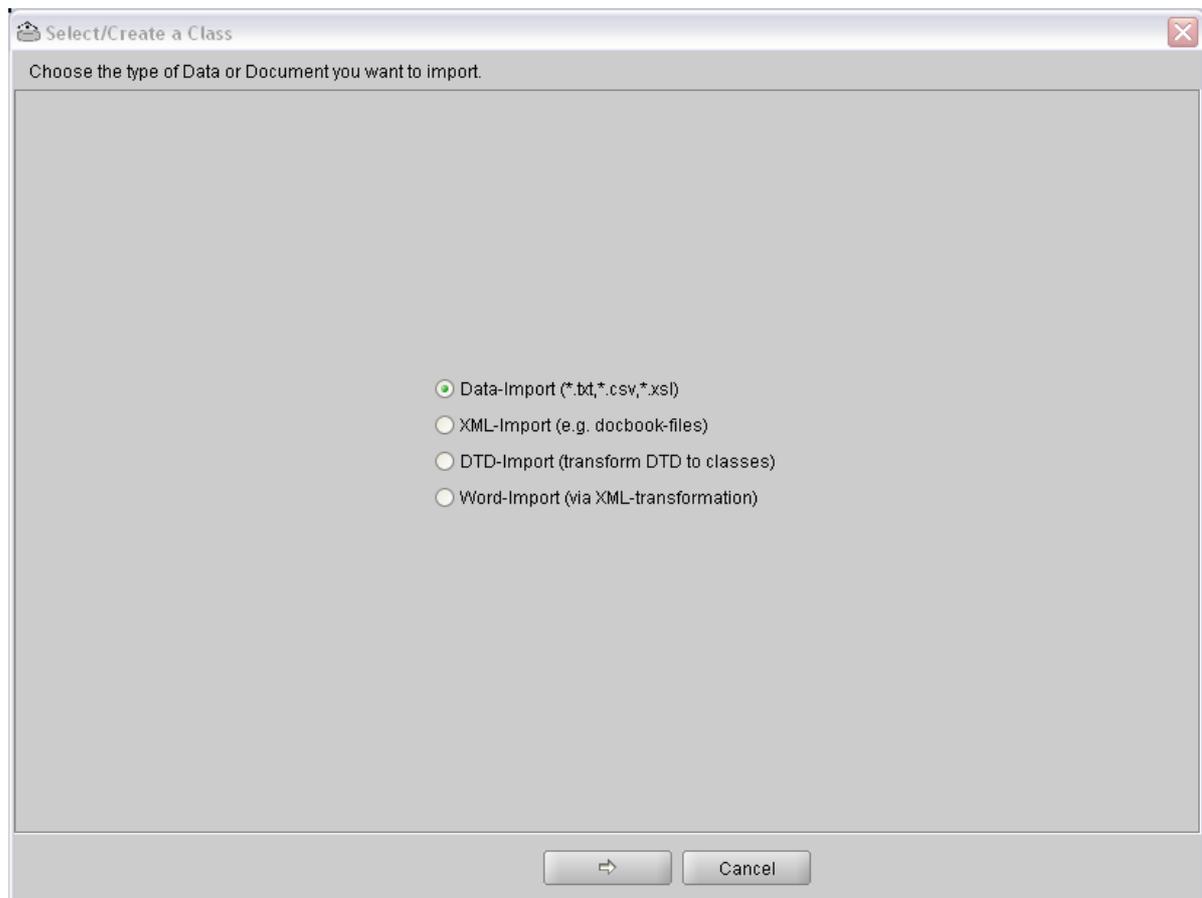


Abb. 29: Import-Wizard

5.3.1 Skalierbare objektrelationale Speicherschicht

Die Grundlage für die Datenspeicherung in SCIFORG ist die objektrelationale Speicherung von Frames (das Persistenzmodell). Dieses wird parallel zum Standard-Datenbank-Persistenzmodell von Protégé verwendet. Sie stellt einen sog. „Framestore“ zur Speicherung von Klassen, Instanzen, etc. zur Verfügung. Protégé kann mehrere Framestores verwalten und kombinieren. Verschiedene Framestores können verkettet werden, indem sie Anfragen selber bedienen und delegieren können. Zusätzlich existieren „Merging“-Framestores, die mehrere Framestores zusammenführen und diese nach außen über die Standard-Schnittstelle als einen Framestore darstellen. Diese Architektur erlaubt es, verschiedene unterschiedliche Speichertechniken zu kombinieren.

Der zusätzliche Framestore zeichnet sich dadurch aus,

- § dass er die Möglichkeiten von relationalen Datenbanken zur Datenverwaltung besser ausnutzt, indem etwa komplexere Suchanfragen auf der Ontologie-Ebene auf komplexe Suchanfragen auf der Datenbank-Ebene (etwa über sog. „Joins“) abbilden kann;

- § dass sehr große Datenmengen durch die Aufteilung auf mehrere Tabellen besser unterstützt werden und damit eine höhere Skalierbarkeit gewährleistet werden kann;
- § dass bestimmte Suchanfragen auf Ontologie-Ebene sehr effizient bedient werden können;
- § dass bestehende Datenbank-Schemata (mit Einschränkungen) abgebildet werden können;

Auf der anderen Seite unterliegt der zusätzliche Datenspeicher einigen Einschränkungen. So kann nicht die volle Ausdrucksmächtigkeit des Frame-Modells von Protégé genutzt werden. Diese Einschränkungen werden später in diesem Abschnitt erörtert.

Anwender können in SCIFORG auf der Ebene von Klassen und Template-Slots entscheiden, welches Persistenzmodell (welcher Framestore) verwendet werden soll. Dabei werden Daten für eine Wertbelegung in der Art „Instanz X hat Wert Y für Eigenschaft Z“ nur dann im zusätzlichen Datenbank-Framestore gespeichert, wenn (i) X eine Instanz ist, (ii) für die Klasse der Instanz die spezielle Speicherung explizit festgelegt wurde, (iii) die Metadaten (Datenbank-Verbindung, Tabellename, etc.) erfasst wurden und die Eigenschaft Z entweder einem einfachen Datentyp (Zeichenkette, etc.) oder einer Instanz entspricht, für die ebenfalls die spezielle Speicherung festgelegt wurde. Dabei ist es durchaus möglich, für eine Klasse prinzipiell den speziellen Framestore zu verwenden, einige Eigenschaften aber über die Standard-Speicherung zu verwalten. Dies ist jedoch nur dann sinnvoll, wenn für diese Eigenschaften keine großen Datenmengen anfallen.

Grundsätzlich können folgende Auswahlkriterien angenommen werden:

- § Der zusätzliche Framestore eignet sich für Klassen,
 - zu denen große Instanzmengen gehören;
 - für die effiziente und skalierbare Zugriffsmechanismen benötigt werden;
 - auf die potentiell andere Applikationen zugreifen, für die mit klassischen Methoden Schnittstellen erstellt werden sollen (Datenbank-Middleware);
 - die typischer Weise für die reine Datenebene wissenschaftlicher Ressource verwendet werden, d.h. etwa die Speicherung von Messdaten;
 - für die das Schema zu dem diese Klassen gehören, nur eingeschränkt verändert wird, etwa durch Hinzufügen zusätzlicher Unterklassen (siehe Abschnitt 5.3.1.2);
- § Der Standard-Framestore eignet sich für Klassen,
 - für die die volle Ausdrucksmächtigkeit des Protégé Frame-Modells benötigt wird;
 - die Teil eines (potentiell) sehr dynamischen Schemas sind, indem Klassen hinzugefügt, gelöscht und in andere Teile des Klassenbaums umgelagert werden;
 - die sowohl die Daten- als auch die Metadaten-Ebene repräsentieren, z.B. zur Darstellung von Zusatzinformationen über Messdaten, berechnete Daten, etc.

Die Vorteile der verschiedenen Framestores lassen sich grundsätzlich verbinden. Dabei sollten jedoch aus Gründen der Effizienz des Datenzugriffs größere Bereiche innerhalb einer Ontologie gewählt werden, die vollständig auf einen Framestore abgebildet werden.

5.3.1.1 Prinzip der Objekt-relationalen Abbildung

Der zusätzliche Framestore nutzt das Prinzip der objekt-relationalen Abbildung (vgl. Balzert, 1999). Dabei werden objekt-orientierte (in diesem Fall frame-basierte) Datenmodelle auf Datenbank-Schemata abgebildet. Datenbanken dienen dabei als Persistenzschicht, d.h. Datensätze der Datenbank werden angelegt und verwaltet, um die Eigenschaften von Objekten zu speichern.

Diese Abbildung ist auf unterschiedliche Art und Weise möglich. So lässt sich beispielsweise das Prinzip der Vererbung durch mehrere Techniken mit Hilfe von Tabellen und Schlüsseln darstellen. Die objekt-relationale Abbildung beinhaltet jedoch immer einen konzeptionellen Bruch, da sich die Datenmodelle von Grund auf unterscheiden. Tabelle 5 listet einige der zentralen Unterschiede auf.

Tab. 5: Zentrale konzeptionelle Unterschiede zwischen relationalen und objektorientierten Datenmodellen

	Objektorientierte/Frame-basierte Datenmodelle	Relationale Datenmodelle
Schlüssel	<ul style="list-style-type: none"> § Das Prinzip von Schlüsseln analog zu Datenbanken existiert nicht; § In einigen Sprachen bzw. Datenmodellen existiert das Prinzip der „unique name assumption“, nach dem ein Bezeichner ein Element eindeutig identifiziert; in Relationen zwischen Elementen spielt der Bezeichner keine Rolle; 	<ul style="list-style-type: none"> § Schlüssel sind ein zentrales Element relationaler Datenmodelle; § Über Schlüssel werden Bezüge zwischen Entitäten modelliert;
Vererbung	<ul style="list-style-type: none"> § Das Prinzip der Vererbung ist ein zentrales Element; § In einigen Sprachen bzw. Datenmodellen können Vererbungshierarchien implizit definiert werden; 	<ul style="list-style-type: none"> § Das Prinzip der Vererbung existiert nicht explizit; § Vererbung kann nachgebildet werden, ist aber nicht eindeutig als solche im Modell ablesbar;
Schema	<ul style="list-style-type: none"> § Schemata (Klassenhierarchien und Eigenschaften von Klassen) haben typischer Weise dynamischen Charakter; § Im Frame-Modell von Protégé lässt sich die Klassen- und Instanzebene mischen, d.h. dass z.B. Klassen mögliche Werte in Slots von Instanzen darstellen; 	<ul style="list-style-type: none"> § Schemata sind meist statisch; § Bei implementierten (auf Datenbanksystemen installierten) Modellen existieren grundsätzlich verschiedene Mechanismen für Schemata (Tabellen) und Instanzen (Datensätze)
Integrität	<ul style="list-style-type: none"> § Die Integrität wird über verschiedene Modellierungselemente wie etwa Kardinalitätsbedingungen modelliert; § Protégé lässt die Verletzung von Integritätsbedingungen z.T. zu, d.h. es können „ungültige“ Ontologien abgespeichert werden 	<ul style="list-style-type: none"> § Die Integrität der Daten kann mit Integritätsbedingungen modelliert werden; § Bei implementierten Modellen wird die Integrität vom System garantiert bzw. erzwungen.
„Lesbarkeit“ / Domänennähe	<ul style="list-style-type: none"> § Objektorientierte und frame-basierte Datenmodelle minimieren den konzeptionellen Bruch zwischen der Sichtweise von Anwendern und Domänenexperten und der Repräsentation von Daten 	<ul style="list-style-type: none"> § Datenbank-Schemata sind nicht als für den Endanwender „lesbare“ Modelle konzipiert; § Typischer Weise ist das Schema durch eine sog. Middleware gekapselt.

Grundsätzlich bestehen vielfältige Möglichkeiten zur Darstellung von objektorientierten Modellen mit Hilfe von Tabellen und Schlüsseln. Die Abbildung ist dabei nie eindeutig, d.h. aus dem Tabellenschema kann nicht auf das objektorientierte bzw. das frame-basierte Modell geschlossen

werden. Das Protégé-interne Schema z.B. illustriert dies. Es kann dabei als eine besondere Form der Abbildung angesehen werden (siehe Abschnitt 5.2.1.6 „Schwachstellen des Systems“ auf Seite 106). Hier wird besonders deutlich, dass die Interpretation des Schemas willkürlich ist und explizit festgelegt werden muss.

Auch bei einer Abbildung, die die für Datenbanksysteme typischen Modellierungs-Ansätze berücksichtigt, ist die eindeutige Interpretation nicht möglich. Dies wird im folgenden Abschnitt deutlich. Dennoch ist das hier gewählte Schema stärker im Einklang mit den üblichen Methoden zur Umsetzung domänen- oder anwendungs-spezifischer Modelle auf relationale Schemata. Dadurch ergibt sich die Möglichkeit, zumindest einen eingeschränkten Teil vorhandener Datenbankschemata in SCIFORG wieder zu verwenden. Darüber hinaus ist die Abbildung so konzipiert, dass der parallele Zugriff auf die Daten durch externe Systeme einfacher so realisiert werden kann, dass die Integrität der Daten sichergestellt ist und die korrekte Darstellung von Datensätzen der Datenbank im Frame-Modell von Protégé gewährleistet ist. Allerdings ist dieses Szenario nur für Teilbereiche von Ontologien realistisch. Grundsätzlich wird im Folgenden davon ausgegangen, dass auf die Daten nur durch das SCIFORG-System (bzw. durch Protégé als Kern desselben) zugegriffen wird.

5.3.1.2 Schema der Objekt-relationalen Abbildung

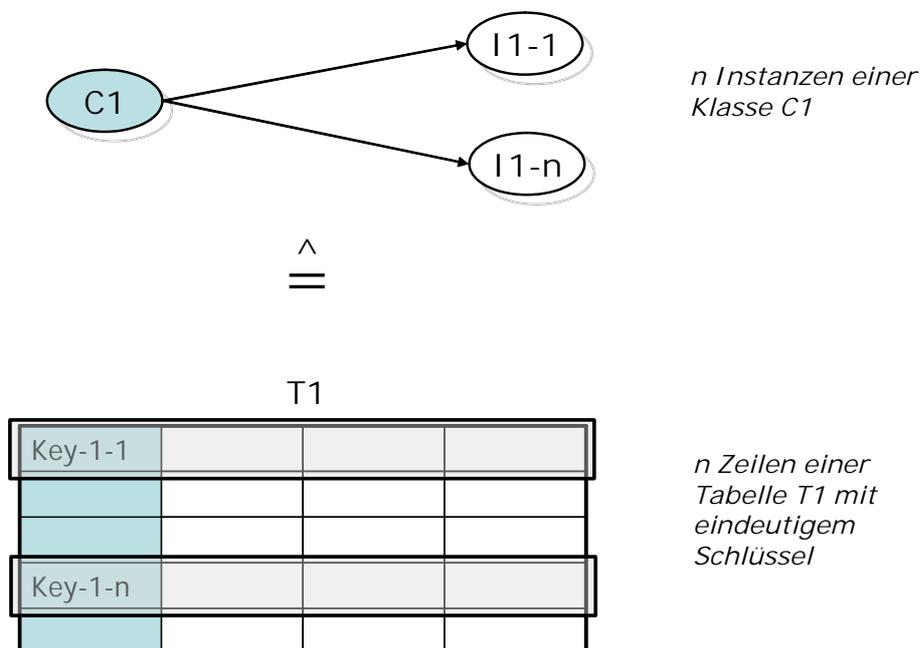


Abb. 30: Abbildung von Klassen auf Tabellen (Entsprechung von Datensätzen und Instanzen)

Bei der hier gewählten Form der objekt-relationalen Abbildung wird jede Klasse der Ontologie mit Hilfe einer Tabelle dargestellt (siehe Abbildung 30). Dabei wird ein spaltenorientierter Ansatz verfolgt. Jede Eigenschaft einer Klasse wird durch eine Spalte repräsentiert. Dies steht im Gegensatz zum zeilenorientierten Ansatz des Standard-Datenbank-Framestores von Protégé.

Diese Form der Abbildung erlaubt einen wesentlich effizienteren Zugriff auf große Instanzmengen mit deren Eigenschaften. Sie hat aber auch den Nachteil, dass die Verwaltung von Klassen wesentlich weniger dynamisch gestaltet werden kann. Dies wird besonders bei der Abbildung von Vererbungshierarchien deutlich (Abbildung 34).

Das hier beschriebene Schema impliziert, dass der beschriebene Datenbank-Framestore nie ausschließlich (ohne gleichzeitige Verwendung des Standard-Framestores) verwendet werden kann. Das liegt an dem Umstand, dass die Information über Instanzen und deren Eigenschaften auf Datensätze in der Datenbank abgebildet werden kann, nicht aber die Information über die Klassen selbst, d.h. welche Klassen existieren, wie ist deren Bezeichner, etc. – kurz: sämtliche Informationen, die aufgrund des verfügbaren Metadaten-Schemas zu einer Klasse erfasst werden können.

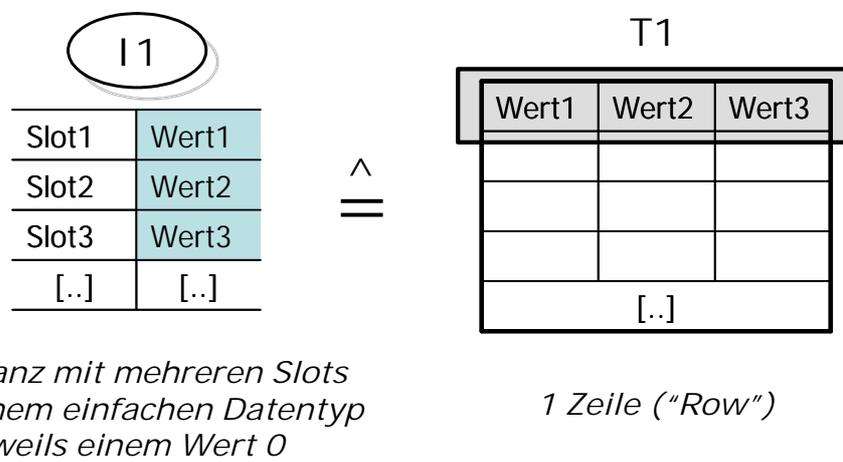


Abb. 31: Abbildung von Slots der Kardinalität eins und einfachen Datentypen als Wertebereich auf Tabellen-Spalten

Abbildung 31 zeigt, wie einfache Datentypen auf Werte in Tabellen abgebildet werden. Dabei werden die in Protégé unterstützten Datentypen auf Datentypen der Datenbank übertragen. Dies macht deutlich, dass der umgekehrte Prozess, die Erzeugung eines Frame-Modells aus einem vorhandenen Datenbankschema, Beschränkungen unterlegen ist, da nicht jeder Datentyp aus Datenbanken einem äquivalenten Datentypen im Protégé-Frame-Modell zugeordnet werden kann.

Eine Sonderstellung nehmen Slots vom Typ „Symbol“ ein (siehe Abbildung 32). Der Wertebereich ist zwar durch einfache Zeichenketten darstellbar, beschränkt sich aber auf eine bestimmte Menge von möglichen Werten. Durch diese Art Slot ließen sich beispielsweise die SI-Einheiten mit den entsprechenden Symbolen „m“, „kg“, etc. modellieren. Um die Bedingungen des Frame-Datenmodells möglichst gut auf Integritätsbedingungen abzubilden, wird eine extra Tabelle verwendet. In dieser Tabelle werden alle möglichen Werte des Wertebereichs verwaltet, wobei jede Zeile einem Symbol entspricht. Der entsprechende Slot wird auf eine Spalte mit einem Fremdschlüssel abgebildet, der auf die Symbol-Tabelle verweist.

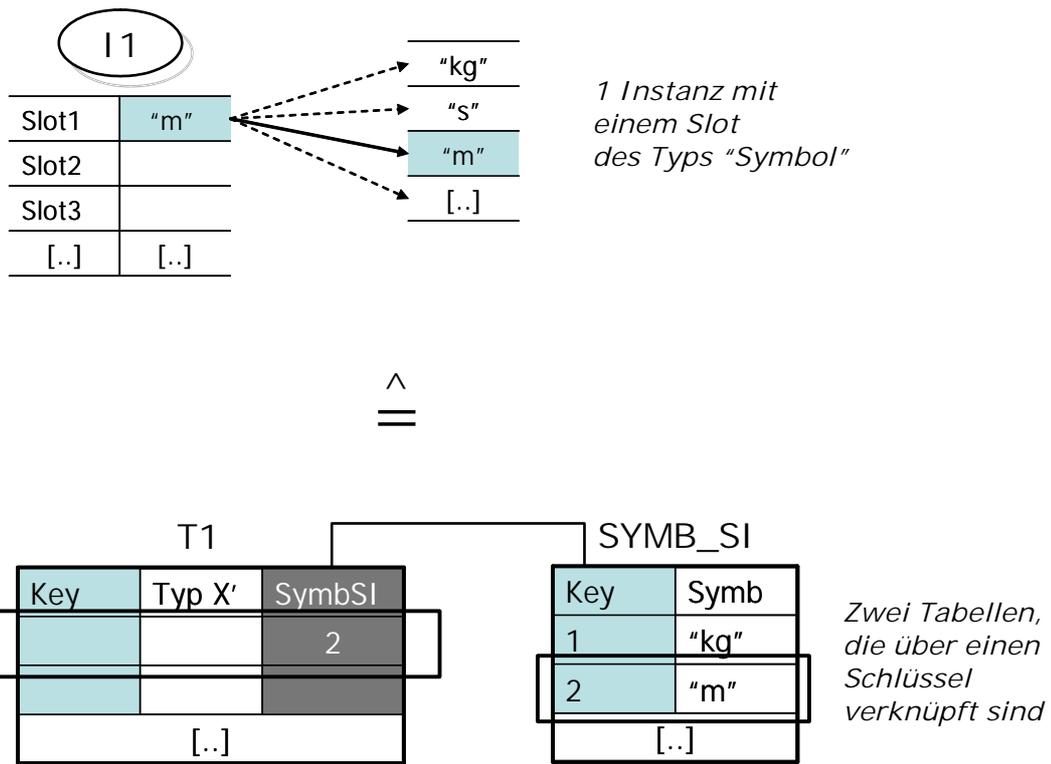


Abb. 32: Abbildung von Slots des Typs „Symbol“ auf eine extra Tabelle

Kardinalitätsbedingungen aus Protégé werden bis zu einem gewissen Grad in der Datenbank reflektiert. Die maximale Kardinalität für diese Art der Abbildung ist eins. Slots mit einer Mindestanzahl von eins werden auf Pflichtfelder (Spalten mit dem Zusatz „NOT NULL“) abgebildet (siehe Abbildung 33).

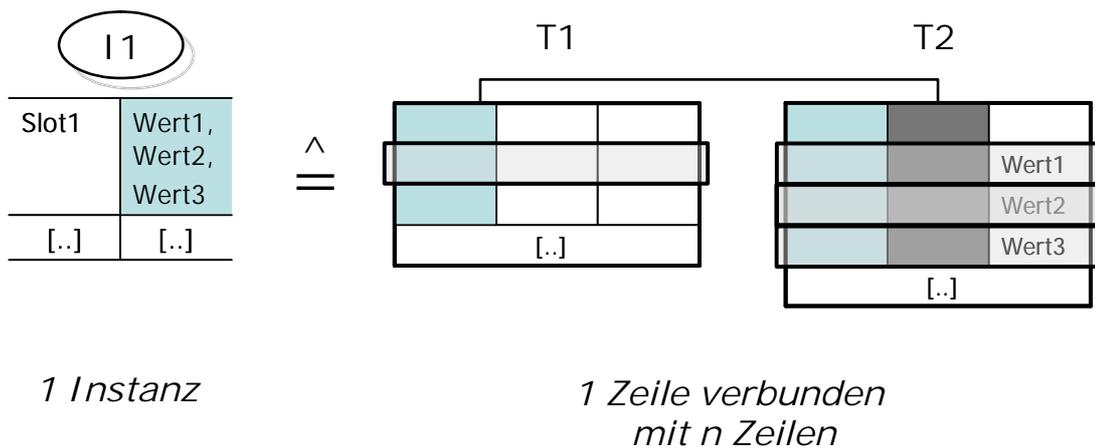


Abb. 33: Abbildung von Slots der Kardinalität n mit einfachen Datentypen als Wertebereich auf eine zusätzliche Tabelle über den Primärschlüssel

Abbildung 34 zeigt für die Darstellung von Vererbungshierarchien, wie die Möglichkeiten zur Definition von referentieller Integrität ausgenutzt werden, um konsistente Daten schon auf der

Datenbank-Ebene zu erzwingen. Eine Klasse C3, die Unterklasse einer Klasse C1 ist, verwendet einen Primärschlüssel, der gleichzeitig Fremdschlüssel ist. Damit kann es keinen Eintrag in der Tabelle T3 geben, ohne dass es einen dazu gehörigen Eintrag in T1 gibt.

Die Vererbung von Attributen ergibt sich aus den Spalten der jeweiligen Tabelle. Das bedeutet, dass für eine neue Klasse C3 eine Tabelle T3 angelegt wird, die ausschließlich die direkten (nicht vererbten) Eigenschaften von C3 repräsentiert. Eine alternative Form der Darstellung von Vererbung in Tabellen besteht darin, in jeder Tabelle sämtliche Eigenschaften einer Klasse abzubilden.

Beide Möglichkeiten haben ihre Vor- und Nachteile. Der alternative Weg bietet eine größere Flexibilität, da die Tabellen grundsätzlich unabhängig voneinander sind. Dagegen ist die Integrität der Daten (im Sinne der Klassenhierarchie) nur mit größerem Aufwand außerhalb des Datenbanksystems herstellbar. Dies ist auf die redundante Darstellung von Attributen zurückzuführen (Attribute werden zwischen Tabellen nicht „vererbt“ und müssen daher für jede Klasse durch entsprechende Spalten repräsentiert werden).

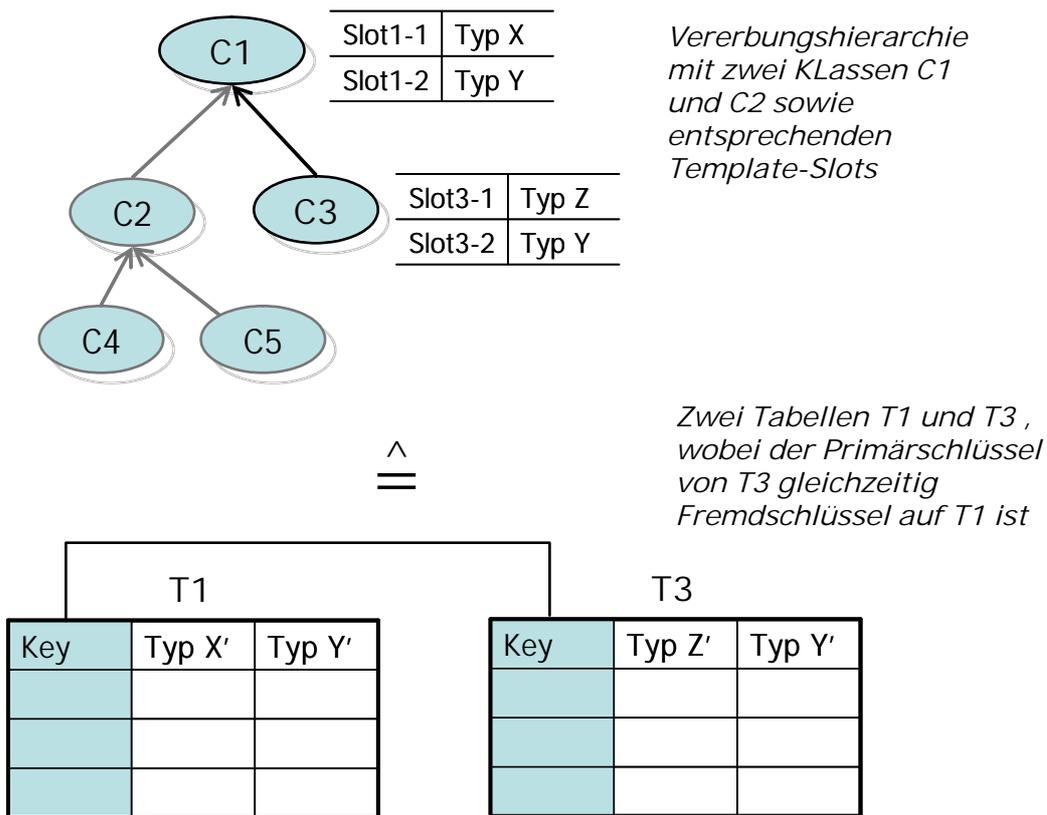


Abb. 34: Abbildung des Vererbungsprinzips mit Hilfe von Primär- und Fremdschlüsseln

Relationen zwischen Instanzen bzw. Slots mit dem Wertebereich „Instanz“ werden mit Hilfe von Integritätsbedingungen zwischen den Tabellen, die die jeweiligen Klassen repräsentieren, dargestellt. Abbildung 35 illustriert dies für Slots mit dem Wertebereich Instanz und einer Maximal-Kardinalität von eins. In der Tabelle, die den Grundbereich des Slots (also die „besitzende Klas-

se“) repräsentiert, wird eine Spalte mit einem Fremdschlüssel angelegt, die auf die Tabelle, die den Wertebereich des Slots repräsentiert, verweist.

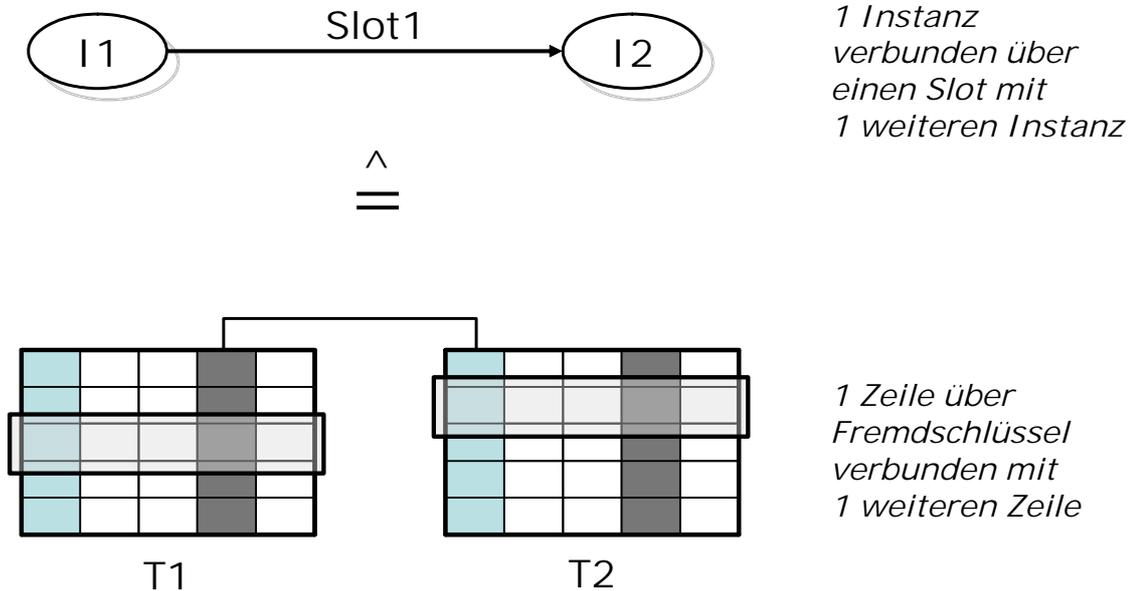


Abb. 35: Abbildung von Slots der Kardinalität eins und dem Wertebereich „Instanz“ über (einen oder mehrere) Fremdschlüssel

Für Slots mit einer Maximal-Kardinalität größer als eins muss eine separate Tabelle angelegt werden. Dies ist in Abbildung 36 dargestellt. Hier handelt es sich um eine klassische M:N-Relation, wie sie im Bereich von Datenbanksystemen üblich ist (vgl. Heuer, 2000).

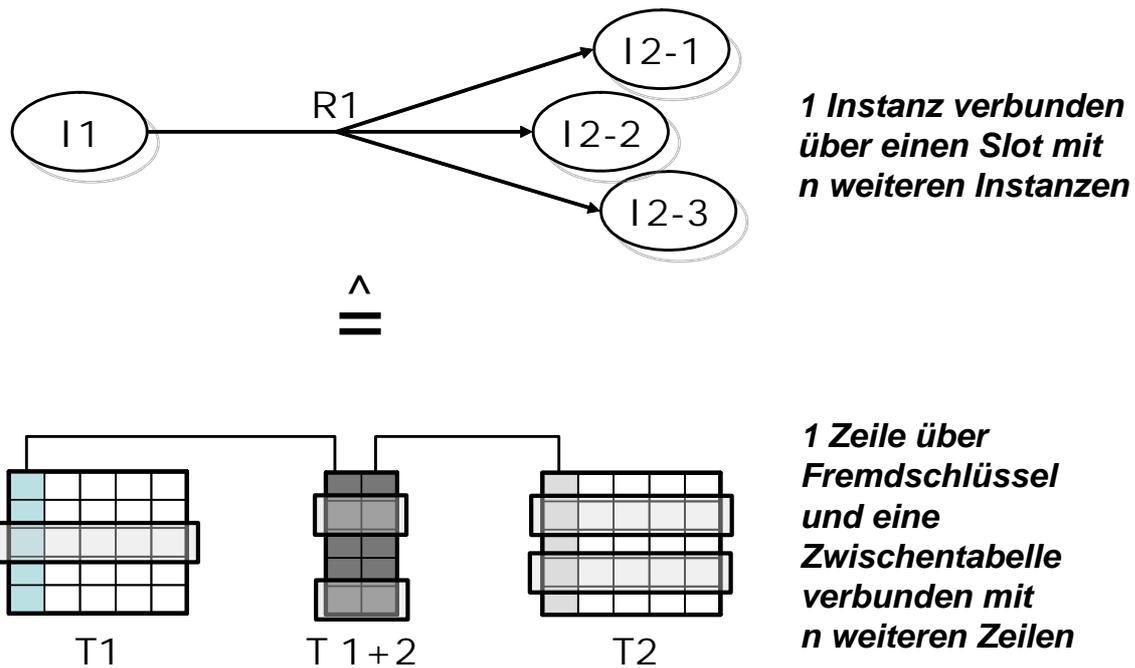


Abb. 36: Abbildung von Slots der Kardinalität n mit dem Wertebereich „Instanz“ (sog. M:N Relation über eine Zwischentabelle)

5.3.1.3 Limitierungen der gewählten Abbildung

Die beschriebene Form der objekt-relationalen Abbildung hat Limitierungen für die Modellierung in Protégé zu Folge. Diese Einschränkungen gelten für die Bereiche einer Ontologie, die dem beschriebenen Prinzip entsprechend auf ein Datenbankschema abgebildet werden. Alle weiteren Teile der Ontologie können ohne Einschränkungen modelliert werden, selbst wenn Querbezüge auf Klassen oder Instanzen (etwa über Wertebereiche von Slots) bestehen, die in der Datenbank repräsentiert sind. Die folgenden Einschränkungen gelten für Klassen und Template-Slots, für die das Datenbankschema bereits erzeugt wurde und für die bereits Instanzen angelegt wurden.

§ Eingeschränkte Änderungen am festgeschriebenen Schema

Wie bereits erwähnt, können Änderungen an der Klassenhierarchie und an Slots nur eingeschränkt vorgenommen werden. So ist es möglich, Unterklassen hinzuzufügen, sowie Template-Slots hinzuzufügen und zu löschen. Nicht möglich ist jedoch das Löschen von Klassen, die Unterklassen haben, oder das Verschieben von Klassen innerhalb der Hierarchie. Metadaten von Slots können ebenfalls nicht verändert werden (dies betrifft z.B. Wertebereich und Kardinalität).

§ Kein Wertebereich „any“

Der Wertebereich „any“ erlaubt es in Protégé Slots anzulegen, die mit jeder Art Wert belegt werden können (einfacher Datentyp, Instanz, ...). Dies bedeutet, dass der

Wertebereich vollkommen offen ist. Für die beschriebene Art der objektrelationalen Abbildung ist das nicht zulässig.

§ Keine Mehrfachvererbung

Mehrfachvererbung wird für die beschriebene Abbildung nicht unterstützt. Es wäre prinzipiell möglich, auch Mehrfachvererbung abzubilden. Nur würde dies einen relativ hohen Aufwand bei der Datenverwaltung und beim Datenzugriff durch den Framestore bedingen.

Limitierung bei der Abbildung vorhandener Schemata

Grundsätzlich unterstützt das SCIFORG-System auch den umgekehrten Weg: Das Erstellen eines frame-basierten Modells auf der Grundlage einer existierenden relationalen Datenbank. An dieser Stelle ist zu beachten, dass das SCIFORG-System nicht primär als Datenintegrationssystem konzipiert ist. Es bestehen eine Reihe von Möglichkeiten zum Daten-Import und die Möglichkeit, Datenbank-Schemata mit externen Applikationen parallel zu verwenden – unter bestimmten Voraussetzungen.

Bei der Anwendung der beschriebenen objekt-relationalen Abbildung auf bestehende Datenbankschemata sind folgende Einschränkungen zu berücksichtigen:

§ Fixes Abbildungs-Schema

Das Schema der objekt-relationalen Abbildung in SCIFORG ist fix. Datenbank-Schemata können auf der Grundlage dieses Schemas für das SCIFORG-System zugänglich gemacht werden. In vielen Fällen wird diese Abbildung der intendierten Semantik und der Interpretation des Schemas durch die dazugehörige Applikation nicht gerecht. Es sind zwar grundsätzlich Anpassungen des SCIFORG-Framestores möglich, diese erfordern aber Änderungen/Ergänzungen am Programmcode und sind nicht zur Laufzeit anwendbar.

§ Einschränkungen des Frame-Modells von Protégé

Das Datenmodell von Protégé ist nur eine spezielle Möglichkeit, Datenbank-Schemata zu interpretieren. Datenbank-Schemata, die im Rahmen von fachspezifischen Applikationen zur Anwendung kommen, können eine Interpretation erfordern, die nicht direkt auf das Protégé-Modell übertragbar ist. Als Beispiel seien hier parametrisierte Relationen genannt.

§ Nur Untermenge von Datentypen

Wie bereits erwähnt, unterstützt die beschriebene Abbildung nur eine Untermenge von Datentypen (vgl. Seite 118). Grundtypen wie Zeichenketten und Fließkommazahlen können üblicher Weise durch eine ganze Reihe von Datentypen abgebildet werden. Dies schließt auch Hersteller-spezifische Spalten-Formate ein. Die Problematik von Datentypen wird bereits am Beispiel von Spalten mit booleschen Werten

(„wahr“/„falsch“) deutlich. Diese sind nicht standardisiert. In einigen Datenbank-Managementsystemen müssen solche Werte als Zahl dargestellt werden. Diese Problematik ist vor allem für Datenintegrationssysteme relevant.

5.3.2 Anwender-Schnittstellen für die Prozesswissenschaften

Für das SCIFORG-System wurden eine Reihe von graphischen Schnittstellen geschaffen. Damit wird das System den besonderen Anforderungen der Prozesswissenschaften gerecht. Um das Erfassen und Abrufen von Informationen einfach und produktiv zu gestalten, sind fachspezifische Visualisierungen bzw. Oberflächen erforderlich. Dies betrifft die Darstellung und das Editieren von mathematischen und chemischen Formeln, die Unterstützung physikalischer Einheiten, die Visualisierung von Datenreihen und graphen-basierte Oberflächen für Ontologie-Ausschnitte.

Für die fachspezifischen Erweiterungen der Oberfläche wurden die bestehenden Schnittstellen von Protégé ausgenutzt. Für Slots, die mathematische oder chemische Formeln beinhalten, wurden mit Hilfe bestehender Bibliotheken „Widgets“ entwickelt, die diese Formeln visualisieren und entsprechend editierbar machen. Für physikalische Einheiten wurde ein System entwickelt, bei dem für Slots physikalische Größen definiert und diesen zugeordnet werden können.

Die Widgets fügen sich in die Standard-Oberfläche ein und können den Möglichkeiten von Protégé entsprechend dynamisch konfiguriert bzw. zugewiesen werden.

5.3.2.1 Mathematische Formeln

Mathematische Formeln werden durch eine erweiterte Version des Gleichungs-Editors JOME (siehe Abschnitt 5.2.5 auf Seite 111) unterstützt. Wie in Abschnitt 4.5.1 „Daten- und Metadaten-Ebene“ auf Seite 88 erörtert, werden mathematische Formeln nicht im Sinne einer eigenen Ontologie repräsentiert.

Das hat zur Folge, dass mathematische Ausdrücke nicht für semantische Suchanfragen verwendet werden können. Es ist beispielsweise nicht möglich Anfragen in der Art „liefere alle Modelle mit Modellgleichungen, in denen eine Differentialgleichung mit trigonometrischen Funktionen auftritt“ zu bedienen. Allerdings ist es möglich, mathematische Ausdrücke für rein syntaktische Suchen für Zeichenfolgen-Vergleiche zu verwenden.

Abbildung 37 zeigt den Editor für mathematische Gleichungen oder Formeln, wie er von Anwendern verwendet werden kann. Der Editor unterstützt eine Reihe von Funktionen, Operationen und Symbolen.

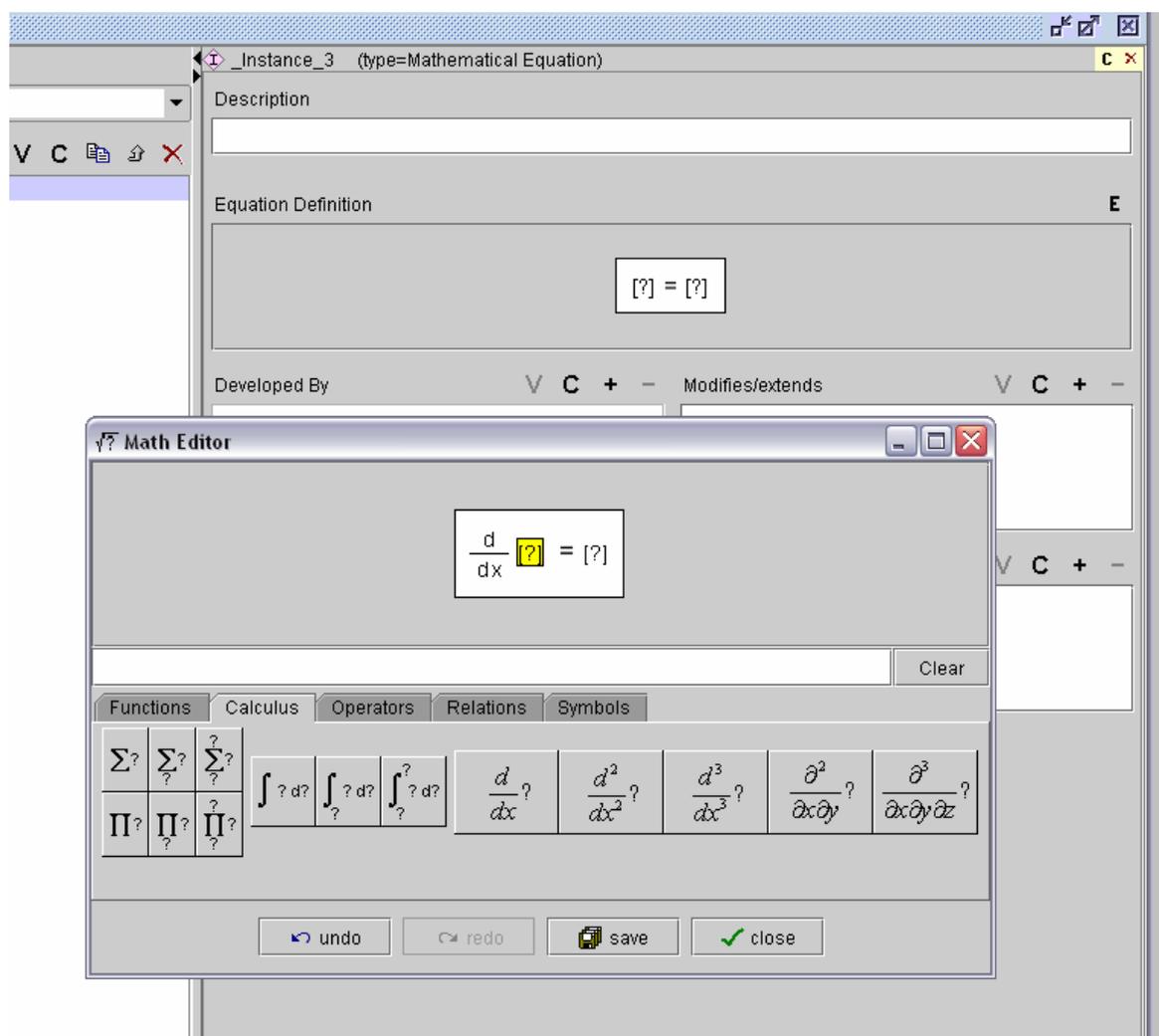


Abb. 37: Erfassen von mathematischen Gleichungen/Formeln

Aktiviert wird der Editor für eine gewählte Instanz und einen Slot. Letzterer muss als Slot für mathematische Formeln definiert sein. Dies geschieht über eine besondere Metaklasse (vgl. Abschnitt 5.2.1.2) und das Zuweisen eines speziellen „Widgets“.

Das entwickelte Widget erlaubt es, Gleichungen mit Hilfe eines Slots eindeutig zu beschreiben und zu visualisieren. Die Parameter der Gleichung können anschließend separat mit Hilfe des Frame-Modells von Protégé Ontologie-basiert beschrieben werden. Diese Methode wird durch das System (bzw. das Widget) durch einen einfachen Abgleich unterstützt. Ein zweiter Slot enthält sämtliche Parameter der Gleichung. Der Wertebereich dieses Slots muss einer ausgezeichneten Klasse entsprechen, die Parameter beschreibt. Auf diese Weise können zusätzliche Informationen zur (physikalischen, phänomenologischen, etc.) Interpretation der Gleichung(en) bzw. Formel(n) erfasst werden.

Es ist auch möglich, Gleichungen textuell zu erfassen. Dabei können Eingaben in MathML erfolgen, wengleich auch die Spezifikation noch nicht komplett unterstützt. Mit Einschränkungen

können Gleichungen damit auch über Kopieren und Einfügen erfasst werden, sofern die Quelle (i.d.R. Textverarbeitung) diesen Standard unterstützt.

5.3.2.2 Chemische Formeln

Die Unterstützung chemischer Formeln ist ähnlich aufgebaut wie die mathematischer Formeln/Gleichungen. Formeln werden als Zeichenfolgen in der Chemical Markup Language erfasst und speziell ausgezeichneten Slots zugewiesen. Für die entsprechenden Slots ist ein auf der Bibliothek JChemPaint (siehe Abschnitt 5.2.5) basierendes Slot-Widget als Standard eingestellt.

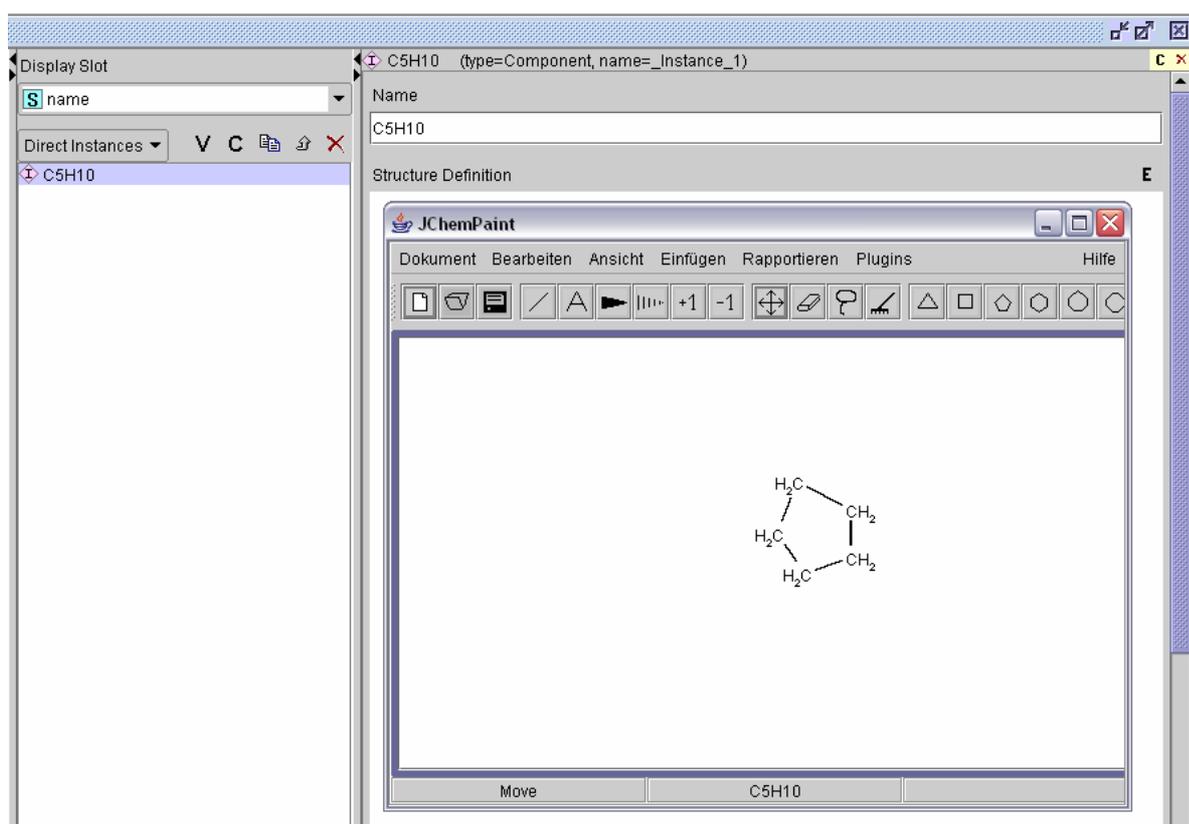


Abb. 38: Erfassen von chemischen Formeln

Chemische Formeln können graphisch zwei-dimensional erfasst und bearbeitet werden (siehe Abbildung 38). JChemPaint unterstützt neben CML noch weitere Formate, in denen Formeln exportiert und importiert werden können. Diese werden jedoch zurzeit nicht berücksichtigt. Beim Export einer Wissensbasis aus SCIFORG wird grundsätzlich die CML-Serialisierung verwendet.

Wie auch bei mathematischen Ausdrücken können chemische Formeln nicht für eine Suche auf Basis der Struktur herangezogen werden, sondern nur für einen rein syntaktischen Abgleich. D.h. es können z.B. Element-Symbole und Fragmente von Formeln in der Textsuche verwendet werden, um entsprechende Verbindungen zu suchen. Es ist also keine echte Struktursuche sondern ein rein textueller Abgleich.

5.3.3 Einheiten-Verwaltung

Die Einheiten-Verwaltung ist in das SCIFORG System integriert. Sie erlaubt es:

- § Physikalische Größen als Kombination von Basisgrößen (den SI-Einheiten entsprechend) zu definieren;
- § Attribute (Slots) von Klassen als physikalische Größe zu definieren;
- § Diesen Attributen eine Standard-Einheit zuzuweisen;
- § Einheiten automatisch umzurechnen;
- § Einheiten beim Daten-Import zu berücksichtigen.

Um einen Slot als Repräsentation einer physikalischen Größe zu definieren, muss dieser Instanz einer besonderen Metaklasse sein (vgl. Abschnitt 5.2.1.2).

Einheiten werden als einfache Zeichenfolge erfasst und definiert, wie z.B. „kg/m³“. Zu diesem Zweck stehen Symbole für die SI-Einheiten sowie für die erforderlichen Operationen zur Verfügung (siehe Tabelle 6).

Tab. 6: Definition von Einheiten über Zeichenfolgen

Symbol/Ausdruck	Interpretation
m	SI-Einheit für Länge „Meter“; etc.; Sämtliche SI-Einheiten sind durch entsprechende Symbole repräsentiert
kg	SI-Einheit für Massen „Kilogramm“ (s.o.)
s	SI-Einheit für die Zeit „Sekunde“ (s.o.)
A	SI-Einheit die Stromstärke „Ampère“ (s.o.)
K	SI-Einheit für die thermodynamische Temperatur „Kelvin“ (s.o.)
mol	SI-Einheit für Substanzmengen „Mol“ (s.o.)
Cd	SI-Einheit für die Lichtstärke „Candela“ (s.o.)
*	Symbol für die Multiplikation;
/	Symbol für die Division
^	Symbol für Potenzen

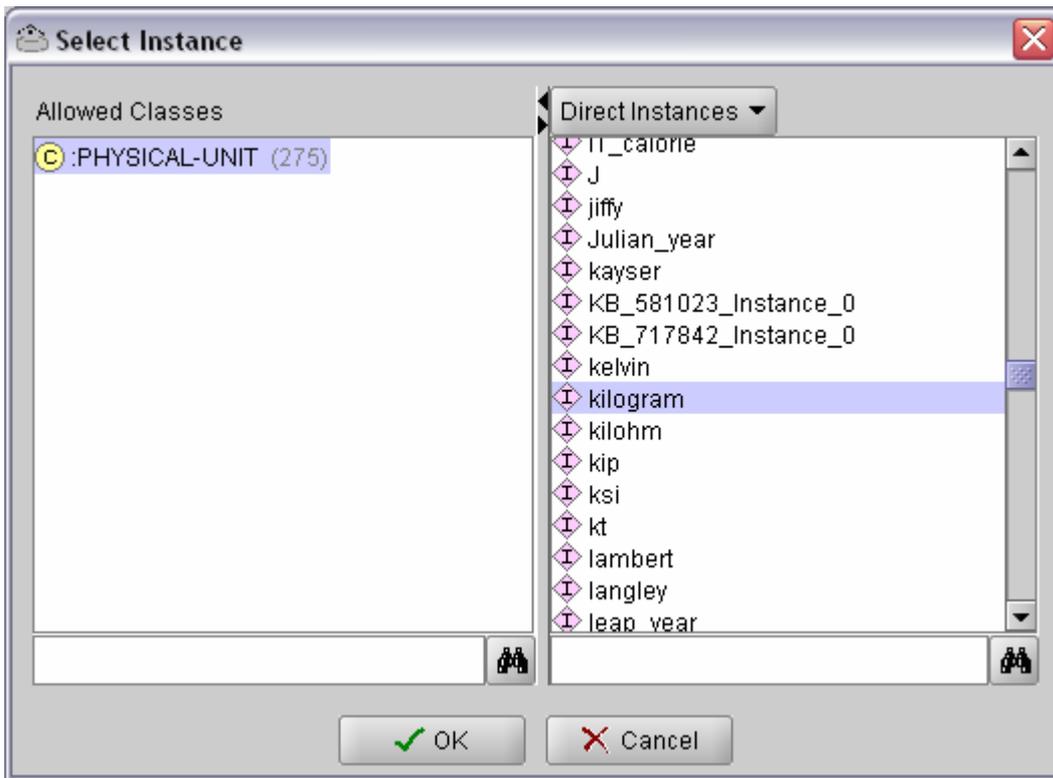


Abb. 39: Auswahl der physikalischen Einheit

Abbildung 39 zeigt die Auswahl von physikalischen Einheiten, die im System von Grund auf verfügbar sind.

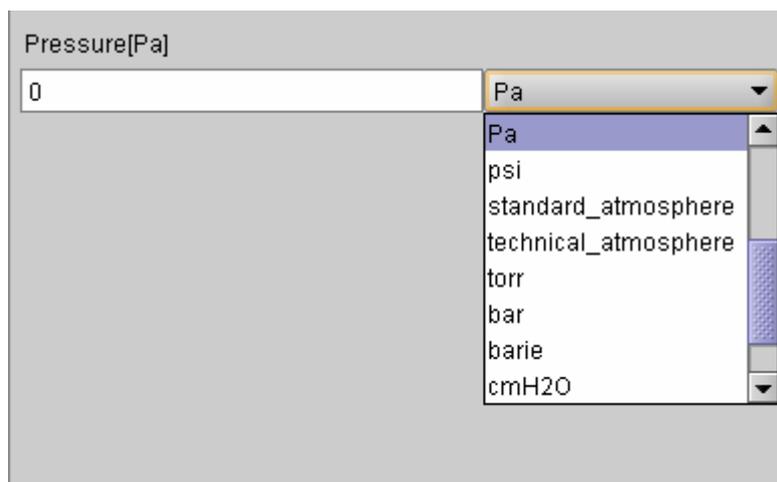


Abb. 40: Auswahl der Einheit in Slot-Widgets

Beim Erfassen und Bearbeiten von Instanzen wird für Slots, die eine physikalische Größe repräsentieren, die entsprechende Standard-Einheit angezeigt. Ein Auswahlfeld ermöglicht es Anwendern, eine alternative Einheit für die entsprechende Größe auszuwählen (siehe Abbildung 40).

5.3.4 Werkzeuge für Regeln, Suchanfragen und zur Navigation

5.3.4.1 Graphische Schnittstelle

SCIFORG verfügt über eine graphische Schnittstelle zur graphischen Formulierung regelhafter Zusammenhänge oder Suchanfragen bzw. zur Navigation in der Wissensbasis. Diese basiert auf der Bibliothek JGraph (siehe Abschnitt 5.2.4) und einem eigens entwickelten Symbol-Editor. Letzterer erlaubt es, Symbole für Instanzen bestimmter Klassen festzulegen, die dann als Knoten im Graphen visualisiert werden.

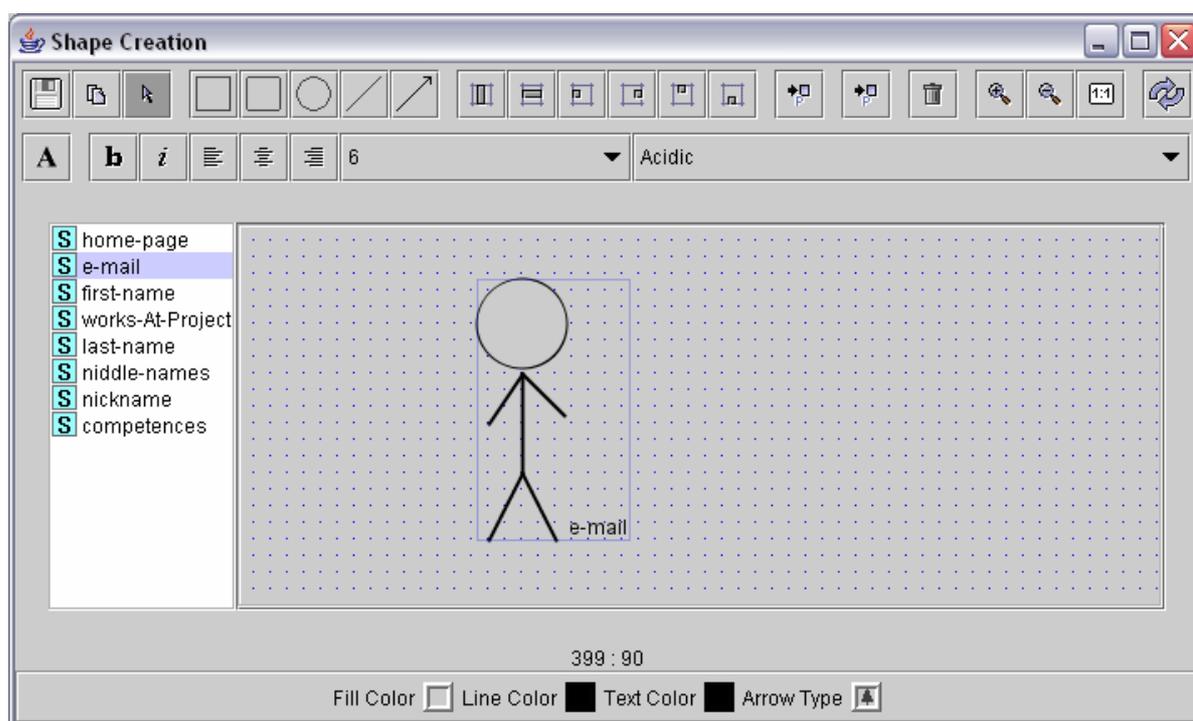


Abb. 41: Symbol-Editor für domänen-spezifische Symbole

Für die Bibliothek JGraph wurde ein Adapter entwickelt, der das Frame-Modell von Protégé als Datenmodell für Graphen hinterlegt. Dieser bildet Elemente einer gewählten Ontologie auf einen Graphen ab, wie in Tabelle 7 aufgelistet. Dabei können sowohl existierende Fakten als auch Muster für Suchanfragen visualisiert werden. Zu diesem Zweck wurde das interne Datenmodell erweitert. Zum Frame-Typ „Instanz“ existiert ein Äquivalent, das ein Muster für die zu suchenden Instanzen repräsentiert. Mit Hilfe der Slots dieses Typs lassen sich baumartig strukturierte Suchen graphisch aufbauen.

Tab. 7: Visualisierung in JGraph

Ontologie-Element	JGraph-Visualisierung
Instanz	Knoten (mit Klassen-spezifischem Symbol)
Relation (Slot mit Wertebereich Instanz)	Kante (mit entsprechender Kanten-Beschriftung)
Einfaches Attribut (Slot mit Wertebereich Zeichenkette, Zahl, etc.)	Label innerhalb eines Klassen-Symbols (s.o.)
Klasse	Analog zu Instanz

Suchanfragen

Zu Formulierung einer Suchanfrage über die graphische Schnittstelle werden Elemente (Klassen oder Instanzen) per „Drag and Drop“ aus dem Klassenbaum auf die entsprechende Fläche des Suchwerkzeuges gezogen. Wird eine Klasse auf die Fläche gezogen, so wird ein Repräsentant dieser Klasse erzeugt, der für alle zu suchenden Instanzen steht. Implizit wird damit eine Variable erzeugt. Wird eine Instanz auf die Fläche gezogen, so steht dies für eine Konstante.

Sobald eine Suchanfrage graphisch beschrieben ist, kann sie an das System abgesetzt werden. Dabei wird intern eine strukturierte Anfrage in der Syntax des Algernon-Systems erzeugt.

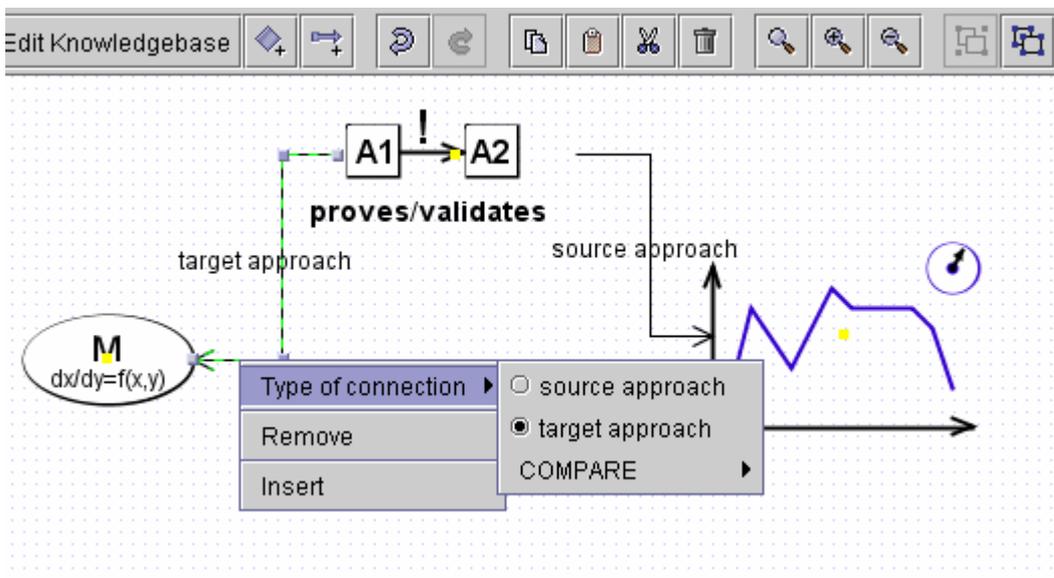


Abb. 42: Graphische Suchanfrage: Bestimmung der Bedeutung von Verknüpfungen

Zwischen den Knoten (Konstanten oder Variablen) können ebenfalls per „Drag and Drop“ Kanten erzeugt werden, die diese Verknüpfen. Dabei wird der Kante ein Slot zugeordnet. Die Zuordnung muss so erfolgen, dass die Domäne des Slots der Klasse des einen Knotens entspricht

und der Wertebereich der Klasse des zweiten Knotens. Das System bietet kontext-sensitiv nur die möglichen Slots zu Auswahl an. Dies ist in Abbildung 42 für die Klassen „Mathematical Model“, „Claim“ und „Experimental Approach“ dargestellt. Ein Claim im Sinne des SchoolOnto-Ansatzes verbindet zwei Ansätze. Um diese Verbindung formal auszudrücken, stehen für jeden Claim zwei Slots zur Verfügung, die als Wertebereich „Approach“ haben.

Kanten können eine direkte Verbindung darstellen (wie in Abbildung 42) oder für Vergleichsoperationen verwendet werden.

Abbildung 42 zeigt eine Suchanfrage, die ausschließlich Variablen verwendet. Diese Suchanfrage liefert als Ergebnis alle mathematischen Modelle, zu denen generell eine experimentelle Validierung existiert.

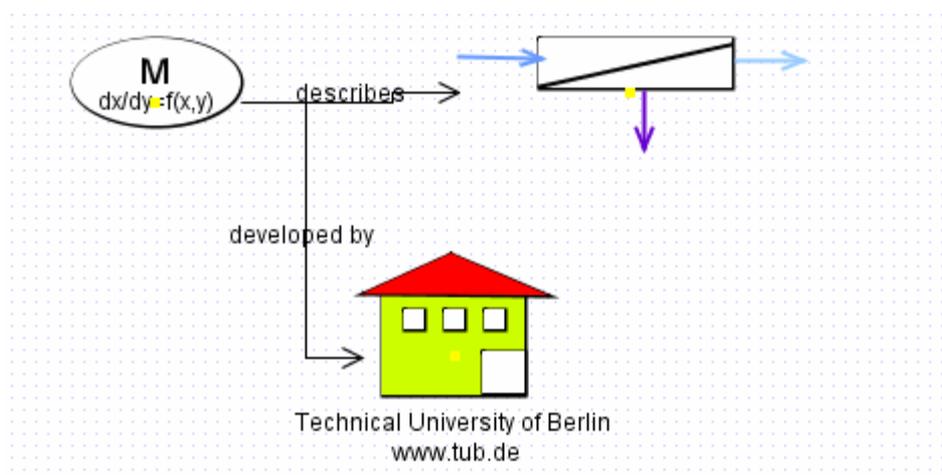


Abb. 43: Graphische Suchanfrage: Welche mathematischen Modelle zur Beschreibung von Membran-(Modul-)Systemen wurden in einer bestimmten Institution entwickelt?

Abbildung 43 zeigt eine Suchanfrage, die sowohl Variablen als auch Konstanten verwendet. Als Konstante wird die Institution „Technische Universität Berlin“ bestimmt. Dies geschieht, indem die entsprechende Instanz aus der angezeigten Liste auf das Suchfeld gezogen wird. Für mathematische Modelle und das technische System „Membranmodul“ wird jeweils eine Variable erstellt, indem die betreffenden Klassen auf die Suchfläche gezogen werden. Für die Kanten wird festgelegt, dass sie den Slot „describes“ (entsp. „Model“-> „describes“-> „System“) bzw. den Slot „developed by“ (entsp. „Model“-> „developed by“-> „Institution“) repräsentieren. Die Suchanfrage steht für alle mathematischen Modelle, die an der Technischen Universität Berlin entwickelt wurden und das Verhalten von Membranmodulen beschreiben.

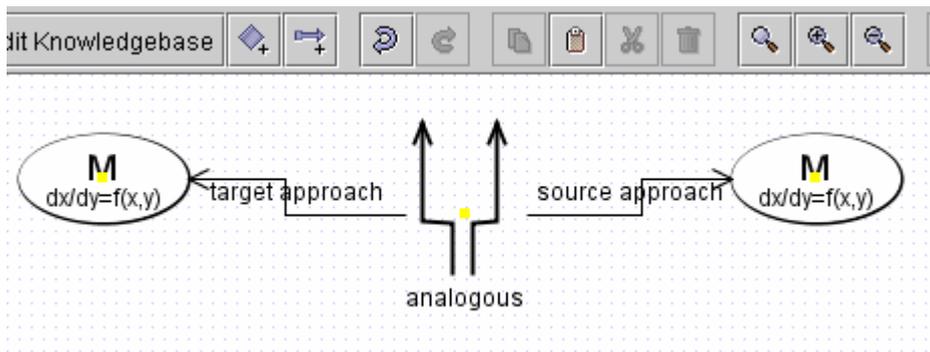


Abb. 44: Graphische Suchanfrage: Suche von analogen Modellen

Abbildung 44 zeigt eine weitere Anfrage für Claims in Sinne des SchoolOnto-Ansatzes. Hier werden analoge Ansätze gesucht.

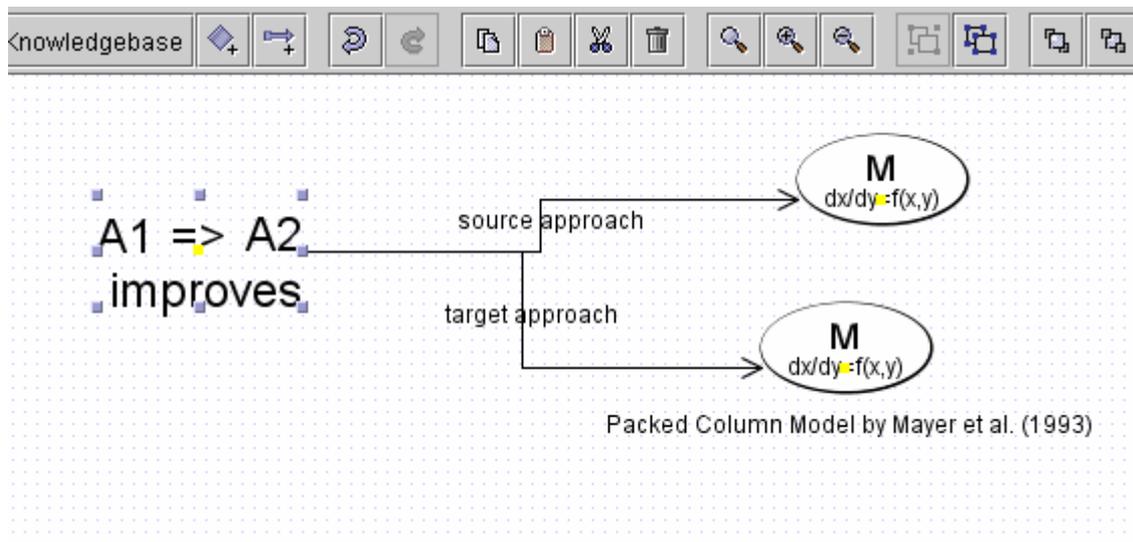


Abb. 45: Welche Modelle stellen eine Weiterentwicklung von einem bestimmten Modell X dar?

Abbildung 45 zeigt eine Anfrage, die nach Verbesserungen/Weiterentwicklungen eines bestimmten Modells (wieder im Sinne des SchoolOnto-Ansatzes) sucht. Das betreffende Modell ist als Konstante erfasst (untere rechte Hälfte der Abbildung). Dementsprechend ist die Eigenschaft bzw. der Slot „Name“ mit einem Wert belegt. Das Symbol für mathematische Modelle ist so angelegt, dass der Name mit angezeigt wird.

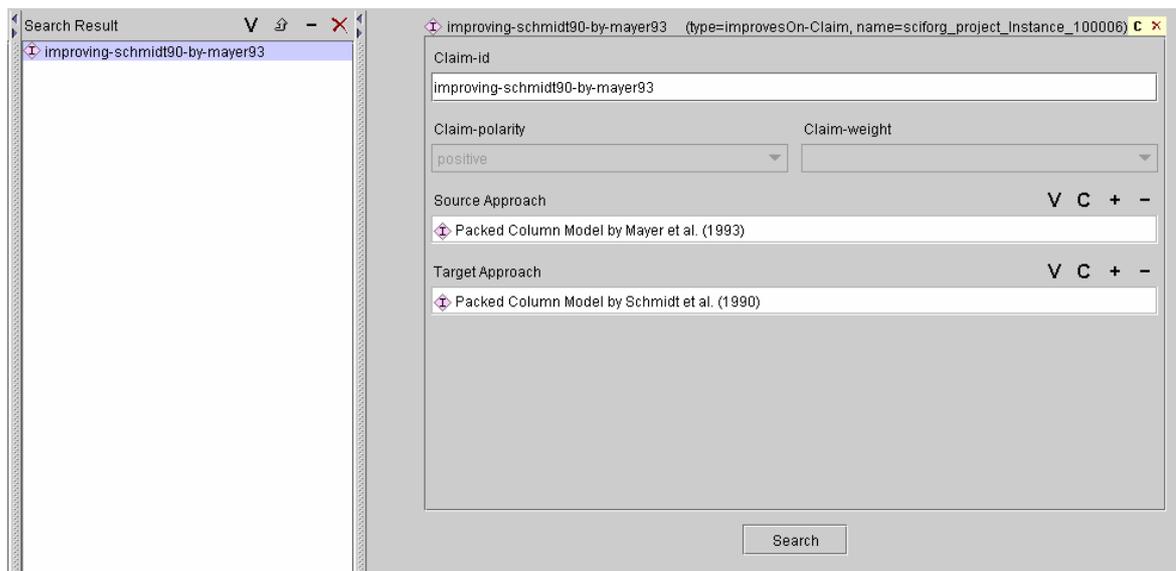


Abb. 46: Darstellung von Suchergebnissen in der Listen-/Formular-Ansicht

Suchergebnisse können in der Graphen- oder der Listen-Ansicht dargestellt werden. Letztere eignet sich für große Ergebnismengen. Die Graphen-Ansicht hat dagegen den Vorteil, dass Ergebnisse auf die gleiche Art visualisiert werden wie Suchanfragen und dass in der Ergebnisdarstellung navigiert werden kann (s.u.).

Abbildung 46 zeigt die Listen-Ansicht für ein einzelnes Suchergebnis vom Typ „Claim“ (im Sinne des SchoolOnto-Ansatzes). Sobald ein Ergebnis in der Liste markiert wird, werden dessen Daten in einem Formular angezeigt.

Komplexere Anfragen und Wertevergleiche

Mit Hilfe der graphischen Schnittstelle sind auch komplexere Suchausdrücke und Vergleichsoperationen möglich. Dabei können die Eigenschaften (bzw. Slots) von Instanzen für die Definition von Vergleichskriterien verwendet werden.

Abbildung 47 zeigt eine komplexere, noch nicht vollständig spezifizierte Suchanfrage anhand derer die Funktionalität für Wertevergleiche verdeutlicht werden soll. Das Ziel der Anfrage soll sein, mathematische Modelle zu suchen, die analog zu einem bestimmten Modell zu sehen sind, sich aber auf ein System mit unterschiedlichen Eigenschaften beziehen. In diesem Fall besteht der Unterschied in einer Stoffeigenschaft.

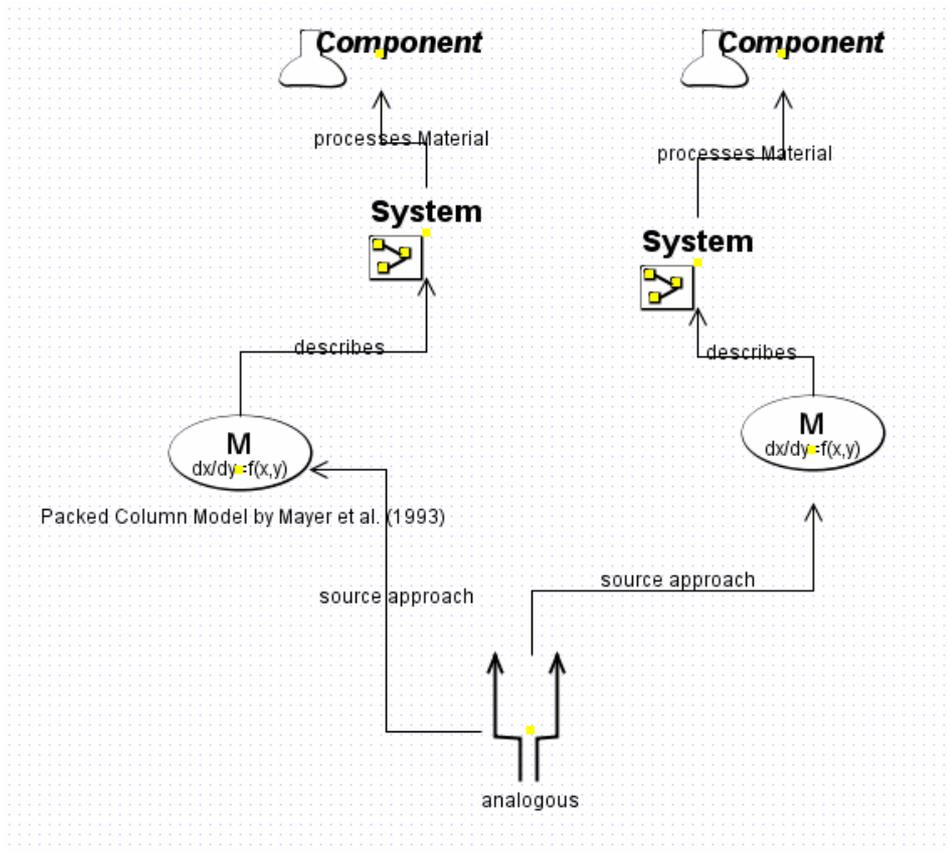


Abb. 47: Komplexere Suchanfrage (unvollständig)

In Abbildung 47 ist ein zu suchender Claim (bzw. mehrere Claims) dargestellt. Dieser bezieht sich auf ein existierendes Modell (eine Konstante) und ein Platzhalter für ein weiteres Modell (eine Variable). Beide beschreiben ein (verfahrenstechnisches) System, in dem eine Komponente verarbeitet wird. Sowohl die Systeme als auch die Komponenten sind als Variablen erfasst. In der unvollständigen Form haben die Variablen für System und Komponente keinen Einfluss auf die Suche, abgesehen davon dass ein System und mindestens eine Komponente existieren müssen.

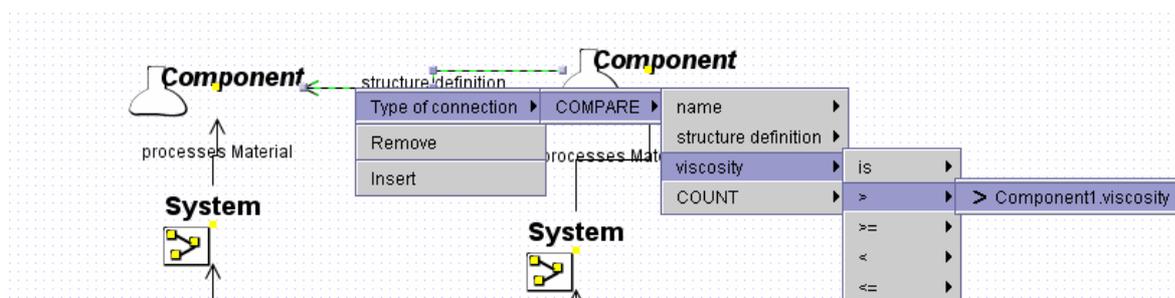


Abb. 48: Definieren eines Wertevergleichs in graphischen Suchanfragen

Abbildung 48 zeigt, wie ein Wertevergleich für die Eigenschaft „viscosity“ der beiden Komponenten (-Platzhalter) über ein Kontextmenü definiert werden kann. Kontextsensitiv werden Ver-

gleiche angeboten, die zum Wertebereich des jeweiligen Slots passen. Hier handelt es sich um einen Zahlenvergleich, dementsprechend werden die Operatoren „größer“, „kleiner“, etc. angeboten.

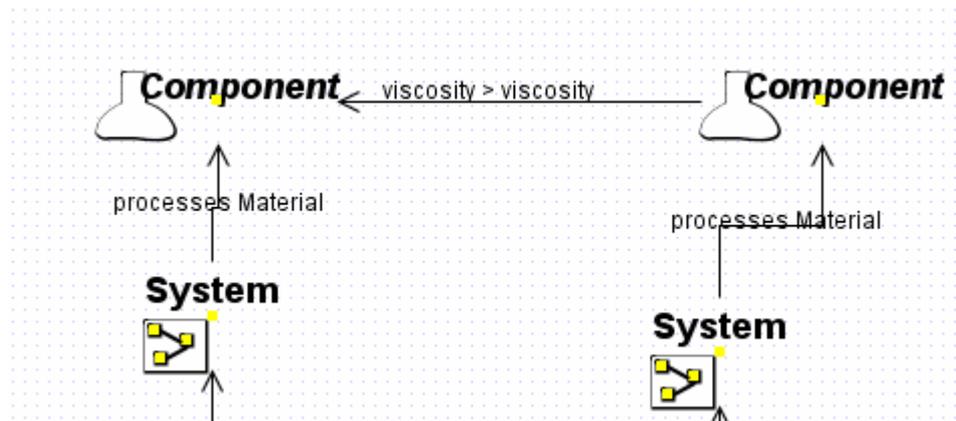


Abb. 49: Anzeige von Vergleichskriterien in graphischen Suchanfragen

Abbildung 49 zeigt, wie der Wertevergleich visualisiert wird, sobald er über das Kontextmenü ausgewählt wurde.

Die resultierende Suchanfrage sucht Modelle, in Analogie zu dem ausgewählten Modell zu sehen sind, sich aber auf eine Systembeschreibung beziehen, die eine Komponente höherer Viskosität beinhaltet. Dabei ist zu beachten, dass nicht ausgeschlossen ist, dass das zu suchende Modell mit dem für die Anfrage ausgewählten Modell übereinstimmt. Die graphische Schnittstelle unterstützt keine „ungleich“-Verknüpfung, über die definiert werden könnte, dass nur Modelle berücksichtigt werden, die nicht dem ausgewählten Modell entsprechen. Jedoch bezieht sich die Suche auf Claims und innerhalb eines solchen werden unterschiedliche Modelle verglichen.

Limitierungen

Über die graphische Schnittstelle können keine verschachtelten Ausdrücke erstellt werden, die konjunktiv und disjunktiv verknüpft sind. Darüber hinaus werden keine Negationen und keine Aggregationen (Summenbildung, etc.) oder Funktionen („procedural attachments“) unterstützt.

Navigieren in der Wissensbasis

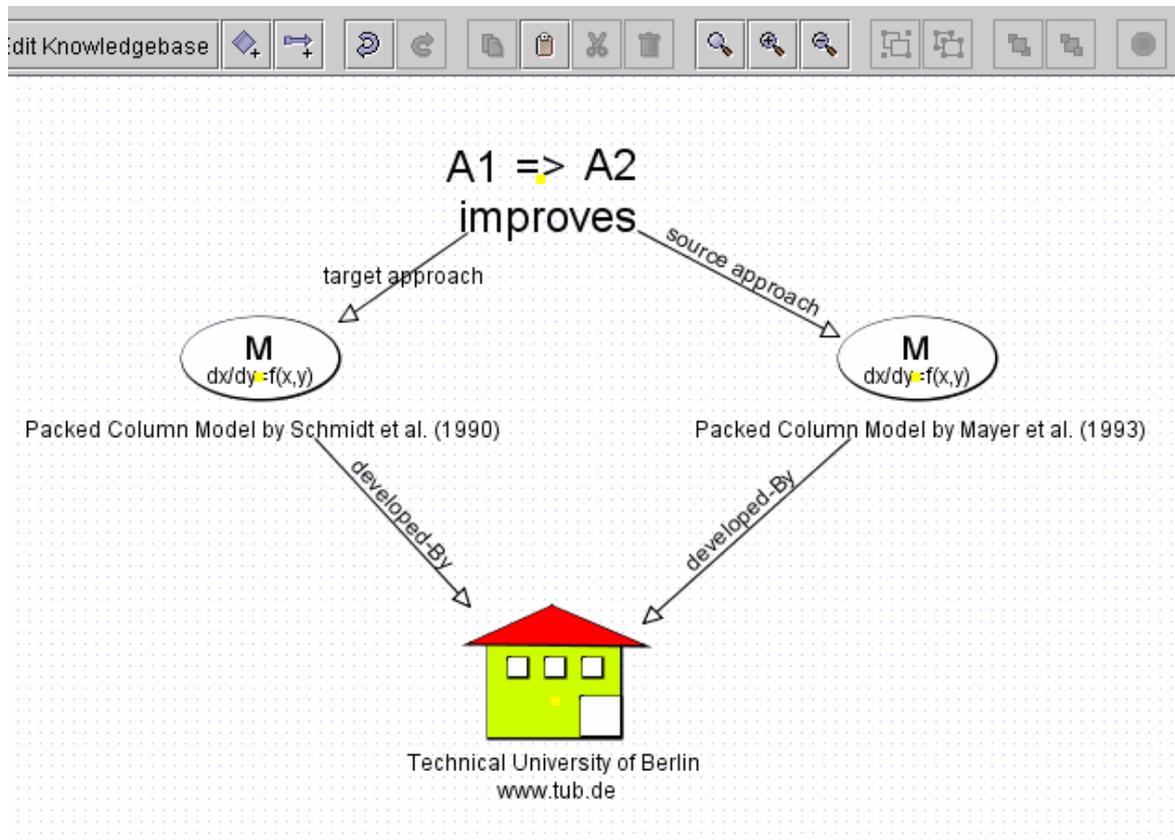


Abb. 50: Visualisierung von Instanzen über die graphische Schnittstelle

Abbildung 50 zeigt die Visualisierung von Fakten (Instanzen) über die graphische Schnittstelle für einen bestimmten „Claim“. Instanzen werden mit dem Symbol der entsprechenden Klasse dargestellt. Wenn dieses Symbol Platzhalter für Wertbelegungen von Slots enthält, werden diese im Symbol visualisiert. Wertbelegungen für Slots deren Wertebereich wiederum Instanzen umfasst, werden als weitere Kanten und Knoten im Graphen angezeigt. Dies geschieht bis zu einer einstellbaren maximalen Tiefe. Im vorliegenden Beispiel beträgt diese Tiefe zwei Kanten. Für den dargestellten Claim werden die mathematischen Modelle (erster Schritt bzw. Tiefe eins) und deren Verbindung zu weiteren Instanzen, in diesem Fall von der Klasse „Institution“ (Tiefe zwei) angezeigt.

Sobald der Fokus vom Anwender auf eine andere Instanz gesetzt wird, verschiebt sich die Graphendarstellung, so dass alle mit dieser Instanz verknüpften Instanzen in der Tiefe zwei dargestellt werden. Auf diese Weise kann durch die Wissensbasis navigiert werden.

Ein Doppelklick mit der linken Maustaste auf eines der dargestellten Symbole öffnet die Formularansicht für die betreffende Instanz (vgl. Abbildung 46).

5.3.4.2 Formulare für Suchanfragen

SCIFORG ermöglicht es, die in Protégé für die Wissensakquisition verwalteten Formulare für Suchanfragen zu verwenden.

Abb. 51: Formular-Suche für Claims

Abbildung 51 zeigt ein Formular für die Klasse „Claim“. Das Formular entspricht dem Standard-Formular für die Visualisierung und für das Editieren von Klassen und Instanzen in Protégé, mit dem einzigen Unterschied dass zusätzliche Schaltfelder zur Verfügung stehen.

Für einfache Slots (d.h. solche mit dem Wertebereich Zahl, Zeichenfolge oder boolescher Wert) steht eine Schaltfläche für Vergleichsoperatoren wie „ist gleich“ zur Verfügung. Die Operatoren werden wie bei der Graphen-basierten Suche kontext-sensitiv in Abhängigkeit vom Wertebereich angeboten (s.o.). Der Operand wird in das entsprechende Formularfeld eingetragen.

Für Slots mit dem Wertebereich „Instanz“ kann wie in der Graphen-basierten Suche entweder eine Konstante oder eine Variable (Platzhalter für eine weitere Instanz) erfasst werden. Eine Konstante wird analog zum Zuweisen von Werten beim Editieren von Instanzen eingetragen.

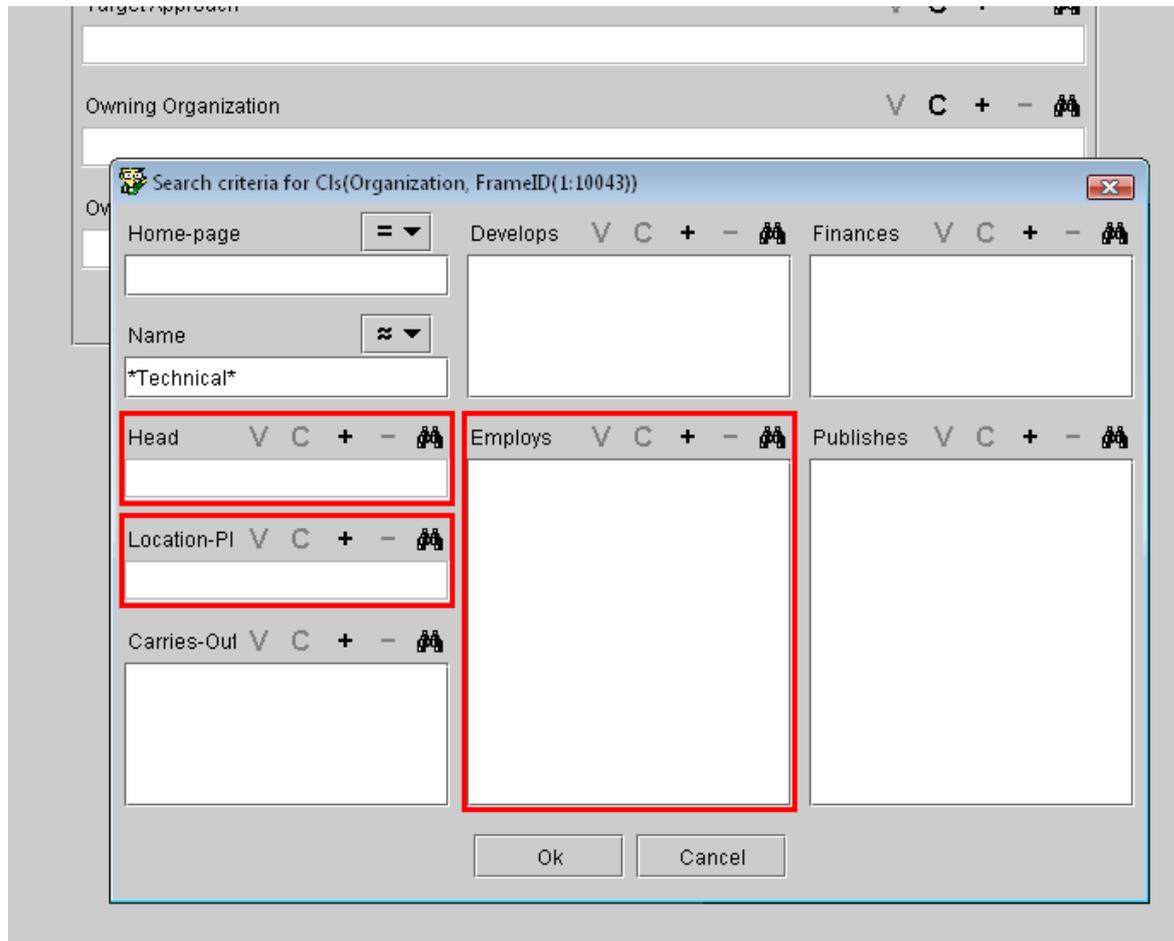


Abb. 52: Erfassen eines Platzhalters für eine Suchbedingung

Um einen Platzhalter zu erfassen, wird beim Formularfeld für den entsprechenden Slot die Schaltfläche mit dem Fernglas-Symbol aktiviert. Abbildung 52 zeigt, wie für die Eigenschaft „owning organization“ des bzw. der zu suchenden Claims ein Platzhalter für eine Organization erfasst wird. Für diese Organisation wird als zusätzliches Suchkriterium festgelegt, dass ihr Name die Zeichenfolge „Technical“ enthalten soll.

5.3.5 Rohdatenverwaltung und Datenvisualisierung

SCIFORG verfügt über Möglichkeiten zur Verwaltung und Visualisierung großer Datensätze auf der Grundlage von Formular-basierten Anfragen. Die Visualisierung bildet Instanzwerte auf Datenreihen ab. Eine Datenreihe entspricht jeweils dem Ergebnis einer Suchanfrage (z.B. den Daten, die innerhalb eines vom Anwender spezifizierten Zeitintervalls liegen). Eine Menge von Instanzen entspricht einer Menge von Datenreihen, wobei die Werte für jeweils einen Slot eine Datenreihe ergeben. Es können jeweils mehrere Anfragen gleichzeitig visualisiert werden.

Abbildung 54 auf Seite 142 illustriert beispielhaft die Visualisierung von Datenreihen in SCIFORG. In dieser Ansicht können Datenreihen hinzugefügt, gefiltert und exportiert werden.

Das Hinzufügen von Datenreihen geschieht über die formularbasierte Suche (siehe Abschnitt 5.3.4.2 „Formulare für Suchanfragen“ auf Seite 137). Als Filterfunktion steht eine Glättung mit Hilfe eines gleitenden Durchschnitts zur Verfügung (auf Abbildung 55 auf Seite 143 beispielhaft dargestellt). Darüber hinaus können für jede Datenreihe Maximum, Minimum und Durchschnitt angezeigt werden (siehe Abbildung 56 auf Seite 144).

Die Funktionalitäten von SCIFORG sind vor allem für die Verwaltung von Rohdaten geeignet. Das Programm bietet (bewusst) nicht die Möglichkeiten von Tabellenkalkulationsprogrammen. Im Gegensatz zu letzteren ist es jedoch für sehr große Datenmengen optimiert. Durch die Import- und Export-Funktionen und durch die in SCIFORG für wissenschaftliche Daten verfügbaren Grundfunktionen (Definition von physikalischen Größen, Verwaltung von Einheiten) bietet die Datenvisualisierung eine Ergänzung zu anderen Werkzeugen.

Im Folgenden wird beispielhaft eine typische Abfolge von Arbeitsschritten zur Anwendung von SCIFORG-Funktionen auf Rohdaten beschrieben. Diese besteht aus den folgenden Schritten:

- § Importieren der Daten in das System,
- § Selektieren/Filtern von Daten;
- § Visualisieren und Analysieren;
- § Exportieren zwecks weiterer Verarbeitung in einem externen Programm.

Übernahme von Daten mit Hilfe der Import-Funktion

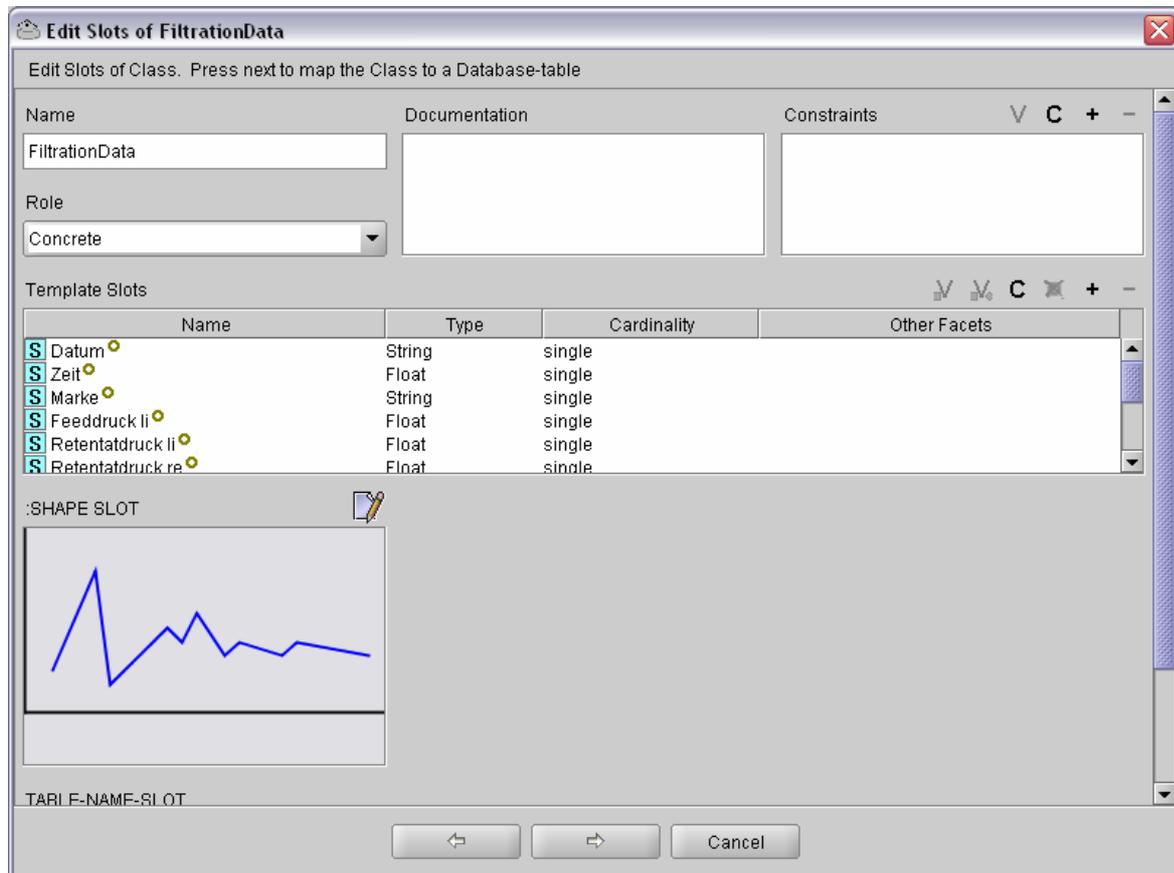


Abb. 53: Import-Wizard

Mit Hilfe des Import-Wizards werden experimentell ermittelte Daten oder Simulationsdaten importiert (siehe Abbildung 53). Voraussetzung ist, dass die Daten tabellarisch als MS Excel™ oder als Textdatei vorliegen (Spalten getrennt durch Kommata, Leerzeichen oder ein beliebiges anderes Zeichen). Dabei wird für jede Messgröße ein Slot verwendet. Slots können automatisch eingerichtet werden.

Der Import setzt voraus, dass Metainformationen bzgl. der Bezeichnung und der physikalischen Einheit von Parametern jeweils den Kopfzeilen angegeben sind.

Slots müssen vom Typ Float, Integer, String oder Boolean sein. Die Bedeutung von Parametern wird über die Zuordnung zu einem Slot festgelegt. Dabei kann einem Slot eine physikalische Einheit zugeordnet werden (siehe auch Abschnitt 5.3.3 „Einheiten-Verwaltung“ auf Seite 127). Beim Import werden Einheiten automatisch umgerechnet. Dies setzt voraus, dass die Symbole, die in der zu importierenden Datei verwendet werden, einer Einheit zugeordnet werden können. Dabei sind auch Kombinationen von bekannten Einheiten möglich. Bei neuen Einheiten müssen diese zunächst angelegt werden.

Wenn noch keine entsprechenden Slots für einen Import zur Verfügung stehen, kann SCIFORG diese automatisch anlegen. Ist der Import abgeschlossen, stehen die importierten Daten innerhalb von SCIFORG zur Verfügung. Sie können anschließend gefiltert, visualisiert und wieder exportiert werden.

Filtern, Visualisieren und Exportieren

Bis auf den Datenimport und der Einheiten-Verwaltung können alle weiteren Funktionalitäten zu Mess- und Simulationsdaten in der Visualisierungsansicht vorgenommen werden (siehe Abbildung 54 auf Seite 142).

In der Visualisierungsansicht können Datenreihen hinzugefügt und entfernt werden (über die „+“ bzw. die „-“-Schaltfläche). Eine Datenreihe entspricht einer Menge von Instanzen für eine zu wählende Klasse. Diese wird durch eine formularbasierte Suche bestimmt. Für jede Datenreihe kann aus den zur Verfügung stehenden Slots ein (im Sinne der Visualisierung) unabhängiger und ein abhängiger Parameter bzw. Grundbereich und Wertebereich bestimmt werden. Wenn für einen Satz gemessener oder berechneter Daten mehrere unterschiedliche Parameter über demselben Parameter dargestellt werden sollen, muss derselbe Datensatz mehrmals als Datenreihe über das Menü hinzugefügt werden, wobei jeweils der gleiche Parameter als unabhängige Variable verwendet wird.

Die Wahl von unabhängiger und abhängiger Variable kann direkt in der Visualisierungsansicht geändert werden. Auf diese Art können einfache Korrelationsanalysen mit großen Datenmengen durchgeführt werden.

Über die Menüleiste können für jede selektierte Datenreihe folgende Funktionen verwendet werden: Export in eine Datei, Abrufen von statistischen Informationen und Entfernen der Datenreihe aus der Ansicht. Darüber hinaus kann, wie bereits erwähnt, eine logarithmische Skalierung gewählt und eine Glättung über den gleitenden Durchschnitt eingestellt werden. Schließlich besteht die Möglichkeit, den Grundbereich durch eine Zoom-Funktion weiter einzuschränken.



Abb. 54: Visualisierung von Datenreihen



Abb. 55: Anwendung der Glättungsfunktion (gleitender Durchschnitt)

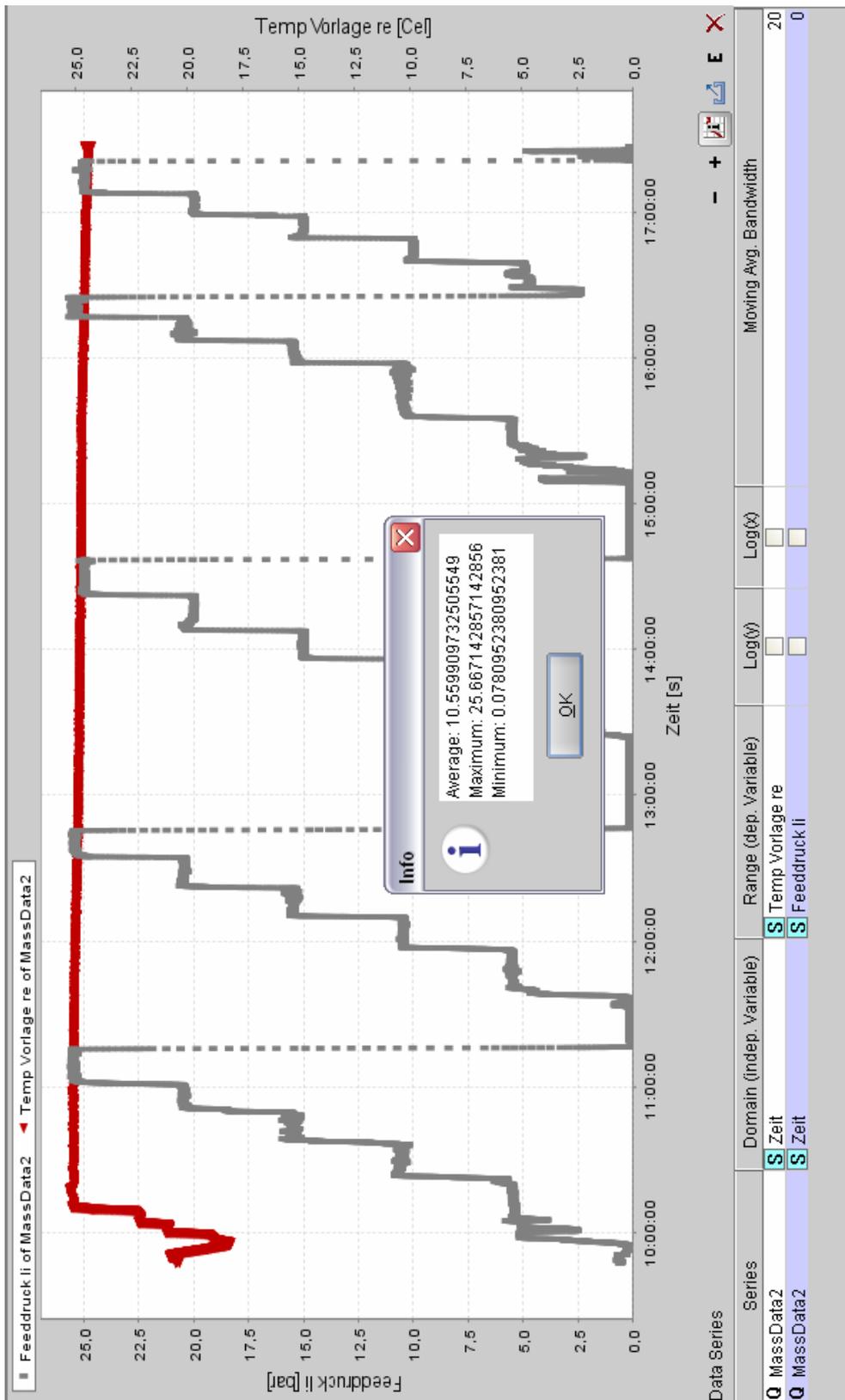


Abb. 56: Anzeige von statistischen Informationen

6 Anwendungsszenarien

Die Möglichkeiten des SCIFORG-Systems lassen sich am besten durch das im Folgenden beschriebene Basis-Szenario illustrieren. Dabei lassen sich verschiedene Funktionalitäten des Systems auch außerhalb dieses Szenarios sinnvoll nutzen. Dementsprechend handelt es sich in erster Linie um eine beispielhafte Darstellung, die einen Überblick vermitteln soll. Spezifischere Einsatzmöglichkeiten sind in den anschließenden Unterkapiteln geschildert.

Basisszenario

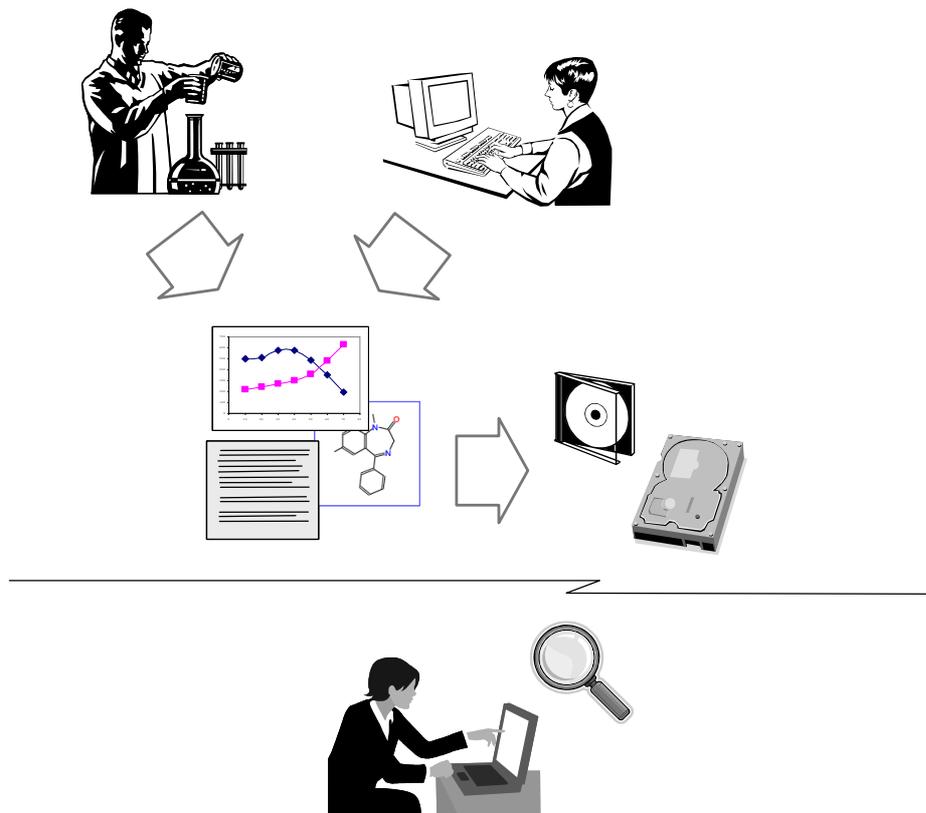


Abb. 57: Grundlegendes Anwendungsszenario für SCIFORG

Das Basisszenario ist in Abbildung 57 illustriert. Es betrifft Institutionen mit Forschungsausrichtung im Bereich der Prozesswissenschaften. Das Szenario geht von der folgenden Situation aus:

- § Im Rahmen von wissenschaftlichen Arbeiten werden verschiedene Untersuchungen durchgeführt.

- § Es werden sowohl induktive als auch deduktive Ansätze verfolgt, in denen über einen längeren Zeitraum und unter Ausbildung thematischer Schwerpunkte (bzgl. der Art der betrachteten Prozesse, der angewandten Methodik, etc.)
 - physikalische/chemische/biologische Wirkzusammenhänge analysiert werden;
 - mathematische Modelle entwickelt und validiert werden;
 - vorhandene Ergebnisse aufgegriffen und erweitert werden.
- § Im Labor sowie im Technikumsmaßstab finden experimentelle Untersuchungen statt, in denen kontinuierlich Messdaten anfallen.
- § Durch die Anwendung von mathematischen Modellen und numerischen Methoden auf Problemstellungen (z.B. im Bereich der Stofftrennung) werden kontinuierlich Daten generiert.
- § Die in den verschiedenen Ansätzen erzeugten Daten werden aufbereitet, bewertet und in Veröffentlichungen, Berichten, etc. referenziert.
- § Die experimentellen Untersuchungen stehen im unmittelbaren Zusammenhang mit der Modellentwicklung und dem damit verbundenen Erkenntnisgewinn.

Im Folgenden werden die Möglichkeiten des SCIFORG-Systems vor dem Hintergrund der oben skizzierten Situation beispielhaft beschrieben.

Mitarbeiter A hat eine Messreihe auf der Grundlage einer bestimmten Versuchskonfiguration abgeschlossen. Die konstante Versuchskonfiguration schließt den Versuchsaufbau und die eingesetzten Stoffe ein. Die erhaltenen Messdaten für diverse Systemparameter stehen im Anschluss in digitaler Form zur Verfügung. Mitarbeiter A bereitet die Daten so auf, dass sie im ASCII-Format vorliegen und in ein Tabellenkalkulationsprogramm übernommen werden können.

Zunächst startet A jedoch den Import in das SCIFORG-System. Letzteres ermöglicht es ihm, einfache Korrelationsanalysen durchzuführen und effizient die für bestimmte Fragestellungen relevanten Teile des (großen) Datensatzes zu identifizieren. Der Mitarbeiter legt im System die erforderlichen Parameter an (soweit nicht vorhanden) und beschreibt durch deren Zusammenstellung formal die Struktur des Messdatensatzes. Anschließend werden die Daten importiert. Mitarbeiter A nutzt die Visualisierung, um verschiedene Parameter im Verlauf darzustellen und ggf. zu glätten. Er vergleicht die Daten mit einer Datenreihe, die mit einer ähnlichen Versuchskonfiguration entstanden und bereits im System erfasst ist. Dazu verwendet er zunächst die Suchfunktion des Systems und anschließend erneut die Datenvisualisierung.

Mitarbeiter A ergänzt die neu erfassten Daten durch Metadaten. Darin beschreibt er v.a. die Versuchskonfiguration. Schließlich exportiert er einen Teil der Daten wieder aus dem System, um sie in ein Tabellenkalkulationsprogramm zu übernehmen. Dort werden die Daten aufbereitet. Be-

rechnete Daten werden unter Anwendung mathematischer Formeln hinzugefügt (etwa durch Bilanzierung oder durch Modellansätze). Ein Teil der Daten wird extrahiert und zu einem neuen Datensatz verdichtet. Dieser stützt eine bestimmte Sichtweise oder Aussage, wie z.B. eine beobachtete Korrelation oder die quantitative Übereinstimmung mit Modellrechnungen.

Mitarbeiter A importiert auch die Daten aus dem Tabellenkalkulationsprogramm. Er erfasst die Metadaten, die die aufbereiteten Daten mit den Rohdaten verbinden und beschreibt die Art der Aufbereitung.

Mitarbeiter A schreibt eine Veröffentlichung zu den Ergebnissen der Versuchsreihe. Er erstellt ein standardisiertes Dokument. Dabei verwendet er eine Vorlage, die den Layout-Vorgaben der veröffentlichenden Institution entspricht. Gleichzeitig erstellt er mit dieser Vorlage ein Dokument, das sich in einer vordefinierten Struktur im XML-Format abspeichern lässt. Dieses Dokument importiert er wiederum in das SCIFORG-System.

Mitarbeiter B entwickelt im Rahmen eines Nachfolgeprojektes ein mathematisches Modell, welches das von Mitarbeiter A untersuchte System beschreibt. Mitarbeiter B verwendet für Teile des Systems Modellansätze aus der Literatur. Dies betrifft etwa die Beschreibung eines Adsorptionsprozesses innerhalb einer Systemkomponente.

Mitarbeiter B verwendet eine entsprechende Software (ein kommerzielles Simulationsprogramm oder eigens programmierten Code) um auf der Grundlage numerischer Methoden Rechnungen durchzuführen. Dabei erzeugt er Datensätze für eine Modell-Konfiguration. Eine Konfiguration steht dabei für eine bestimmte Parametrisierung.

Zunächst verschafft sich Mitarbeiter B einen Überblick zu den Arbeiten von Mitarbeiter A. Als schriftliche Ressourcen stehen ihm Veröffentlichungen und Berichte seines Vorgängers zur Verfügung. Über das SCIFORG-System kann sich Mitarbeiter B über die Navigation in der Ressourcen-Ontologie vorhandene Ressourcen aus den Ergebnissen von Mitarbeiter A anzeigen lassen. Dazu startet eine Suche für Instanzen der Klasse „Report“. Durch Navigieren in der angezeigten Liste der Berichte und Anzeigen von Einzelinformationen findet Mitarbeiter B heraus, dass die Messungen (nach Meinung von Mitarbeiter A) im Einklang stehen mit einem Ansatz, der auf einem semi-empirischen Modell für den Stofftransport basiert.

Mitarbeiter B startet eine zweite Anfrage, mit deren Hilfe er Ansätze ermittelt, die das betreffende Modell verwenden bzw. erweitern. Er prüft, ob unter den aufgelisteten Ergebnissen ein Modellansatz ist, der sich als Grundlage für die eigene Arbeit eignet. Die Angaben zu jedem der Ansätze geben Aufschluss über Publikationen, Personen und Institutionen.

6.1 Erfassen und Abrufen von Modell-Informationen

Mit Hilfe der vorhandenen Ontologien können Modelle formal beschrieben bzw. klassifiziert werden.

- § Welche Modelle sind Weiterentwicklungen von Modell X?
- § Welche Modelle mit den Merkmalen X,Y sind in Institution A verwirklicht worden?
- § Auf welche Daten stützt sich Ansatz X?
- § [...]

Tatsächlich können in Anfragen sämtliche Informationen ausgenutzt werden, die zusammenhängend modelliert sind. Dabei können sich Anwender in Bezug auf Verbindungen zwischen Ansätzen, Ergebnissen, Ressourcen, etc. auf die sehr generische Ebene des SchoolOnto-Ansatzes stützen, oder (sofern vorhanden) speziellere Kategorien verwenden.

Im Ergebnis kann damit eine „Landkarte“ vorhandener Modelle erstellt und verwendet werden.

6.2 Erfassen und Abrufen von Simulations- und Messdaten (sowie Metadaten)

Die Beschreibung und das Erfassen von Ressourcen wie Messdaten ist nicht gebunden an die Beschreibung von mathematischen Modellen. Das System kann daher auch ausschließlich für die Verwaltung von Mess- und Simulationsdaten eingesetzt werden. Fragestellungen, die explizit abgebildet werden können, wären z.B.:

- § Welche Datensätze hängen zusammen (etwa durch Arbeitsprozesse wie Filtern oder Anreichern, siehe Abschnitt 4.4.1.2 „Ressourcen“ auf Seite 86)?
- § Welche Messergebnisse oder Simulationsdaten wurden in einer spezifischen Veröffentlichung beschrieben? Wo wurden diese Daten noch verwendet?

Darüber hinaus kann das System als Vorstufe (betrifft die Speicherung und Selektierung von Daten) zur Datenbearbeitung mit Tabellenkalkulationsprogrammen, Datamining-Werkzeugen, etc. eingesetzt werden.

6.3 Konsistenzprüfungen für Metadaten

SCIFORG integriert die Inferenzmaschine Algernon (siehe Abschnitt 5.2.3 „Algernon“ auf Seite 110) und stellt graphische Schnittstellen für das Erstellen von Regeln und Abfragen bereit (siehe Abschnitt 5.3.4 „Werkzeuge für Regeln, Suchanfragen und zur Navigation“ auf Seite 129).

Auf dieser Grundlage können regelhafte Bedingungen für die Konsistenz (bzw. die Inkonsistenz) der formalisierten Aussagen über Modelle erstellt werden. Im Folgenden sind einige Beispiele für inkonsistente Beschreibungen von Modellen beschrieben:

Inkonsistenz/Konflikt bzgl. Erweiterung

Erweiterungen können z.B. über die „uses-applies-isEnabledBy“ oder die „improves on“ – Beziehung dargestellt werden. Dies hängt davon ab, welcher Aspekt herausgestellt werden soll. Ob ein Modell eine Verbesserung darstellt ist grundsätzlich abhängig von der Art der Anwendung. So kann sich in einem konkreten Fall eine theoretische bzw. konzeptionelle Verbesserung ergeben, weil eine oder mehrere vereinfachende Annahmen weggelassen werden können, die aber einen hohen Aufwand für die mathematische oder numerische Lösung nach sich zieht.

Grundsätzlich werden mit der hier beschriebenen Methode mögliche Inkonsistenzen bzw. Konflikte bzgl. der Sichtweisen von Modellen (und damit verbundenen Daten) beschrieben.

Beispiel 1:

Modell M_2 stellt eine Erweiterung von Modell M_1 dar. Modell M_1 verwendet Theorie T_1 . Modell M_2 ist wiederum als gegensätzlich zu Theorie T_1 gekennzeichnet. Daraus ergibt sich insgesamt ein Widerspruch.

Beispiel 2:

Modell M_2 stellt eine Erweiterung von Modell M_1 dar. Modell M_1 ist als „empirisch“ klassifiziert, während M_2 als „theoretisch“ eingestuft ist. In diesem Falle ist zu überprüfen, wie ein Modell, auf einem empirischen Modell aufbauend, vollständig theoretisch begründet sein kann. Dies kann z.B. darauf zurück zu führen sein, dass die Modelle auf der gleichen Systembeschreibung basieren und für ähnliche Anwendungsfälle entwickelt wurden. Damit besteht zwischen M_1 und M_2 kein Bezug im engeren Sinne einer Erweiterung. Unter den Anwendern ist in diesem Falle die Interpretation der entsprechenden formalen Beziehung zu klären.

Der umgekehrte Fall ist ebenfalls problematisch. Wenn ein rein empirisches Modell auf einem vollständig theoretischen Modell aufbaut, ist das ein Widerspruch in sich. Möglich ist z.B., dass die Klassifizierung semi-empirisch hätte verwendet werden müssen. Auch hier kann die Konsistenzprüfung den wissenschaftlichen Diskurs stützen bzw. anstoßen.

Bewertung von Konflikten/Inkonsistenzen

Inkonsistente Daten können - wie z.T. bereits erwähnt - unterschiedliche Ursachen haben:

- § Es handelt sich schlicht um einen Fehler bei der Eingabe der Daten. Dies kann auf ein Versehen oder auf eine Fehlinterpretation der Ontologie zurückgehen. Letzteres wäre (sofern viele Inkonsistenzen eintreten) ein Hinweis darauf, dass die Ontologie ihrer Zielgruppe nicht gerecht wird.
- § Die Inkonsistenz geht auf eine unterschiedliche Interpretation der fachlichen Ebene zurück. Aufgrund der Tatsache, dass die Aussagen von unterschiedlichen Anwendern erfasst wurden und es sich zunächst um individuelle Angaben handelt, sind solche Wi-

dersprüche möglich. Sie bedeuten, dass keine Einigkeit bzgl. der formalen Beschreibung der Modelle und ihrer Bezüge zueinander herrscht. Durch die Möglichkeit, die Beschreibungen auszuwerten, können diese Widersprüche explizit gemacht werden.

In jedem Falle bietet die automatisierte Suche nach (möglichen) Inkonsistenzen oder Konflikten eine nützliche Unterstützung:

- § Fehler können leichter identifiziert und ggf. behoben werden;
- § Der wissenschaftliche Diskurs wird unterstützt, da unterschiedliche Sichtweisen leichter explizit bzw. transparent gemacht werden können.

Die Suche nach Inkonsistenzen bzw. das Erstellen von Regeln zum Definieren derselben zeigt, dass eine „schlanke“, generische Ontologie als Basis -wie das SchoolOnto-Modell- auch den Vorteil bietet, dass Zusatzwissen ebenfalls sehr generisch formuliert werden kann.

7 Fazit und Ausblick

Die nachhaltige Verfügbarkeit wissenschaftlicher Ergebnisse und der Zugriff auf dieselben stellt eine wachsende Herausforderung dar. Bereits verfügbare Technologien, die eine schnelle und hoch skalierbare Volltextsuche in verteilten Systemen ermöglichen, werden dieser Problematik nicht vollständig gerecht. Aufgrund der Mehrdeutigkeit der natürlichen Sprache und des Variantenreichtums zur natürlich-sprachlichen Beschreibung von Sachverhalten sind diesem Ansatz Grenzen gesetzt. Es handelt sich nicht um eine formal interpretierbare Suche, die auch Zusammenhänge berücksichtigt. Die Klassifizierung und Zuordnung von Forschungsergebnissen zueinander erschließt sich nur durch die Interpretation der Texte selbst – die wiederum der Anwender vornehmen muss. Bei einer großen Anzahl von Ressourcen und einem hohen Grad an Spezialisierung ergibt sich die Schwierigkeit, Ergebnisse in Teilbereichen eines Fachgebietes gezielt herauszusuchen.

Das SCIFORG-System bietet einen Ansatz zur Verwaltung von Forschungsergebnissen im Bereich der Prozesswissenschaften, der es Anwendern ermöglicht, diese auf der Grundlage ihrer Beschreibung aufzufinden und einzuordnen. Die Beschreibung der entsprechenden Ressourcen ist maschinenlesbar angelegt, so dass in Suchanfragen im Gegensatz zur reinen Stichwortsuche explizite Zusammenhänge ausgenutzt werden können. Grundlage hierzu sind der Ontologie-Ansatz und die damit verbundenen semantischen Technologien. Sie erlauben es, Domänenmodelle maschineninterpretierbar zu beschreiben. Die formale Semantik von Ontologiesprachen ermöglicht den Austausch und die Wiederverwendung von Datenmodellen.

Die vorliegende Arbeit versteht sich auch als Transferleistung. Semantische Technologien bieten grundsätzlich die Möglichkeit, Informationen maschinenlesbar zu erfassen und abzufragen. Dabei sind sie –im Gegensatz zu relationalen Datenbanken oder XML- von vornherein so angelegt, dass die zugrunde liegenden Datenmodelle für Anwender nachvollziehbar bleiben. Die vorliegende Arbeit übernimmt diesen Ansatz für die Prozesswissenschaften und erweitert ihn auf der Ebene der technischen Umsetzung, um den domänen-spezifischen Anforderungen gerecht zu werden. Für das Erstellen und Auswerten von Ontologien steht eine Reihe von Werkzeugen zur Verfügung, die jedoch in erster Linie eine generische Unterstützung für den „Knowledge-Engineer“ bieten. Das SCIFORG-System hält Funktionalitäten bereit, die speziell Anwender aus den Prozesswissenschaften bedienen und kapselt dabei die Komplexität der zugrunde liegenden Technologie. Als integriertes System lässt es sich ohne weitere Anpassungen einsetzen. Welcher Teil der Funktionalität genutzt wird, bleibt Anwendern überlassen. Dadurch gliedert es sich in die vorhandene Palette von Werkzeugen zur Datenauswertung und Datenaufbereitung ein, ohne bestimmte Prozessketten oder Arbeitsabläufe zwingend vorauszusetzen.

SCIFORG ist damit das erste integrierte System, das Anwendern der Prozesswissenschaften eine Plattform zur wissensbasierten Verwaltung von Forschungsergebnissen auf der Basis von Ontologien bietet – einschließlich domänenspezifischer Funktionalitäten wie die Verwaltung physikalischer Einheiten oder die Datenvisualisierung.

Die Frage der Nachvollziehbarkeit, der Akzeptanz und der effizienten Nutzung von Datenmodellen zur Beschreibung wissenschaftlicher Ressourcen ist allerdings nicht rein technischer Natur. Um einen einfachen Einstieg in das Prinzip der Ontologie-basierten Ressourcenverwaltung zu gewährleisten, sollten möglichst „leichtgewichtige“ Modelle zur Verfügung stehen, die nach Bedarf erweitert werden können. Der SchoolOnto-Ansatz bietet einen guten Ausgangspunkt für die Beschreibung wissenschaftlicher Daten. Im Gegensatz zu anderen Ontologie-basierten Ansätzen im Bereich der Modellentwicklung, der Verfahrenstechnik oder verwandten Gebieten wird in dieser Arbeit der Schwerpunkt auf die Beschreibung von Ressourcen in Form von Messdaten, Simulationsdaten und Dokumenten gelegt. Als zentraler Ansatzpunkt für eine Anwendung und Erweiterung des SchoolOnto-Ansatzes eignet sich der Bereich mathematischer Modelle. Viele der in SchoolOnto generisch formulierten Bezüge lassen sich auf mathematische Modelle und damit verbundene Datensätze, Dokumente, etc. anwenden und verfeinern.

Wie in Abschnitt 3.4 deutlich geworden ist, handelt sich bei der Modellbildung um einen Lernprozess. Dieser ist nicht allein als individueller Prozess zu sehen, sondern als einer, der Gruppen, Institutionen und Fachdisziplinen betrifft. Den wissenschaftlichen Diskurs zu unterstützen und dabei eine Sicht auf wissenschaftliche Ergebnisse zu ermöglichen, die Zusammenhänge berücksichtigt stellt einen wesentlichen Beitrag zur Bewältigung des steigenden Informationsaufkommens gerecht zu werden.

Damit bietet SCIFORG eine Möglichkeit, das ontologie-basierte Anreichern von wissenschaftlichen Ressourcen direkt umzusetzen, ohne das Verständnis komplexer Modelle vorauszusetzen. Die Erweiterung der vorhandenen Modelle kann jedoch einen zusätzlichen Mehrwert bringen. So wäre es beispielsweise möglich, die mathematische Beschreibung formal abzubilden und auszuwerten. Damit könnten Informationen über die Beschaffenheit eines Modells (z.B. die Klassifizierung als „stationär“) aus der Beschreibung der Parameter und der Repräsentation der Gleichungen abgeleitet werden. In diesem Zuge könnten mathematische Modellrepräsentationen direkt aus Werkzeugen übernommen werden und anschließend Ontologie-basiert angereichert werden. Auch wäre es möglich, detailliertere Beschreibungen mathematischer Modelle und verwandter Bereiche (analog zu den Vorarbeiten von Morbach et al., 2007) zu integrieren und damit präzisere Darstellungen von Domänenkonzepten zu erreichen.

Die SCIFORG-Implementierung zeigt, dass es möglich ist, mit vorhandenen Werkzeugen im Bereich von semantischen Technologien, Datenbanken und Datenvisualisierung eine Unterstützung zur ontologie-basierten Verwaltung von Forschungsergebnissen in Form eines integrierten Systems anzubieten. SCIFORG bietet Anwendern bereits bei der Verwendung als Einzelplatzsystem einen Mehrwert in Bezug auf die Organisation von Informationen und Daten.

Auf der Seite der software-technischen Umsetzung gezeigt, dass das gewählte System Protégé eine gute Grundlage für ein domänen-spezifisches Werkzeug bietet. Allerdings war ein relativ hoher Aufwand bzgl. der Erweiterung für eine höhere Skalierbarkeit und eine effiziente Unterstützung physikalischer Einheiten erforderlich. Dieser Aufwand könnte sich zukünftig verringern,

sofern semantische Technologien an Popularität gewinnen und domänen-spezifische Ausprägungen von Werkzeugen in größerem Ausmaß verfügbar werden.

Im Rahmen dieser Arbeit wurden der konzeptionelle Aspekt und der technische Aspekt bzgl. der Beschreibung und des Auffindens wissenschaftlicher Ressourcen behandelt. Damit wurde eine Daten- bzw. Informationssicht zugrunde gelegt. Ein gleichfalls wichtiger Aspekt betrifft die Prozesssicht. Das SCIFORG-System verzichtet –wie bereits erwähnt- bewusst darauf, bestimmte Abläufe in der wissenschaftlichen Arbeit zwingend vorzuschreiben. Unter Beibehaltung dieses Ansatzes könnte ein explizites Einbeziehen der Prozesssicht den Nutzen des Ansatzes verstärken. Diese könnte sich direkt am zentralen Thema Modellentwicklung orientieren. Wie in Abschnitt 3.4 beschrieben, stellt die Entwicklung mathematischer Modelle einen iterativen Prozess dar. Indirekt wird die Prozesssicht bereits jetzt abgebildet, indem die Verwendung vorhandener Ansätze und Ergebnisse beschrieben wird. Durch eine explizite Beschreibung von Prozessen und durch eine entsprechende Unterstützung im System (etwa im Rahmen von „Wizards“ oder Assistenzsystemen mit speicherbaren Zuständen) könnte eine verstärkte Führung des Anwenders durch bestimmte Arbeitsvorgänge entwickelt werden. Die Prozesssicht müsste für eine Etablierung des SCIFORG-Ansatzes auf breiter Ebene betrachtet werden. Dies betrifft etwa die Vergabe und Abwicklung von Forschungsprojekten, verbunden mit Anreizsystemen für die nachhaltige Verwaltung von Daten.

SCIFORG ist als Desktop-Werkzeug ausgelegt, das in einem gewissen Umfang mit Mehrbenutzerfähigkeit ausgestattet ist. Die gemeinsame Nutzung erfolgt durch Einrichten eines zentral zugängigen Datenbankservers. Damit sind der Kooperation über die Grenzen von Institutionen hinweg Grenzen gesetzt, etwa durch die Sicherheitsproblematik. Darüber hinaus geht das SCIFORG-System davon aus, dass für Dokumente, die außerhalb des Systems verwaltet werden, aber dennoch referenziert werden sollen, ein Dokumentenserver zur Verfügung steht. Für die institutionsübergreifende Kooperation müsste sowohl für Dokumente als auch für Datensätze eine Lösung etabliert werden, die von Grund auf eine verteilte Architektur voraussetzt. Webservice-Technologien und so genannte Datengrids bieten Ansätze für den verteilten Datenzugriff, die auch das Sicherheitsproblem berücksichtigen.

Der beschriebene Ansatz verwendet eine Frame-basierte Wissensrepräsentation. Diese kann zu einem großen Teil auf Standards wie RDF(S) abgebildet werden. Die Unterstützung der Ontologiesprachen RDF(S) und OWL ist Schwerpunkt für die meisten kommerziellen und akademischen Entwicklungen im Bereich Semantischer Technologien. Insofern bleibt zu untersuchen, inwieweit diese Sprachen geeignete Paradigmen für die Beschreibung und „Kapselung“ von Daten im Bereich der Prozesswissenschaften im Sinne des SCIFORG-Ansatzes darstellen und welche Werkzeug-Unterstützung notwendig ist. Die bestehenden Import- und Export-Möglichkeiten des Systems erlauben in eingeschränkter Form bereits den Datenaustausch im RDF(S)-Format.

Das SCIFORG-System bietet in beschränktem Umfang die Möglichkeit, vorhandene Datenbanken einzubinden. Dabei wird die Struktur der Datenbank auf eine bestimmte Art interpretiert und auf das Frame-basierte Datenmodell abgebildet. Für viele Systeme kann sich dieser Ansatz

als zu starr erweisen. Zwar ist es prinzipiell möglich, mit regelbasierten Auswertungs- und Anfragesystemen wie Algernon die resultierende Struktur auf eine neue Struktur abzubilden. Jedoch muss dieser Prozess einhergehen mit entsprechenden Optimierungen des Auswertungsprozesses (internes Umschreiben von Regeln, Minimieren von Datenbankabfragen). Darüber hinaus umfassen mögliche Datenquellen für wissenschaftliche Ressourcen mehr als die hier unterstützten relationalen Datenbanken und Text- sowie Tabellen-Dokumente. Hier ist ein genereller Integrationsansatz gefragt. Dieser würde jedoch den Rahmen dieser Arbeit sprengen. Methoden und Technologien für die semantische Datenintegration existieren jedoch bereits (vgl. Wache et al., 2001) und können für Weiterentwicklungen des SCIFORG-Ansatzes und des SCIFORG-Systems übernommen werden.

Anhang

Dieser Abschnitt beschreibt einige der wesentlichen Schritte zur Installation und Konfigurierung des SCIFORG Systems und die Verwendung der Dokumentenvorlage. Er versteht sich jedoch nicht als Ersatz für ein vollständiges Handbuch.

A 1 Konfigurieren des SCIFORG-Systems

A 1.1 Installation und Konfigurierung

SCIFORG ist zu 100% in Java implementiert. Dementsprechend ist das System auf allen Plattformen lauffähig, die Java mindestens in der Version 1.4 unterstützen. Darüber hinaus muss eine Java-Laufzeitumgebung bereits installiert sein und eine der unterstützten relationalen Datenbanksysteme lokal oder über das Netzwerk zur Verfügung stehen.

Zur Installation ist es lediglich erforderlich, SCIFORG bei Beibehaltung der Verzeichnisstruktur auf den Zielrechner zu kopieren. Anschließend kann das System über die Java-Laufzeitumgebung gestartet werden. Ein entsprechendes Skript enthält bereits den erforderlichen Aufruf.

Wenn SCIFORG keine oder keine vollständige Konfigurationsdatei findet, wird der Anwender zur Konfiguration aufgefordert (siehe Abbildung 58).

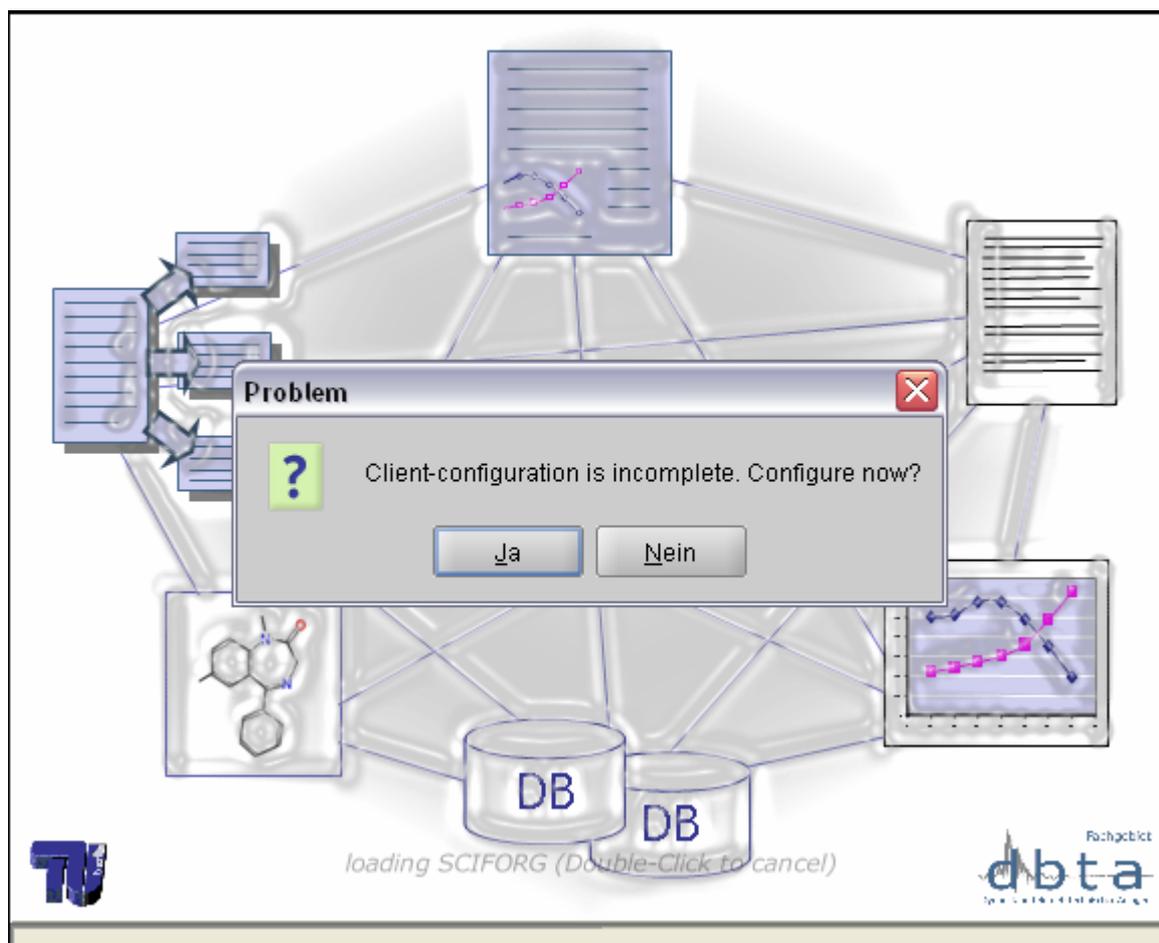


Abb. 58: SCIFORG Splash-Screen und Aufforderung zur Konfiguration

Zur Konfiguration sind in erster Linie Metadaten zur Datenbankverbindung erforderlich (siehe Abbildung 59).

Configure SCIFORG-Client

Database Connection Parameters

Url

User Name Password

Driver File

Choose file:

JDBC-Driver

Type of Driver

Test Connection

Additional Parameters (Test Connection first)

Default varchar length

Default type for XXL-Strings

Additional Column-SQL (e.g. byte-size for CLOB)

Storage Parameters

Project Table Name Classes Table Name

Abb. 59: SCIFORG Konfigurations-Dialog

Neben den Angaben zur Datenbankverbindung muss der Datentyp für überlange Zeichenketten bestimmt werden.

Der Datenbanktreiber wird durch die Auswahl eines Java-Archivs und die anschließende Angabe eines Treiber-Typs spezifiziert (siehe Abbildung 60). Letzterer stellt sicher, dass Datenbankabfragen und –kommandos korrekt generiert werden. Dies geschieht zur Laufzeit im Wesentlichen durch die SimpleORM-Bibliothek (siehe Abschnitt 5.2.2 „SimpleORM“ auf Seite 109).

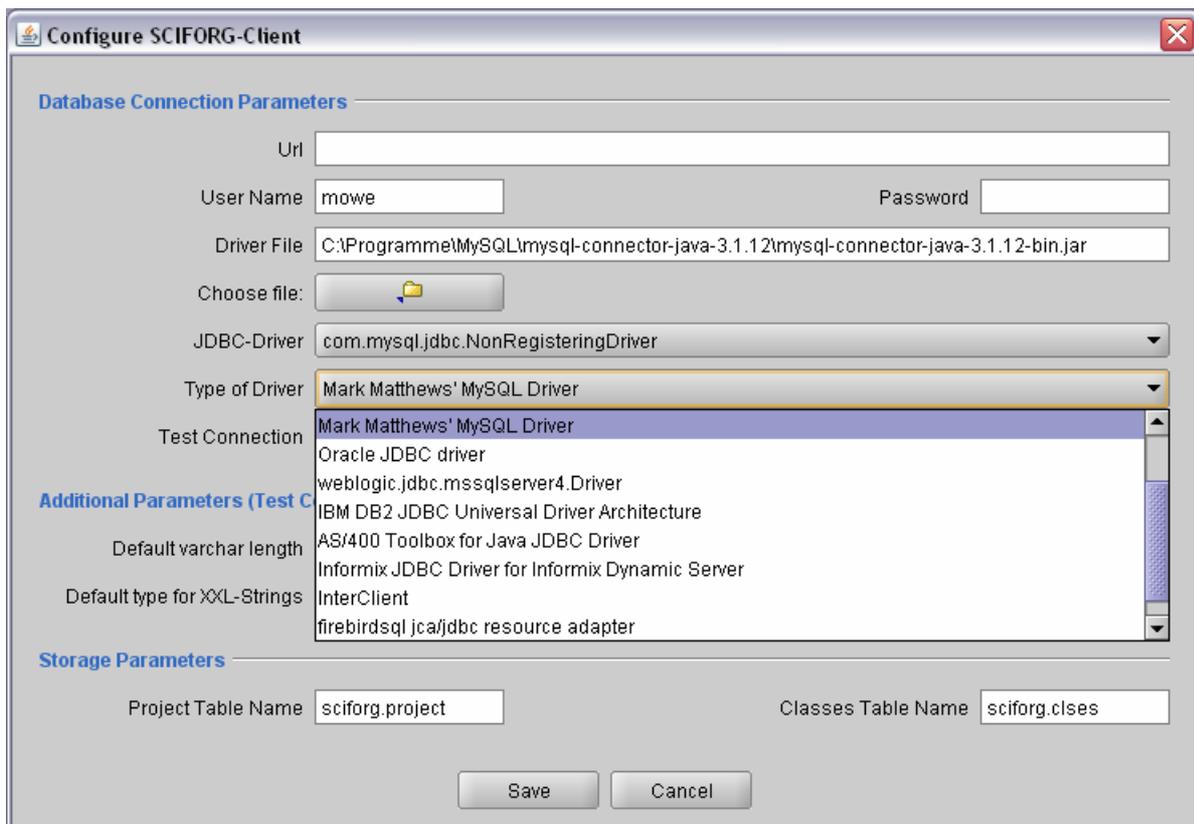


Abb. 60: Auswahl des Datenbank-Treibers

A 1.2 Erweitern des Kernmodells

Grundsätzlich ist es möglich, die vordefinierten Ontologien von SCIFORG zu erweitern und zu verändern. Einschränkungen sind nur für die Teile der Ontologie gegeben, die über ein objektrelationales Mapping an eine Datenbank gebunden sind (siehe auch Abschnitt 5.3.1.2 „Schema der Objekt-relationalen Abbildung“ auf Seite 117). So wird für diese Teile der Ontologie beispielsweise keine Mehrfachvererbung unterstützt. Zudem sind Änderungen nur in beschränktem Umfang und in Abhängigkeit von der zugrunde liegenden Datenbank möglich.

A 1.3 Einstellungen für das Objektrelationale Mapping

Erstellen von Datenbank-Mappings und Anzeige von Mapping-Eigenschaften

Bestehende Klassen können halb-automatisch auf eine Datenbank-Tabelle abgebildet („gemappt“) werden (vgl. Abschnitt 5.3.1 Skalierbare objektrelationale Speicherschicht auf Seite 113). Diese muss sich nicht notwendiger Weise in derselben Datenbank oder im gleichen Schema befinden, wie die Systemtabellen von SCIFORG. Voraussetzung für ein Mapping ist

- § dass keine Mehrfachvererbung verwendet wird;

- § dass keine mehrfachen Wertebereiche möglich sind, d.h. dass jeder Slot entweder vom Wertebereich Zeichenfolge, oder Zahl, etc. ist und im Falle von Instanzen als Wertebereich nur eine mögliche Klasse angegeben ist;
- § dass die Oberklasse entweder über kein Datenbank-Mapping verfügt oder auf eine Tabelle im gleichen Datenbankschema abgebildet wurde
- § und dass die zu erzeugenden Tabellen und Spalten nicht bereits existieren.

Für letzteren Fall kann ein so genanntes „reverse mapping“, d.h. das Erzeugen von Klassen und Slots aus einem Datenbankschema durchgeführt werden.

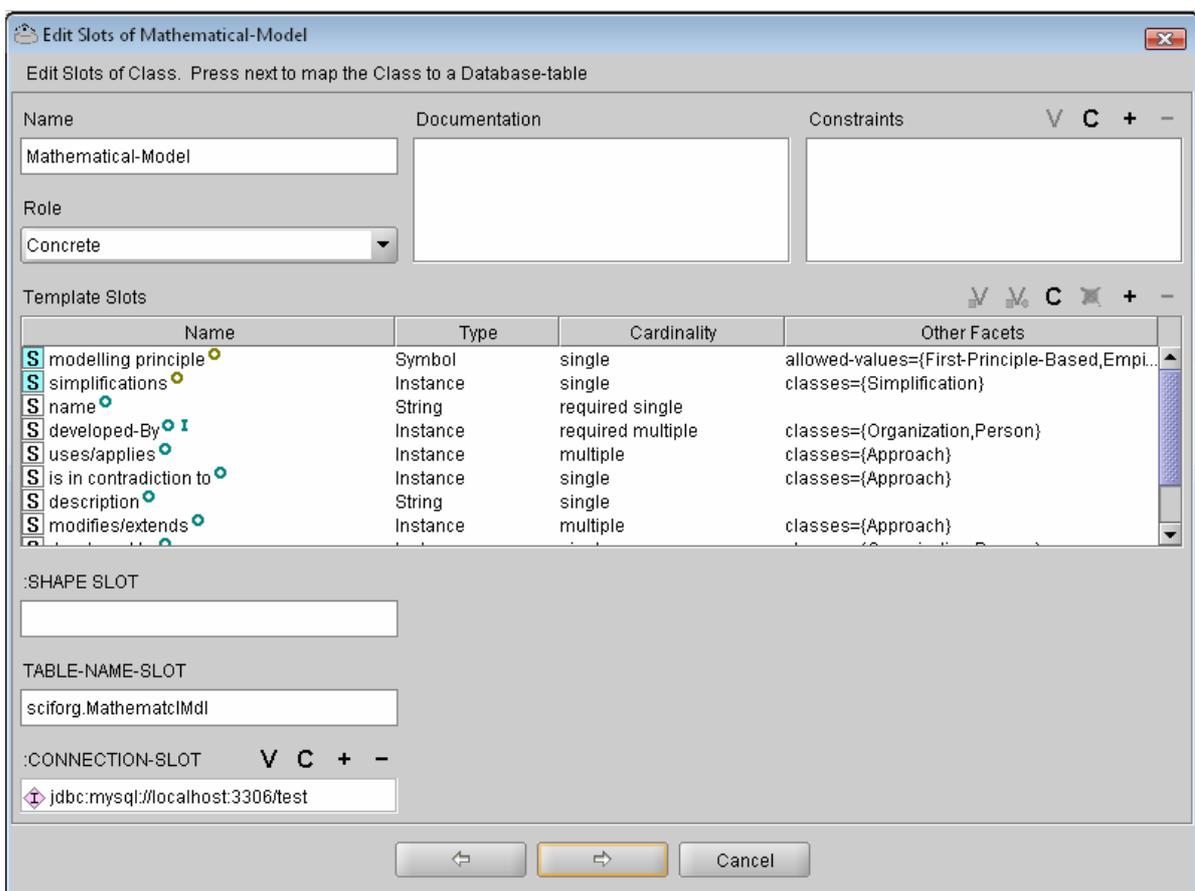


Abb. 61: Dialog zum Einstellen von Mapping-Parametern

Um eine Klasse auf eine Tabelle abzubilden, wird der entsprechende Dialog über die Schaltfläche mit dem -Symbol gestartet. Anschließend kann das Mapping konfiguriert werden (siehe Abbildung 61). Dies betrifft u.a. die Auswahl (evtl. auch das Anlegen von) einer Datenbank-Verbindung, den Namen der zu erzeugenden Tabelle und die Einstellungen für die zu generierenden Spalten. Letztere können angepasst werden, indem das Formular des entsprechenden Slot aus der „Template Slots“-Tabelle aktiviert wird (durch einen Doppelklick auf die jeweilige Zeile).

Sobald die Konfiguration bestätigt wird, generiert das System den entsprechenden SQL-Befehl für die Datenbank und zeigt diesen an (Abbildung 62).

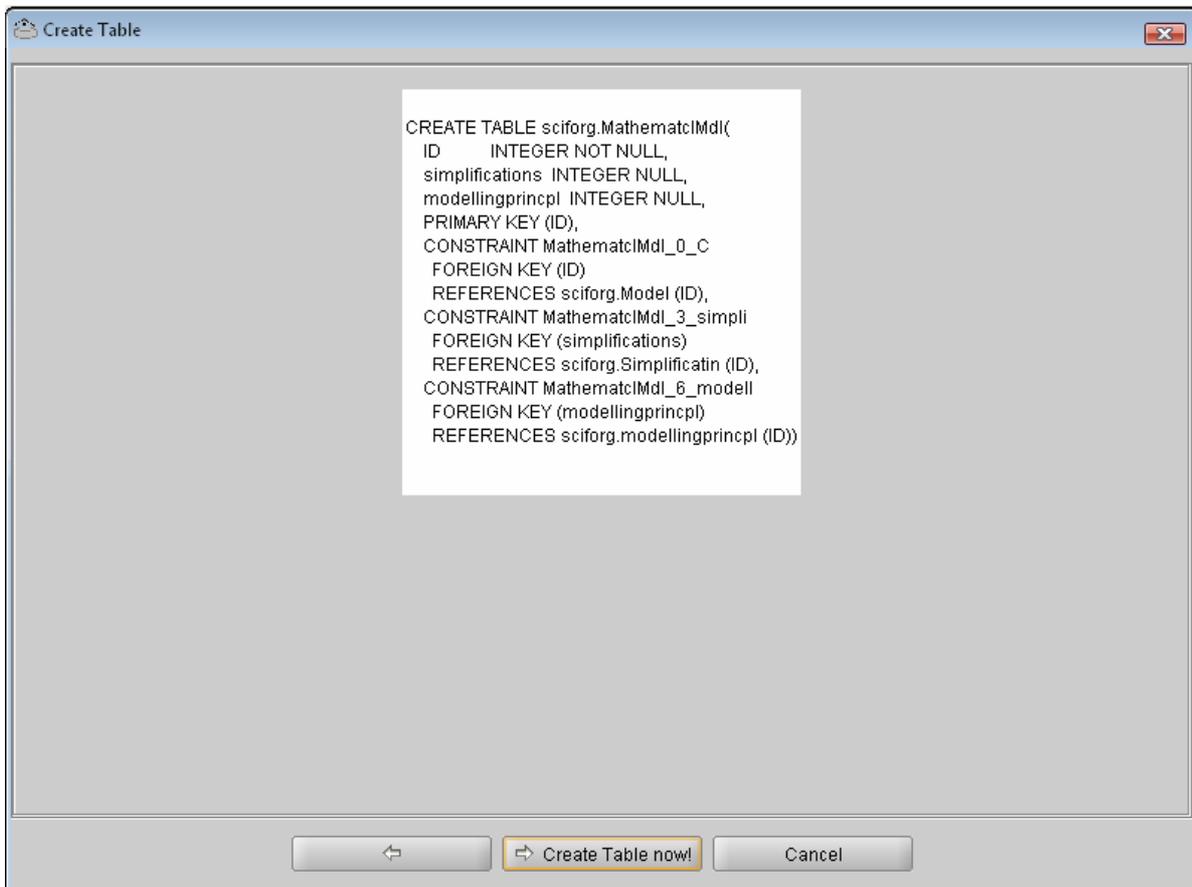


Abb. 62: Anzeige des automatisch generierten SQL-Befehls zum Anlegen einer Tabelle für die objektrelationale Abbildung.

General Mapping Parameters

Create new DB-Connection

Use DB-Connection

Auto-Mapping

General Class Parameters

:CONNECTION-SLOT V C + - TABLE-NAME-SLOT

Slot-Mapping

Slot-Name	Is mapped	Mapping-Type	VALUE-COL...	TABLE-NAM...	:COLUMN-P...	FOREIGN-K...	:KEY-FOR-S...	INTEGRITY-...	KEY-F...
uses/applies	<input checked="" type="checkbox"/>	m:n	usesapplies	sciforg.Appr...	[not required]		Approach	<input checked="" type="checkbox"/>	<input type="checkbox"/>
is in contrad...	<input checked="" type="checkbox"/>	1:n	isincontradi...	[not required]	[not required]	[not required]	[not required]	<input checked="" type="checkbox"/>	<input type="checkbox"/>
description	<input checked="" type="checkbox"/>	column	description	[not required]	[not required]	[not required]	[not required]	<input checked="" type="checkbox"/>	<input type="checkbox"/>
modifies/ext...	<input checked="" type="checkbox"/>	m:n	modifiesext...	sciforg.Appr...	[not required]		Approach	<input checked="" type="checkbox"/>	<input type="checkbox"/>
developed by	<input checked="" type="checkbox"/>	1:n	developedby	[not required]	[not required]	[not required]	[not required]	<input checked="" type="checkbox"/>	<input type="checkbox"/>
is an examp...	<input checked="" type="checkbox"/>	m:n	isanexampl...	sciforg.Appr...	[not required]		Approach	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Abb. 63: Anzeige von Datenbank-Mapping Eigenschaften für eine ausgewählte Klasse

Für erstellte objektrelationale Abbildungen können die Mapping-Eigenschaften über die „ORM-Settings“ angezeigt werden. Die ORM-Settings werden über die Schaltfläche mit dem  Symbol aktiviert. Dies ist beispielhaft in Abbildung 63 dargestellt.

Protokollieren von Datenbankzugriffen



Abb. 64: Anzeige von Datenbankzugriffen

Datenbankzugriffe können im System angezeigt werden. Dies geschieht über den „ORM-Settings“ Dialog, der über die Werkzeugleiste aufgerufen werden kann. Wie in Abbildung 63 gezeigt wird, kann eingestellt werden, welche Arten von Datenbankzugriff angezeigt werden sollen.

A 1.4 Dokumenten-Import

Für den Import von Word™-Dokumenten müssen die folgenden Voraussetzungen erfüllt sein:

- § Die Dokumente müssen im Richtext-Format vorliegen (in MS Word™ über das Menü „Datei“-> „speichern unter“ erreichbar);
- § Der Dokumententyp, für Word-Dateien „simplified docbook“, muss als Klassenstruktur vorhanden sein, so dass die Datei als Menge von Instanzen im System abgelegt werden kann.

Der zweite Schritt lässt sich so vorkonfigurieren, dass das System bereits mit den entsprechenden Klassen ausgeliefert wird. Zu Demonstrationszwecken ist im folgenden Unterabschnitt beschrieben, wie diese Klassenstruktur hergestellt wird.

Dokumententypen-Import

Sämtliche Elemente eines Dokumentes, wie etwa Abschnitte oder Tabellen, werden einer Klasse zugeordnet. Die entsprechenden Klassen werden aus der Dokumententypen-Definition (DTD) automatisch angelegt. Dabei werden keine Vererbungsbeziehungen erzeugt, d.h. alle Elemente

befinden sich in der gleichen Hierarchieebene. Allerdings wird eine gemeinsame Oberklasse für alle Elemente festgelegt. Im folgenden Beispiel wird davon ausgegangen, dass dies die Klasse „DocumentItem“ ist.

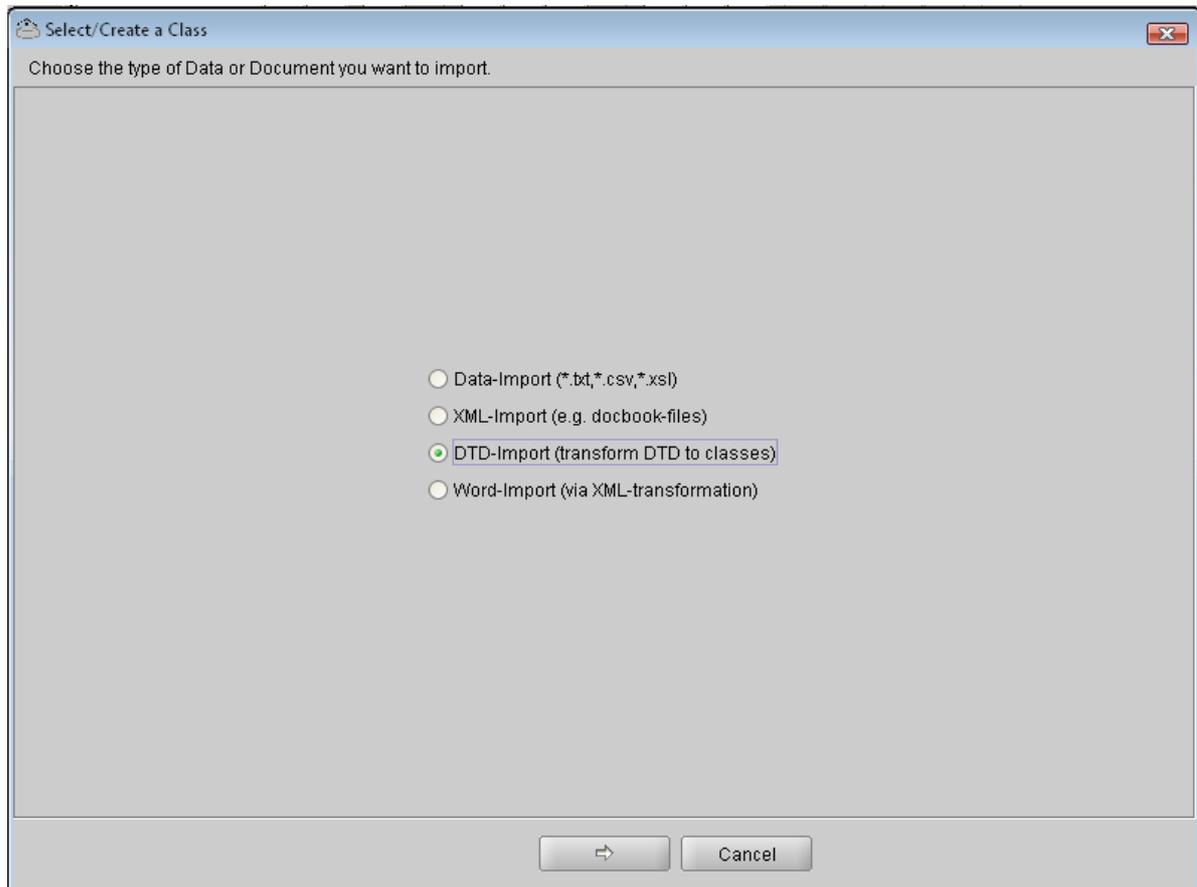


Abb. 65: Auswahl des Dokumententypen-Imports

Zunächst wird die Import-Auswahl über die Schaltfläche mit dem Import-Symbol in der Werkzeugleiste gestartet. Anschließend wird der DTD-Import gewählt, wie in Abbildung 65 gezeigt.

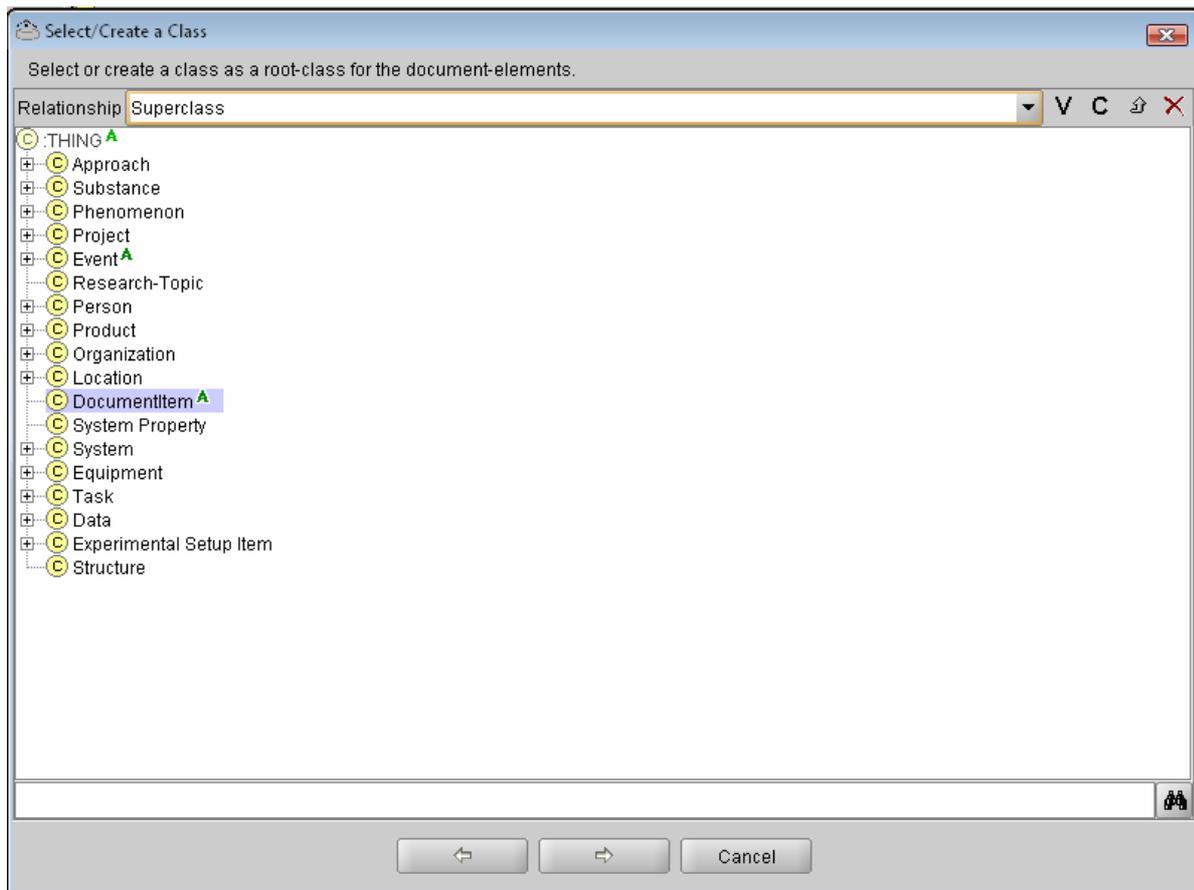


Abb. 66: Auswahl der Oberklasse für Dokumenten-Elemente

Im nächsten Schritt wird die Oberklasse gewählt, unter der die Klassen für Dokumenten-Elemente entsprechend der Dokumententypen-Definition angelegt werden sollen (siehe Abbildung 66). Anschließend werden alle erforderlichen Klassen und Attribute erzeugt und sind im System verfügbar (siehe Abbildung 67).

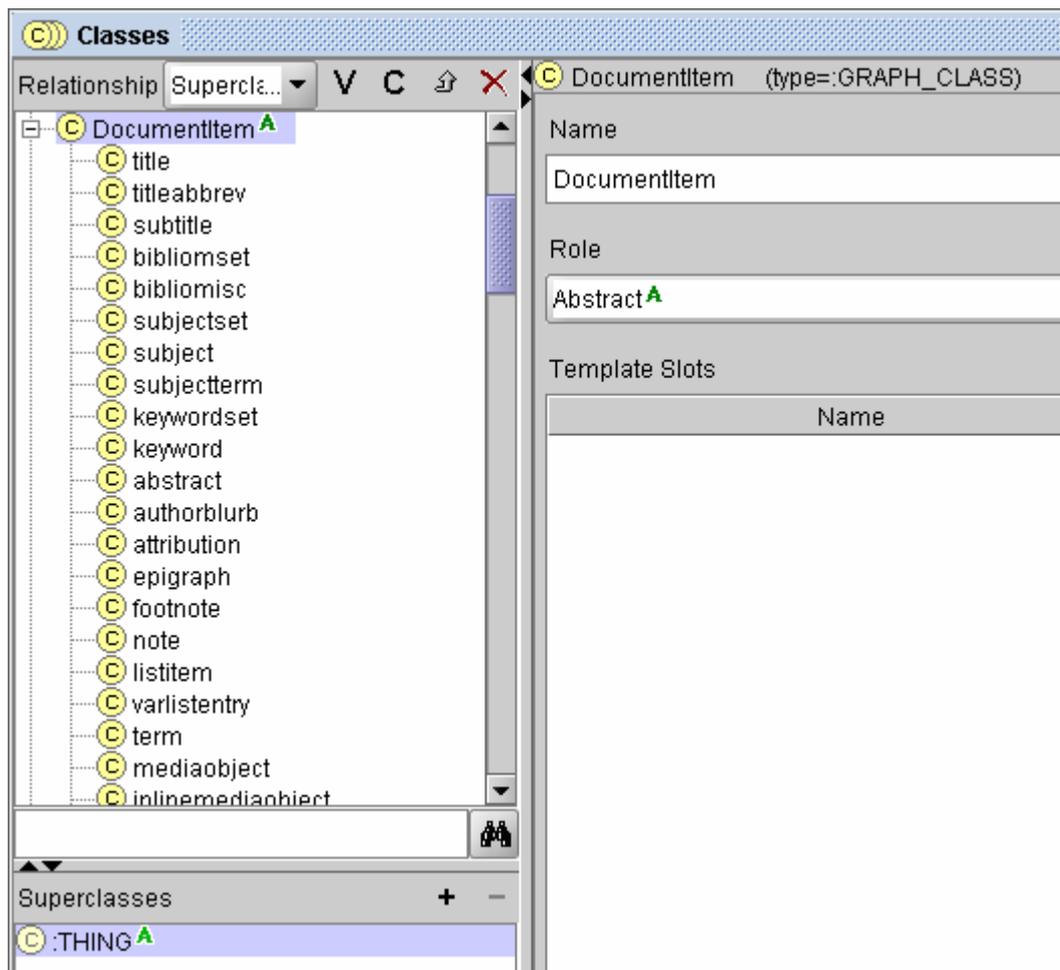


Abb. 67: Aus einer DTD automatisch erzeugte Klassen

Für die Anzeige von importierten Dokumenten im docbook-Format steht ein spezielles „Widget“ zur Verfügung. Dieses muss über die Bearbeitung der von generierten Protégé Formulare dem „content“-Slot der anzuzeigenden Elemente zugewiesen werden. Dabei wird die Standard-Funktionalität von Protégé zum Anpassen von Formularen verwendet. Das spezielle Widget wird vom SCIFORG-System bereitgestellt (siehe Abbildung 64).

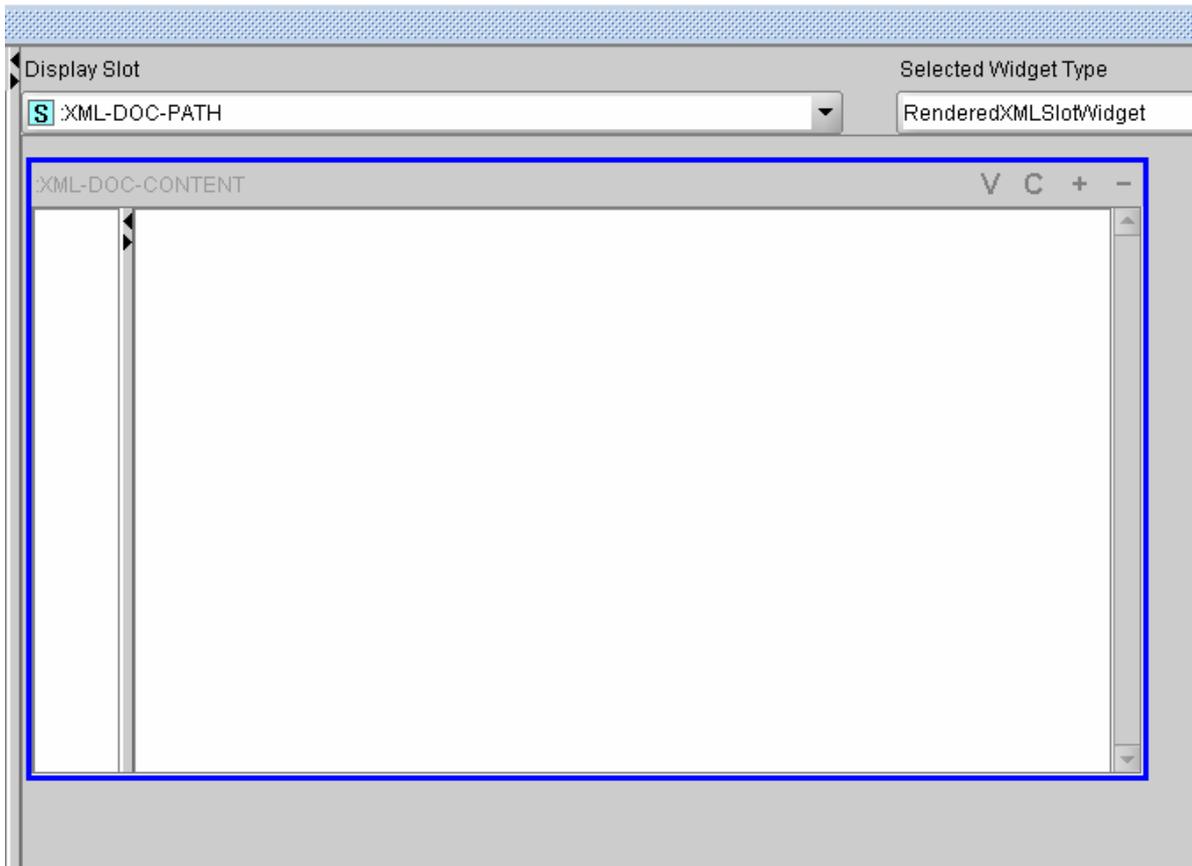


Abb. 68: Spezielles Widget für die Anzeige von importierten XML-Dateien

Import von Richtext-(Word-)Dokumenten

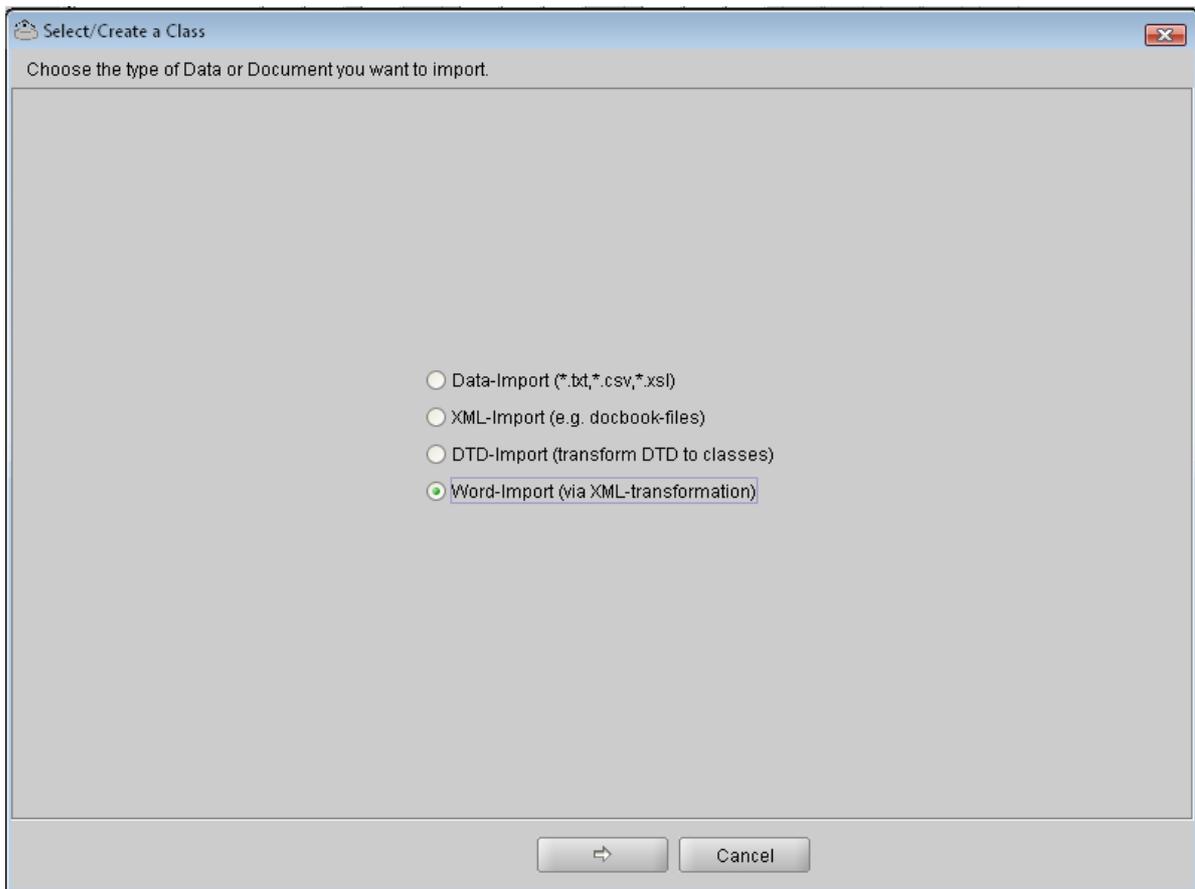


Abb. 69: Dokumenten-Import Schritt 1: Auswahl des Import-Wizards

Nach dem Aktivieren der Import-Auswahl wird der Word-Import gewählt, wie in Abbildung 69 gezeigt

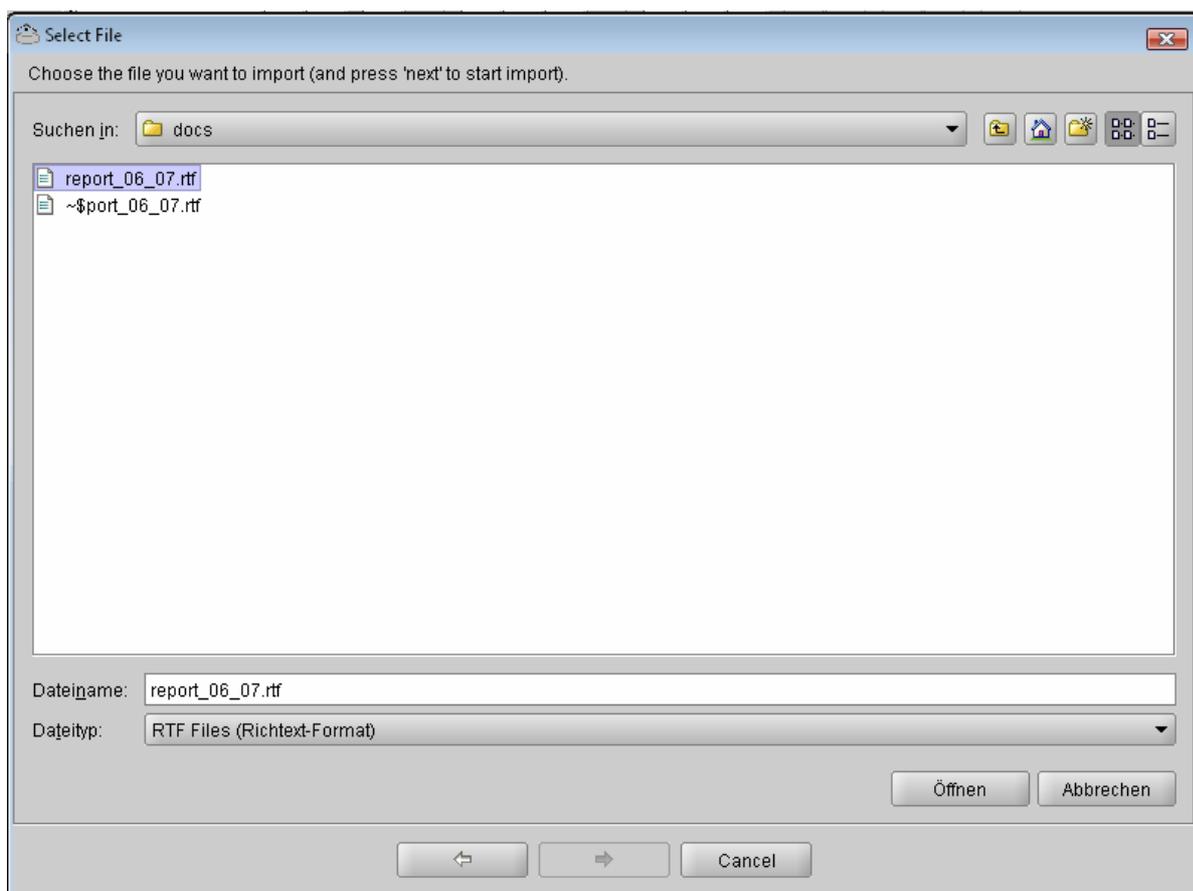


Abb. 70: Dokumenten-Import Schritt 2: Auswahl einer Richtext-Datei

Im folgenden Schritt wird die zu importierende Datei gewählt. Der Dateiname muss die Endung „.rtf“ haben.

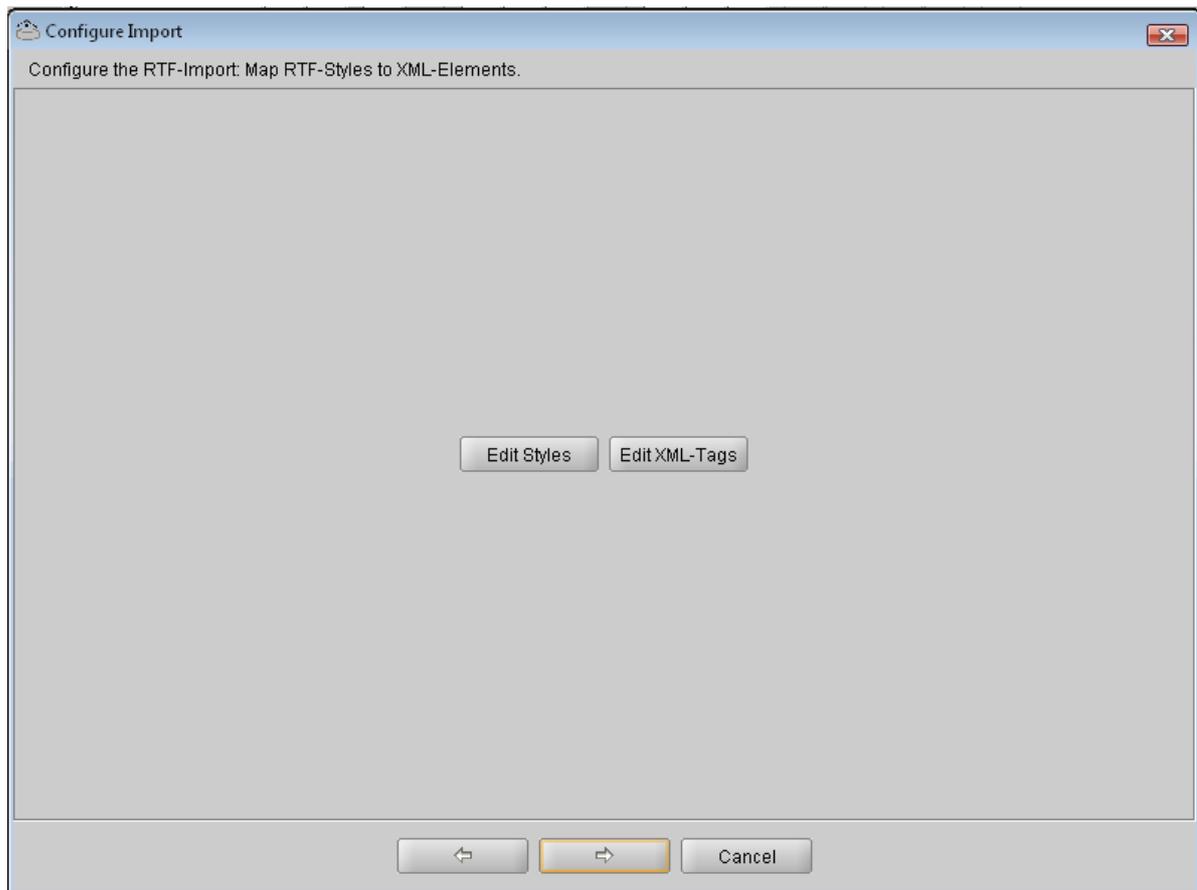


Abb. 71: Dokumenten-Import Schritt 3: Anpassen des Imports

Bei Bedarf kann der Import angepasst werden. Dies betrifft die Umwandlung von Formatvorlagen in XML-Primitive und deren Bezeichnung. Die Funktionalität wird im Wesentlichen durch das Majix-System bereitgestellt und wurde in SCIFORG nur leicht erweitert.

Abbildung 72 zeigt den Dialog zur Anpassung der Behandlung von Formatvorlagen in Majix. Abbildung 65 zeigt den entsprechenden Dialog für die Bezeichnung von XML-Tags. Bei einem vollständig vorkonfigurierten SCIFORG-System ist diese Anpassung nicht erforderlich, es sei denn, der Word™-Formatvorlage wurden weitere Absatz- oder Zeichenformate hinzugefügt.

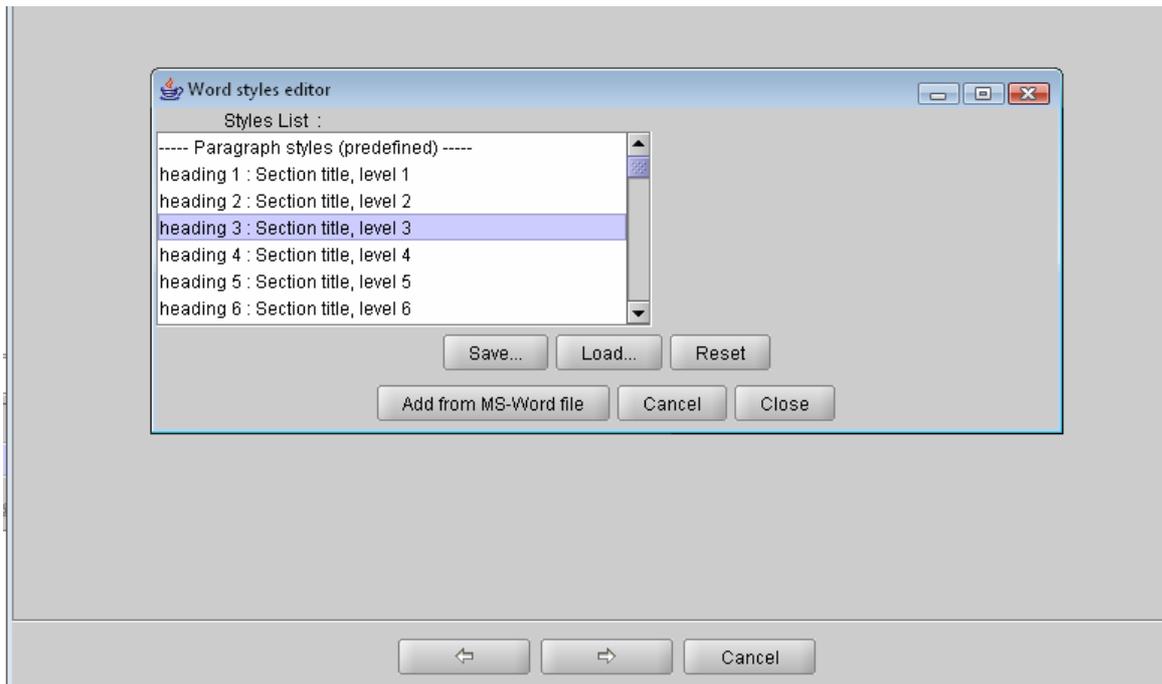


Abb. 72: Dokumenten-Import: Anzeige des Majix Konfigurations-Dialoges zum Editieren von Formatvorlagen

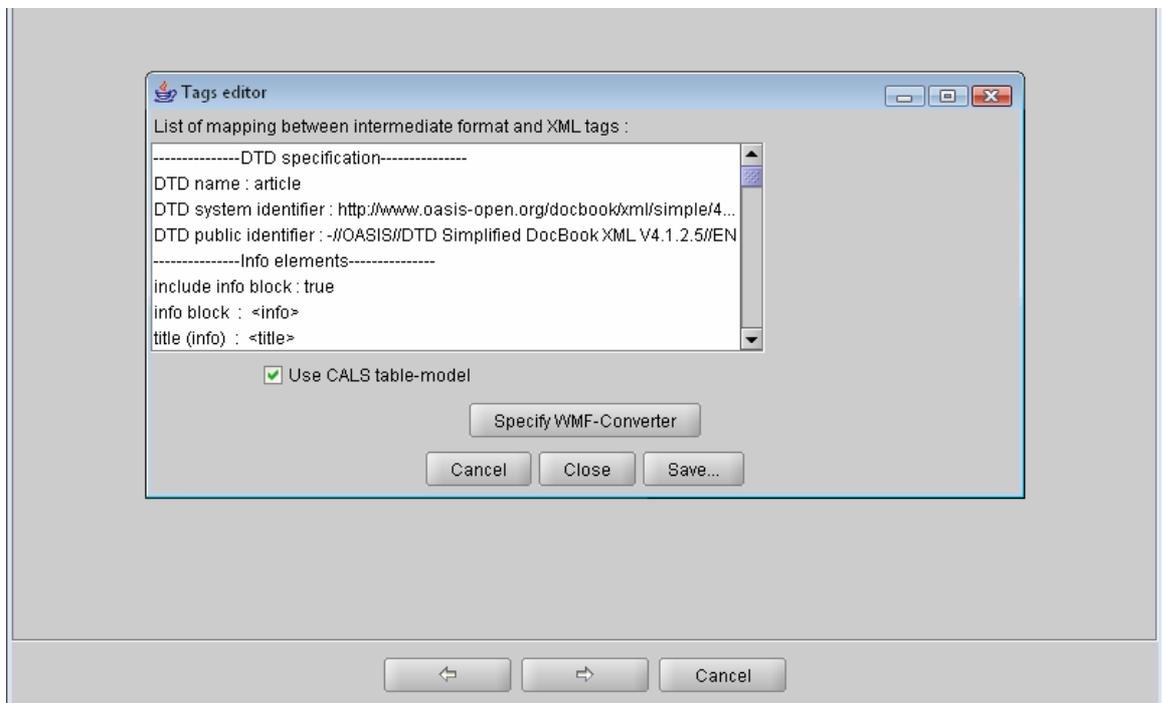


Abb. 73: Dokumenten-Import: Anzeige des Majix Konfigurationsdialoges zum Editieren von XML-Tags

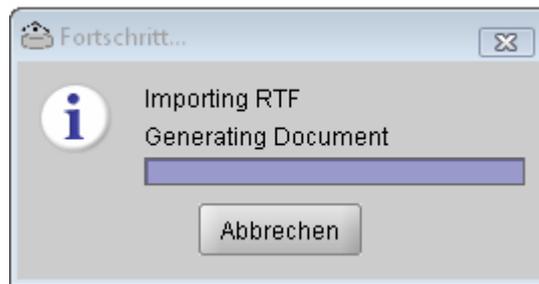


Abb. 74: Dokumenten-Import Schritt 4: Statusanzeige für den Dokumentenimport

Nach Bestätigen von Schritt 3 wird der Import des Dokumentes gestartet. Abbildung 75 zeigt beispielhaft die Darstellung von Dokumenten im SCIFORG-System über das XML-Widget (siehe vorhergehender Abschnitt).

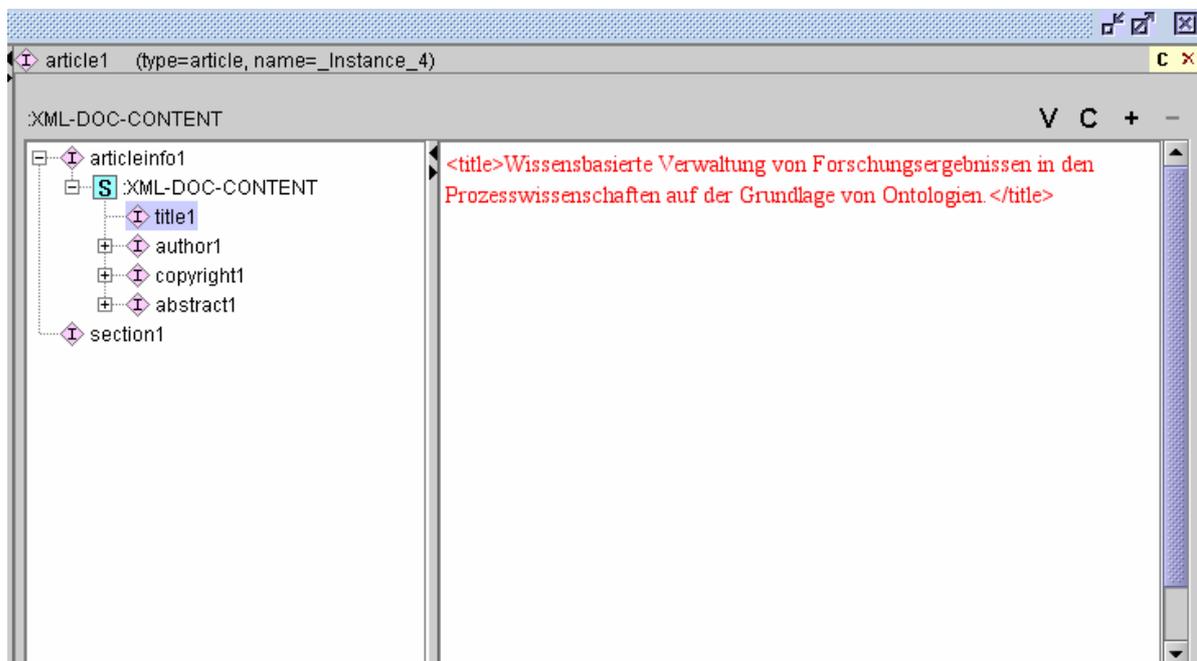


Abb. 75: Darstellung von Dokumenten-Inhalten im SCIFORG.

A 2 Verwenden der Dokumentenvorlage

Die Dokumentenvorlage enthält sämtliche Absatz- und Zeichenformate für alle typischen Elemente von Berichten, Veröffentlichungen, etc.⁹ Darüber hinaus sind diverse Arbeitsschritte durch die Vorlage (teil-)automatisiert. Dies betrifft u.a.:

⁹ So wurde auch die vorliegende Arbeit mit Hilfe der Vorlage erstellt.

- § das Datenbank-gestützte Anlegen und Pflegen eines Literaturverzeichnisses sowie das Erzeugen von entsprechenden Verweisen im Text,
- § das Einrichten bestimmter Zitierweisen,
- § und das Importieren und Formatieren von Graphiken und Tabellen.

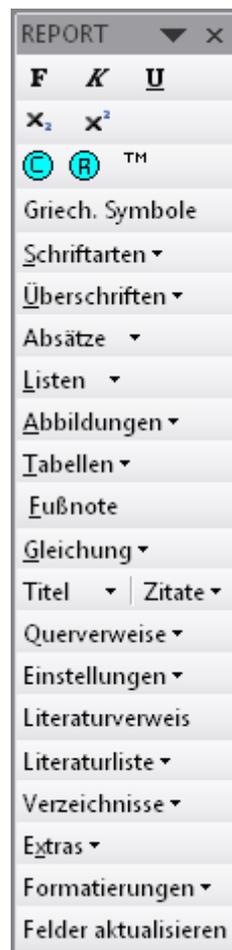


Abb. 76: Werkzeugleiste der Dokumentenvorlage

Sämtliche Funktionen sind über eine zentrale Werkzeugleiste verfügbar (siehe Abbildung 76). Im oberen Bereich der Werkzeugleiste befinden sich vor allem Menüs für die Zuweisung von Zeichen- und Absatzformaten. Der untere Bereich enthält v.a. Funktionen für Verweise, Verzeichnisse und Layout.

Glossar

API	„Application Programming Interface“ – Programmier-Schnittstelle, die ein System offen für Entwickler macht, etwa um lesend auf das interne Datenmodell zuzugreifen oder die Funktionalität des Systems zu erweitern.
CAE	„Computer Aided Engineering“ – Systeme zur Unterstützung von Arbeitsprozessen im ingenieurstechnischen Bereich.
HTML	„Hypertext Markup Language“ – textbasierte Auszeichnungssprache für Inhalte, die in einem Browser dargestellt werden.
LIMS	„Labor Informations Management Systemste“ – Systeme zur Verwaltung von Labordaten und zur Steuerung von Kontrollprozessen zur Qualitätssicherung.
MathML	„Mathematical Markup Language“ – eine auf XML (s.u.) basierende Sprache zur Beschreibung und Darstellung mathematischer Inhalte.
Objekt- relationale Ab- bildung	Ein Konzept zur Abbildung von relationalen Schemata auf objektorientierte Datenmodelle.
OWL	„Ontology Web Language“ – auf Beschreibungslogiken basierende Ontologiesprache, die vom W3C standardisiert wurde (siehe Abschnitt 2.2 „Semantische Technologien“ auf Seite 26)
Persistenzmodell	Ein Konzept zur persistenten (d.h. über die Laufzeit einer Applikation hinweg verfügbaren) Datenhaltung sowie eine dazu passende Implementierung (vgl. Abschnitt 5.3.1 „Skalierbare objektrelationale Speicherschicht“ auf Seite 113).
RDF(S)	„Resource Description Framework“ – auf einem Graphenmodell basierende Ontologiesprache, die vom W3C standardisiert wurde (siehe Abschnitt 2.2 „Semantische Technologien“ auf Seite 26)
UML	„Unified Modeling Language“ – Standard zur Erstellung von Datenmodellen als Grundlage für Applikationen, der nicht an eine bestimmte Programmiersprache gebunden ist (vgl. Balzert, 1999)
URI	„Uniform Ressource Identifier“ – Standard zur Identifizierung von abstrakten oder physischen Ressourcen (Webseiten, etc.).

W3C	„World Wide Web Consortium“ – Standardisierungsgremium, welches u.a. XML und RDF standardisiert hat.
XML	„eXtensible Markup Language“ – Datenformat das es erlaubt, Metadaten mit Hilfe von sog. „Markup“ hierarchisch darzustellen und zu standardisieren (siehe auch Abschnitt 4.2.1). XML ist durch das W3C standardisiert.

Abbildungsverzeichnis

Abb. 1:	Prozess der Modellentwicklung	28
Abb. 2:	Zusammenhang Realität - System – Modell.....	31
Abb. 3:	Zeitverhalten von Modellen	35
Abb. 4:	Raumverhalten von Modellen	36
Abb. 5:	Induktive und deduktive Modellbildung.....	38
Abb. 6:	Induktive Modellbildung nach Möller (1992)	39
Abb. 7:	Vorgehensweise bei der Modellbildung und Simulation nach Lugner und Bub (1990)	40
Abb. 8:	Modellbildung nach Lunze (1995).....	41
Abb. 9:	Modellentwicklung entsprechend Top und Akkermans (1994)	42
Abb. 10:	Strukturierung der konzeptionellen Modellbildung nach Liebl (1992)	43
Abb. 11:	„Blackbox Modell“	44
Abb. 12:	Modellentwicklung in einer Fachdomäne: Beispiel Rohrströmung.....	45
Abb. 13:	Beschreibung der Systemgrößen	48
Abb. 14:	Beschreibung der Modellgrößen	48
Abb. 15:	Verifikation und Validierung entsprechend Lugner und Bub (1990).....	50
Abb. 16:	Metainformationen in der Modellentwicklung und resultierende Fragestellungen	52
Abb. 17:	Erkenntnisgewinn im Rahmen von Forschungsprojekten.....	66
Abb. 18:	Ontologie-basierter Zugriff auf heterogene Informationen	67
Abb. 19:	XML-Formate für den naturwissenschaftlichen Bereich	69
Abb. 20:	SchoolOnto-Ansatz nach Buckingham et al. (2000).....	71
Abb. 21:	SchoolOnto-Ansatz für die Modellentwicklung.....	77
Abb. 22:	Anschauliche Darstellung der wesentlichen Schritte zur Informationsverwaltung auf Ontologie-Basis.....	82
Abb. 23:	Übersicht der wichtigsten Elemente zur Beschreibung mathematischer Modelle	84
Abb. 24:	SCIFORG Haupt-Ansicht	93
Abb. 25:	Grundlegende SCIFORG-Komponenten.....	94
Abb. 26:	Interfaces der Protégé Frames API (Ausschnitt ; ohne Methoden)	97
Abb. 27:	Umsetzung komplexerer Anfragen an das Datenmodell mit Datenbank- Schnittstelle	107
Abb. 28:	Haupt-Symbolleiste.....	112
Abb. 29:	Import-Wizard	113
Abb. 30:	Abbildung von Klassen auf Tabellen (Entsprechung von Datensätzen und Instanzen)	117
Abb. 31:	Abbildung von Slots der Kardinalität eins und einfachen Datentypen als Wertebereich auf Tabellen-Spalten.....	118
Abb. 32:	Abbildung von Slots des Typs „Symbol“ auf eine extra Tabelle	119

Abb. 33:	Abbildung von Slots der Kardinalität n mit einfachen Datentypen als Wertebereich auf eine zusätzliche Tabelle über den Primärschlüssel.....	119
Abb. 34:	Abbildung des Vererbungsprinzips mit Hilfe von Primär- und Fremdschlüsseln	120
Abb. 35:	Abbildung von Slots der Kardinalität eins und dem Wertebereich „Instanz“ über (einen oder mehrere) Fremdschlüssel	121
Abb. 36:	Abbildung von Slots der Kardinalität n mit dem Wertebereich „Instanz“ (sog. M:N Relation über eine Zwischentabelle).....	122
Abb. 37:	Erfassen von mathematischen Gleichungen/Formeln	125
Abb. 38:	Erfassen von chemischen Formeln	126
Abb. 39:	Auswahl der physikalischen Einheit.....	128
Abb. 40:	Auswahl der Einheit in Slot-Widgets	128
Abb. 41:	Symbol-Editor für domänen-spezifische Symbole.....	129
Abb. 42:	Graphische Suchanfrage: Bestimmung der Bedeutung von Verknüpfungen	130
Abb. 43:	Graphische Suchanfrage: Welche mathematischen Modelle zur Beschreibung von Membran-(Modul-)Systemen wurden in einer bestimmten Institution entwickelt?.....	131
Abb. 44:	Graphische Suchanfrage: Suche von analogen Modellen	132
Abb. 45:	Welche Modelle stellen eine Weiterentwicklung von einem bestimmten Modell X dar?.....	132
Abb. 46:	Darstellung von Suchergebnissen in der Listen-/Formular-Ansicht	133
Abb. 47:	Komplexere Suchanfrage (unvollständig).....	134
Abb. 48:	Definieren eines Wertevergleichs in graphischen Suchanfragen.....	134
Abb. 49:	Anzeige von Vergleichskriterien in graphischen Suchanfragen	135
Abb. 50:	Visualisierung von Instanzen über die graphische Schnittstelle.....	136
Abb. 51:	Formular-Suche für Claims.....	137
Abb. 52:	Erfassen eines Platzhalters für eine Suchbedingung.....	138
Abb. 53:	Import-Wizard.....	140
Abb. 54:	Visualisierung von Datenreihen	142
Abb. 55:	Anwendung der Glättungsfunktion (gleitender Durchschnitt)	143
Abb. 56:	Anzeige von statistischen Informationen	144
Abb. 57:	Grundleges Anwendungsszenario für SCIFORG.....	145
Abb. 58:	SCIFORG Splash-Screen und Aufforderung zur Konfiguration	156
Abb. 59:	SCIFORG Konfigurations-Dialog	157
Abb. 60:	Auswahl des Datenbank-Treibers.....	158
Abb. 61:	Dialog zum Einstellen von Mapping-Parametern.....	159
Abb. 62:	Anzeige des automatisch generierten SQL-Befehls zum Anlegen einer Tabelle für die objektrelationale Abbildung.....	160
Abb. 63:	Anzeige von Datenbank-Mapping Eigenschaften für eine ausgewählte Klasse	161
Abb. 64:	Anzeige von Datenbankzugriffen.....	162
Abb. 65:	Auswahl des Dokumententypen-Imports	163

Abb. 66:	Auswahl der Oberklasse für Dokumenten-Elemente.....	164
Abb. 67:	Aus einer DTD automatisch erzeugte Klassen.....	165
Abb. 68:	Spezielles Widget für die Anzeige von importierten XML-Dateien.....	166
Abb. 69:	Dokumenten-Import Schritt 1: Auswahl des Import-Wizards	167
Abb. 70:	Dokumenten-Import Schritt 2: Auswahl einer Richtext-Datei	168
Abb. 71:	Dokumenten-Import Schritt 3: Anpassen des Imports.....	169
Abb. 72:	Dokumenten-Import: Anzeige des Majix Konfigurations-Dialoges zum Editieren von Formatvorlagen	170
Abb. 73:	Dokumenten-Import: Anzeige des Majix Konfigurationsdialoges zum Editieren von XML-Tags	170
Abb. 74:	Dokumenten-Import Schritt 4: Statusanzeige für den Dokumentenimport.....	171
Abb. 75:	Darstellung von Dokumenten-Inhalten im SCIFORG.....	171
Abb. 76:	Werkzeugleiste der Dokumentenvorlage.....	172

Tabellenverzeichnis

Tab. 1:	Gründe für den Einsatz mathematischer Modelle	29
Tab. 2:	Gegenüberstellung von mathematischen Prozessmodellen und semantischen Modellen	58
Tab. 3:	Elemente des Datemodells von Protégé Frames.....	100
Tab. 4:	Anwender-Rollen für Protégé bzw. Protégé-basierte Werkzeuge.....	104
Tab. 5:	Zentrale konzeptionelle Unterschiede zwischen relationalen und objektorientierten Datenmodellen.....	116
Tab. 6:	Definition von Einheiten über Zeichenfolgen	127
Tab. 7:	Visualisierung in JGraph.....	130

Literaturverzeichnis

ALBERTS, L. 1994. Ymir A sharable ontology for the formal representation of engineering design knowledge. In: GERO, J.S.; TYUGU, E. Hrsg. Formal Design Methods for CAD. IFIP Transactions, Amsterdam, 3-32.

ALEXA T. MCCRAY, MARIE E. GALLAGHER, MICHAEL A. FLANNICK. 1999 Extending the Role of Metadata in a Digital Library System. Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries '99.: IEEE Computer Society, 190-199.

ARIS, R. 1991. Manners makyth modellers. Trans. Inst. Chem. Eng. 69, 165-174.

BAGGA, J. UND HEINZ, A. 2002. JGraph - A Java Based System for Drawing Graphs and Running Graph Algorithms. In: MÜTZEL, P.; JÜNGER, M.; LEIPERT, S. Hrsg. Graph Drawing : 9th International Symposium, GD 2001 Vienna, Austria, September 23-26, 2001. Lecture Notes in Computer Science, Berlin, Heidelberg, New York: Springer, 459-460.

BALCI, O. 1988. Credibility assesment of simulation results. In: BALCI, O. Hrsg. Proceedings of the conference on methodology and validation. SCS Simulation Series, Vol. 19, #1, Orlando, FLA, 19-25.

BALZERT, H. 1999. Lehrbuch der Objektmodellierung. Analyse und Entwurf, Berlin, Heidelberg: Spektrum Akademischer Verlag.

BATRES, R. AOYAMA, A. NAKA, Y. 2001. A Life-Cycle Approach for Model Reuse and Exchange. In: GANI, R. JØRGENSEN, S. B. Hrsg. European Symposium on Computer Aided Process Engineering – 11. Lyngby, Dänemark, 87-92.

BATRES, R. UND NAKA, Y. 2000. Process Plant Ontologies based on a multi-dimensional Framework. AIChE Symposium Series 96, AIChE Symposium Series 96 (323), 433-437.

BAYER, B. SCHNEIDER, R. UND MARQUARDT, W. 2000. Integration of Data Models for Process Design - first Steps and Experiences. Computers Chemical Engineering 24, 599-605.

BAYER, B.; MARQUARDT, W. 2004. Towards Integrated Information Models for Data and Documents. Computers and Chemical Engineering 28, 1249-1266.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. 2001. The Semantic Web: A New Form of Web Content That Is Meaningful to Computers Will Unleash a Revolution of New Possibilities. Scientific American 284 (5), 28-37.

BLASS, ECKHART, 1997. Entwicklung verfahrenstechnischer Prozesse. Methode, Zielsuche, Lösungssuche, Lösungsauswahl. 2.Aufl. Berlin, Heidelberg: Springer Verlag.

BOGUSCH, R, LOHMANN, B. UND MARQUARDT, W. 1997. Process-centered Modeling and Simulation environments: reducing the cost of developing models and performing simulation experiments. In: BREITENECKER, F. UND HUSINSKY, I. Hrsg. S N E, Eurosim. 8-10.

BOGUSCH, R. LOHMANN, B. UND MARQUARDT, W. 2001. Computer-aided process modeling with ModKit. Computers and Chemical Engineering 25, 963-995.

- BOGUSCH, R. UND MARQUARDT, W. 1997. A formal representation of process model equations. *Computers Chemical Engineering* 21 (10), 1105-1115.
- BORST, P. AKKERMANS, H. UND TOP, J. 1997. Engineering Ontologies. *International Journal of Human Computer Studies* 46, 365-406.
- BORST, W.N. 1997. CONSTRUCTION OF ENGINEERING ONTOLOGIES FOR KNOWLEDGE SHARING AND REUSE. CTIT Ph. D-series No. 97-14, Enschede, NL.
- BOSEL, HARTMUT, 1994. Modellbildung und Simulation. Konzepte, Verfahren und Modelle zum Verhalten dynamischer Systeme. 2. Auflage mit verbesserter Software. Braunschweig, Wiesbaden: Friedr. Vieweg & Sohn Verlagsgesellschaft mbH.
- BUNEMAN, P.; KHANNA, S.; TAJIMA, K.; WANG-CHIEW, T. 2004. Archiving Scientific Data. *ACM Transactions on Database Systems (TODS)* 29 (1), 2-42.
- BUB, W UND LUGNER, P. 1992. Systematik der Modellbildung. Teil 1: Konzeptionelle Modellbildung. *Modellbildung für Regelung und Simulation*. Düsseldorf: VDI-Verlag GmbH, 1-18.
- BUB, W UND LUGNER P. 1992. Systematik der Modellbildung. Teil 2: Verifikation und Validation. *Modellbildung für Regelung und Simulation*. Düsseldorf: VDI-Verlag GmbH, 19-44.
- BUCKINGHAM, S. MOTTA, E. UND DOMINGUE, D. 2000. ScholOnto: an ontology-based digital library server for research documents and discourse. *International Journal on Digital Libraries* 3, 237-248.
- BUNEMAN, P.; KHANNA, S.; WANG-CHIEW T. 2001. Why and Where: A Characterization of Data Provenance. In: BUSSCHE, D. V. J.; VIANU, V. Hrsg. *Database Theory - ICDT 2001, 8th International Conference, London, UK, January 4-6, 2001, Proceedings*. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer.
- CECCARONI, L. CORTÉS, U. UND SÁNCHEZ-MARRÈ, M. 2001. OntoWEDSS - An Ontology-based Environmental Decision-Support System for the management of Wastewater treatment plants. Dissertation Universitat Politècnica de Catalunya.
- CHAUDHRI, K. V.; FARQUHAR, A.; FIKES, R.; KARP, D. P.; RICE, J. 1998. OKBC: A Programmatic Foundation for Knowledge Base Interoperability. In: AMERICAN ASSOCIATION FOR ARTIFICIAL INTELLIGENCE (AAAI), Hrsg. *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*. Madison, Wisconsin: AAAI Press/The MIT Press, 600-607.
- CRAWFORD, J.M.; KUIPERS, B.J. 1991. Algernon - a tractable system for knowledge representation. *SIGART Bulletin* 2 (3), 35-44.
- DACONTA, M.C.; OBRST, L.J.; SMITH, K.T.;, 2003. *The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management*. Indianapolis, USA: Wiley.
- DAVENPORT, T. H.; PRUSAK, L. 1998. *Working Knowledge: How Organizations Manage What They Know*. Boston, USA.

- DECKER, S.; MELNIK, S.; HARMELEN, V. F.; FENSEL, D.; KLEIN, A. C. M.; BROEKSTRA, J.; ERDMANN, M.; HORROCKS, I. 2000. The Semantic Web The Roles of XML and RDF. *IEEE Internet Computing* 4 (5), 63-74.
- DIRAT, L. 2000. JOME, a software component for interactive and distributed mathematics. *ACM SIGSAM Bulletin* 34 (2), 38-42.
- DOWES, S. 2004. Resource Profiles. *Journal of Interactive Media in Education*. 2004 (5).
- EBERSBACH, A.; GLASER, M.; HEIGL, R. 2005. Wiki Web Collaboration. Berlin, Heidelberg, New York: Springer.
- ERDMANN, M. UND STUDER, R. 2001. How to structure and access XML documents with ontologies. *Data & Knowledge Engineering* 36, 317-335.
- FAYYAD, U. 1997. Data Mining and Knowledge Discovery in Databases: Implications for Scientific Databases. In: HANSEN, D. UND IOANNIDIS, Y. Hrsg. Ninth International Conference on Scientific and Statistical Database Management. Olympia, Washington: IEEE Computer Society, 2-11.
- EGGERSMANN, M. VON WEDEL, L. UND MARQUARDT, W. 2002. Verwaltung und Wiederverwendung von Modellen im industriellen Entwicklungsprozess. *Chemie Ingenieur Technik* 74 (8), 1068-1078.
- FRANSEN, S.M. UND NYRUP, A. 1998. Implementation of a 'standard' LIMS. *Laboratory Automation & Information Management* (33), 227-233.
- GAINES, B. R. UND SHAW M. L. G. 1997. Knowledge Management for Research Communities. Proceedings of the AAAI Spring Symposium on Artificial Intelligence in Knowledge Management. Stanford, USA, 55-62.
- FOSS, B. A. LOHMANN, B. UND MARQUARDT, W. 1998. A field study of the industrial modeling process. *Journal of Process Control* 8 (5/6), 325-338.
- GIARRATANO, C. J.; RILEY, G. 1994. Expert Systems Principles and Programming. Boston, MA, USA: PWS Publishing Co.
- GOLOMB, S.W. 1970. Mathematical Models - Uses and Limitations. *Simulation* 4 (14), 197-198.
- GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M.; CORCHO, O. 2004. Ontological Engineering. with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web, Berlin, Heidelberg, New York: Springer.
- GORDON, GEOFFREY, 1972. Systemsimulation. München: R. Oldenbourg Verlag.
- GREEN, DAVID G. 1990. Syntactic modelling and simulation. *Simulation* (6), 281-286.
- GRÖTSCHEL, M. UND LÜGGER, J. 1999. Scientific Information Systems and Metadata. In: LOCAREK-JUNGE, H. UND GAUL, W. Hrsg. Classification in the Information Age. Berlin: Springer, 3-20.

- GRUBER, R. T.; OLSEN, R. G. 1994. An Ontology for Engineering Mathematics. In: DOYLE, J.; SANDEWALL, E.; TORASSO, P. Hrsg. Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94). Bonn: Morgan Kaufmann, 258-269.
- GRUBER, T. R. 1995. Towards Principles for the Design of Ontologies used for Knowledge Sharing. *International Journal of Human Computer Studies* 43 (5/6), 907-928.
- GRUBER, T.R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5, 199-220.
- GRUBER, T.R. 1993. Model formulation as a problem-solving task: computer assisted engineering modelling. *International Journal of Intelligent Systems* 8, 105-127.
- GUIDI, F. UND SCHEINA, I. 2003. A Query Language for a Metadata Framework about Mathematical Resources. In: ASPERTI, A. BUCHBERGER, B. UND DAVENPORT, J.H. Hrsg. *Mathematical Knowledge Management*. Berlin, Heidelberg: Springer, 105-118.
- GULBINS, J. SEYFRIED, M. UND STRACK-ZIMMERMANN, H. 1999. *Dokumentenmanagement. Vom Imaging zum Business-Dokument*, Berlin, Heidelberg, New York: Springer.
- GUPTA, A. UND BARU, C. 1999. An extensible information model for shared scientific data collections. *Future Generation Computer Systems* 16, 9-20.
- HANGOS, K. M. UND CAMERON, I, T. 2001. The formal representation of process system modelling assumptions and their implications. *Computers Chemical Engineering* 21, 237-255.
- HARTMANN, S. 2003. Scientific Models. In: SARKAR, S.; PFEIFER, J. Hrsg. *The Philosophy of Science: An Encyclopedia*. New York: Routledge Press, 740-749.
- HARTMANN, S. UND BAILER-JONES, D. 1999. Modelle. In: H.-J. SANDKÜHLER ET AL. Hrsg. *Enzyklopädie der Philosophie*. Hamburg, 854-859.
- HESSE, W. 2002. Ontologie(n). *Informatik-Spektrum* 25 (6), 477-480.
- HEUER, A. 2000. *Datenbanken: Konzepte und Sprachen*. Bonn: MITP.
- HEUMESSER, B.D. SEIPEL, D.A. UND GÜNTZER, U. 2003. An Expert System for the Flexible Processing of Xml-Based Mathematical Knowledge in a Prolog-Environment. In: ASPERTI, A. BUCHBERGER, B. UND DAVENPORT, J.H. Hrsg. *Mathematical Knowledge Management*. Berlin, Heidelberg: Springer, 133-146.
- HILL, L. L. CROSIER, S. J. SMITH, T. R. UND GOODCHILD, M. 2001. A Content Standard for Computational Models. *D-Lib Magazine* 7 (6) (Internet Veröffentlichung).
- HINKE, T.H. RUSHING, J. KANSAL, S. GRAVES S.J. UND RANGANATH, H. 1997. For Scientific Data Discovery: Why Can't the Archive be More Like the Web?. In: HANSEN, D. UND IOANNIDIS, Y. Hrsg. *Ninth International Conference on Scientific and Statistical Database Management*. Olympia, Washington: IEEE Computer Society, 96-99.
- HINTON, M. 1995. LIMS in the manufacturing environment. *Laboratory Automation & Information Management* (31), 109-113.

- HOLSAPPLE, C.W. UND JOSHI, K.D. 2000. An investigation of factors that influence the management of knowledge in organizations. *Journal of Strategic Information Systems* 9, 235-261.
- HORROCKS, I.; LI, L.; TURI, D.; BECHHOFFER, S. HAARSLEV, VOLKER ; MOLLER, RALF, Hrsg. 2004. The Instance Store DL Reasoning with Large Numbers of Individuals. .
- HOUSTIS, C. NIKOLAOU, C. LALIS, S. KAPIDAKIS, S. CHRISTOPHIDES, V. SIMON, E. UND THOMASIC, A. 1999. Towards a next generation of open scientific data repositories. *CWI Quarterly (Centrum voor Wiskunde en Informatica)* 12 (2), 111-132.
- HUNTER, J.; LITTLE, S. 2005. A framework to enable the semantic inferencing and querying of multimedia content. *International Journal of Web Engineering and Technology* 2(2/3), 264-286.
- JONES, D. M. UND PATON, R. C. 1999. Toward principles for the representation of hierarchical knowledge in formal ontologies. *Data & Knowledge Engineering* 31, 99-113.
- JURISTO, N. UND MORENO, A.M. 2000. Introductory paper: Reflections on Conceptual Modelling. *Data & Knowledge Engineering* 33, 103-117.
- KAESTLE, G. SHEK, E.C. UND DAO, S.K. 1999. Sharing Experiences from Scientific Experiments. In: OZSOYOGLU, Z.M. OZSOYOGLU, G. UND HOU, W.-C. Hrsg. 11th International Conference on Scientific and Statistical Database Management. Cleveland, Ohio, USA: IEEE Computer Society, 168-177.
- KAO, D.T. BERGERON, R.D. UND SPARR, T.M. 1994. An Extended Schema Model for Scientific Data. In: LEE, J. P. Hrsg. Database Issues for Data Visualization, IEEE 1993. Berlin: SPRINGER, 69-82.
- KAPETANIOS, E. 1996. Extracting and Providing Knowledge within an Object-Centered Scientific Information System for Atmospheric Research. In: SVENSSON, P. UND FRENCH, J.C. Hrsg. Eighth International Conference on Scientific and Statistical Database Management. Stockholm, Sweden: IEEE Computer Society, 198-207.
- KEDEM, O.; KATCHALSKY, A. 1960. A Physical Interpretation of the Phenomenological Coefficients of Membrane Permeability. *Journal of General Physiology* 45(1), 143-179.
- KENT, J.-P. UND SCHUERHOFF, M. 1997. Some thoughts About a Metadata Management System. In: HANSEN, D. UND IOANNIDIS, Y. Hrsg. Ninth International Conference on Scientific and Statistical Database Management. Olympia, Washington: IEEE Computer Society, 174-185.
- KIFER, M. UND LAUSEN, G. 1995. Logical foundations of object-oriented and frame-based language. *Journal of the ACM* 42 (4), 741-843.
- KIFER, M. UND LAUSEN, G. 1997. FLogic: A higher-order language for reasoning about objects. *SIGMOD Record* 18(6), 134-146.
- KINGSTON, JOHN K.C. 1998. Designing knowledge based systems: the CommonKADS design model. *Knowledge Based Systems* 11, 311-319.
- KIRCZ, J. G. 1998. Modularity: the next form of scientific information presentation?. *Journal of Documentation* 54 (2), 210-235.

KIRCZ, J. G. UND ROOSENDAL, H. E. 1996. Understanding and shaping scientific information transfer. In: SHAW, D. UND MOORE, H. Hrsg. Electronic publishing in science. Proceedings of the joint ICSU Press - UNESCO Expert Conference. Paris, 106-116.

KOKOSSIS, A.; BAÑARES-ALCÁNTARA, R. 2003. Dynamic Information Management for Web-Enabled Environments in the Chemical Process Industries. In: GROSSMANN, I. E.; McDONALD, C. M. , Hrsg. Proceedings of the Fourth International Conference on Foundations of Computer-Aided Process Operations. Coral Springs, Florida, 547-550.

KOWALK, WOLFGANG P. 1996. System - Modell - Programm. Vom GOTO zur objektorientierten Programmierung. Heidelberg, Berlin, Oxford: Spektrum Akademischer Verlag.

KRAINES, S.; BATRES, R.; KOYAMA, M.; WALLACE, D.; KOMIYAMA, H. 2005. Internet-Based Integrated Environmental Assessment. Using Ontologies to Share Computational Models. Journal of Industrial Ecology 9 (3), 31-50.

KRAUSE, S.; WILLIGHAGEN, E.; STEINBECK, C. 2000. JChemPaint - Using the Collaborative Forces of the Internet to Develop a Free Editor for 2D Chemical Structures. Molecules 5, 93-98.

KROBB, C. UND HACKENBERG, J. 2000. Modellierung und Unterstützung verfahrenstechnischer Modellierungsprozesse. Informatik 2000. Berlin.

LANGER, H. UND NAUMANN, S. 2000. Parsing natürlicher Sprache. In: GÖRZ, G. ROLLINGER, C.-R. UND SCHNEEBERGER, J. Hrsg. Handbuch der Künstlichen Intelligenz. 3. Aufl. München, Wien: Oldenbourg, 711-738.

LEE, T. BOZKAYA, T. KUO, H.-C. ÖZSOYOGLU, G. UND ÖZSOYOGLU, Z.M. 1996. A Scientific Multimedia Database System for Polymer Science Experiments. In: SVENSSON, P. UND FRENCH, J.C. Hrsg. Eighth International Conference on Scientific and Statistical Database Management. Stockholm, Sweden: IEEE Computer Society, 86-95.

LIEBL, 1992. Simulation. München, Wien: R. Oldenbourg Verlag.

LOBIN, H. 2000. Informationsmodellierung in XML und SGML. Berlin, Heidelberg, New York: Springer-Verlag.

LUGNER, P. UND BUB, W. 1990. Modellbildung und Simulation in der Automatisierungstechnik. Systematik der konzeptionellen Modellbildung In: BREITENECKER, F. TROCH, I UND KOPACEK, P. Hrsg. Simulationstechnik. Braunschweig: Vieweg, 62-66.

LUNZE, J. 1995. Künstliche Intelligenz für Ingenieure. Band 2: Technische Anwendungen, München: Oldenbourg.

MARQUARDT, WOLFGANG, 1992. Rechnergestützte Erstellung verfahrenstechnischer Prozessmodelle. Chemie Ingenieur Technik (64), 25-40.

LOHMANN, B. UND MARQUARDT, WOLFGANG, 1996. On the systematization of the process of model development. Computers Chemical Engineering 20 (Suppl.), 213-218.

MARQUARDT, WOLFGANG, 1995. Modellbildung als Grundlage der Prozeßsimulation. In: SCHULER, HANS, Hrsg. Prozeßsimulation. Weinheim: VCH Verlagsgesellschaft mbH, 3-34.

- MARQUARDT, WOLFGANG, 1995. Towards a process modeling methodology. In: BERBER, RIDVAN, Hrsg. *Methods of model based process control*. Dordrecht, Boston, London: Kluwer Academic Publishers, 3-40.
- MARQUARDT, WOLFGANG, 1996. Trends in computer-aided process modeling. *Computers Chemical Engineering* 20 (6/7), 591-609.
- MASON, E.A. 1986. Generalization of Membrane Reflection Coefficients for Nonideal, Nonisothermal, Multicomponent Systems with External Forces and Viscous Flow. *Journal of Membrane Science* 28, 229-267.
- MASSEY, K.D. KERSCHBERG, L. UND MICHAELS, G. 1997. VANILLA: A Dynamic Data Schema for A Generic Scientific Database. In: HANSEN, D. UND IOANNIDIS, Y. Hrsg. *Ninth International Conference on Scientific and Statistical Database Management*. Olympia, Washington: IEEE Computer Society, 104-107.
- MIHAILA, A. G.; RASCHID, L.; VIDAL, M. 2002. Source selection and ranking in the websemantics architecture using quality of data metadata. *Advances in Computers* 55, 87-118.
- MCDOWALL, R.D. 1999. The role of laboratory information management systems (LIMS) in analytical method validation. *Analytica Chimica Acta* (391), 149-158.
- MINSKY, M.L. 1968. Matter, Mind and Models. In: MINSKY, M.L. Hrsg. *Semantic Information Processing*. Cambridge / Massachusetts: MIT Press, 425-32.
- MÖLLER, DIETMAR P.F. 1992. *Modellbildung, Simulation und Identifikation dynamischer Systeme*. Berlin, Heidelberg, New York: Springer.
- MÖLLER, R.; HAARSLEV, V.; WESSEL, M. FREKSA, C.; KOHLHASE, M. Hrsg. 2006. *On the Scalability of Description Logic Instance Retrieval*. .
- MORBACH, J.; YANG, A.; MARQUARDT, W. 2007. OntoCAPE - A large-scale ontology for chemical process engineering. *Engineering Applications of Artificial Intelligence (EAAI)* 20(2), 147-161.
- MURRAY-RUST, P. UND RZEPA, H. S. 2001. Chemical Markup, XML and the World-Wide Web. 2. Information Objects and the CMLDOM. *J. Chem. Inf. Comput. Sci.* 41, 1113-1123.
- NOY, N. F. SINTEK, M. DECKER, S. CRUBEZY, M. FERGERSON, R. W. UND MUSEN, M. A. 2001. Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems* 16(2), 60-71.
- PUPPE, F. STOYAN, H. UND STUDER, R. 2000. Knowledge Engineering. In: GÖRZ, G. ROLLINGER, C.-R. UND SCHNEEBERGER, J. Hrsg. *Handbuch der künstlichen Intelligenz*. München, Wien: Oldenbourg Verlag, 599-641.
- SCHAIBLE, M. 2001. Searching scientific databases for guides to experiment and theory. *Computing in Science and Engineering* 4, 30-39.
- NOY, N.F. FERGERSON, R.W. UND MUSEN, M.A. 2000. The knowledge model of Protégé 2000: combining interoperability and flexibility. In: DIENG, R. UND CORBY, O. Hrsg. *Knowledge Engineering and Knowledge Management*. Berlin: Springer, 17-32.

- SCHUCHARDT, L. 1998. LIMS am Scheideweg - Entwicklung und Tendenzen. Nachrichten aus Chemie, Technik und Labor 46 (4), A 20- 26.
- SCHWARZE, G. 1990. Digitale Simulation. Konzepte - Werkzeuge - Applikationen. Berlin: Akademie-Verlag.
- SHANNON, R.E. 1975. Systems Simulation: The Art and Science. Englewood Cliffs, NJ: Prentice-Hall.
- SHIPMAN, F. UND MCCALL, R. 1999. Supporting Incremental Formalization with the Hyper-Object Substrate. ACM Transactions on Information Systems 17 (2), 199-227.
- SÖDERMAN, U. 1995. Conceptual modelling of mode switching physical systems. Dissertation an der Linköping Universität (Dissertation no. 375).
- SOLDAR, SMITH, G. UND D. 2002. An Architecture for Retrieval of RDF-Described Scientific Data Semantics. In: CHAUDHRI, B. K. UNLAND, R. DJERABA, C. UND LINDNER, W. Hrsg. XML-Based Data Management and Multimedia Engineering - EDBT 2002 Workshops, EDBT 2002 Workshops XMLDM, MDDE, and YRWS, Prague, Czech Republic, March 24-28, 2002, Revised Papers. Lecture Notes in Computer Science, Berlin, Heidelberg, New York: Springer, 501-511.
- STAFFORD, J.E.H. 1998. LIMS: An automating or informing technology?. Laboratory Automation & Information Management (33), 163-168.
- SUGIMOTO, S. BAKER, T. UND WEIBEL, S.L. 2002. Dublin Core: Process and Principles. In: LIM, E.-P. FOO, S. KHOO, C. CHEN, H. FOX, E. URS, S. UND COSTANTINO, T. Hrsg. Digital Libraries: People, Knowledge, and Technology. Berlin, Heidelberg: Springer, 25-35.
- TOP, J. UND AKKERMANS, H. 1994. Tasks and ontologies in engineering modelling. International Journal of Human Computer Studies 41, 585-617.
- TUDORACHE, T. 2006. Employing Ontologies for an Improved Development Process in Collaborative Engineering. Dissertation an der Technischen Universität Berlin, Fakultät IV - Elektrotechnik und Informatik.
- UNBEHAUEN, ROLF, 1980. Systemtheorie. Eine Darstellung für Ingenieure. 3. Aufl. München, Wien: R. Oldenbourg Verlag.
- VAN GIGCH, J.P. 1991. System design modeling and metamodeling. New York: Plenum Press.
- VDI-GESELLSCHAFT FÖRDERTECHNIK MATERIALFLUSS LOGISTIK, Hrsg. 2000. Simulation von Logistik-, Materialfluß- und Produktionssystemen - Grundlagen. VDI-Richtlinie: VDI 3633 Blatt 1.
- VÖLKELE, M.; KRÖTZSCH, M.; VRANDEIC, D.; HALLER, H.; STUDER, R. 2006. Semantic Wikipedia. Proceedings of the 15th international conference on World Wide Web (WWW 06). Edinburgh: ACM Press, 585-594.

- WACHE, H.; VÖGELE, T.; VISSER, U.; STUCKENSCHMIDT, H.; SCHUSTER, G.; NEUMANN, H.; HÜBNER, S. 2001. Ontology-based Integration of Information - A Survey of Existing Approaches. In: STUCKENSCHMIDT, H. Hrsg. Proceedings of the IJCAI-01 Workshop: Ontologies and Information Sharing. Seattle, WA, 108-117.
- WALSH, N. UND MUELLNER, L. 1999. DocBook. The Definitive Reference. Sebastopol, CA: O'Reilly.
- WEININGER, D. 1988. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. 28 (1), 31-36.
- WEITEN, M. UND WOZNY, G. 2004. Advanced information management for process sciences: knowledge-based documentation of mathematical models. International Journal of Internet and Enterprise Management 2(2), 178-190.
- WEITEN, M. WOZNY, G. UND GOERS, B. 2002. A Data Model for Mathematical Models in a Knowledge Based System. In: LUCZAK, H. CAKIR, A. E. UND CAKIR, G. Hrsg. Proceedings of the 6th International Conference on Work With Display Units. Berlin: ERGONOMIC GmbH, 624-626.
- WELTY, C.; IDE, N. 1999. Using the right tools: enhancing retrieval from marked-up documents. Journal Computers and the Humanities 33 (10), 59-84.
- WELTY, C.A. UND JENKINS, J. 1999. Formal ontology for subject. Data & Knowledge Engineering 31, 155-181.
- YANG, A.; MORBACH, J.; MARQUARDT, W. 2004. From Conceptualization to Model Generation: the Roles of Ontologies in Process Modeling. In: FLOUDAS, C.; AGRAWAL, R. Hrsg. Proceedings of the Sixth. International Conference on Foundation of Computer Aided Process Design. (FOCAPD 2004), Princeton, 591-594.