

Diplomarbeit

Kollaborative Ontologien und ihre
Visualisierung in
kooperationsunterstützenden
Systemen

vorgelegt bei
Juniorprofessor Dr. rer. nat. Thorsten Hampel
Heinz Nixdorf Institut
Universität Paderborn
im Rahmen des Hauptstudiums
für den Diplomstudiengang
Medienwissenschaften (Schwerpunkt Medieninformatik)

eingereicht von
Michael Vonruden
Matrikelnr.: 6024447
Hermannstr. 15
33102 Paderborn
11.05.2005

Kurzfassung: Wissen ist eine der zentralen Säulen der heutigen Gesellschaft. Neue Techniken, Wissen auch maschinell fassbar zu machen deuten an, welche strategische Bedeutung dem Wissensbegriff zukommt. Im Kontext des Semantic Webs ist oft die Rede von der Ontologie als Struktur zur Darstellung und Speicherung von Semantik und Wissen. Losgelöst von der maschinellen Interpretation, analysiert diese Arbeit den Einsatz von Ontologien zur Strukturierung von Wissen in einer Mensch-Mensch Relation. Da die Wissensgewinnung nicht als singulärer sondern als gemeinschaftlicher Prozess anzusehen ist, wird der Fokus hier insbesondere auf eine kooperative Erstellung von Ontologien gelegt. Dieser Fokus gliedert sich in einen konzeptionellen und anwendungsorientierten Bereich.

Konzeptionell werden drei verschiedene Aspekte erörtert. Neben einer Einordnung von Ontologien als Wissensstruktur und der Notwendigkeit einer kooperativen Bearbeitung wird der eigentliche Prozess des Aufbaus von Ontologien ebenso beschrieben, wie eine generelle Vorgehensweise zur Visualisierung dieser Struktur. Während für die grafische Darstellung des ontologischen Konzepts mögliche Herangehensweisen abgewogen werden, geht die Beschreibung des Aufbauprozesses auf Probleme und Notwendigkeiten einer kooperativ gestützten Wissenserarbeitung ein. Besondere Kernelemente bilden hier die Findung eines Konsenses, die gegenseitige Wahrnehmung und das Nachvollziehen des Wissensprozesses.

Neben dem Entwurf möglicher Einsatz-Szenarien wird eine entwickelte Anwendung zum kooperativen Aufbau einer Ontologie vorgestellt. Basierend auf dem aufgestellten Konzept wird eine Nutzer-Applikation beschrieben, die mit Hilfe einer grafischen Metapher die ontologische Struktur visualisiert. Als generelle und organisatorische Komponente für die Handhabung von Ontologien wird abschließend ein Server-Modul für die `openTeam`-Umgebung präsentiert.

Inhaltsverzeichnis

1	Einleitung	1
2	Ontologien	4
2.1	Begriffsbestimmung	4
2.2	Aufbau einer Ontologie	5
2.3	Ontologien und das Semantische Web	7
2.4	Grundannahmen	8
3	Kollaborative Ontologien zur Strukturierung von Wissen	10
3.1	Kooperative Ontologienbildung	10
3.1.1	Rollenverteilung	11
3.1.2	Medienfunktionen	12
3.1.3	Ontologien als transportierbare Wissensseinheiten	13
3.1.4	Formen kooperativer Ontologien	14
3.1.5	Entwurfsmethoden und Inhalt	15
3.1.6	Konsens	16
3.1.7	Nachvollziehbarkeit und Erweiterung	18
3.1.8	Gegenseitige Wahrnehmung	19
3.2	Grafische Strukturierung	21
3.2.1	Trennung von Exploration und Editierung	23
3.2.2	Darstellung der Modellstruktur	25
3.2.3	Gruppierung von Zusammenhängen	32
3.2.4	Alternative Darstellungsform	32
4	Szenarien - Einsatz von Ontologien	34
4.1	Unterstützung von Lehr- Lernprozessen	34
4.1.1	Kooperative Organisation von Lehrveranstaltungen	34
4.1.2	Wissenserarbeitung im Team	35
4.1.3	Ontologien als Übungen	38
4.1.4	Erschließung des kooperativen Systems	39
5	visCOntE - Eine Umgebung zur Erstellung kooperativer Ontologien	42
5.1	Visueller Browser und Editor	42
5.1.1	Die Konzept Metapher	42
5.1.2	Verschiedene Sichten	45

5.1.3	Awareness	51
5.1.4	Anlegen einer Ontologie	52
5.1.5	Editierung	52
5.1.6	Diskussion	54
5.1.7	Orientierung und Bedienungselemente	55
5.1.8	Konfiguration und Internationalisierung	57
5.1.9	Technische Umsetzung - Ein Überblick	58
5.2	Einbettung in ^{open} sTeam	61
5.2.1	Datenformat für Ontologien	61
5.2.2	Modul Aufbau	66
5.2.3	Die Hilfsklasse ID	72
5.2.4	Client-Server Interaktion	73
5.2.5	Annotationen	76
5.2.6	Speicherung und Archivierung	76
6	Ausblick und Zusammenfassung	77

Abbildungen und Tabellen

Abbildungsverzeichnis

2.1	Modell-Ausschnitt einer Ontologie mit mehrfacher (links) und einfacher (rechts) Vererbung	6
2.2	Gegenseitige Abhängigkeiten innerhalb einer Ontologie	7
3.1	Formen grafischer Strukturierung (Quelle: Hampel u. a., 2004)	22
3.2	Struktur - Beschränkungen einer Ontologie	23
3.3	Bildschirmfoto von Protégé – Ansicht auf die Hierarchie einer Ontologie	24
3.4	Hierarchieausschnitt in drei Ebenen	26
3.5	Hierarchie als Treemap (Quelle: Shneiderman, 1992)	27
3.6	Hierarchie in 3D Darstellung mit Hilfe von Cone Trees (Quelle: Robertson u. a., 1991)	27
3.7	Ausschnitt aus der Ontologie Visualisierung in OntoEdit©	29
3.8	Denkbare Konstellation einer Darstellung als Graph	30
3.9	Darstellung einer Clustermap (Quelle: Fluit u. a., 2002)	33
4.1	Ausschnitt der Ontologie <i>Vorlesung</i>	35
4.2	Ontologie zu Lizenzmodellen. Links: Modellstruktur, Rechts: Individuenstruktur	37
4.3	Verteilung von Kompetenzen für bestimmte Fächer	39
4.4	Thematisch verwandte Fächer, mit impliziter Verwandtschaft einzelner Gruppen	40
5.1	Die Flüssigkeitsmetapher am Beispiel der Modellhierarchie	43
5.2	Vergrößerte Ansicht eines Modells	44
5.3	Die Übersicht der Ontologie als Treemap	46
5.4	Darstellung der einzelnen Individuen als Tropfen	48
5.5	Eigenschaften des Individuums	49
5.6	Filterung der Tropfen nach Assoziationen	50
5.7	Protokollfenster mit Liste der ausgeführten Aktionen	51
5.8	Aufruf der Ontologie aus der ^{open} sTeam-Weboberfläche	52
5.9	Bearbeitungsmaske für das Hinzufügen eines Individuums	53
5.10	Elementbasierte Annotationen	55
5.11	Navigations- und Bedienelemente der grafischen Anwendung	56

5.12	Genereller Ablauf der dynamischen Grafikerzeugung	59
5.13	Die einzelnen Schichten des Servermoduls	66
5.14	Übersetzungsfunktion der Kontrollstruktur	67
5.15	UML-Diagramm: Gegenseitige Bezüge in der Klassenstruktur	68
5.16	Aufbau der Klassenstruktur des Aktionen-Modells	71
5.17	Sequenzdiagramm der Aktion <i>Modell ableiten bzw. subclass</i>	74

Tabellenverzeichnis

3.1	Primäre (individuelle und kooperative) Medienfunktionen	12
-----	---	----

1 Einleitung

Wissen ist sicherlich der strategische Faktor unserer modernen Gesellschaft. Wissenserwerb kann als kontinuierlicher Prozess betrachtet werden, der über die Grenzen der täglichen Arbeitswelt hinweg auch in das Privatleben reicht. Schlagworte wie *lebenslanges Lernen* oder *Wissensarbeiter* unterstreichen die Wichtigkeit dieses Prozesses und sind gleichzeitig Indikatoren für dessen ganzheitliche Ausprägung auf das tägliche Leben. Bei einem kritischen Blick auf die Kontexte in denen Wissensprozesse stattfinden, ist festzustellen, dass zunehmend Gemeinschaften des Wissenserwerbs gegenüber dem individuellen Lernen in den Vordergrund treten. Damit wird die Wissensorganisation, also das gemeinsame Strukturieren, gegenseitige Bereitstellen und Pflegen von Wissen zur Kernaufgabe kooperativer wie individueller Lern- und Arbeitsprozesse. Wissen begründet sich somit nicht mehr nur in dem reinen Verständnis einer speziellen Thematik, sondern wird zunehmend auch um das Wissen von Zusammenhängen und organisatorischen Komponenten erweitert.

Gerade im Bereich der rechner- bzw. genauer webgestützten Wissensstrukturierung bzw. -vermittlung nimmt dieses Metawissen eine entscheidende Rolle ein. Die kaum messbare Anzahl an möglichen Ressourcen (z.B. Internet) für eine potenzielle Wissensaufnahme und Strukturierungsmittel (z.B. Ordnerstrukturen), die zur Bewältigung dieser Mengen nur bedingt hilfreich sind, zeigen dass neue Formen der Strukturierung – insbesondere bei einer gemeinschaftlichen Wissenserzeugung – notwendig sind. Speziell in einem kooperativen Prozess, in dem mehrere Personen diverse Inhalte produzieren, muss jeder Beteiligte über den derzeitigen Wissensstand der Gruppe ausreichend informiert sein. Fragen wie *Wer in meiner Gruppe verfügt über Spezialwissen in einem Wissensgebiet, Wo platziert sich ein Dokument in der betrachteten Wissensdomäne* oder *Welche Dokumente sind mit dem Vorliegenden thematisch verwandt* sind in so einem Prozess durchaus vorstellbar, aber mit klassischen Strukturierungsmitteln, wie den oben angesprochenen Ordnerstrukturen, nicht ohne weiteres zu beantworten. Zwangsläufig wird die Beantwortung dieser Fragen außerhalb dieser Strukturierung stattfinden, was für die Prozessbeteiligten einen enormen Koordinierungsaufwand bedeutet.

Um diesen Aufwand zu minimieren bedarf es eines Konzeptes, das es erlaubt die eigentlichen Wissenscontainer bzw. -speicher wie z.B. Dokumente oder auch Ordner in einen semantischen Bezug zu setzen. Dies alleine reicht aber nicht aus. Weiter muss dieses Konzept die Möglichkeit zulassen Begrifflichkeiten, Gegenstände (z.B. das oben angesprochene Wissensgebiet oder eine Person) oder kurz Entitäten zu definieren, die Zusammenhänge und Bedeutungen eines Wissensgebietes für alle beteiligten Personen verständlich machen. Das Konzept muss also eine Metaebene

1 Einleitung

anbieten, die auf Basis der vorhandenen Struktur in der Form von z.B. Ordnern genügend Freiraum bietet selbstdefinierte Kontexte zu bilden.

Eine mögliche Form der Strukturierung stellt in dieser Hinsicht das Konzept der Ontologie bereit. Mit Hilfe von Ontologien ist es möglich bestimmte Begriffe zu klassifizieren, gegenseitig zu assoziieren und auch abzuleiten. Ferner werden auf Basis der definierten Begriffe tatsächliche bzw. real existierende Objekte konstruiert, die den abstrakten Entitäten entsprechen. Das Konzept der Ontologie hat seinen Ursprung in der Philosophie und beschäftigt sich mit der Abgrenzung vom Abstrakten und Realen. Obwohl dieses Konzept schon längere Zeit im Kontext der Informatik zu finden ist, erfährt es durch die von Tim Berners-Lee (vgl. Berners-Lee u. a., 2001) aufgeworfene Idee des Semantischen Netzes eine neue Bewertung. Kerngedanke dieses Netzes ist es, auf Basis von Ontologien Rechnern zu ermöglichen, Sinnzusammenhänge wie z.B. den Inhalt einer Webseite zu ergründen und dem Menschen den Umgang mit Informationen zu erleichtern. Im Grunde stellen diese Ontologien des Semantischen Webs somit Verknüpfungen zwischen verschiedenen Ressourcen (z.B. Webseiten) her und können mit Hilfe der klassifizierten Begrifflichkeiten verglichen werden. Dabei muss eine Ressource nicht zwangsläufig ein Dokument sein, sondern kann ebenso eine Person oder ein Fachgebiet beschreiben.

Eine Frage, die sich in diesem Zusammenhang geradezu aufdrängt, lautet zunächst wer diese Zusammenhänge definiert und strukturiert? Die vorliegende Arbeit gibt hierzu eine klare Antwort: Der Mensch. Grundthese ist also, dass der Mensch persönliches Wissen und Weltanschauung zunächst in eine Ontologie und somit in den Rechner überführt. Hierbei wird insbesondere davon ausgegangen, dass Ontologien nicht einseitig in einer Autor-Nutzer Rolle erstellt werden, sondern jede beteiligte Person beide Rollen ausführt. Daraus erwachsen neue Anforderungen, die den Kern dieser Ausarbeitung bilden. Mit der primären Betrachtung der Ontologie als direktes Austauschformat von Sinnzusammenhängen zwischen Menschen, oder genauer zwischen Personen deren Ziel die gemeinsame Strukturierung eines Wissensfeldes ist, ergeben sich neue Anforderungen, die analysiert werden müssen.

Die Hauptanforderung bildet hierbei die Notwendigkeit einer kooperativen Erstellung dieser Struktur. Kooperative Erstellung bedeutet in diesem Zusammenhang nicht nur ein gemeinsames Strukturieren und Arrangieren von Dokumenten, sondern vielmehr – und dies begründet sich in der Mischung aus semantischen Verbindungen und Ressourcen – die Findung eines Konsenses aller beteiligten Personen über die entstehende Wissensstruktur. Ein Großteil dieser Arbeit widmet sich somit der Analyse und Ausarbeitung von Strukturen, die es erlauben Ontologien in Kooperation zu konstruieren.

Aufgrund der Betrachtungsweise Ontologien primär für die Kommunikation von Wissen in einer direkten Mensch zu Mensch Beziehung zu nutzen, und die maschinelle Auswertung dieser Struktur als einen nachgelagerten Prozess anzusehen, muss für diesen Austausch eine geeignete Plattform angeboten werden. Für die Konstruktion einer solchen Plattform wird das Konzept der Ontologien auch

1 Einleitung

daraufhin analysiert wie dessen Struktur in eine Visualisierung überführt werden kann.

Neben einer Skizzierung möglicher Einsatzzwecke für kooperative Ontologien wird auf Basis der oben aufgeworfenen und im Verlauf der Arbeit gewonnenen Erkenntnisse im Abschluss dieser Arbeit ein Werkzeug für die kooperative Konstruktion von Ontologien vorgestellt. Bestimmt für den Einsatz in der ^{open}Team-Umgebung soll dieses Hilfsmittel die Nutzer bei der Strukturierung von Dokumenten und Zusammenhängen unterstützen und somit Ausdrucksmittel für eine gemeinsame Wissensbasis bzw. Wissensstruktur sein.

2 Ontologien

Der Begriff der Ontologie taucht seit einiger Zeit vermehrt im Kontext des Semantischen Webs und des Wissenmanagements auf. In diesem Kapitel soll zum einen der Begriff abgegrenzt bzw. im Sinne eines computergestützten Einsatzes verdeutlicht werden, und zum anderen die konzeptuelle Nutzung dieser Struktur im Bezug auf das Semantic Web dargelegt werden.

2.1 Begriffsbestimmung

Der Duden definiert den Begriff der Ontologie folgendermaßen: »*Ontologie die: - :Lehre vom Sein, von den Ordnungs-, Begriffs- u. Wesensbestimmungen des Seienden.*« (vgl. Duden, 2000) Das Konzept der Ontologie hat seinen Ursprung in der Philosophie und thematisiert den Objektbegriff, wobei verschiedene metaphysische Ansätze zur Anwendung kommen. Um die Komplexität der philosophischen Definition exemplarisch darzulegen, sollen an dieser Stelle kurz zwei grundverschiedene Ansätze vorgestellt werden, die den weiten philosophischen Raum einer Ontologie unterstreichen sollen. Für eine mögliche Definition des Objektbegriffes stellen Wolfgang Hesse und Hubert von Braun Realismus und Idealismus als jeweilige Grundposition bei der Definition von Objekten und deren Begrifflichkeit gegenüber. Während der Realismus grundsätzlich von der Annahme ausgeht, dass Objekte von sich aus implizite Unterscheidungen aufweisen, bezieht sich der Idealismus auf eine subjektive, vom Menschen definierte Unterscheidung einzelner Begrifflichkeiten (vgl. Hesse und von Braun, 2001).

Diese Thematik kann und soll an dieser Stelle nicht erschöpfend behandelt werden, sondern zeigen inwieweit das Konzept der Ontologie in Philosophie und Metaphysik verankert ist. Vielmehr soll die Begrifflichkeit der Ontologie für das weitere Verständnis innerhalb dieser Arbeit geklärt und präzisiert werden, wobei der philosophische Charakter einer Ontologie auch einen Einfluss auf die informatrische Auslegung dieses Konzeptes beinhaltet. Wie nahe sich Informatik und Philosophie – im Sinne einer Definition des Objektbegriffs – stehen, wird deutlich am Beispiel der *Objekt Orientierten Softwareentwicklung*. Während diese sich mit *Klassen* und *Objekten* befasst, legt die philosophische Herangehensweise die Bestimmung von *Universalien* und *Einzel dingen* (vgl. Scheffe, 2001) zugrunde. Der später beschriebene Aufbau einer Ontologie zeigt, dass das Konzepte der *Objekt Orientierten Softwareentwicklung* und somit der Unterscheidung von Abstraktem und Realem mit der informatrischen Ausprägung der Ontologie gewisse Gemeinsamkeiten aufweist.

2 Ontologien

Im Bezug auf das Wissensmanagement beschreibt T. R. Gruber eine Ontologie wie folgt »*An ontology is an explicit specification of a conceptualization*« (vgl. Gruber, 1995). Eine Ontologie stellt also eine konkrete Spezifizierung eines Konzeptes dar. Was aber ist nun ein Konzept in diesem Kontext? Nach Gruber ist ein Konzept »[...] *an abstract, simplified view of the world that we wish to represent for some purpose.*«, also eine vereinfachte Sicht der dargestellten Zusammenhänge. In diesem Kontext erscheint auch häufig der Begriff der »Wissensdomäne« der andeutet, dass Wissen in einer Ontologie eine spezielle Abgrenzung erfährt. Eine Ontologie beschreibt somit nicht die Welt als Ganzes, sondern immer nur speziell fokussierte Ausschnitte aus der Welt bzw. im Sinne der Wissensrepräsentation einen definierten Wissensbereich. Bei einer Ontologie handelt es sich also im informatorischen Sinne um eine Struktur zur Darstellung von Wissen und dessen Zusammenhängen untereinander. Entitäten in Form von Taxonomien und Eigenschaften, sowie Relationen bzw. Beziehungen von abstrakten und realen Dingen dienen dazu, diese Zusammenhänge in maschinell lesbarer Form zur Verfügung zu stellen.

2.2 Aufbau einer Ontologie

Zur weiteren Klärung was sich hinter dem Konzept der Ontologie verbirgt wird an dieser Stelle eine kurze Beschreibung über die internen Zusammenhänge dieser Struktur gegeben. Dabei entspricht diese Erläuterung einer informatorischen Sichtweise bzw. im engeren Sinne der des später beschriebenen Semantischen Webs. Eine Ontologie lässt sich in zwei Aufbauformen unterteilen. Zum einen besteht die Ontologie aus einer Taxonomie von Begriff zu Oberbegriff bzw. einer abstrakten Klassifizierung von Entitäten (Modellen) und deren Beziehungen untereinander. Dem gegenüber steht die Ableitung vom Modell zu real existierenden Objekten bzw. die Einsetzung von konkreten Entitäten (Individuen) basierend auf den im Modell definierten Eigenschaften und Beziehungen.

Ein Modell zeichnet sich zunächst durch die Definition von Eigenschaften, die Bildung von Assoziationen und eine Vererbungs- bzw. »*ist ein*« Beziehung zu anderen Modellen aus. Im Bezug auf die Vererbung werden von einem generischen Oberbegriff bzw. Modell ausgehend immer speziellere Modelle abgeleitet, die definierte Eigenschaften und Assoziationen des Ober- bzw. Vatermodells übernehmen und um eigene Definitionen erweitern können. Zudem besteht die Möglichkeit neben der einfachen Vererbung (jedes Modell hat genau eine »*ist ein*« Beziehung) auch Ableitungen in einer mehrfachen Vererbung (jedes Modell hat mehrere »*ist ein*« Beziehungen) zu konstruieren. Im Folgenden richtet sich der Fokus ausschließlich auf die einfache Ableitung. Dies begründet sich zum einen durch die stark verminderte Komplexität als auch durch die Möglichkeit mit Hilfe von Assoziationen und Eigenschaften diese mehrfachen Ableitungen zu kompensieren und besser zu verdeutlichen. Abbildung 2.1 soll diesen Sachverhalt anhand eines Ausschnitts einer erdachten Ontologie eines *Zeitungsverlages* grafisch näher bringen. Die beiden Ausschnitte zeigen zwei unterschiedliche Möglichkeiten das Modell *Arbeitnehmer*

2 Ontologien

einer *Tätigkeit* zuzuordnen. Die linke Abbildung in 2.1 baut auf einer mehrfachen Vererbung auf und besagt das ein *Redakteur* sowohl ein *Arbeitnehmer* als auch ein *Autor* ist. Die rechte, auf eine einfache *ist-ein* Beziehung aufbauend, zeigt ebenfalls dass ein *Redakteur* ein *Arbeitnehmer* ist. Der Unterschied besteht aber in der Bildung der Assoziation *übt aus*, die jedem Arbeitnehmer (und somit auch jedem *Redakteur* oder *Grafiker*) eine Tätigkeit zuweist, die bei einer Konstruktion von Individuen (auch Instanzen genannt) z.B. in die Formulierung *Sven Gerber* (als Autor) *übt aus* (als Assoziation) *Autor* (als Tätigkeit) gebracht werden kann. Hierbei wird deutlich, dass es zum einen unterschiedliche Wege in der Modellierung einer Ontologie gibt, und zum anderen, dass die einfache Vererbung durch das Bilden von Assoziationen die Zusammenhänge besser verdeutlichen kann. Der resultierende Aufbau

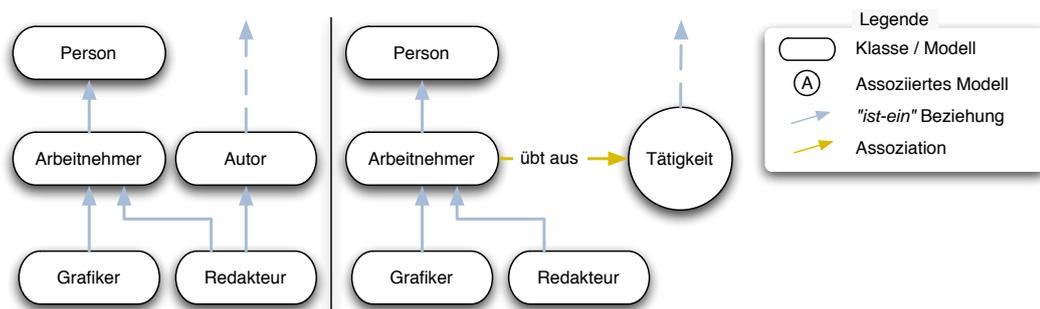


Abbildung 2.1: Modell-Ausschnitt einer Ontologie mit mehrfacher (links) und einfacher (rechts) Vererbung

einer einfachen Vererbung spiegelt die Struktur einer strikten Hierarchie bzw. (graphentheoretisch) eines gewurzelten Baumes wieder. Einzelne Modelle können zu anderen Modellen Assoziationen aufbauen die, entsprechend der Sichtweise, die Hierarchie in eine Netzstruktur transformieren. Ebenso wie Modelle können auch Eigenschaften und Assoziationen abstrakt definiert und in eine Vererbungshierarchie überführt werden. Dieser Ansatz ist besonders für eine spätere maschinelle Auswertung interessant, da hiermit gegenseitige Beziehungen von Eigenschaften und Assoziationen nachvollzogen werden können. So kann z.B. die Assoziation »*ist Vater von*« als eine Spezialisierung der Assoziation »*ist Elter von*« gesehen werden (vgl. Ziegler, 2003).

Dem Modellaufbau stehen auf der anderen Seite, wie oben erwähnt, die definierten Individuen gegenüber. Diese Individuen stellen eine konkrete Ausprägung des Modells dar, die einerseits die im Modell definierten und geerbten Eigenschaften mit tatsächlichen Werten belegen, und andererseits festgelegte und geerbte Assoziationen mit anderen Individuen verknüpfen (vgl. Abbildung 2.2). So kann beispielsweise das Individuum *La Traviata* als konkrete Einsetzung für das Modell *Oper* mit einer Assoziation *hat Komponisten* und einer Eigenschaft *Uraufführung am* beschrieben werden. Während das Modell *Oper* über die Assoziation *hat Komponisten* mit dem Modell *Komponist* verbunden wird, stellt sich im Sinne der Individuen

2 Ontologien

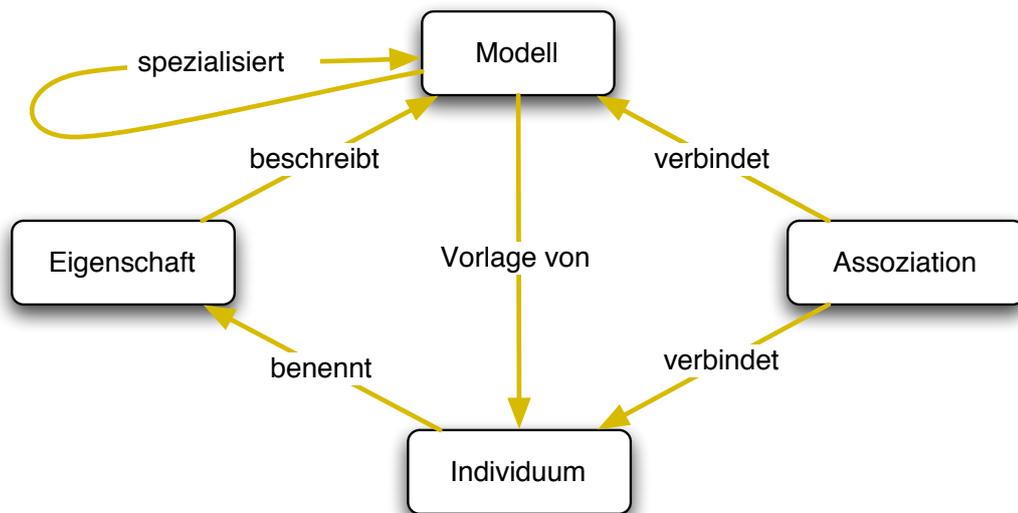


Abbildung 2.2: Gegenseitige Abhängigkeiten innerhalb einer Ontologie

diese Struktur in der Form *La Traviata hat Komponisten Verdi* dar. Dies zeigt, dass die definierten Assoziationen aus der Modellstruktur auch in die Struktur der Individuen einfließen. Anders als beim Aufbau der Modellstruktur stellen sich die Beziehungen der Individuen untereinander als reine Netzstruktur dar, was eine hierarchische Zuordnung zunächst erschwert. Durch den Bezug auf das als Vorlage dienende Modell können aber Rückschlüsse auf die Einordnung innerhalb der Modellhierarchie gezogen werden, was eine semantische Loslösung bzw. Trennung von Individuum und Modell ausschließt.

2.3 Ontologien und das Semantische Web

»The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.« (vgl. Berners-Lee u. a., 2001)

Das derzeitige World Wide Web zielt auf die Bedürfnisse einer menschlichen Betrachtung von Informationen ab. Während Webinhalte für den Menschen als strukturierte Information präsentiert und nachvollzogen werden können, erscheint die – für HTML¹ typische – Vermischung von Darstellung und Inhalt als unüberwindbare Hürde für ein computerisiertes bzw. automatisches Erschließen von Bezügen innerhalb dieser Inhalte. Diese maschinelle Hürde spiegelt sich nach mehr als einer Dekade des World Wide Webs in der kaum beherrschbaren Dichte an möglichen

¹HTML Hyper Text Markup Language - Auszeichnungssprache für Webinhalte

2 Ontologien

Informationen und Inhalten wider. Stichwortbasierte Suchmaschinen zeigen alltaglich, dass trotz diverser Optimierungen ein maschinelles Suchen den Informationsraum nur bedingt darstellen kann. Eines der Kernelemente des Semantischen Webs bildet somit die – auf das derzeitige WWW aufbauende – computerisierte Erschliessung der dargelegten Inhalte. Hierbei bilden die oben beschriebenen Ontologien das notige Konzept zur Definition der inhaltlichen Zusammenhange. Basierend auf der Festschreibung von Entitaten und Relationen konnen Ontologien als Datenformat in den bestehenden Kontext des World Wide Webs eingebunden werden. Die jeweilige Ontologie stellt eine Taxonomie von abstrakten Modellen bzw. Konzepten, deren Beziehungen und Ableitungen untereinander und zu real existierenden Objekten bereit (vgl. Kapitel 2.2).

Basierend auf diesem Konstrukt liee sich nun z.B. eine Suche nach einem bestimmten Themengebiet stark eingrenzen, indem speziell die Inhalte bzw. Ergebnisse prasentiert werden, die auf ein bestimmtes Modell bzw. eine bestimmte Ontologie verweisen oder den jeweiligen Kontext der Suche gleichzeitig mit darstellen (vgl. Berners-Lee u. a., 2001). Aufgrund des Beziehungsgeflechts ermoglichen Ontologien auch Prozesse die eine Erweiterung derzeitiger und zukunftiger Web-Dienste begunstigen konnen. Denkbar ist hier z.B. der Einsatz eines Agentenprogramms das aufgrund der in einer Ontologie aufgestellten Regeln bzw. Zusammenhange (unter Berucksichtigung definierter Vorlieben eines Reisenden) eine komplette Geschaftsreise (also Flug, Hotel, Mietwagen etc.) bucht (vgl. Daconta u. a., 2003). Die Liste der Beispiele, die auf Basis dieses Konzeptes konstruiert werden konnen, liee sich sicherlich erweitern. Deutlich soll an dieser Stelle zunachst nur eines werden: Das es moglich erscheint – nach Berners-Lee auch ohne Kunstliche Intelligenz (vgl. Berners-Lee u. a., 2001) – Bezuge die fur den Menschen inherent gegeben sind, durch Computer erschlieen zu konnen. Um dies zu erreichen bedarf es aber formaler Regeln die logische und berechenbare Schlusse zulassen. Ob die Vision eines Semantischen Webs nicht doch in den Raum der Kunstlichen Existenz hineinreicht, soll hier aber weder bestatigt noch negiert werden.

2.4 Grundannahmen

Es stellt sich vielmehr die Frage, wie semantische Verbindungen uberhaupt Einzug in die Welt der Computer erhalten konnen. Hierzu sollen nun einige Grundannahmen getroffen werden, die dieser Arbeit als Ausgangsbasis dienen. Die Aufstellung von semantischen Bezugen – so profan es klingen mag – werden nicht von Maschinen fur Maschinen gemacht, sondern von Menschen fur Menschen. Im Sinne des Semantischen Webs schafft der Mensch also Sinnzusammenhange, die der Computer als Mediator anderen Menschen zur Verfugung stellt. Die folgenden Annahmen werden daraus abgeleitet:

1. Der Mensch uberfuhrt sein Wissen als primaren Schritt in den Computer. Dieses Wissen soll bzw. kann in Form von Ontologien in eine persistente Struktur

2 Ontologien

überführt und so anderen zugänglich gemacht werden. Diese Speicherung von Wissen kann abstrahierend von der maschinellen Interpretation als Ausgangsbasis für das Entstehen von neuem Wissen genutzt werden.

2. Das ontologische Konzept wird primär für den Austausch von Wissen und Zusammenhängen (von z.B. Dokumenten) in einer Mensch-Mensch Beziehung adaptiert. Als Schwierigkeit erweist sich hierbei, dass Wissen ebenso individuell ist wie der jeweilige Mensch und verschiedene Weltansichten zwangsläufig zu einem Miss- bzw. Unverständnis der repräsentierten Sinnzusammenhänge führen müssen.
3. Es genügt somit nicht das Wissen eines Einzelnen in eine Ontologie zu überführen, vielmehr muss dieses auf einer gemeinschaftlichen bzw. kooperativen Basis mit der Gesamtheit aller Prozessbeteiligten stattfinden. Hierfür bedarf es geeigneter Strukturierungsmittel, die diesen gemeinschaftlichen Prozess unterstützen und den Wissenskontext verständlich abbilden.
4. Für eine gemeinsame Betrachtung und Erstellung einer Ontologie in einer Mensch-Mensch Beziehung müssen mögliche Formen für die grafische Repräsentation des entstehenden Wissensraumes gefunden werden. Weiter bedarf es für die kooperative Bearbeitung einer Ontologie weiterer Mittel für die Kommunikation der beteiligten Personen.

3 Kollaborative Ontologien zur Strukturierung von Wissen

Seit je her versucht der Mensch sein persönliches Wissen zu externalisieren und anderen zugänglich zu machen. Sei es nun durch orale Kommunikation, oder durch Auslagerung auf ein Speichermedium. Es stellt sich jedoch die Frage, in welchem Maße persönliches Wissen auf bzw. über ein Medium übertragbar ist, und wie es von anderen nachvollzogen werden kann. Ebenso wie der Erwerb ist auch die Weitergabe von Wissen eine sehr individuelle Handlung, die sicherlich nicht mit einer bloßen Addition von Wissen gleichzusetzen ist.

Schlagworte wie *Wissensgesellschaft* oder *Informationsgesellschaft* gehen oft einher mit – oder basieren gar auf – der fortschreitenden Vernetzung von digitalen Systemen und deren zunehmende Bedeutung in allen Bereichen des täglichen Lebens. Die Betonung von *Wissen* in Bezug auf *Gesellschaft* deutet an, welcher Stellenwert dem Faktor Wissen zukommt. Nina Degele beschreibt die zentrale Rolle der Wissensproduktion als Rekursiven Prozess, als eine »[...] Anwendung von Wissen auf Wissen« (vgl. Degele, 2000). Wie aber kann nun dieses Wissen auf Wissen unzählige Informationen, wie sie beispielsweise das Internet bereithält, in eine Form von explizitem Wissen transformieren bzw. produzieren? Und inwieweit können kooperative Prozesse zur gemeinsamen Strukturierung diesem Ziel dienen?

Die netzgestützte Strukturierung von Wissen geht einher mit einer Vielzahl von Hilfsmitteln, wie der häufigen lexikografischen Erschließung durch Stichworte und Metadaten und in vermehrter Form mit dem Einsatz von grafischen Repräsentationsformen, die vom Benutzer direkt arrangiert und editiert werden können. Im Folgenden wird nun ein möglicher Rahmen beschrieben, der es erlaubt Wissen mit Hilfe von Ontologien kooperativ zu erstellen und diesen Prozess mit grafischen Strukturierungsmitteln zu unterstützen.

3.1 Kooperative Ontologienbildung

Abstrahierend von der maschinellen Interpretation einer Ontologie, wird die direkte Erstellung von Ontologien auf der Nutzerseite fokussiert. Die Grundannahme dieser Arbeit ist, dass der Mensch für eine mögliche spätere maschinelle Erschließung von Semantik in Form von Ontologien der bestimmende Faktor ist. Um eine kooperative Wissensstrukturierung in Form von Ontologien zu ermöglichen, müssen aber zunächst verschiedene Herangehensweisen untersucht werden, die zum einen den Nutzen einer kooperativen Ontologie-Bildung unterstreichen und zum

anderen Funktionalitäten benennen, die einen kooperativen Aufbau unterstützen können.

3.1.1 Rollenverteilung

Aus Nutzersicht bietet die Strukturierung von Wissen in einer Ontologie die Möglichkeit, Wissensfelder immer im Kontext zu anderen Wissensfeldern abzubilden und auch zu identifizieren. Die einzelnen Entitäten einer Ontologie, seien sie nun abstrakt oder mit Bezug auf existierende Objekte, stellen immer nur eine Form von Metadaten dar, oder anders ausgedrückt eine Art Schablone, die den gegenseitigen Bezug verschiedener Objekte zueinander beschreibt. In einer Ontologie nehmen existierende Objekte nur eine Rolle an, die sich von der Rolle (des gleichen Objekts) in einer anderen Ontologie komplett unterscheiden kann. Als Beispiel für eine mögliche Unterscheidung könnten hier die vielfältigen Rollen des Objektes *Student* wie z.B. als Teilnehmer einer Vorlesung, als Verwaltungsgegenstand oder als Autor einer Diplomarbeit genannt werden. Die Ontologie bildet somit immer nur definierte Kontexte ab, deren Festlegung rein subjektiv entschieden wird.

Eine Vertiefung des Begriffes *Rolle* wirft die Fragen auf, *Wer* diese Rollen vergeben soll, *Warum* ein definiertes Objekt gerade eine bestimmte Rolle übernehmen soll und für *Wen* diese Rollenverteilung festgelegt wird? Klassisch würde diese Frage sicher folgendermaßen beantwortet: Wenige (Personen) konstruieren, viele (Personen) rezipieren. Während die Beantwortung dieser Frage für ein Theater-schauspiel noch sinnvoll erscheinen mag, ist die Gültigkeit für den Prozess des Wissensaufbaus nicht ohne weiteres gegeben. Gerade im Kontext des Semantischen Webs, in dem es ja um die Konstruktion von Zusammenhängen geht, wird dieser Fragestellung zu wenig Beachtung geschenkt (vgl. Konferenzbände: Horrocks und Hender, 2002; Fensel u. a., 2003). Da es sich bei der Aneignung von Wissen um einen kommunikativen Prozess handelt, kann aus der Beantwortung der letzteren Frage (Wen) auch die Antwort der Frage abgeleitet werden, wer eine Ontologie aufbaut bzw. die Rollen verteilt. An dieser Stelle bietet sich eine Einbindung des kollaborativen Ansatzes zur Erstellung von Ontologien an. In einer teamgestützten Wissenserschließung bezieht sich der Aufbau einer Struktur nicht auf die Ausarbeitungen eines Einzelnen, sondern auf eine gemeinsame Bewegung hin zu einer vielschichtigen Betrachtungsweise der beobachteten Wissensdomäne.

Dies negiert zwar nicht, dass Ontologien auch zur Repräsentation von Wissen in einer »*Ein Autor für Viele Rezipienten*« Relation geeignet sind, doch ist es geradezu unentbehrlich – sollen die Wissens-Verknüpfungen von den Nutzern nachvollzogen werden – diese auch in den Prozess des Wissensaufbaus mit einzubinden. Eine Betrachtung der aktuellen Diskussion im Bezug auf das Semantische Web zeigt, dass zwar kooperative Ansätze zur Bildung von Ontologien existieren (vgl. Sure u. a., 2002), jedoch der Fokus im Bezug auf die *Autor zu Nutzer-Beziehung* eher einseitig ist und eine Wechselwirkung von Autoren- und Nutzer-Rolle kaum in Betracht gezogen wird.

Der Ansatz Ontologien basierend auf einer »*Viele Autoren für Viele Rezipienten*«

Relation aufzubauen, bietet einen stetigen, wie oben angeführt, kommunikativen Prozess, der das Validieren und Anfügen neuer Erkenntnisse erleichtert. Die omnipräsente Frage *Warum* ein spezieller Zusammenhang definiert wurde, bildet hierbei den Kern der kooperativen Herangehensweise.

3.1.2 Medienfunktionen

Für den Prozess eines kooperativen Ontologie-Aufbaus stellt sich auch die Frage, in welcher Form eine Ontologie modifiziert werden kann. Da es sich bei der hier beschriebenen Form von Ontologien auch um ein syntaktisches Gebilde in Form von Daten handelt, wird hier eine Analyse möglicher Funktionen auf Basis dieser Daten beschrieben. Tabelle 3.1 soll darüber Aufschluss geben welche primären Medienfunktionen die Ontologie einerseits als Ganzes (1. Spalte), andererseits in Bezug auf die Entitäten und Beziehungen erfüllen (●) bzw. nicht erfüllen (-) kann. Neben den individuellen Medienfunktion wie z.B. dem Erzeugen, Löschen oder Arrangieren von Dokumenten werden auch die speziell für kooperative Prozesse nötigen Medienfunktionen in Hinblick auf das Übertragen an andere Kooperationspartner, dem Einräumen spezieller Berechtigungen und die Möglichkeit einer gemeinsamen Sicht auf die Struktur untersucht (vgl. Hampel, 2002). Ferner werden im dritten Tabellenabschnitt weitere Medienfunktion benannt, die speziell den kooperativen Aufbau einer Ontologie unterstützen und erweitern können.

Funktionen	Ontologie	Modell	Assoziation	Eigenschaft	Individuum
Erzeugen	●	●	●	●	●
Löschen	●	●	●	●	●
Editieren	●	●	●	●	●
Arrangieren	●	●	-	-	-
Verknüpfen	●	●	-	-	●
Übertragen	●	-	-	-	-
Zugreifen	●	-	-	-	-
Synchronisieren	●	●	●	●	●
Kommentieren	●	●	●	●	●
Archivieren	●	●	●	●	●

Tabelle 3.1: Primäre (individuelle und kooperative) Medienfunktionen

Die dargelegte Auflistung zeigt, dass sich die Funktionalität der Ontologie als Ganzes nicht in allen Bereichen auf die jeweiligen Entitäten und Beziehungen übertragen lässt. So kann zwar die Ontologie als Datei übertragen werden, eine Übertragung einzelner Entitäten und Beziehungen kann aufgrund der zusammenhängenden Struktur aber ausgeschlossen werden. Ebenso verhält es sich in Bezug auf die Gewährung von Zugriffen auf eine bestimmte Entität oder Beziehung. Sicherlich ist es denkbar, das Anlegen und Editieren von gewissen Elementen zu reglementieren, würde aber innerhalb der Ontologie die kooperative Arbeitsweise zu sehr

einschränken und eine Betrachtung des Gesamtkontextes unmöglich machen. Im weiteren Verlauf dieser Arbeit werden die einzelnen hier dargelegten Medienfunktionen in Bezug auf Nutzung und kooperative Unterstützung genauer analysiert.

3.1.3 Ontologien als transportierbare Wissensseinheiten

Das Konzept von Ontologien basiert auf den Bezügen zwischen tatsächlich vorhandenen Objekten, sowie deren Beziehung und Einstufung bzw. Klassifizierungen untereinander. Hierbei bildet die Ontologie als Datei einen in sich abgeschlossenen Datenraum, geteilt in Sinnzusammenhänge auf der einen Seite und Verweise auf externe Quellen (wie z.B. Dokumente) auf der anderen Seite. Die Trennung zwischen (im informatorischen Sinne) Daten und deren Relationen gestattet es somit, dass Semantik und Wissensstruktur transportiert werden können und die eigentliche Wissenscontainer wie z.B. Texte, Bilder etc. an ihrem ursprünglichen Ort verbleiben. Um die externen Referenzen zu z.B. Dokumenten von jedem Standort (der Datei) zu gewährleisten, bedarf es einer eindeutigen Identifizierung bzw. Adresse (z.B. mit Hilfe einer Webadresse) auf die verwiesene Ressource. In gleicher Weise benötigen – obwohl eher für eine maschinelle Auswertung interessant – auch die einzelnen Entitäten einer Ontologie eine eindeutige Adresse um gegenseitig vergleichbar zu sein. Eine genaue Betrachtung bezüglich dieser eindeutigen Adressierung findet sich in Kapitel 5.2. Mit Hilfe dieser eindeutigen Adressierung von Konzept und insbesondere der verbundenen Dokumente wird die Ontologie als Datei unabhängig vom jeweiligen Standort und somit austausch- bzw. transportierbar. Der Austausch bzw. Transport von Sinnzusammenhängen ist somit nicht nur innerhalb eines (kooperativen) Systems möglich, sondern auch über dessen Grenzen hinaus und kann somit auch Dritten zugänglich gemacht werden.

Basierend auf der thematischen Ausprägung und dem Grad der Spezialisierung kann diese Struktur in einen bestehenden Prozess der Wissenserarbeitung einfließen, als erweiterte Referenz dienen oder sogar als Ausgangsbasis bzw. Vorlage für eine weitere Spezialisierung der Struktur benutzt werden. Im Sinne der Medienfunktionen bedeutet dies, dass die Ontologie – in Verbindung mit entsprechenden Zugriffsrechten – für einen gegenseitigen Austausch zwischen Nutzern und Systemen geeignet ist.

Ein wichtiges Kriterium bei der Adaption fremder Wissensbestände ist aber immer die Frage nach der Validität bzw. Richtigkeit der dargelegten Zusammenhänge. Es handelt sich bei einem Transport von Wissen also immer um ein Vertrauensverhältnis zwischen einer referenzierten Wissensquelle und einem Wissensempfänger (vgl. Kuhlen, 2003). Insbesondere wenn keine explizite Institution oder Person hinter der Quelle zu erkennen ist, gestaltet sich der Aufbau von Vertrauen bzw. die Annahme der Richtigkeit als schwierig. Im Bezug auf die Ontologie als Wissensquelle kommt der kooperativgestützte Aufbauprozess als vertrauensbildende Referenz zum Tragen. Stellt sich die Ontologie auch als anonymes Konstrukt in Bezug auf Institution und Person dar, kann diese aber aufgrund der gemeinschaftlich erarbeiteten Wissensbasis durchaus den Anspruch auf Validität besitzen.

Bei einer Betrachtung über Sinn und Zweck eines Transportes von Wissen wird die Wichtigkeit dieser Eigenschaft einer kooperativ erstellten Ontologie deutlich. Sinn und Zweck kann hierbei nur die geplante oder gewünschte Adaption des in der Ontologie dargelegten Wissensbestandes sein. Um die oben genannte Richtigkeit bzw. Validität dieses Wissens leichter anzuerkennen, kann der Bezug auf das im kooperativen Prozess herausgebildete gemeinschaftliche Verständnis von Vorteil sein.

3.1.4 Formen kooperativer Ontologien

Aufgrund der offenen thematischen Struktur einer Ontologie lassen sich viele mögliche Einsatz-Szenarien konstruieren. Neben konkreten Anwendungsfällen, die im Kapitel 4 behandelt werden, soll hier beschrieben werden, welche Arten von Ontologien in einem kooperativen System denkbar sind.

Wird beispielsweise versucht ein System wie *openTeam* mit Nutzern, Dokumenten, Gruppen usw. in einen semantischen Bezug zu setzen, kann der Prozess der Ontologien-Bildung als globale Aufgabe angesehen werden, in dem jeder die Möglichkeit erhält seinen persönlichen Nutzerraum in die Ontologie mit einzubringen. Denkbar in diesem Fall sind beispielsweise Organigramme, die versuchen verschiedene Nutzer in einen systembasierten Gesamtkontext zu setzen. Dies kann sich z.B. in *Nutzer zu Nutzer Beziehungen*, oder *Nutzer zu System- bzw. Dokumentenzusammenhängen* widerspiegeln. Ein fortgeschrittener Fall wäre hierbei die Einbindung von Nutzern zu speziellen Themen bzw. Fachgebieten oder sogar einer Ontologie, die über alle im System vorhandenen Ontologien – als eine weitere Ebene des Metawissens – informiert.

Eine weitere Form, die auf speziellere Wissensdomänen eingeht, ist die Form einer geteilten bzw. gruppenorientierten Ontologie. Diese Form lässt sich am besten in den Lehr-Lernprozess integrieren, da hier aufgrund der gleichen Zieldefinition der Lernenden die Struktur feinere Unterscheidungen von Wissen zulässt als in einer systemweiten Ontologie. Vorstellbare Szenarien wären hier z.B. die Aufarbeitung spezieller Vorlesungsstoffe zum Verständnis des Inhaltes und dessen Zusammenhang zu anderen Thematiken, oder auch die Einordnung eines Themengebietes in die gesamte Vorlesung. Dem Nutzer bzw. dem Lernenden wird hier in besonderem Maße die Möglichkeit gegeben sich und seine erstellten Materialien, wie z.B. Dokumente, in Bezug zur Vorlesung und der kooperativen Wissensstruktur zu sehen.

Die Untergliederung dieses Gruppenansatzes kann zu einer weiteren Form von privaten bzw. wenige Personen betreffenden Ontologien führen, die von Kleingruppen erstellt werden. In diesen können sich Prozesse ausbilden, die der Erarbeitung einer gemeinsamen (Wissens-) Basis dienen. Dies kann zum einen, um das Beispiel der Vorlesung aufzugreifen, als eine Art Übungsevaluierung dem Lehrenden dazu dienen, den Kenntnisstand von wenigen besser zu bemessen bzw. nachzuvollziehen. Auf Seiten der Lernenden kann diese Form der Bearbeitung das gegenseitige Verständnis der Gruppenteilnehmer untereinander und die des behan-

delten Lehrstoffes verbessern.

Die hier unterschiedenen Facetten möglicher Ontologie-Formen zeigen, dass sich je nach Einsatz und Anzahl von Nutzern verschiedene Ausprägungen differenzieren lassen. Die Unterteilung von globalen, geteilten und privaten Ontologien deutet aber auch an, dass diese in einer *Teil Ganzes* Beziehung stehen können und nicht zwangsläufig in einem parallelen Prozess zueinander entstehen müssen. Die jeweils gewählte Thematik entscheidet jedoch stark über das Ineinandergreifen der aufgezeigten Formen.

3.1.5 Entwurfsmethoden und Inhalt

Vor dem eigentlichen Prozess der Wissensstrukturierung muss zunächst die thematische Zuordnung getroffen werden, d.h. welche Wissensdomäne mit welcher Zielrichtung bzw. Betrachtungsweise die Ontologie repräsentieren soll. Diese Metadaten über die Ontologie können sowohl generell darüber Auskunft geben welches Thema behandelt wird, als auch eine Vorabdefinition von Begrifflichkeiten liefern, die den Kern der Ontologie bilden. Des Weiteren können Referenzen zu bereits existierenden Ontologien gezogen werden, oder im Sinne der Adaption von Wissen, auf welche Ontologie das entstehende Wissenskonstrukt aufbauen soll.

Die Formulierung eines Schlüsselkonzeptes definiert auch zum Teil die Breite bzw. Tiefe der entstehenden Wissensstruktur. Geht das Schlüsselkonzept von generellen Begrifflichkeiten aus, wie z.B. *Beschreibung von universitären Forschungsgebieten*, lässt dies eine breite bzw. offene Herangehensweise in Bezug auf die Spezialisierung zu, z.B. *Naturwissenschaften, Wirtschaftswissenschaften etc.*, die sich im weiteren Verlauf vertiefen lassen zu z.B. *Betriebswirtschaftslehre, Produktionswirtschaft etc.* Mit der Formulierung des Schlüsselkonzeptes wird – in einem gewissen Rahmen – die Spezialisierung bzw. Komplexität einer Wissensstruktur vorgeben.

Der eigentliche Aufbau einer einer Ontologie kann in mehrere Ansätze unterteilt werden (vgl. Noy und McGuinness). Da das Grundkonzept einer Ontologie von einem abstrakten Modell bzw. Konzept (in englischer Literatur auch oft als »*Thing*« bezeichnet) ausgeht, kann die Integration von erweiterten Modellen als ein in der Hierarchie abwärts gerichteter Ansatz (*top-down*) verstanden werden, in dem schrittweise eine immer speziellere Wissensdomäne aufgebaut wird. Dem gegenüber steht der Ansatz, ein gerade betrachtetes Konzept direkt in die Ontologie zu integrieren und dann sukzessive die generellen, hierarchisch höher liegenden Elemente (*bottom up*) zu definieren. Der dritte Ansatz, und dieser ist insbesondere bei einem kooperativen Prozess wichtig, ist eine Mischform der beiden Ansätze (*middle out approach*). Dieser Ansatz kombiniert die oben genannten Entwurfsmethoden indem eine Annäherung zu den sowohl höheren als auch tieferen Ebenen in der Hierarchie – durch eine Einsetzung dieser Ebenen – zu einem späteren Zeitpunkt erfolgt. *Uschold und Grunniger* messen diesem Ansatz die bessere Methodik bei (vgl. *Uschold und Grunniger, 1996*). Hinter dem *middle out approach* steht die Methode, mit den wichtigsten Begriffen oder Modellen der betrachteten Domäne den Ontologieaufbau zu beginnen. Der Fokus in diesem Ansatz verlagert den Auf-

bau von generellen und spezielleren Modellen in den weiteren Verlauf des Strukturierungsprozesses bzw. setzt auf ein schrittweises Modellieren des umgebenden Kontextes. Die Wahl der Modellierung von unten nach oben (*top-down*) hingegen kann zu einer sehr tiefen Struktur von aufeinander aufbauenden Modellen führen, da der jeweilige Spezialisierungsgrad für das nächstfolgende Modell unter Umständen sehr klein sein kann. Dies begünstigt eher eine Teilung der Konzepte bzw. Modelle, als eine Fokussierung auf die semantischen Verbindungen. Die Herangehensweise von unten nach oben (*bottom-up*) indes birgt die Gefahr, einen zu hohen Grad an Details (in Form von Eigenschaften) zu einem sehr frühen Stadium in die Wissensstruktur einzufügen, was zu einem erhöhten Bedarf nachträglicher Bearbeitung führen kann (vgl. Uschold und Gruninger, 1996).

Je nach persönlicher Sicht auf ein bestimmtes Wissensfeld kann also ein anderer Ansatz zum Aufbau der Wissensstruktur gewählt werden. Um jedoch den ersten Schritt beim Aufbau einer Ontologie zu erleichtern sollten in einer neu angelegten Ontologie bereits bestimmte Elemente integriert sein. So ist es sicherlich von Vorteil, gerade im Hinblick auf eine visuell gestützte Editierung, ein abstraktes Obermodell (*Thing*) zu definieren, das als Anknüpfungspunkt für weitere Spezialisierungen dient. Ferner ist es auch denkbar für verschiedene Einsatzkontexte (z.B. Strukturierung von Dokumenten) bereits detaillierte Vorlagen anzubieten.

3.1.6 Konsens

Eine Erarbeitung von Wissensstrukturen in einer kollaborativ gestützten Herangehensweise bedeutet auch immer, verschiedene Charaktere mit unterschiedlichen Wissensspektren an einem Tisch zu versammeln. Da die individuelle Sicht bzw. das persönliche Verständnis eines Wissensfeldes in Bezug auf Semantik (Sinn) und Syntax (Vokabular) (vgl. Studer u. a.) mitunter stark divergiert, stellt dies primär eine der Hauptschwierigkeiten in der Erarbeitung einer gemeinsamen Wissensbasis dar. Kommunikation nimmt somit eine Schlüsselstellung ein, um den Austausch über ein gewisses Gebiet zu ermöglichen. Luhmann (vgl. Degele, 2000) definiert den Wissenserwerb aus einer Synthese der drei Bausteine: Information, Mitteilung und Verstehen. Diese Kernmerkmale der Kommunikation geben ebenfalls Aufschluss über die Notwendigkeit eines Prozesses, der das gegenseitige Verstehen unterstützt.

Ein Konsens bezüglich der semantischen Zusammenhänge eines Wissensfeldes und der Findung einer gemeinsamen Sprache bzw. eines gemeinsamen Vokabulars, sei es nun interdisziplinär oder fachintern, ist essenziell für die Etablierung bzw. Initialisierung eines gemeinsamen Aufbaus von Wissen als Struktur (vgl. Aschoff u. a.). Mechanismen für einen detaillierten Diskurs über die betrachtete Wissensdomäne ermöglichen zum einen eine stetige Revision des Kontextes bzw. Wahrheitsgehaltes und reduzieren damit die Gefahr von gegenseitigen Missverständnissen in Bezug auf Vokabular auf der einen und semantischen Vorstellungen auf der anderen Seite (vgl. Studer u. a.). Auch im Hinblick auf eine spätere Adaption der Ontologie und somit von Wissen ist ein konsensbasierter Aufbau von Vorteil.

3 Kollaborative Ontologien zur Strukturierung von Wissen

Mit der Ambivalenz von *Autor und Nutzer Rolle* bzw. der Loslösung von einer *Experte zu Laien Relation*, ergibt sich die Situation eines »kopflosen« und somit »Runden Tisches«, der die Findung eines Konsenses aufgrund der flachen Hierarchie der Beteiligten erleichtern kann. Die mehrheitliche Übereinstimmung innerhalb einer Gruppe bezüglich eines semantischen Zusammenhanges erleichtert es nachfolgenden bzw. darauf aufbauenden Wissensprozessen zum einen diesen Konsens mitzutragen, und zum anderen (vgl. Kapitel 3.1.3) die Validität bzw. Richtigkeit der Wissenszusammenhänge anzuerkennen. Das Anerkennen der Richtigkeit bezieht sich jedoch nicht zwingend auf die fachlich korrekte Darlegung der Zusammenhänge sondern vielmehr auf das gruppeninterne Verständnis von Richtigkeit in Bezug auf den erstellten Wissensraum. Dies belegt auch eine Untersuchung mit 55 Probanden an der University of South Carolina (vgl. Stephens und Huhns, 2001). In diesem Experiment sollten einzelne Studenten die Wissensdomäne *Mensch* klassifizieren und in eine ontologische Form bringen. Die Auswertung zeigte, dass die Probanden zum Teil sehr differenziert aufbauten und fachlich nicht korrekte Beziehungen, wie z.B. die Einordnung von *Tier* als Spezialisierung von *Säugetier*, des Öfteren konstruiert wurden. Dieses Ergebnis unterstreicht zum einen den Ansatz der kooperativen Erstellung von Ontologien, da es sich hier um 55 einzelne Ontologien handelte. Zum anderen zeigt es aber auch, dass trotz dieses Ansatzes die Möglichkeit besteht, fachlich falsche aber gruppenintern richtige Zusammenhänge aufzubauen. Dies muss aber nicht zwangsläufig ein Nachteil sein. Zum einen ist die fachliche Korrektheit immer auch in Zusammenhang mit der – im universitären Sinne – jeweiligen Disziplin zu sehen. Was für die Einen fachlich korrekt erscheint, mag für die Anderen schlicht falsch sein. Genau hier ist aber auch der Vorteil einer Konsens-basierten Ontologie zu sehen, verschiedene (interdisziplinäre) Gruppen können mit Hilfe einer Ontologie die jeweiligen Sichtweisen besser darlegen, so das Richtig und Falsch im Sinne von Standpunktverschiebungen neu bewertet werden können.

Auf der anderen Seite ist es in Lehr- Lernprozessen natürlich von außerordentlicher Wichtigkeit fachlich korrekte Zusammenhänge wiederzugeben. Aber auch hier sollten eher die Chancen als die Nachteile gesehen werden. Auf Seiten des Lehrenden bietet sich somit die Möglichkeit zum einen Defizite in Bezug auf das gruppeninterne Verständnis zu erkennen, und zum anderen, aufgrund der diskursiven Konsensbildung, auch den Weg zu diesem (Miss-) Verständnis nachvollziehen zu können. Auf Seiten der Lernenden hat dies den Vorteil, dass in einer begleitenden Lehrübung speziell auf die Probleme von Gruppen bzw. der Lernenden in ihrer Gesamtheit eingegangen werden kann.

Welche Funktionalität kann nun aber bei der Findung eines Konsenses hilfreich sein? Bei einer Betrachtung des ontologischen Modells fällt auf, dass jegliche Entität, sei es nun Modell, Individuum, Assoziation, Eigenschaft oder Hierarchie Raum für Diskussionen eröffnet. Um also die Findung eines Konsenses nicht auf die Ontologie als Ganzes zu beschränken, muss jegliche Entität und Beziehung die Möglichkeit bieten, Fragen, Kritik und Unverständnis anzufügen um schließlich einen Gesamtkonsens der semantischen Bezüge zu finden. Für die Unterstützung der

3 Kollaborative Ontologien zur Strukturierung von Wissen

Konsensfindung sollte also ein elementarierter und im Bezug auf die netzgestützte Editierung, asynchroner Annotations- bzw. Diskussionsmechanismus zum Einsatz kommen.

Ähnlich wie die seitenbasierte Diskussion in Wikisystemen¹ sollten die Annotationen immer im Kontext der jeweils betrachteten Entität und Assoziation aufrufbar sein. Des Weiteren sollten die Annotationen von jedem Teilnehmer in ihrer Gesamtheit editierbar sein, Auskunft über Datum und Nutzer geben und andere Prozessbeteiligte über Änderungen bzw. Neuerungen informieren. Darüber hinaus sind weitere Informationen in Form von Kategorien, der Priorität eines Kommentars, oder ob eine Annotation sich positiv oder negativ in Bezug auf die Entität äußert, denkbar (vgl. Weng und Gennari, 2004).

Als Erweiterung dieser asynchronen Konsensfindung können auch externe, synchron gestützte Kommunikationsmechanismen wie z.B. geteilte Anwendungen (so genannte *shared whiteboards*²) oder Diskussionssysteme (Chat³) zum Einsatz kommen. Der synchrone Austausch bietet sich an, um auf der einen Seite die Findung eines gemeinsamen Standpunktes zu beschleunigen, und auf der anderen Seite aufgrund der tatsächlichen Gesprächssituation die gegenseitigen Sichtweisen detaillierter darlegen zu können. Der Einsatz von synchronen Kommunikationsmitteln dient hierbei aber eher als Erweiterung der asynchronen Formen, da die gewonnenen Einsichten aufgrund der Flüchtigkeit nicht bzw. nur erschwert der Gruppe in Ihrer Gesamtheit zugänglich gemacht werden können. Protokolle der synchronen Diskussion, grafische Zusammenfassungen (Schnappschüsse) aus einer geteilten Anwendung oder das speziell in der *open'sTeam*-Umgebung eingesetzte *shared whiteboard* – das sowohl synchrone als auch asynchrone Zusammenarbeit unterstützt (vgl. Hampel u. a., 2004) – können dieser Flüchtigkeit entgegenwirken.

3.1.7 Nachvollziehbarkeit und Erweiterung

Die Beschreibung *Wissen auf Wissen* anzuwenden, oder dies als rekursiven Prozess anzusehen (vgl. Degele, 2000), bedeutet zum einen, über die Möglichkeit einer Rückwärtsbewegung hin zu den Basiskonzepten einer Wissensstruktur zu verfügen, und zum anderen die Unendlichkeit einer solchen Anwendung zu berücksichtigen. Insbesondere in einer teamgestützten Wissensproduktion ist es unerlässlich, Autoren und Nutzern geeignete Strukturen anzubieten, die ein Nachvollziehen des Wissensprozesses unterstützen.

Ontologien verfügen aufgrund der Klassifizierung von Begriffen und Konzepten über eine interne Struktur, die das Verständnis der betrachteten Wissensdomäne zum Teil autonom verdeutlichen kann. Aber gerade in der Situation einer mehrköpfigen Herangehensweise ist es unbedingt erforderlich, nicht nur den Kontext

¹Wikis (hawaiianisch für *schnell*) sind verknüpfbare Seitensammlungen im WWW die direkt gelesen und editiert werden können.

² Gemeinsam Sicht auf eine *elektronische Wandtafel*, die von allen Nutzern manipulierbar ist.

³Der Chat bzw. das chatten (engl. für *plaudern*) dient zum synchronen Austausch von (Text)Nachrichten mehrerer Personen.

und Zusammenhang einer derzeitigen Wissensstruktur nachzuvollziehen, sondern auch den Weg wie ein spezieller Kontext gefunden wurde und welche Konsense der Autoren zu dessen Findung nötig waren. Nicht nur im Bezug auf das gegenseitige Verständnis der direkt beteiligten Autoren, sondern auch für Nutzer, die sich zu einem späteren Zeitpunkt diesem Prozess anschließen ist die Möglichkeit eines »Abschreiten des Weges« zu jedem Zeitpunkt von außerordentlicher Wichtigkeit.

Um dies zu gewährleisten muss jeder Schritt in der evolvierenden Struktur in Relation zu dem oben genannten Annotationsmechanismus aufgezeichnet werden. Dies dient neben der Nachvollziehbarkeit der Grundkonzeption und des Konsenses noch weiteren Strategien. Die Möglichkeit den Fortschritt der Wissensstruktur detailliert zu betrachten, kann – um bei der *Weg-Metapher* zu bleiben – auch zur Gabelung bzw. Abspaltung dieser Struktur nach einer bestimmten Schrittlänge benutzt werden. Basierend auf dem bereits vorhandenen Konstrukt, kann so ein thematisch differenzierter Aufbau erfolgen, ohne jedoch die Grundkonzeption zu verlieren. Neben dieser Verzweigung dient das Zurückspringen zu einem beliebigen Zustand der Ontologie auch der Zurücknahme einer bereits definierten Struktur, um somit ein Anknüpfen an eine bereits vorhandene Struktur.

Die im Kapitel 3.1.2 beschriebene Medienfunktion Archivierung bzw. Versionsverwaltung kann sowohl auf die Ontologie als Ganzes, als auch auf deren Entitäten abzielen. Die Archivierung der ganzen Ontologie nach der Ausführung einer individuellen Medienfunktion wie z.B. Löschen oder Erzeugen, bedeutet zwar einerseits eine redundante Speicherung der Struktur, auf der anderen Seite ist aber für jede erfolgte Änderung eine kohärente und vollständige Ontologie jederzeit verfügbar. Der Ansatz nur diejenige Entität zu archivieren, die auch einer tatsächlichen Änderung unterzogen wurde vermeidet diese Redundanz, ist aber als einzelnes Bedeutungsatom in der semantischen Struktur nicht transportierbar und kompliziert aufgrund der unterschiedlichen Versionen einzelner Entitäten den Aufbau von Vorgängerversionen. Dieser versionsgestützte Ansatz – sei er nun Entitäts- oder Ontologiebezogen – ermöglicht einen flexiblen und reintegrativen Umgang mit der ontologischen Struktur und ermöglicht es dem Autor bzw. Nutzer zum einen in einer fehlertoleranten und reversiblen Umgebung zu arbeiten und zum anderen die Nachvollziehbarkeit des Wissens für Dritte zu erleichtern.

3.1.8 Gegenseitige Wahrnehmung

In Bezug auf eine gemeinschaftliche Konsensfindung und den Überblick über die evolvierende Wissensstruktur ist es wichtig, die Prozessbeteiligten auch in spezieller Art und Weise wahrzunehmen. Diese gegenseitige Wahrnehmung (im englischen auch als *Awareness* bezeichnet) unterstützt den einzelnen Nutzer in dem Verständnis nicht alleine, sondern in einer Gruppe an einer Ontologie bzw. Struktur zu arbeiten. *Awareness* bzw. Wahrnehmung in Bezug auf ein kooperatives System kann verschiedene Ausprägungen annehmen und soll verschiedene Funktionen erfüllen. Diese Funktionen reichen von der reinen Darstellung gegenwärtiger Kooperationspartner (*Group Awareness*) bis hin zu einer Wahrnehmung über

3 Kollaborative Ontologien zur Strukturierung von Wissen

die derzeitigen Handlungen dieser Partner (Workspace Awareness) (vgl. Liechti, 2000). Die Wahrnehmung gegenseitiger Handlungsweisen kann sowohl synchron als auch asynchron erfolgen und ist für einen kooperativen Aufbau einer Ontologie geradezu essentiell. Awareness kann einerseits durch den oben beschriebenen Annotationsmechanismus erfüllt werden, andererseits muss aber die unterstützende Applikation die Nutzer in den Gesamtkontext der Ontologie rücken. Um dem Nutzer zu verdeutlichen das ein anderer die Ontologie mitgestaltet muss zunächst der Frage nachgegangen werden, in welcher Form die Existenz der Mitwirkenden angedeutet wird. Wichtig im Kontext des kollaborativen Ontologie-Aufbaus ist es einen Überblick über die veränderte Struktur anzubieten. Es stellt sich jedoch die Frage welche Informationen wichtig bzw. unwichtig für diesen Überblick sind. Eine Annäherung zum separieren der nötigen Informationen kann die Beantwortung der folgenden Fragestellung erbringen: »*Wer hat Was mit Wem Wann gemacht*«. Im Folgenden wird nun diese Frage in Bezug auf die benötigten Informationen erörtert.

Zunächst gibt das *Wer* Aufschluss darüber, welcher Nutzer eine bestimmte Aktion in der Ontologie ausgelöst bzw. initiiert hat. Neben der namentlichen Nennung des Nutzers sollten weitere Informationen wie Kontaktmöglichkeiten und Engagement in der Ontologie bereitgestellt werden. Letzteres kann beispielsweise durch das Gruppieren der einzelnen Aktionen und Annotationen eines einzelnen Nutzers geschehen, wobei sich auch ein Profil über die Interessen im Kontext der Ontologie erkennen lassen könnte. Die Frage *Was* der jeweilige Nutzer in der Ontologie editiert hat ist im Bezug zu den möglichen (individuellen) Medienfunktionen zu beantworten. So kann z.B. angedeutet werden das ein bestimmter Nutzer ein neues Modell angelegt hat oder ein Individuum mit einer Annotation versehen hat. Dies leitet auch über zu der Beantwortung des *Wem*. Wem in diesem Fall bezieht sich auf die konkret editierte Entität oder Beziehung und sollte ebenfalls neben einer namentlichen Nennung wie z.B. Vorlesung weitere Interaktionsmöglichkeiten aufweisen. So ist denkbar, dass das editierte Modell direkt angesprungen werden kann oder weitere Bezüge wie z.B. *ist abgeleitet von* oder andere involvierte Nutzer aufgezeigt werden.

Da der Prozess des Ontologie-Aufbaus sowohl synchron als auch asynchron erfolgen kann, ist der Zeitbezug bei der Beantwortung des *Wann* von besonderer Wichtigkeit. In beiden Situationen lässt sich so die Aktion einerseits ihrer Aktualität zuordnen und andererseits wird nachvollziehbar, über welchen Zeitraum sich der Strukturierungsprozess erstreckt. Um dies zu gewährleisten muss der kooperative Prozess neben der oben erwähnten Möglichkeit zur Speicherung einzelner Ontologie-Versionen auch die vom Nutzer initiierten Aktionen in Bezug zu den einzelnen Versionen setzen und fortlaufend aufgezeichnet werden. Diese Aufzeichnung muss dem Nutzer zum einen während des Explorierens zugänglich sein oder kann zum anderen zusätzlich über andere Kanäle wie z.B. RSS-Newsfeeds⁴ greif-

⁴RSS (Rich Site Summary) ist ein XML-Format und kann zur Darstellung von Nachrichten in speziellen Betrachtungsprogrammen angeboten

bar werden.

3.2 Grafische Strukturierung

Für eine kooperative Erstellung einer Wissensstruktur ist es notwendig eine gemeinsame Sicht der beteiligten Personen auf eben diese Struktur bereitzustellen. Insbesondere eine Ontologie benötigt eine grafische Darstellung, die es ermöglicht sowohl die komplexe interne Struktur als auch die konstruierten Wissenszusammenhänge für die beteiligten Personen zu verdeutlichen. In diesem Kapitel wird daher die interne Struktur einer Ontologie analysiert und verschiedene Möglichkeiten für eine Visualisierung aufgezeigt. Anzumerken ist an dieser Stelle, dass die hier erläuterten Visualisierungs-Merkmale eher genereller Natur sind und erst mit der Verschmelzung der in Kapitel 3.1 aufgestellten Eigenschaften eine kooperative Erstellung von Ontologien ermöglichen.

In einer ganzen Reihe von wissenschaftlichen Arbeiten der letzten Jahre wurde nach Möglichkeiten gesucht, Wissen kooperativ zu gestalten und zu strukturieren. Wikis, Groupware Systeme und kooperative Systeme, wie ^{open}sTeam zur Strukturierung von Wissen zeigen, welche unterschiedlichen Formen kooperativer Zusammenarbeit existieren und welches Potenzial diese netzgestützte Zusammenarbeit bietet. Im Folgenden wird, mit Bedacht auf neue, grafisch-semantische Formen der Wissensstrukturierung in Gruppen, ein kurzer Überblick über derzeitige Strukturierungsmöglichkeiten gegeben. Hierbei muss sowohl unterschieden werden zwischen automatisch generierten oder von Nutzern erstellten Strukturen, als auch zwischen freien bzw. offenen oder vorgegebenen Strukturen zur Repräsentation eines Wissensraumes (vgl. Rubart und Hampel, 2003). Neben einer individuellen Strukturierung von Wissensräumen mit Hilfe von Ordnern und Verweisstrukturen, oder dem Generieren und Weitergeben von zurückgelegten Pfaden in einem Netzwerk wie z.B. dem WWW, kommen auch in Kooperation erstellte Semantische Karten – die es z.B. zulassen Zusammenhänge räumlich abzubilden – als eine freie und offene Strukturierung eines Wissensraumes in Betracht. Dem gegenüber stehen vorgegebene Strukturen wie z.B. automatisch generierte Karten, die einen entsprechenden Systemzustand darstellen, oder Repräsentationen, die von einem einzelnen Autor für eine Vielzahl von Nutzern im Vorhinein strukturiert wurden. Für eine weitere Betrachtung möglicher Strukturierungsformen sei an dieser Stelle auf den Artikel »*Semantische Räume – Von der Navigation zur kooperativen Wissensstrukturierung*« (vgl. Hampel u. a., 2004) verwiesen. Abbildung 3.1 zeigt eine Taxonomie netzgestützter Methoden der semantisch-grafischen Strukturierung und gibt darüber Auskunft in welchen Zusammenhängen (z.B. automatisch oder kooperativ erzeugt) grafische Strukturen entstehen können.

Da in dieser Arbeit generell von einer kooperativen Erstellung einer Ontologie und somit auch der grafischen Struktur ausgegangen wird, soll an dieser Stelle vielmehr die Ontologie anhand der Möglichkeiten einer offenen bzw. vorgegebenen Strukturierung untersucht werden. Da die Ontologie in ihrer Form nicht als

3 Kollaborative Ontologien zur Strukturierung von Wissen

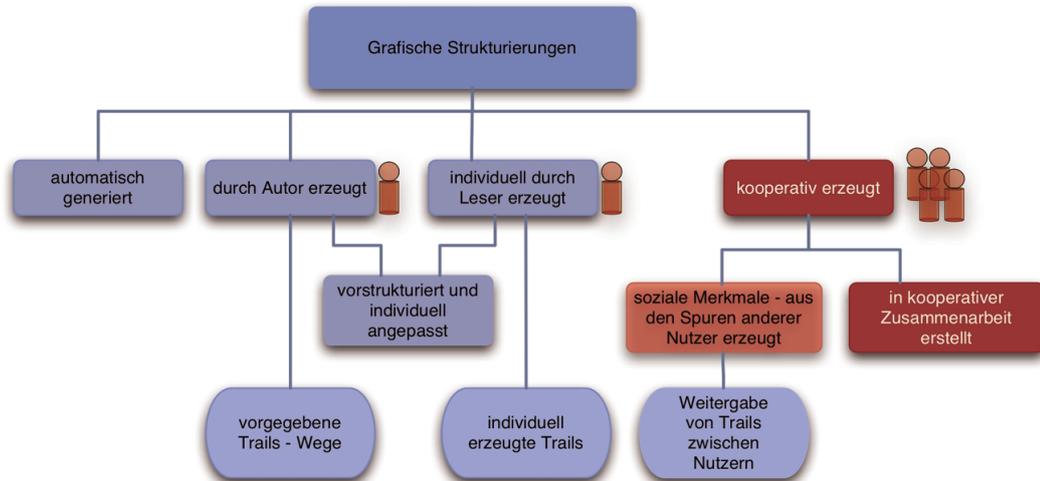


Abbildung 3.1: Formen grafischer Strukturierung (Quelle: Hampel u. a., 2004)

grafisches Konzept, sondern als Rahmenwerk für die Speicherung von Sinnzusammenhängen konzeptioniert ist, stellt dies zunächst eine offene Herangehensweise in der Gestaltung einer grafischen Repräsentation dar. Die freie Gestaltung und Strukturierung wird jedoch durch die interne Struktur einer Ontologie begrenzt. Vielmehr präsentiert sich die Ontologie als ein Teils offenes und Teils vorgegebenes Hilfsmittel zur Strukturierung eines Wissensfeldes. Abbildung 3.2 verdeutlicht dies für die einzelnen Ontologie-Elemente anhand von umschließenden Freiräumen. Während sich das Anlegen neuer Modelle als freies Arrangieren abstrakter Wissenszusammenhänge ausbildet, ist dies gleichzeitig der Schritt hin zu einer fest vorgegebenen Struktur. Diese äußert sich bereits in dem Anlegen neuer Eigenschaften und Assoziationen, wobei z.B. letztere nur auf Basis von mindestens zwei vorhandenen Modellen gebildet werden können. Das Anlegen der so genannten Individuen bildet innerhalb der Ontologie das am weitesten reglementierte Element. Die Vorgaben von Modell, Eigenschaft und Assoziation bilden hierbei den vorab strukturierten Rahmen, wobei die Individuen durch das Einsetzen von Werten in ihrer Ausprägung unterscheid- bzw. abgrenzbar werden. Dieses Vorgeben von Assoziationen und Eigenschaften ist auch im Hinblick auf eine mögliche spätere maschinelle Erschließung notwendig. Wäre das Anlegen von neuen Individuen nicht an die vom Modell vorgegebene Struktur gebunden, könnte ein maschineller Einsatz auf Grundlage der ontologischen Struktur nicht stattfinden. Diese Einschränkungen wirken sich wechselseitig auf die grafische Repräsentation – z.B. ist ein Individuum ohne zugehöriges Modell nicht denkbar – und die Strukturierung eines Wissensfeldes aus.

Wie aber kann das Konzept der Ontologien nun als grafisches Hilfsmittel zur Repräsentation von Wissen genutzt werden? Der hier vorgestellte Ansatz soll eine strukturelle Herangehensweise beschreiben, wie das Konzept der Ontologie in ei-

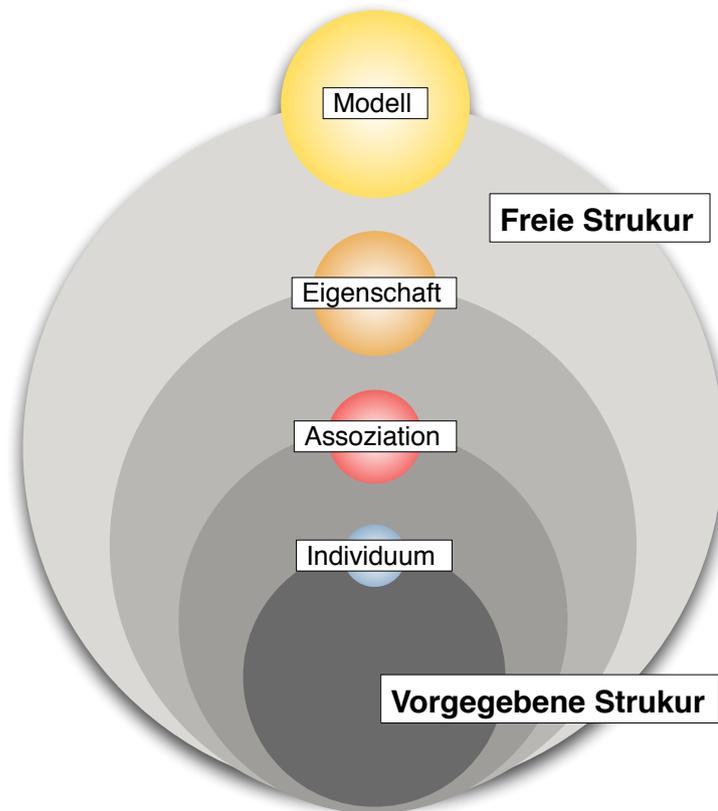


Abbildung 3.2: Struktur - Beschränkungen einer Ontologie

ne grafische Struktur überführt werden kann. Aufgrund des komplexen Aufbaus einer Ontologie ist es von besonderer Wichtigkeit, einen intuitiven Zugang zum ontologischen Konzept auf der einen Seite, und auf der anderen Seite eine übersichtliche Navigation und kontextbezogene Editierfunktionen für diese Struktur bereitzustellen.

Basierend auf den in Kapitel 2.2 beschriebenen Facetten interner Verknüpfungen muss eine mögliche Visualisierung zum einen das Aufbaukonzept einer Ontologie vermitteln können, zum anderen aber auch die Komplexität insoweit vereinfachen, dass zukünftige Nutzer nicht mit der enormen Fülle möglicher Funktionen überfordert sind.

3.2.1 Trennung von Exploration und Editierung

Ein erster Ansatz, der dies unterbinden kann, besteht in der Trennung von Exploration bzw. Navigation und Editierumgebung. Diese Trennung sollte jedoch nicht so strikt ausfallen wie z.B. in den Ontologie-Editoren OntoEdit© (vgl. Sure u. a., 2002) oder Protégé (vgl. Noy u. a., 2001). In diesen Anwendungen werden Editierung

3 Kollaborative Ontologien zur Strukturierung von Wissen

und Visualisierung der Ontologie getrennt behandelt (vgl. Abbildung 3.3), wobei die visuelle Repräsentation keine Möglichkeiten für eine direkte Bearbeitung der Ontologie bietet. Um dies jedoch zu ermöglichen bedarf es einer Kombination von Visualisierung und Bearbeitung. Um trotzdem die Übersicht zu wahren ist es vorstellbar, die Bearbeitungsfunktionen in der Visualisierung in den Hintergrund zu stellen und erst nach einem expliziten Aufruf durch den Nutzer in Erscheinung treten zu lassen. Dies bietet zum einen den Vorteil aufgrund der verminderten Komplexität eine schnellere Einarbeitung in das Konzept der Ontologien bzw. in das grafische Werkzeug zu gewährleisten, und zum anderen die grafische Darstellung auch als reines Instrument der Informationssuche zu nutzen. Das Editieren der On-

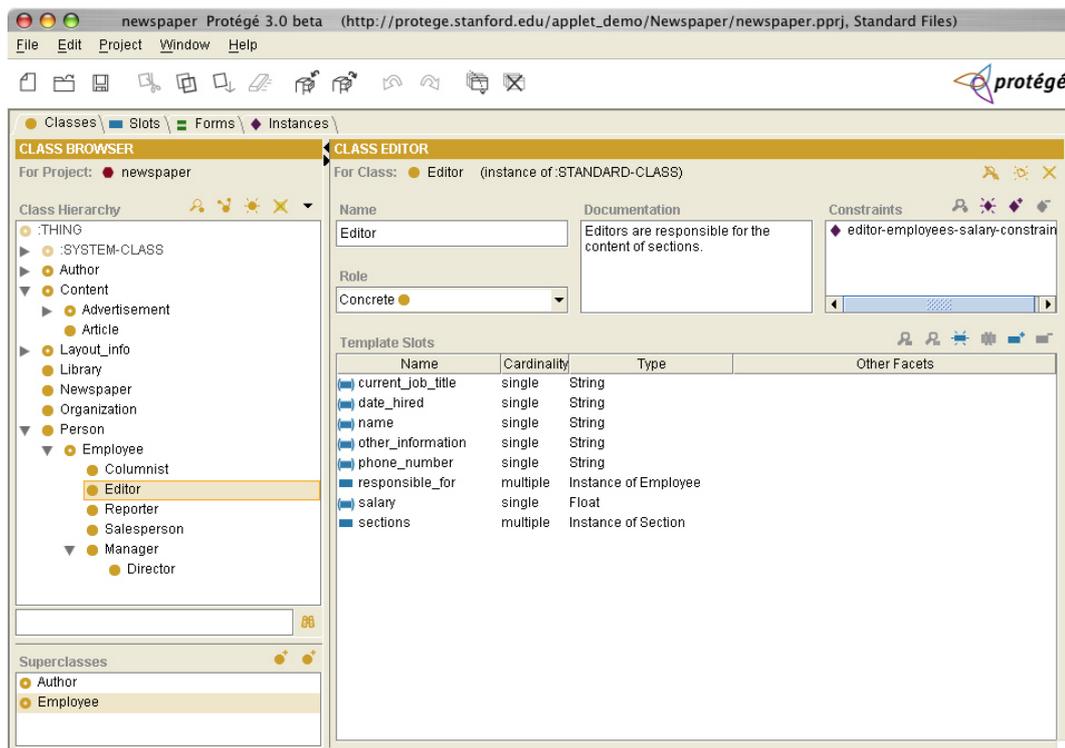


Abbildung 3.3: Bildschirmfoto von Protégé – Ansicht auf die Hierarchie einer Ontologie

tologie kann in dieser Form auf zwei verschiedenen Ebenen stattfinden. Diese beiden Ebenen unterscheiden sich jeweils in der Ausprägung der direkten Manipulation der grafischen Symbolik. Hierbei gestaltet sich das direkte Editieren als ein Anfügen und Ändern von Attributen (z.B. Assoziation oder Eigenschaft) mit einer Mehrzahl von grafischen Mitteln und einer geringen Anzahl an textbasierten Editierfunktionen. Als vorteilhaft in diesem Fall gestaltet sich die Wahrnehmung des Nutzers, die Bearbeitung als direkte Manipulation eines grafischen Objekts zu interpretieren. Aufgrund der Fülle möglicher Funktionen und Elemente in einer

Ontologie⁵ muss für eine direkte Manipulation auch jeweils eine entsprechende Funktion bzw. ein grafisches Symbol zur Verfügung stehen. Dies kann zu einer gesteigerten Komplexität der gesamten Anwendung bzw. Darstellung führen. Eine Bündelung der möglichen Funktionen erscheint hierbei sinnvoll. Dies leitet über zu der zweiten Ebene des verdeckten Editierens. Dem ersten Ansatz ähnelnd werden in diesem Ansatz Funktionsgruppen gebildet, die zusammenhängende Aktionen (z.B. Löschen, Anfügen oder Ändern einer Assoziation) beschreiben und ebenfalls symbolisiert angeboten werden. Die einzelnen Funktionen können nach einer Aktivierung der jeweiligen Funktionsgruppe durch den Nutzer über geleitete Dialoge textbasiert ausgeführt werden. Dies hat den Vorteil eine halbgrafische Bearbeitung zur Verfügung zu stellen, und dabei die Anzahl möglicher Funktionen durch das Gruppieren zu verringern.

3.2.2 Darstellung der Modellstruktur

Auf Seiten der Navigation und Darstellung der Ontologie sollte, basierend auf der Shneiderman Formel zur Informationssuche »*Overview first, zoom and filter, then details on demand*« (vgl. Shneiderman, 1991) die Visualisierung zum einen aufeinander aufbauen und mit Bedacht auf die engen Verknüpfung innerhalb der Ontologie bzw. deren Entitäten auch einen visuellen Kontext bilden. Zur Analyse dieses Kontextes erscheint ein weiteres Aufbrechen der ontologischen Struktur nötig. Basierend auf der in Kapitel 2.2 beschriebenen hierarchischen Struktur, werden in diesem Abschnitt zwei Ansätze herausgearbeitet, die auf verschiedene Intentionen bzw. Sichtweisen in Bezug auf eine mögliche Darstellung der Modellstruktur einer Ontologie eingehen. Eine mögliche Visualisierung muss zum einen die gegenseitigen Abhängigkeiten der Modelle in Bezug auf Ableitung bzw. Vererbung darstellen, und zum anderen die Assoziationen als Verbindungen zwischen diesen Modellen unterstreichen. Der Verbund dieser beiden Anforderungen stellt sich aufgrund der hierarchischen Abhängigkeit zunächst als – im Sinne der Graphentheorie – Baumstruktur mit einer definierten (abstrakten) Wurzel dar. Durch das Anfügen der Assoziationen wird dieser Baum in eine Netzform transformiert.

Basierend auf diesen Eigenschaften lassen sich nun zwei grundlegende Ansätze der Repräsentation unterscheiden. In diesem Zusammenhang ist zu bemerken, dass diese Ansätze jeweils von einer einfachen Vererbung bzw. Spezialisierung der Modelle ausgehen. Die Möglichkeit der Mehrfachvererbung wird aufgrund der erhöhten Komplexität der Visualisierung nicht weiter untersucht.

Eine Fokussierung auf die Baumstruktur kann als vertikaler Ansatz betrachtet werden. In diesem Ansatz legt die Visualisierung den Akzent auf die hierarchischen Abhängigkeiten der Struktur und unterstreicht so die Vererbungslinie bzw. Spezialisierung innerhalb der Ontologie. Bei dieser Herangehensweise sollten zwei Dinge in Betracht gezogen werden. Zum einen muss eine einmal definierte Richtung (von oben nach unten, links nach rechts etc.) des Baumes beständig beibehal-

⁵Dies sind allein für das Anfügen, Löschen und Editieren aller Elemente 12 einzelne Funktionen

ten werden, und zum anderen stellt sich die Schwierigkeit, dass der Baum in Breite (Anzahl der Knoten auf einer Ebene) und Tiefe (Anzahl der Ebenen) stark variieren kann. Eine andere Alternative bildet der horizontale Ansatz. Hierbei wird der Fokus weniger auf die Vererbungslinien gerichtet, sondern die Assoziationen werden in den Vordergrund gestellt. Aus diesem Ansatz ergibt sich die Notwendigkeit die Gesamtheit aller Knoten in einer Übersicht zu präsentieren. Eine vollständige Darstellung beider Ansätze kann aufgrund der beschränkten Darstellungsfläche auf einem Bildschirm nicht garantiert werden. Im Folgenden wird daher versucht sich dem Ziel einer vollständigen Repräsentation anzunähern.

Vertikale Ausrichtung der Modellstruktur

Ein möglicher Ansatz zur Unterstreichung der vertikalen bzw. der »ist ein« Beziehung ist die Darstellung eines hierarchischen Ausschnitts in einer *drei Ebenen Anordnung* (vgl. Abbildung: 3.4). Diese Anordnung zeigt ein aktuell selektiertes

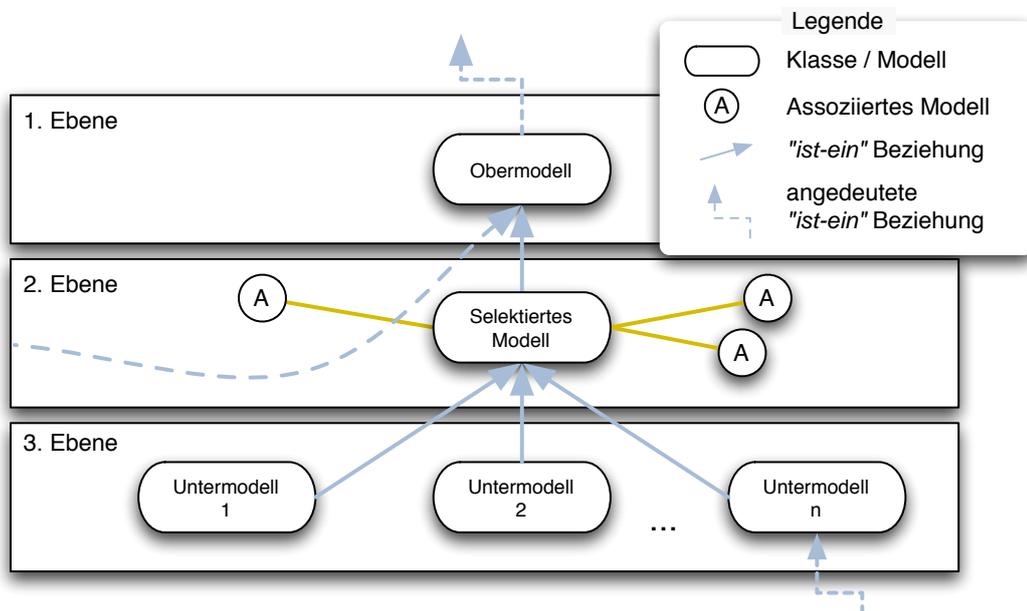


Abbildung 3.4: Hierarchieausschnitt in drei Ebenen

Modell in seiner Beziehung zum direkten Obermodell und zu den direkten Folge- bzw. Kindmodellen. Da die unterschiedlichen Positionen der Assoziationen innerhalb der Hierarchie mitunter stark divergieren und der dargestellte Ausschnitt der Hierarchie sehr klein ist, werden assoziierte Modelle in dieser Darstellung nicht direkt sichtbar sein, was eine weitere Modifikation der Ansicht nach sich zieht. In der Ebenendarstellung werden die assoziierten Modelle nicht vollständig in ihrem Gesamtkontext präsentiert, sondern nur als eine anwählbare Referenz angedeutet. Die Anzeige der Assoziationen kann bzw. sollte nur auf das aktuell selektierte Modell

3 Kollaborative Ontologien zur Strukturierung von Wissen

beschränkt werden, um eine bessere Übersicht zu garantieren. Eine Verknüpfung und Anzeige der Ober- und Folgemodelle mit den jeweiligen Assoziationen würde aufgrund der Vererbung Redundanzen aufweisen und die Sicht auf die derzeit relevanten Zuordnungen erschweren.

Dieser kleine, auch *local map* (vgl. Utting und Yaneklovich, 1989) genannte, Ausschnitt zieht allerdings eine Erweiterung der Navigation mit sich, um zum einen Sprünge innerhalb der Hierarchie zu beschleunigen und zum anderen die Orientierung innerhalb der Ontologie zu verbessern. Insbesondere die Frage der Orientierung sollte Fragen wie *Wie kam ich hierher* oder *Wohin kann ich von hier aus gehen* beantworten (vgl. Utting und Yaneklovich, 1989). Für diese Erweiterung könnten

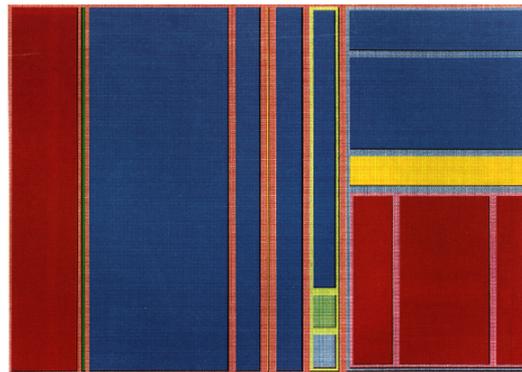


Abbildung 3.5: Hierarchie als Treemap (Quelle: Shneiderman, 1992)

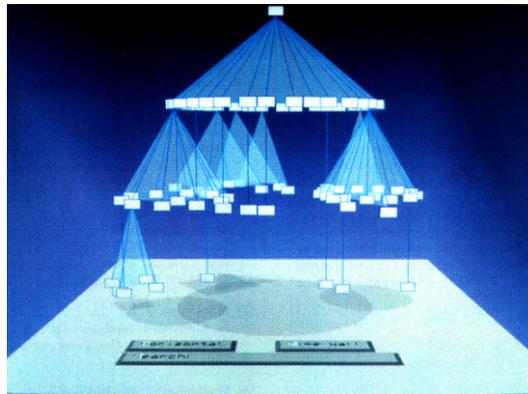


Abbildung 3.6: Hierarchie in 3D Darstellung mit Hilfe von Cone Trees (Quelle: Robertson u. a., 1991)

neben anderen der Treemap-Ansatz (vgl. Shneiderman, 1992), die bekannten Ordnerstrukturen aus Dateisystemen (z.B. Windows Explorer) oder die Visualisierung mit der dreidimensionalen Darstellung in Form eines so genannten Cone Tree (vgl. Robertson u. a., 1991) in Betracht gezogen werden.

3 Kollaborative Ontologien zur Strukturierung von Wissen

Es gilt jedoch die einzelnen Visualisierungs- bzw. Navigationsansätze zu unterscheiden. Die Darstellung als Treemap (vgl. Abbildung 3.5) bietet den Vorteil, eine Hierarchie zu jeder Zeit in ihrer gesamten Tiefe und Breite platzsparend darzustellen. Die Sicht auf diese Struktur stellt, anders als die folgenden Visualisierungen, die einzelnen Hierarchieebenen als ineinanderliegende Rechtecke dar, wobei unterschiedliche Farben die einzelnen Tiefen und unterschiedliche Flächengrößen die jeweilige Gewichtung darstellen. Fläche bzw. Gewichtung werden in dieser grafischen Struktur durch eine rekursive Teilung der Baumstruktur bestimmt, wobei die Flächengröße durch die Anzahl der folgenden Hierarchieebenen und/oder das jeweilige Gewicht einer Ebene definiert wird. Das An- bzw. Auswählen eines dieser Rechtecke resultiert in einem neuen Aufbau der Tree Map, hierbei bildet der ausgewählte Knoten das umfassende Rechteck und nachfolgende Knoten (Rechtecke) werden ebenfalls neu dargestellt. Diese ungewohnte Betrachtungsweise – mit einer Vogelperspektive vergleichbar – erfordert von den zukünftigen Nutzern eine gewisse Einarbeitungszeit, die Untersuchungen zufolge zwischen 10-15 Minuten liegt (vgl. Shneiderman, 1998).

Cone Trees als Darstellungsform haben den Vorteil, bis zu einer gewissen Tiefe die Hierarchie ebenfalls in einer sehr kompakten Form repräsentieren zu können und dabei die einzelnen Ebenen voneinander abzusetzen. Robertson, Mackinlay und Card begrenzen die mögliche Darstellung auf bis zu 1000 Knoten und 10 Ebenen in der Hierarchie (vgl. Robertson u. a., 1991). Aufgrund der dreidimensionalen Darstellung muss beim Einsatz dieser Technik, neben einem erhöhten Rechenaufwand, auch die Möglichkeit von überlagerten Knoten bedacht werden. Wie die Abbildung 3.6 zeigt, ist aufgrund der konischen Anordnung der einzelnen Bauebenen eine Überlagerung bei schon wenigen Knoten sichtbar. Die Überlagerung der Knoten und die erschöpfende Anzahl darstellbarer Ebenen bedeuten somit Einbußen für die Gesamtübersicht der ontologischen Hierarchie.

Die hier dargelegten Ansätze bieten allerdings nur zu einem Teil eine Antwort auf die oben gestellten Fragen *Wohin kann ich von hieraus gehen* und *Wie kam ich hierher*. So zeigt die Treemap zwar alle Unterknoten, die von der aktuellen Position erreichbar sind, obere Knoten sind jedoch nicht direkt sichtbar. Daher sollten ergänzende Navigationsstrukturen in Verbindung mit der Treemap angeboten werden. Diese Ergänzung erübrigt sich zwar bei der Wahl von Cone Tree oder der bekannten Baumstrukturen als Navigation, allerdings zeigen diese Ansätze nicht die Gesamtheit aller möglichen Wege von einem Knoten auf, bzw. repräsentieren nur die nächstmögliche Hierarchieebene.

Um dem Nutzer zu zeigen, wie er zu dem derzeitigen Knoten gelangt ist müssten alle Ansätze in sofern erweitert werden, dass sie den zurückgelegten Pfad deutlich kennzeichnen. Dies kann z.B. durch einen parallel dargestellten Pfad angezeigt werden, der die besuchten Knoten explizit darstellt oder in der Form einer Aufzeichnung der besuchten Knoten als Chronik, wie sie in Webbrowsern zu finden ist.

Horizontale Ausrichtung der Modellstruktur

Für eine Unterstreichung der Assoziationen bzw. eine horizontale Ausrichtung müssten die Modellabhängigkeiten in ihrer Gesamtheit als Netzstruktur dargestellt werden. Diese bedeutet im Einzelnen das sowohl die »ist ein« Beziehungen als auch die verbundenen Assoziationen zu den jeweiligen Knoten angezeigt werden. Diese Struktur setzt die dargestellten Assoziationen zwar in einen stärkeren Bezug zum Gesamtkontext und verringert die Notwendigkeit einer differenzierten Navigation, macht aber auf der anderen Seite eine Unterscheidung von Assoziation und »ist ein« Beziehungen – aufgrund der aufgehobenen hierarchischen Ordnung – ungleich schwerer. Eine andere Schwierigkeit birgt die Positionierung der einzelnen Knoten. Für Hierarchien und deren Aufbau bietet es sich zumeist an, diese in einer Leserichtung darzustellen wie z.B. von oben nach unten, oder von rechts nach links. Da dieser horizontale Ansatz aber von der Darstellung einer Netzstruk-

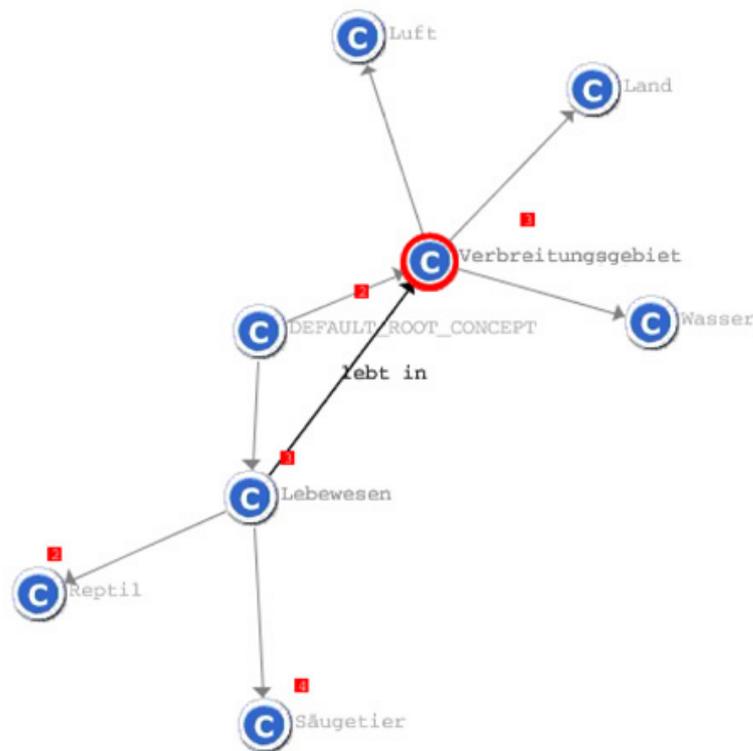


Abbildung 3.7: Ausschnitt aus der Ontologie Visualisierung in OntoEdit©

tur ausgeht, wird diese Leserichtung unterbrochen. Anhand der Abbildung 3.7, die dem Ontologie-Editor OntoEdit©(vgl. Sure u. a., 2002) – der ebenfalls auf eine einfache Vererbungsstruktur ausgelegt ist – entnommen wurde, wird ersichtlich, welche Schwierigkeiten bei einer Darstellung als Netzstruktur auftreten können.

schwerer verständlich wird. Bei einer Fokussierung auf die hierarchische Ordnung (vertikaler Ansatz) hingegen werden zwar nur Teilausschnitte und angedeutete Assoziationen angezeigt, die Unterschiede zwischen einer Vererbungsbeziehung und einer Assoziation werden aber klar herausgestellt. Die Wahl der beiden Ansätze sollte zum einen darauf basieren ob das gesamte Beziehungsgeflecht der Ontologie (horizontal) oder ob gezielt ein spezielles Modell und dessen direkter Kontext (vertikal) wiedergegeben werden soll. Des Weiteren ist von besonderer Wichtigkeit inwiefern die Visualisierung das ontologische Konzept für die zukünftigen Nutzer verdeutlichen soll.

Darstellung der Individuen

Die Darstellung der Individuen muss zwar – der ontologischen Struktur entsprechend – einen sichtbaren bzw. nachvollziehbaren Verweis zur Modellstruktur aufweisen, eine weitere Vermischung von Modell und Individuen würde aber der Übersicht schaden. Vielmehr sollte in Betracht gezogen werden ob die Darstellung der Individuen – dem »[...] *details on demand*« (vgl. Shneiderman, 1991) entsprechend – als zusätzliche Sicht angeboten wird, die eine sichtbare Verbindung zum jeweiligen Modell bzw. zur Modell-Ansicht aufweist. Vorstellbar sind hier Visualisierungen, die eine Teil-Ganzes Beziehung von Individuum und Modell in visueller Form vermitteln.

Anders als die Darstellung der Modellstruktur nehmen die Individuen eine von der hierarchischen Struktur abweichende Position ein. Der Fokus liegt hier auf den Assoziationen und Beziehungen der Individuen untereinander. Aufgrund der resultierenden Struktur als Netz oder Graph muss in dieser Sicht ein möglichst freies Arrangieren der Individuen ermöglicht werden. Ansätze semantisch nahe Entitäten auch räumlich abzugrenzen können hier ebenso eine Anwendung finden wie eine grafische Unterscheidung der Relevanz (Größenunterschied, Transparenzen etc.) eines Individuums in der Ontologie. Diese Ansätze müssen aber berücksichtigen, dass semantische Nähe und Relevanz subjektiv sind und nicht unbedingt den Anspruch auf Allgemeingültigkeit besitzen. Eine Integration dieser Ansätze sollte deswegen auch prüfen, inwiefern eine individuelle und persistente Darstellung der semantischen Nähe auf Nutzerseite zu verwirklichen ist. Ein weiterer Ansatz um zum einen diese semantische Nähe auszudrücken und zum anderen die Individuen geordnet darzustellen, ist eine kontextbezogene und nutzerzentrierte Filterung nach beliebigen Eigenschaften, Modellzugehörigkeiten oder Assoziationen. Diese Herangehensweise zur individuellen Strukturierung der Individuen bzw. Instanzen kann auf die gesuchten Informationen des einzelnen Nutzers eingehen und hierauf basierend eine automatische räumliche Anordnung vornehmen.

Im Sinne der schrittweisen Annäherung an die gesuchten Informationen sollten die benannten Eigenschaften des Individuums zunächst verdeckt werden. Dem Nutzer wird somit gestattet sich zunächst ein Bild über die Zusammenhänge der einzelnen Individuen untereinander zu machen und kann daraufhin selbst entscheiden für welche Entität er genauere Informationen erhalten möchte.

3.2.3 Gruppierung von Zusammenhängen

Aufgrund der komplexen Struktur einer Ontologie bzw. den vielen Unterscheidungsmerkmale (wie z.B. Assoziation von Modellen und Individuen) ist es, wie oben angedeutet, schwer die einzelnen Zusammenhänge dieser Elemente zu verdeutlichen. Besonders in einer grafischen Darstellung bietet es sich an, bestimmte Elemente einer Ontologie in einen gemeinsamen Kontext zu setzen. Hierfür kann eine visuelle Gruppierung in Form einer so genannten Interface-Metapher, wie z.B. die geläufige Schreibtisch- bzw. Desktop-Metapher heutiger Betriebssysteme, sinnvoll sein. Als Beispiel-Metapher für eine Ontologie sei an dieser Stelle auf die im Kapitel 5.1.1 beschriebene *Flüssigkeitsmetapher* verwiesen.

Das Schlüsselement einer Ontologie ist das Modell. Das Modell steht nicht für sich allein, sondern wird durch definierte Eigenschaften und Assoziationen charakterisiert und begrenzt. Modell, Eigenschaft und Assoziation bilden eine gemeinsame Struktur, die eine einzelne Betrachtung ausschließt. Für eine visuelle Darstellung kann eine Gruppierung dieser Elemente zu einer logischen Struktur in Betracht gezogen werden. Ähnlich verhält es sich auch im Bezug auf die einzelnen, von den Modellen abgeleiteten Individuen. Hier gilt es zu beachten, dass ein Individuum ohne Modellzugehörigkeit nicht existent ist, da es die abstrakten Eigenschaften des Modells mit Fakten aus der realen Welt belegt. Dies zeigt, dass auch Individuen Teil des Modells sind und folglich die logische Verbindung auch in der Visualisierung ihre Entsprechung finden muss. Die Herausforderung besteht hierbei in der Findung einer geeigneten Symbolik bzw. Darstellung, die dem Nutzer diese internen Verknüpfungen verdeutlichen kann. Basierend auf dieser Symbolik können für eine einfache Editierung elementbezogene Aktionen bzw. Funktionsgruppen angeboten werden. Aufgrund der Gruppierung kann der Nutzer somit besser ersehen, inwiefern sich eine Aktion auf das repräsentierte Gesamtgefüge auswirkt. Als vorteilhaft erweist sich hierbei – wie im drei Ebenen Modell beschrieben – ein selektiertes Modell oder Individuum in den Fokus des Betrachters zu rücken und es von den umgebenden Objekten abzusetzen.

3.2.4 Alternative Darstellungsform

Abweichend von dem vorherigen Ansatz zur Darstellung einer Ontologie sei an dieser Stelle eine weitere mögliche Darstellungsvariante vorgestellt. Diese vom Grundsatz her für eine mehrfache Vererbung ausgelegte Visualisierung, wird im weiteren Verlauf dahingehend untersucht, wie die Struktur für die einfache Vererbung adaptiert werden könnte. Der von Fluit, Sabou und van Harmeln vorgestellte Cluster-Map-Ansatz stellt eine mögliche Visualisierung für eine Ontologie in einer Mehrfachvererbung dar (vgl. Fluit u. a., 2002). Modelle und Individuen werden bei dieser Visualisierung (vgl. Abbildung 3.9) als gemeinsame Anhäufungen bzw. Cluster repräsentiert, wobei die grün dargestellten Kugeln und Verbindungslinien die einzelnen Modelle mit den jeweiligen Spezialisierungen darstellen (so ist *IT* z.B. eine Spezialisierung einer offenen Arbeitsstelle in einem Stellenmarkt *Job Va-*

3 Kollaborative Ontologien zur Strukturierung von Wissen

cancies). Anders als in den oben beschriebenen Ansätzen findet die Visualisierung

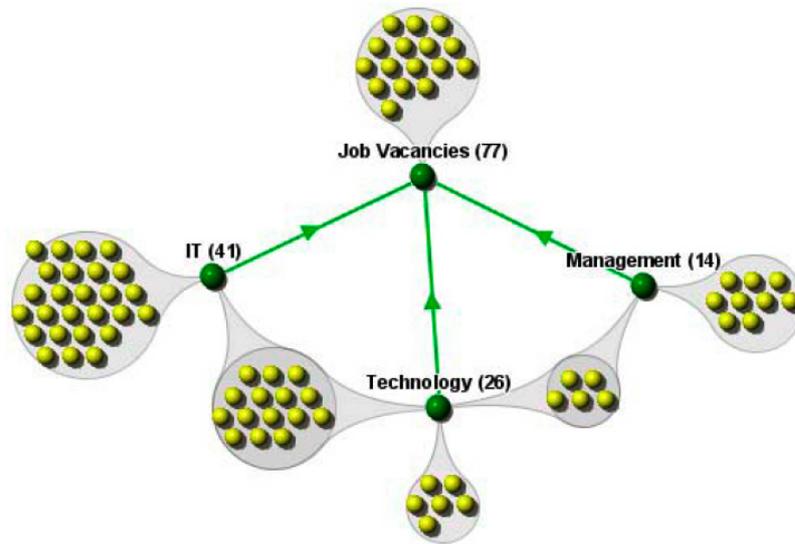


Abbildung 3.9: Darstellung einer Clustermap (Quelle: Fluit u. a., 2002)

gegenseitiger Assoziationen keine Anwendung. Vielmehr werden bei dieser Darstellungsvariante – in Bezug auf die mehrfache Vererbung – die gemeinsamen Individuen mehrerer Modelle dargestellt. Durch die Anzahl der Individuen begrenzt, ergeben sich in unterschiedlicher Größe dargestellte Ballons bzw. Cluster, die die Individuen als Kugeln darstellen. Individuen die auf mehrere Modelle aufbauen werden ebenfalls als Ballons dargestellt und verbinden die einzelnen Hauptmodelle. Der verbindende Cluster zwischen *IT* und *Technology* beinhaltet beispielsweise diejenigen Individuen, die aus beiden Modellen abgeleitet wurden. Die Größe dieser Schnittmengen soll unter anderem die semantische Nähe zweier Modelle widerspiegeln.

Dieser Ansatz weicht aufgrund der Visualisierung von mehrfachen Modellvererbung von dem in dieser Arbeit aufgestellten Ansatz ab. Die Darstellung gegenseitiger Spezialisierung wird ebenso deutlich wie die Menge an existierenden Individuen. Für eine Visualisierung einer kooperativ erstellten Ontologie bietet diese Darstellung jedoch zu wenige Informationen. Assoziationen, die bei der Strukturierung von Wissen eine zentrale Rolle spielen, werden in dieser Darstellung nicht berücksichtigt. Dies erschwert das Strukturieren enorm. Denkbar für den Fall der Einzelvererbung ist jedoch eine Modifikation des dargelegten Ansatzes. So könnte anstatt der Darstellung der Schnittmengen in Form von Ballons eine – auf Anzahl der definierten Assoziationen basierende – in der Größe variierende Gesamtverbindung zweier Modelle stehen. Diese Verbindung könnte hierbei die Stärke des gegenseitigen Bezugs bzw. der Verwandtschaft ausdrücken und durch eine Nutzeraktion als Einzelverbindungen sichtbar werden.

4 Szenarien - Einsatz von Ontologien

Im folgenden Kapitel sollen nun konkrete Einsatzszenarien für die kooperative Erstellung von Ontologien dargestellt werden. Die aufgezeigten Sichtweisen beschränken sich auf die Unterstützung von Lehrsituationen im universitären Rahmen und die Darstellung von Zusammenhängen innerhalb der Grenzen kooperativer Systeme.

4.1 Unterstützung von Lehr- Lernprozessen

Die Möglichkeit, Wissen und Wissens- bzw. Dokumentenzusammenhänge mit Hilfe von Ontologien darstellen zu können, macht dieses Feld für eine Integration in Lehr-Lernprozesse sehr interessant. Die Einsatzmöglichkeiten von Ontologien in Bezug auf die Unterstützung von Vorlesungen und Seminaren können weit definiert werden. Neben der Darstellung von Wissen lässt sich das ontologische Konzept auch für die Darstellung von organisatorischen Zusammenhängen nutzen. Im folgenden werden nun Anwendungsszenarien skizziert, die diesen weitreichenden Einsatz beschreiben.

4.1.1 Kooperative Organisation von Lehrveranstaltungen

Um den Einsatz von Ontologien im Bereich der wissenschaftlichen Lehre zu beschreiben sollten im Vorfeld einzelne Kernelemente von Vorlesungen und Seminaren unterschieden werden. Zunächst bietet sich eine Unterscheidung der an einer Lehrveranstaltung beteiligten Personen (Dozenten, Studenten und eventuell Übungsleiter bzw. Betreuer) an. Jede dieser Personen ist in der Struktur der Veranstaltung unterschiedlich involviert¹. Dozenten und Betreuer bereiten sowohl die Vorlesung als auch die Übungen vor. In dieser Phase, die vor den eigentlichen Veranstaltungen stattfindet, werden sowohl inhaltliche als auch organisatorische Zielrichtungen definiert.

Bereits in diesem frühen Stadium bietet sich der Einsatz von Ontologien an. So können in einem ersten Schritt die organisatorischen Strukturen und Rollen der beteiligten Personen untereinander, und deren Einsatz innerhalb der einzelnen Vorlesung definiert werden. Diese Elemente könnten z.B. die Themen der einzelnen Vorlesungsstunden und/oder Übungen sein. Diese Struktur würde zunächst Aussagen über die Beziehungen zwischen den Personen untereinander, Personen zu

¹Dieses Beispiel ist fiktiv und muss somit nicht unbedingt den tatsächlichen Gegebenheiten entsprechen

4 Szenarien - Einsatz von Ontologien

Themen und Themen zu Vorlesungsstunden bzw. Übungen treffen. Basierend auf diesen Beziehungen könnten dann Personen mit einem gleichen Themengebiet kooperativ an der weiteren Ausarbeitung eines Themengebietes arbeiten. Vorstellbar in diesem Zusammenhang ist z.B. die dargestellte Struktur in Abbildung 4.1. Bei

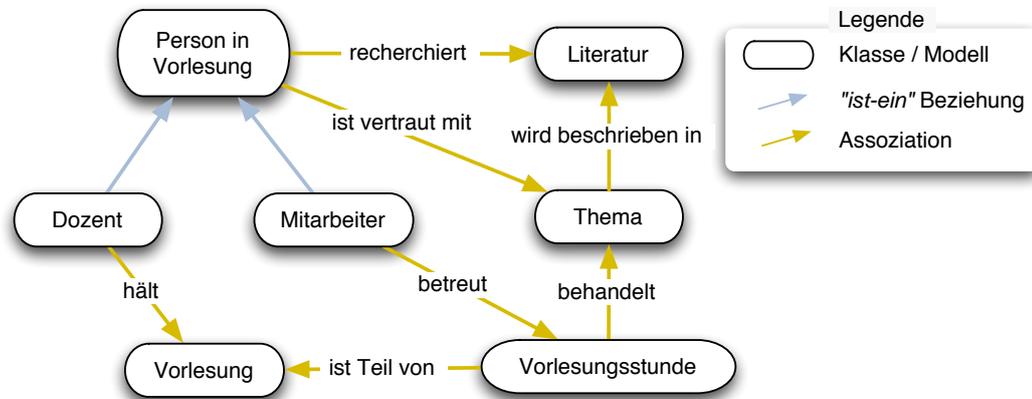


Abbildung 4.1: Ausschnitt der Ontologie *Vorlesung*

einer Übertragung der abstrakten Struktur auf reale Personen und Themen kann hiermit gezielt definiert bzw. herausgefunden werden, wer mit einem speziellen Thema vertraut ist, oder wer eine Vorlesungsstunde betreut. Des Weiteren ist es möglich diverse Literatur-Referenzen einem bestimmten Personenkreis und einem bestimmten Thema zuzuordnen.

Hiermit ergibt sich aufgrund des kooperativen Ansatzes zum einen die Möglichkeit den Rechercheprozess zu beschleunigen, und zum anderen die Möglichkeit eines diskursiven Austauschs über den Einsatz von z.B. einer bestimmten Literatur in der jeweiligen Vorlesung. Aufgrund der Teilung von abstraktem und realen muss diese Ontologie nicht zwangsläufig auf eine Vorlesung beschränkt bleiben, sondern kann z.B. auf eine Ontologie aus dem vorigen Semester aufbauen. So kann z.B. das Anlegen der Beziehung *Vorlesung* → *ist thematisch verwandt mit* → *Vorlesung* äußerst hilfreich sein, um beispielsweise Literatur aus anderen Vorlesungen in die derzeitige zu integrieren. Hierbei kann das Modell vom Typ *Literatur* z.B. Eigenschaften wie Titel, Autor oder Dateiverweis bereitstellen, die ein konstruiertes Individuum (z.B. ein konkretes Buch) mit den jeweiligen Daten bzw. Verweisen benennt.

4.1.2 Wissenserarbeitung im Team

Von der im vorhergehenden Kapitel größtenteils organisatorisch aufgebauten Ontologie ausgehend sollen nun – im Kontext des Beispiels Vorlesung bzw. Seminar – weitere denkbare Ausprägungen dargelegt werden. Ausgehend von der Perspektive der Studierenden bedarf es neben einer organisatorischen Struktur auch einer

Repräsentationsform zur Darstellung von fachlichen Zusammenhängen. In Seminaren ergibt sich oft die Konstellation einer gruppenweisen Erarbeitung eines bestimmten Themengebietes, das zu einem späteren Zeitpunkt innerhalb des Seminars präsentiert werden soll. In interdisziplinären Seminaren entsteht hierbei oft die Schwierigkeit, inwiefern die Gruppenteilnehmer eine gleiche Ausgangsbasis an Wissen in Bezug auf das zu erarbeitende Thema haben, und wie sich der persönlichen Standpunkt bezüglich des Themengebietes ausnimmt.

Das folgende Beispiel soll zum einen den kooperativen Ansatz und zum anderen die Möglichkeit der Strukturierung von Wissen durch Integration externer Dokumente verdeutlichen. Ein fiktives Seminar mit dem Titel »*Verwertungs- und Urheberrechte. Nutzungsmodelle zum Gebrauch digitaler Inhalte.*«, geht der Frage nach, welche Arten von Lizenzmodellen für spezielle digitale Artefakte existieren und welche Rechte diese Modelle garantieren. Die oben angesprochenen Gruppen bzw. Kleingruppen beschäftigen sich jeweils mit einer speziellen Ausprägung von Artefakten z.B. Musik oder Software und untersuchen die möglichen Lizenzmodelle anhand von Beispielen. Alle Kleingruppen arbeiten an einer Gesamtontologie, die sie zunächst gemeinsam aufbauen. In dieser Phase müssen alle Gruppen die (abstrakten) Gemeinsamkeiten der betrachteten Wissensdomäne spezifizieren. Dies bedeutet, im Bezug auf die möglichen Lizenzmodelle und Artefakte gegenseitige Beziehungen und Ableitungen einzelner Begriffe zu definieren (vgl. Abbildung 4.2 links). Basierend auf den getroffenen Ableitungen und Beziehungen der Gesamtgruppe erarbeiten nun die Kleingruppen weitere Spezialisierungen und Beziehungen der einzelnen Modelle und legen, basierend auf den Modellen, konkrete Individuen an.

Die Gruppe die sich mit den Lizenzmodellen digitaler Musiktitel beschäftigt leitet so z.B. das Modell *Musiktitel* von *Artefakt* ab. Bernd, ein leidenschaftlicher Musikliebhaber, definiert für das Modell *Musiktitel* sofort zahlreiche Attribute, wie z.B. Musikrichtung, Titel, Komponist, Albumname und Jahr. Anja und Carla finden jedoch das Titel und Komponist ausreichen um einen Musiktitel in diesem Kontext zu beschreiben. Schließlich einigen sich die drei auf ein weiteres Attribut *Titelinformation*, das weitere Informationen über eine Internetadresse zugänglich macht. Bei der weiteren Recherche entdecken sie das Lizenzmodell der Creative Commons², dass sie als mögliche Lizenzierungsform für wichtig erachten. Bernd stößt bei der Suche nach Beispielen auf den Internet-Musikverlag Magnatune³, der unter anderem das Künstlerduo »Ehren Starks« unter der Creative Commons Lizenz vertreibt. Die Gruppe hält dies für ein anschauliches Beispiel und beschließt das Recht des *freien privaten Gebrauchs* mit der genannten Lizenz zu verbinden (vgl. Abbildung 4.2 links).

Bei einer weiteren Betrachtung entdecken sie, dass aufgrund verschiedener Gesetzgebungen für jedes Land spezielle Lizenzen bestehen. Daraufhin entschließen

²Creative Commons definiert Lizenzierungen für diverse Inhaltsformen <http://creativecommons.org/>

³siehe <http://magnatune.com/>

4 Szenarien - Einsatz von Ontologien

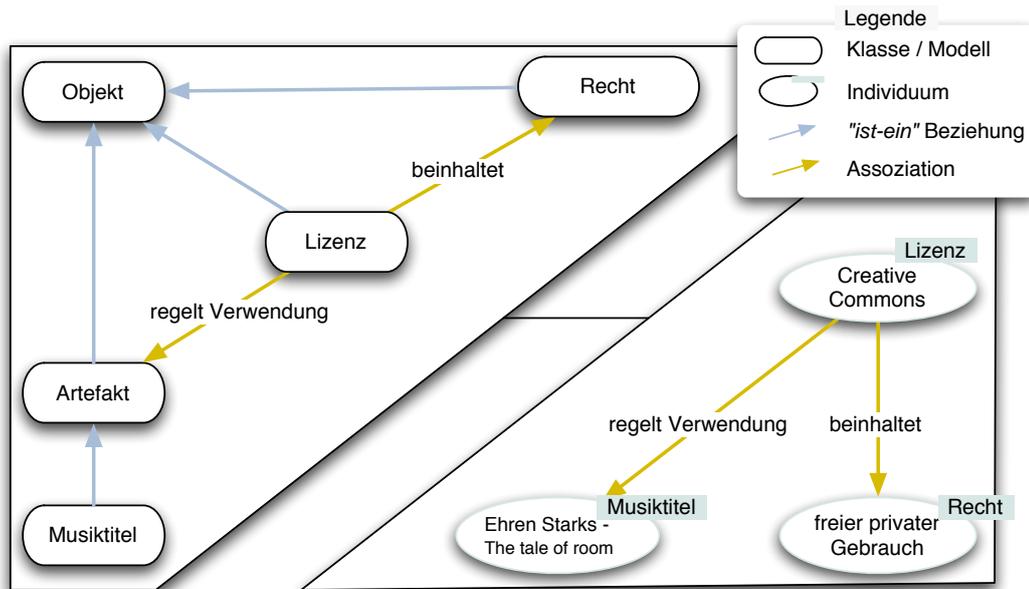


Abbildung 4.2: Ontologie zu Lizenzmodellen. Links: Modellstruktur, Rechts: Individuenstruktur

sie sich ein neues Modell *Land* und die Assoziation *Recht* → *ist gültig in* → *Land* in die Ontologie zu integrieren. Diese Beziehung kollidiert allerdings mit der von einer anderen Gruppe bereits erstellten Assoziation *Lizenz* → *ist gültig in* → *Land*, die sich auf die Gültigkeit von Softwarelizenzen in Bezug auf gerichtliche Auseinandersetzungen bezieht. Bei einem Diskurs der beiden Gruppen, ob nur ein bestimmtes Recht (z.B. Vervielfältigung, Integration in andere Medienprojekte) oder die Lizenz als Ganzes in einem Land Gültigkeit besitzt, einigen sie sich auf die Assoziationen *Lizenz* → *hat gerichtlichen Bestand in* → *Land* und *Recht* → *ist gesetzlich geregelt in* → *Land*. Da Carla die Lizenzbestimmungen zu komplex erscheinen definiert sie für das Modell *Recht* das Attribut *Lizenzauszug* und kann so im Einzelnen zu den von Anja übersetzten (als Dokumente im System vorhandenen) Lizenzbestimmungen verweisen.

Bereits an dieser Stelle wird deutlich, welchen Freiraum bzw. persönlichen Nutzen die Strukturierung von Wissen in Form von Ontologien bietet. Zum einen lassen sich fachliche Bezüge innerhalb dieser Repräsentationsform besser strukturieren und aufgrund der gemeinsamen Sicht auf die Wissensdomäne können inhaltliche Beziehungen einer erneuten Betrachtung unterzogen werden. Auf der anderen Seite bietet sich die Möglichkeit Attribute zu definieren, die ein Modell weiter spezifizieren, oder externe Referenzen wie z.B. Internetseiten oder (multimediale) Dokumente in den gesamten Aufbau mit einzubeziehen, um somit das recherchierte Material direkt mit der Wissensstruktur zu verknüpfen. Des Weiteren wird hierbei deutlich, dass diese Art der Strukturierung einen hohen kognitiven Aufwand bedeutet, der durch ein geeignetes Hilfsmittel unterstützt werden muss, das dem

Nutzer ein leichtes Editieren und Arrangieren, sowie eine eingehende visuelle Erarbeitung ermöglicht.

4.1.3 Ontologien als Übungen

Der Einsatz von Ontologien bietet sich auch als Erweiterung von vorlesungsbegleitenden Übungen an. Besonders wenn es um das Darlegen von erlernten Kontexten geht, kann eine grafisch repräsentierte Ontologie dem Übungsleiter dabei helfen, gezielt etwaige Missverständnisse unter den Lernenden zu beseitigen. Analog zu dem vorherigen Szenario ist die folgende Übungssituation denkbar. Basierend auf der gruppenweiten Ontologie adaptiert nun jede Kleingruppe diese Wissensstruktur (Ontologie) und bearbeitet sie unabhängig weiter. Übungsfragen inwiefern sich die einzelnen Lizenzmodelle unterscheiden oder welche Vor- und Nachteile sich für die *Inhalte-Erzeuger* und *-Verwerter* ergeben scheinen denkbar.

Neben diesen subjektiven Fragestellungen kann auch eine andere Herangehensweise in Betracht kommen. Eine Übung kann auch aus der Vorgabe abstrakter Modelle ohne definierte Assoziationen bestehen, die den Lernenden vorgeben werden. In dieser Ontologie müssen die Lernenden dann eigene Assoziationen und Ableitungen anlegen und gezielt Individuen einsetzen. Durch die gleiche Ausgangslage der einzelnen Übungsgruppen kann so ein objektiver Vergleich über den Kenntnisstand der einzelnen Gruppen geführt werden. Beispiel hierfür wären Thematiken wie der Verlauf der Französischen Revolution als historische Tatsache, die sich aufeinander beziehen und nur richtige oder falsche Assoziationen und Individuen zulassen.

Ein weiterer Einsatz ist die Anwendung von wissenschaftlichen Konzepten die dem Lernenden vermittelt werden sollen. In der Übung müssen nun diese Konzepte auf spezielle und vorher definierte Sachverhalte angewendet werden. Als Beispiel dient hierbei eine Vorlesung über die Visualisierung von Daten. In dieser Vorlesung sollen den Studierenden die Basiskonzepte vermittelt werden, wie bestimmte Daten bzw. Datentypen wie z.B. Jahreszahlen, Temperaturen etc. aussagekräftig visualisiert werden können. Für eine Übung kann eine Ontologie bereits gewisse Modelle wie z.B. *Datentyp*, *Visualisierungsform* bereitstellen und schon konkrete Individuen definieren wie z.B. *Verkaufszahlen*, *Wochen*, *Länder* als Datentypen und *Position*, *Farbe* und *Form* als Visualisierungsform. Anhand der in der Vorlesung vermittelten Konzepte müssen die Studierenden nun den einzelnen Visualisierungsformen über die Assoziation *ist geeignet für* die einzelnen Datentypen zuordnen (so ist z.B. für das Individuum *Verkaufszahlen* die geeignete Visualisierungsform die *Position*). Für den Übungsleiter ergibt sich hier ebenfalls die Möglichkeit, Kenntnis über das derzeitige Verständnis innerhalb der einzelnen Gruppen zu erlangen.

Bei der Unterstützung von Übungen durch Ontologien muss von Fall zu Fall entschieden werden, ob diese eher eine Übung unterstützen oder selbst eine Übung darstellen sollen. Diese Fallunterscheidung bezieht sich im besonderen Maße darauf, ob die Ontologie primär die Repräsentation von Wissen als Ganzes, oder die Strukturierung von externen Dokumenten unterstützen soll bzw. kann.

4.1.4 Erschließung des kooperativen Systems

Denkbar ist der Einsatz einer Ontologie auch für die Darstellung von Zusammenhängen innerhalb eines kooperativen Systems wie z.B. ^{open}sTeam. Die Nutzer eines solchen Systems gestalten, individuell oder in Gruppen, einen virtuellen Datenraum. Am Beispiel des ^{open}sTeam-Systems – das zumeist in der universitären Lehre Anwendung findet – wird deutlich, dass sich Nutzer und Nutzergruppen nicht zufällig zu einem Verbund zusammenschließen, sondern meist aufgrund einer seminar- bzw. vorlesungsbegleitenden Situation. Speziell in Vorlesungen der Informatik ergibt es sich oft, dass die Lernenden sich zu einer Übungsgruppe von drei bis vier Lernenden zusammenschließen, um gemeinsam die gestellten Übungsaufgaben zu lösen. Besonders bei großen Veranstaltungen gestaltet sich diese Gruppenfindung zuweilen als mühsam. Hier geben die privaten Präferenzen der Studierenden untereinander oftmals den Ausschlag für eine Gruppenbildung. Dies muss zwar nicht zwangsläufig schlecht sein, es besteht aber die Gefahr, dass die Kompetenzen der Lernenden nicht optimal bzw. einseitig gebündelt werden. Insbesondere

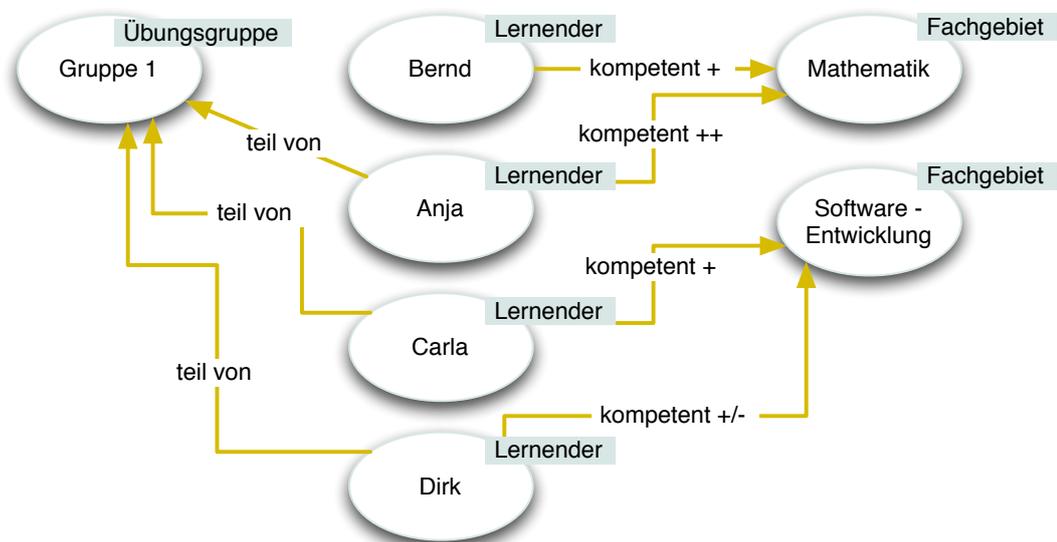


Abbildung 4.3: Verteilung von Kompetenzen für bestimmte Fächer

aufgrund der Vereinigung verschiedener Studienrichtungen innerhalb einer einzelnen Vorlesung (am Beispiel der Informatik: Kern-, Wirtschafts- und Medieninformatik, Lehramt etc.) können diese Kompetenzen unterschiedlich ausgeprägt sein. Vorstellbar ist deshalb der Einsatz einer Ontologie, in der Lernende ihre Kompetenzen (im Sinne von fachlichen Kompetenzen wie z.B. Mathematik oder Softwareentwicklung) subjektiv bewerten. In Abbildung 4.3 wird ein möglicher Ausschnitt dieser Ontologie präsentiert. Hierbei setzen sich die einzelnen Lernenden mit den geforderten Fachgebieten in Beziehung und bewerten die jeweiligen Stärken und Schwächen. Basierend auf diesen Zuordnungen können nun entweder die Vorle-

4 Szenarien - Einsatz von Ontologien

sungsteilnehmer selbst eine Gruppenzuweisung durchführen oder die jeweiligen Übungsleiter übernehmen diesen Prozess. Durch die verschiedenen Kommunikationsstrukturen innerhalb des Systems ist es zudem möglich gegenseitigen Kontakt aufzubauen. Des Weiteren erscheint es denkbar, diesen Prozess auch automatisiert ablaufen zu lassen. Basierend auf den definierten Zuweisungen könnte so das System Gruppenzusammensetzungen konstruieren, die z.B. eine maximale Gruppenstärke (3-4 Personen) und die Verteilung von Kompetenzen optimal darstellen.

Aufbauend auf der Repräsentation der Systemnutzer in einer Ontologie erscheinen weitere Szenarien für eine Nutzer zu Nutzer Verknüpfung denkbar. Eine Erweiterung des Modells *Lernender* bzw. *Nutzer* durch die Assoziation *hat Studienschwerpunkt* ließe es zu, neue Gemeinsamkeiten innerhalb der Nutzerschaft des Systems zu ergründen. Dies könnte in einem konkreten Fall z.B. für die Suche von Spezialisten zu einer bestimmten Thematik seine Anwendung finden. Durch Rückreferenzen von einem Fachgebiet könnten somit alle Entitäten bzw. in diesem Falle Nutzer aufgezeigt werden, die die Assoziation *hat Studienschwerpunkt* mit dem ausgewählten Fachgebiet verknüpfen. Ähnlich der Verknüpfung von Nutzern inner-

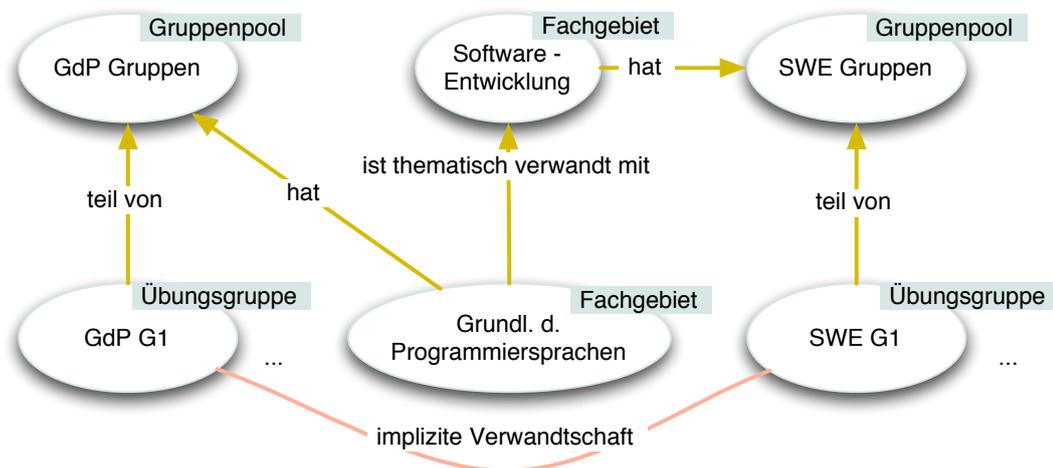


Abbildung 4.4: Thematisch verwandte Fächer, mit impliziter Verwandtschaft einzelner Gruppen

halb des kooperativen Systems ist es auch möglich ganze Nutzergruppen in neue Beziehungen zu setzen. Um bei dem Beispiel der Vorlesung bzw. des Seminars zu bleiben, wäre es vorstellbar alle existierenden Gruppen aus den vergangenen Semestern mit einer Vorlesung zu verknüpfen. Diese Vorlesungen würden wiederum über eine Assoziation *ist thematisch verwandt mit* untereinander in Beziehung gesetzt. Hiermit ergibt sich die Möglichkeit nicht nur die relevanten Gruppen in Bezug auf eine Vorlesung zu identifizieren, sondern auch weiter (in Bezug auf die Vorlesung) entfernte Gruppen zu finden, die aufgrund der thematischen Verwandtschaft ebenfalls relevant erscheinen. Das folgende Beispiel (vgl. Abbildung 4.4) soll dies genauer spezifizieren. Gegeben sind zwei Informatik Vorlesungen mit den Ti-

4 Szenarien - Einsatz von Ontologien

teln »Softwareentwicklung« (1) und »Grundlagen der Programmiersprachen« (2). Während die erstgenannte Vorlesung anhand der Programmiersprache Java⁴ das Konzept der Objekt Orientierten Programmierung vermittelt, gibt die andere Einblicke über Syntax und Konzept existierender Programmiersprachen. Für beide Veranstaltungen existieren Gruppen im ^{open}sTeam System, in denen Vorlesungsfolien, Literaturtipps, Übungen und Programmierbeispiele verwaltet werden. Die Teilnehmer einer Übungsgruppe der 2. Veranstaltung haben nun die Aufgabe eine Übung zu bearbeiten, die auf das Objekt Orientierte Konzept der Sprache Java aufbaut. Um diese Aufgabe zu lösen bedarf es einer zusätzlichen Wissensquelle. Idealerweise existiert nun eine Ontologie in der beide Veranstaltungen als thematisch verwandt gekennzeichnet und die zugehörigen Gruppen und Materialien mit dieser Veranstaltung verbunden sind. Schließlich stößt die Übungsgruppe auf die 1. Veranstaltung innerhalb der Ontologie und kann somit auf das benötigte Wissen zugreifen. Hierbei ergibt sich aufgrund der thematischen Verwandtschaft der beiden Vorlesungen auch eine implizite Verwandtschaft der verbundenen Übungsgruppen, Materialsammlungen oder sonstiger Bezüge.

⁴Java ist eine Objekt Orientierte Programmiersprache

5 visCOntE - Eine Umgebung zur Erstellung kooperativer Ontologien

visCOntE – Visual Cooperative Ontology Environment – ist eine Umgebung zur kooperativen Erstellung und Editierung von Ontologien. Basierend auf der open-source Plattform ^{open}sTeam (vgl. Hampel und Keil-Slawik, 2002) stellt dieses Rahmenwerk zum einen eine Web- bzw. Browserbasierte Applikation zur grafischen Bearbeitung von Ontologien bereit, und zum anderen eine offene Server-Applikation, die der Verwaltung, Strukturierung und Abfrage dient.

5.1 Visueller Browser und Editor

Basierend auf den in Kapitel 3.2 erläuterten Eigenschaften der Darstellung einer Ontologie, wird in diesem Abschnitt die Visualisierung der – auf Skalierbare Vektorgrafiken (SVG) aufbauenden – Nutzer-Applikation sowie integrierter Medienelemente und kooperationsunterstützender Ansätze beschrieben. Ziel der Visualisierung ist, neben der später beschriebenen Navigation und den Editierfunktionen, die Vermittlung des ontologischen Konzeptes in einer nachvollziehbaren Form.

5.1.1 Die Konzept Metapher

Der Aufbau einer Ontologie ist eine komplexe Struktur von ineinandergreifenden Konzepten. Anhand einer detaillierten Betrachtungsweise dieses Konzeptes wird im Folgenden die Hinführung zu einer aufeinander aufbauenden Metapher zur Verdeutlichung des Konzeptes beschrieben.

Bei einer genauen Betrachtung der Vererbungsbeziehungen zwischen den einzelnen Modellen fällt auf, dass die Eigenschaften und Assoziationen von dem generellen bzw. höherliegenden Modell auf die spezialisierten Modelle übergehen. Die Beziehung der Modelle untereinander kann als eine Art Fluss von Attributen (in Form von Eigenschaften und Assoziationen) verstanden werden. Diesem Fluss gleicher Merkmale wurde in der Visualisierung mit einer Flüssigkeitsmetapher Rechnung getragen. Die Verbindung von Modell zu Obermodell wird mit einer Art *Schlauch* dargestellt, der die Vererbungsbeziehung unterstreichen soll (vgl. Abbildung 5.1). Die Repräsentation der Modelle stellt eine Art *Reservoir* dar, in dem sich die einfließende Flüssigkeit sammelt bzw. Gemeinsamkeiten von einem zum anderen Modell fließen. Die einzelnen Modelle geben die definierten Eigenschaften und Assoziationen an die durchfließende Flüssigkeit ab. Diese Flüssigkeit symbolisiert somit einerseits die geerbten Eigenschaften, die in das Modell einfließen, zum anderen

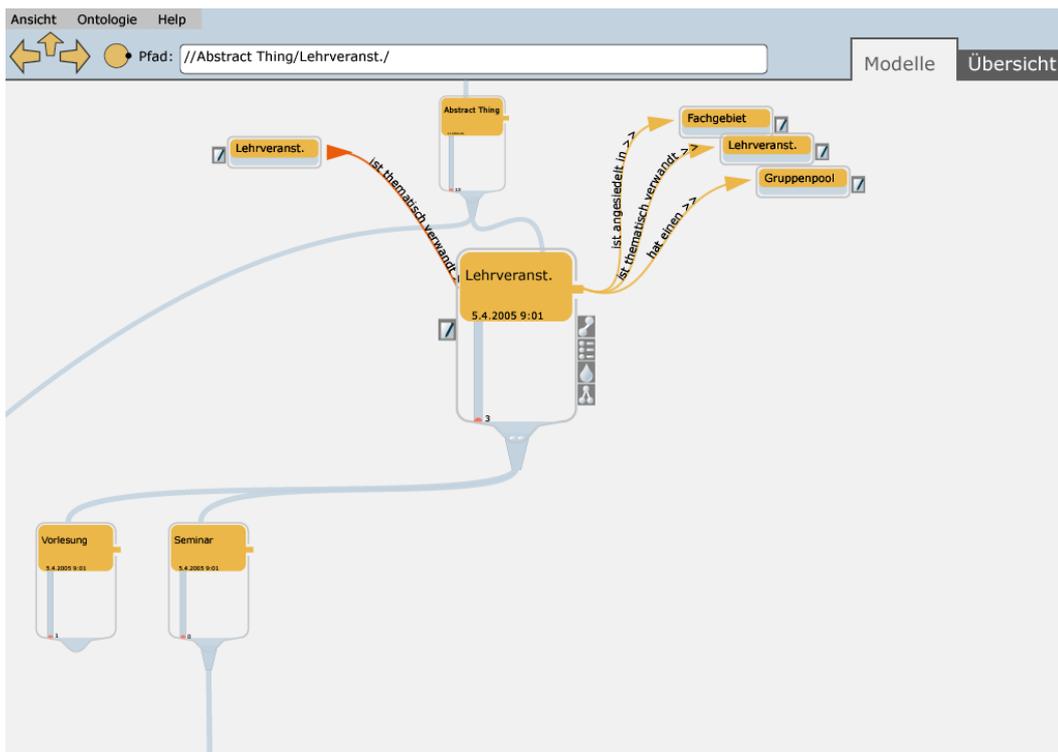


Abbildung 5.1: Die Flüssigkeitsmetapher am Beispiel der Modellhierarchie

bestehen die mit dem Modell verbundenen Individuen eben aus der Summe dieser Eigenschaften und Assoziationen. Als eine konsequente Fortführung der Metapher resultieren Individuen somit in der Darstellung als (einzelne) Tropfen. Die jeweilige Füllhöhe in den Reservoiren gibt die Anzahl der Tropfen bzw. Individuen wieder, die vom aktuellen Modell und dessen Kindmodellen abgeleitet wurden.

Im Folgenden soll anhand der Abbildung 5.2 weiter erklärt werden, wie die Repräsentation eines Modells aufgebaut ist. Die Aktionsknöpfe (1-4) stellen jeweils eine Gruppierung von Aktionen dar, die auf die einzelnen Elemente der Ontologie ausgerichtet sind. Im Einzelnen sind dies: Assoziationsaktionen (1), Eigenschaftsaktionen (2), Individuumaktionen (3) und Hierarchieaktionen (4). Diese Aktionen werden zum einen durch die angezeigten Symbole als auch durch spezielle Zeiger, die auf den jeweiligen grafischen Bereich zielen (z.B. Reservoir), verdeutlicht. Das Reservoir (5) zeigt die Anzahl der Individuen, die sowohl von dem selektierten Modell als auch von den Folgemodellen abgeleitet wurden. Via Mausklick gelangt der Nutzer zu der Darstellung der Individuen in der Form von Tropfen. Das links neben dem Modell platzierte Symbol (6) erlaubt es dem Nutzer dem Modell eine Annotation anzufügen und gleichzeitig die bereits vorhandenen Kommentare zu betrachten. Das Feld im oberen Teil des Modells (7) gibt über den Namen und das Datum der letzten Änderung Auskunft. Des Weiteren sind verdeckt hinter diesem

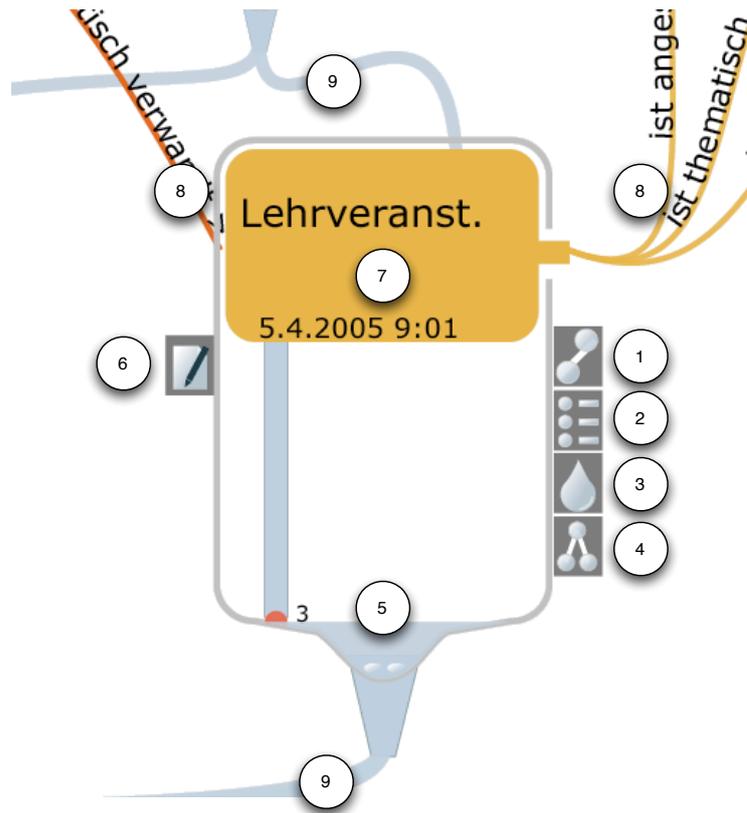


Abbildung 5.2: Vergrößerte Ansicht eines Modells

Etikett die definierten und geerbten Eigenschaften, die ebenfalls via Mausklick als Liste erscheinen. Eingehende (8, links) und ausgehende (8, rechts) Assoziationen werden ebenfalls mit dem oben genannten Feld verbunden. Die bläulich dargestellten Linien (9) symbolisieren die eingehende bzw. ausgehende Vererbungslinie in Form eines Schlauches.

Basierend auf der Shneiderman'schen Formel »*Overview first, zoom and filter, then details on demand*« (vgl. Shneiderman, 1991) ist das Anzeigen der Individuen (im Reservoir) und Eigenschaften (im Etikett) erst nach einer expliziten Aktion des Nutzers möglich. Hiermit wird die Anzeige von weiteren Informationen auf mehrere Ebenen verteilt, was einer besseren Übersicht zugute kommt.

Assoziationen werden neben dem jeweiligen Modell dargestellt (vgl. Abbildung 5.1) und durch Verbindungspfeile mit verkleinerten Modellen verbunden. Die Verbindungspfeile werden mit dem Namen der Assoziation beschriftet und mit einer zusätzlichen Pfeilspitze dargestellt, die die Leserichtung der Assoziation wiedergeben soll. Des Weiteren wird die Leserichtung durch die Anordnung der Assoziationen unterstützt. So sind die auf der rechten Seite dargestellten Modelle Assoziationen die vom betrachteten Modell herausgehen, es handelt sich hierbei um

die Relation *aktuelles Modell* \rightarrow *Assoziiertes Modell*. Auf der linken Seite hingegen verkehrt sich die Richtung in die Relation *Modell* \rightarrow *aktuelles Modell*, also in eine Darstellung der eingehenden Assoziationen.

Diese Darstellung lässt es zu, das Modell in seinem Gesamtkontext in der Ontologie zu betrachten. Abweichend von den hier dargestellten Abbildungen kann z.B. der folgende semantische Kontext anhand des Konzeptes »Vorlesung« veranschaulicht werden. Für eine Vorlesung gibt es sowohl eine Assoziation 1.) *Dozent* \rightarrow *hält* \rightarrow *Vorlesung* als auch die Assoziation 2.) *Student* \rightarrow *nimmt teil an* \rightarrow *Vorlesung*. Des Weiteren existiert die Beziehung 3.) *Vorlesung* \rightarrow *behandelt* \rightarrow *Thema*. Bei einer Beschränkung der Darstellung auf die ausgehenden Beziehungen, würde der Gesamtkontext aufgrund der alleinigen Anzeige von 3. nur ungenügend wiedergegeben. Bei der Berücksichtigung der eingehenden Assoziation (1. und 2.) jedoch bettet sich das Konzept *Vorlesung* besser in den Kontext von *Dozent*, *Student* und *Thema* ein und ermöglicht so eine bessere Übersicht des Modells.

Die Anwendung dieser Metapher in Bezug auf eine intuitive Herangehensweise zum Konzept der Ontologie bietet zum einen eine nachvollziehbare Darstellung des strukturellen Aufbaus, und zum anderen die Vermeidung des konzeptuellen Bruchs zwischen Modell-Struktur auf der einen Seite und abgeleiteten Individuen auf der anderen (vgl. hierzu auch Kapitel 3.2). Die Integration der Individuen (in der Darstellung als Füllmenge) in die grafische Repräsentation des Modells ermöglicht dennoch eine visuelle Trennung des Abstrakten und Realen, was einer besseren Übersichtlichkeit dienlich ist.

5.1.2 Verschiedene Sichten

Für eine verbesserte Darstellung der Ontologie gliedert sich der Aufbau in drei verschiedene Sichten. Neben einer umfassenden Übersicht für die Navigation innerhalb der Ontologie, wird eine eigene Ansicht für die Darstellung der hierarchischen Bezüge bzw. Modelle und der Anzeige der Individuen bereitgestellt.

Darstellung der Übersicht

Die Darstellung der Übersicht basiert auf dem in Kapitel 3.2.2 beschriebenen Treemap-Ansatz (vgl. Shneiderman, 1992), der eine optimale Platzausnutzung von hierarchischen Strukturen aufweist. Anders als etwa die aus der Dateiverwaltung bekannten Dateibäume bietet die Darstellung eines Baumes (Tree) in Form einer Karte (Map) einige Vorteile. Das Prinzip der Treemap basiert auf einer rechteckigen Darstellung einzelner Hierarchie- bzw. Baumebenen, wobei die Wurzel das Ausgangsrechteck bildet. Durch eine rekursive Teilung dieser Struktur in Bezug auf die Gewichtung der Unterbäume entstehen kleinere Rechtecke, deren Fläche die Tiefe der folgenden Ebenen widerspiegelt (vgl. Abbildung 5.3). Diese Darstellung ermöglicht es zum einen die Hierarchie angepasst an den verfügbaren Anzeigebereich darzustellen, und zum anderen einen detaillierteren Überblick über die Tiefe der Struktur. Gerade der letzte Aspekt ermöglicht es einen tiefen Sprung in eine

5 visContE - Eine Umgebung zur Erstellung kooperativer Ontologien

Hierarchie zu vollziehen, ohne jede Ebene explizit – wie in einer Ordnerstruktur – aufzurufen. Im Kontext der Nutzer-Anwendung nimmt der Treemap Ansatz, wie

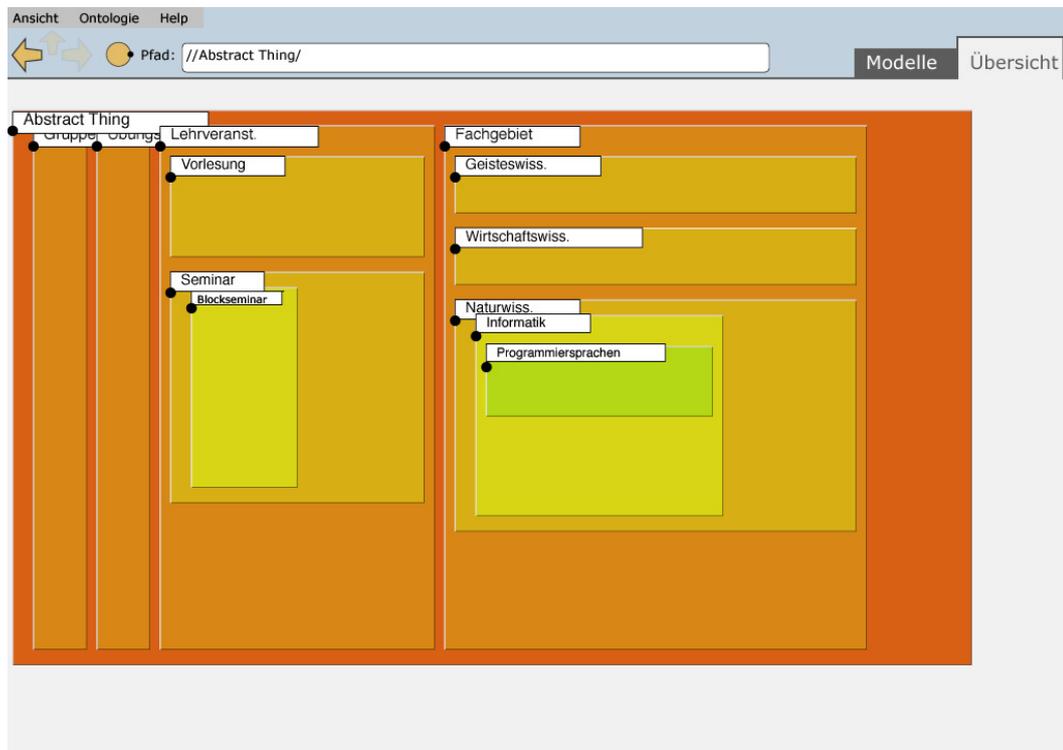


Abbildung 5.3: Die Übersicht der Ontologie als Treemap

oben erwähnt, die Rolle der Gesamtübersicht ein. Die einzelnen Rechtecke stellen die Modelle bzw. Konzepte in der hierarchischen Ordnung dar. Neben der Tiefe der einzelnen Ebenen fließt zusätzlich die Anzahl der mit dem Modell verbundenen Individuen in die Berechnung der Flächengröße mit ein. Darüber hinaus symbolisiert die Farbe der Rechtecke die jeweilige Ebene der einzelnen Modelle. So kann, wie in Abbildung 5.3 zu sehen ist, der Repräsentation entnommen werden, dass Vorlesung und Seminar auf der gleichen Hierarchieebene platziert sind. Die unterschiedlichen Flächengrößen und Farben lassen so Unterscheidungen in Bezug auf Relevanz (Menge der Individuen) und Spezialisierung (Tiefe des Modells) zu. Das Anspringen tiefer liegender Ebenen erfolgt durch das jeweilige Anwählen des Rechteckes, wobei das ausgewählte Rechteck bzw. Modell die Rolle einer lokalen Wurzel übernimmt und die Darstellung des lokalen Unterbaums in vergrößerter Form fortgesetzt wird. Am Beispiel der Abbildung 5.3 würde nach dem Auswählen des Rechteckes *Fachgebiet* dieses zum umfassenden Rechteck und die eingebetteten Rechtecke würden entsprechend größer. Zur namentlichen Identifizierung werden jedem Rechteck Fähnchen mit den entsprechenden Modellnamen hinzugefügt, die jeweils nach einem Kontakt zwischen Maus-Cursor und Rechteck angezeigt bzw.

zur besseren Lesbarkeit nach vorne gebracht werden.

Darstellung der Modelle

Die Auswahl eines Modells in der Übersicht bzw. Treemap steht in wechselseitiger Beziehung mit der Darstellung der Hierarchie als Graphen. Diese Sicht basiert auf dem in Kapitel 3.2 aufgestellten Ansatz einer vertikalen Sichtweise und stellt einen detaillierten Ausschnitt eines Modells mit Bezug auf die umgebenden Modelle dar. Die Darstellung erfolgt durch die Anzeige von genau drei Ebenen aus der Hierarchie. Neben dem aktuell ausgewählten Modell in der Mitte, werden oberhalb das generellere (Vater-) Modell und unterhalb die spezialisierten (Kind-)Modelle angezeigt. Die symbolisierten Verbindungen (vgl. Kapitel 5.1.1) und die vertikale Positionierung der Modelle bzw. Reservoir unterstreichen die Vererbungs- bzw. *ist-ein-Beziehungen* der Modelle untereinander.

Des Weiteren deuten Verbindungen (von Vater- und/oder Kindmodellen ausgehend), deren Endpunkte außerhalb des sichtbaren Bereichs liegen, die Existenz von entweder weiteren Untermodellen (Verbindung unterhalb der Kindmodelle) oder weiteren Modellen auf der gleichen Hierarchieebene des fokussierten Modells (weitere Verbindung unterhalb des Vatermodells) an (vgl. Abbildung 5.1). Jedes der umgebenden Modelle kann durch eine Nutzeraktion ausgewählt werden und somit in den Fokus des Betrachters rücken. Diese Nutzeraktion spiegelt sich, wie oben erwähnt, auch in der Position der Treemap bzw. Übersicht wider, da beide Sichten miteinander verbunden sind. Damit lässt sich im Bezug zur aktuellen Position in der Hierarchie immer feststellen welche Modelle sich noch in einem Unterbaum befinden.

Darstellung der Individuen

Die Individuen oder real existierenden Elemente werden ebenfalls in einer eigenen Ansicht dargestellt, die direkt mit den als Reservoir dargestellten Modellen verbunden ist. Diese wird sichtbar durch einen Klick auf die im Reservoir befindliche Flüssigkeit, wobei eine Animation das aktuelle Modell einklappt und zu der in Abbildung 5.4 dargestellten Repräsentation wechselt (für einen Wechsel zurück in die Modellansicht wird diese Animation umgekehrt ausgeführt). Während der Animation die in die *Tropfen-Ansicht* wechselt, werden die Individuen des ausgewählten Modells und die seiner Untermodelle vom Server geladen und schließlich angezeigt. In dieser Sicht stellen sich die Individuen und deren Assoziationen zu anderen Individuen als verbundene Tropfen dar, die frei positioniert werden können. Durch das Anklicken eines einzelnen Tropfens werden die jeweilig assoziierten (Vor- und Rückbeziehungen) Tropfen bzw. Individuen nachgeladen. Hier spiegelt sich die Farbgebung der ein- und ausgehenden Assoziationen aus der Modellansicht wider (vgl. Abbildung 5.4). Ein selektierter Tropfen hebt sich durch eine Vergrößerung von den anderen sichtbaren Tropfen und Individuen ab. Basierend auf der Selektion können nun weitere Informationen abgefragt werden. In dem Tropfen

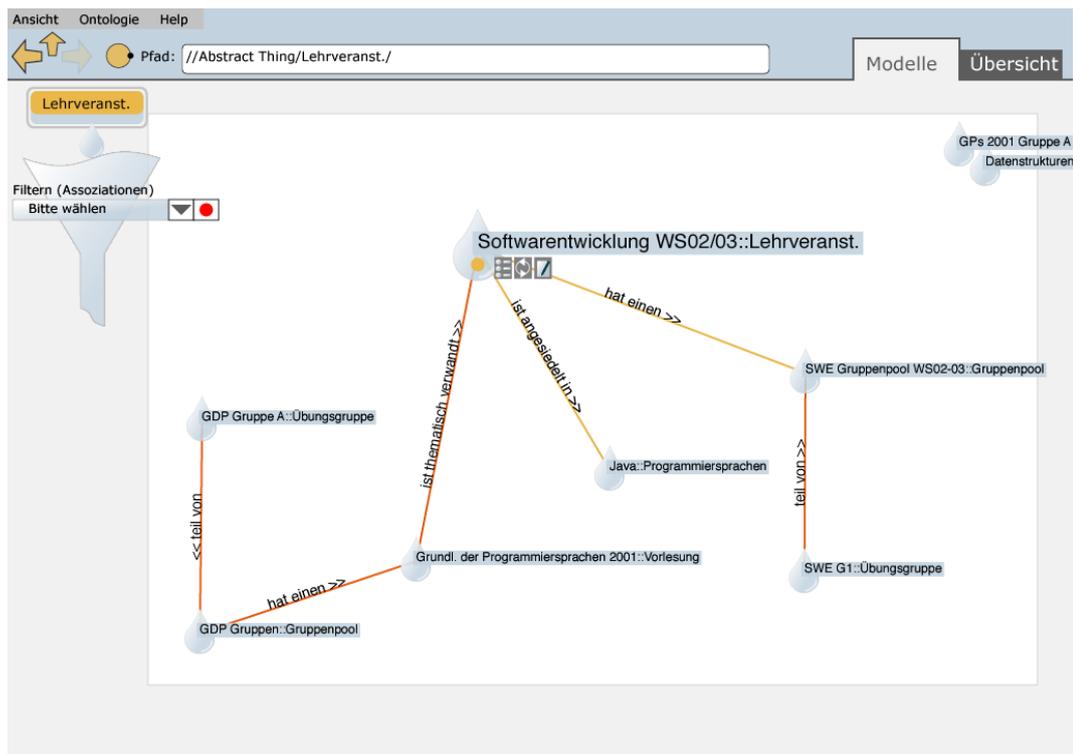


Abbildung 5.4: Darstellung der einzelnen Individuen als Tropfen

selbst sind die einzelnen Eigenschaften des Individuums gesammelt und werden auf einem abgesetzten Kreis dargestellt (vgl. Abbildung 5.5). Da die mögliche Zahl beschreibender Eigenschaften unter Umständen sehr groß sein kann, werden nur drei Viertel des Kreises zur Darstellung genutzt, das verbleibende Viertel wird als verdeckte Ansammlung der nicht darstellbaren Eigenschaften benutzt. Eine Pfeilnavigation ermöglicht eine Rotation der Eigenschaften um diesen Kreis, wobei jede Eigenschaft um genau eine Position nach rechts oder links rückt. Die Eigenschaften werden dabei von der Kreismitte zu den Kreisrändern mit Verbindungslinien dargestellt und symbolisieren somit den Verbund mit dem Tropfen bzw. Individuum. Die Bezeichnungen und Werte der Eigenschaften werden je nach Typ der Eigenschaft (z.B. Verknüpfung, Text etc.) anders dargestellt. Insbesondere Verweise, auch als URLs¹ bezeichnet, bilden den Kern der Strukturierung von externen Ressourcen. Die folgende Liste gibt einen Überblick, welche Typisierungen unterschiedliche Darstellungen erzwingen.

- URLs, deren Protokolle durch einen Webbrowser aufrufbar sind (z.B.: <http://> oder <ftp://>), werden neben der Darstellung als Text auch als externe Verknüpfung angeboten.

¹URL - Uniform Resource Locator - Bezeichnung für eine eindeutige Adressierung von Dateien (zumeist im Kontext des Internets gebraucht)

5 visContE - Eine Umgebung zur Erstellung kooperativer Ontologien

- URLs mit Endung auf eine Grafikdatei wie z.B.: *.png, *.gif oder *.jpg werden neben der aufrufbaren Verknüpfung auch direkt als Bild dargestellt.
- Verweise auf ^{open}sTeam interne Dokumente werden ebenfalls als Verknüpfung angeboten und zusätzlich mit den aus ^{open}sTeam bekannten Symbolen angezeigt. (Beispiel: Diskussionsforum, Container etc.)
- Die Eingabe eines ^{open}sTeam internen Nutzernamens wird direkt mit dessen Heimatomdner verknüpft. Hier können dann weitere Informationen über die betreffende Person abgerufen werden.
- Elemente mit Datum und Zeitbezügen werden spezifisch gekennzeichnet. Eine Verknüpfung bzw. Verarbeitung durch externe Kalenderfunktionen erscheint hierbei denkbar.

Die Liste möglicher Datentypen kann je nach Anwendungskontext im später beschriebenen Server-Modul beliebig erweitert werden. Neben den oben aufgeführten Typisierungen werden auch Wahrheitswerte und verschiedene Zahlenformate unterstützt. Durch die Differenzierung einzelner Typen werden somit Referenzen

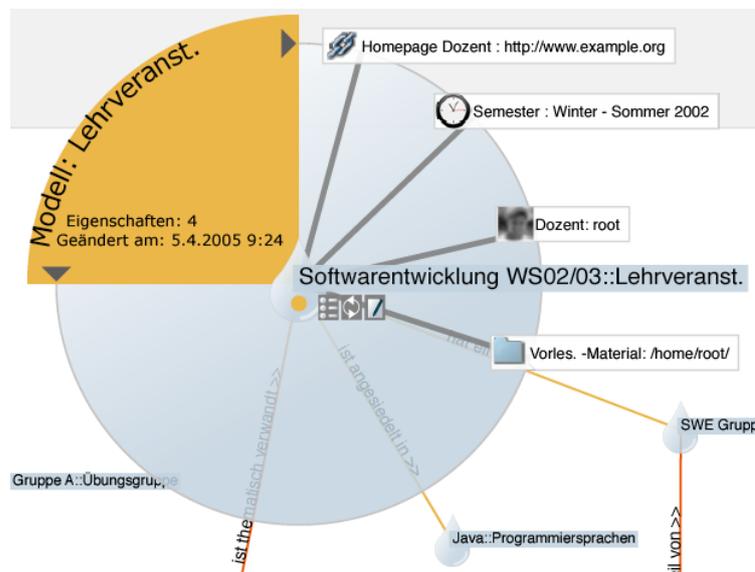


Abbildung 5.5: Eigenschaften des Individuums

zu externen Dokumenten in einer unterscheidbaren Darstellung angeboten. Je nach Einsatz und Voraussetzungen kann eine Typisierung um spezifische Elemente erweitert werden. So ist es z.B. denkbar für den universitären Einsatz einen Typen *Buchnummer* zu definieren, der eine spezifische Nummer eines Buches direkt mit dem Katalog einer Universitätsbibliothek verknüpft.

In Bezug auf die allgemeine Ansicht der Tropfen kann die Anzahl der Individuen mitunter sehr groß sein. Es ist darum unerlässlich, die dargestellten Elemente

nach speziellen Kriterien filtern zu können. Dieses Filtern kann auf verschiedene Art und Weise geschehen. Neben der Filterung nach Modellzugehörigkeit kann auch eine Differenzierung zwischen bestimmten Assoziationen erfolgen. Hierbei kann zwischen generellen und individuen-spezifischen Assoziationen unterschieden werden. Als Beispiel wäre eine generelle Filterung die Assoziation *Professor* → *arbeitet zusammen mit* → *Mitarbeiter*, wobei hier alle Individuen die diese Assoziation erfüllen angezeigt würden. Ein Beispiel für eine spezifische Assoziation wäre *Meier* → *arbeitet zusammen mit* → *Schulz*, dies würde die Filterung auf die Entitäten *Meier* und *Schulz* beschränken und nicht die Gesamtheit der Individuen, die über diese Assoziation verbunden sind. Beide Herangehensweisen bieten Vor- und Nachteile. Während die generelle Filterung in einer Listenansicht überschaubarer ist als die spezielle Betrachtung jedes einzelnen Individuums, bietet die spezielle Filterung eine niedrigere Anzahl möglicher *Tropfen*, was einer besseren Übersicht zugute kommen würde. Hierbei könnte eine Erweiterung der Filterung durch eine stichwortbasierte Suche erfolgen.

Aufgrund der derzeitigen Spezifikation des SVG-Standards (vgl. SVG Working Group, 2003), der weder Texteingabe noch Listen unterstützt, muss diese Funktionalität aber speziell nachgebaut werden, was komplexe Abfragemasken, wie sie in HTML-Seiten der Fall sind, praktisch ausschließt. Für weitere Informationen bezüglich dieser Thematik sei an dieser Stelle auf die Kapitel 5.1.9 und 6 verwiesen. Um trotzdem die Basisfunktionalität einer Filterung anbieten zu können wird in

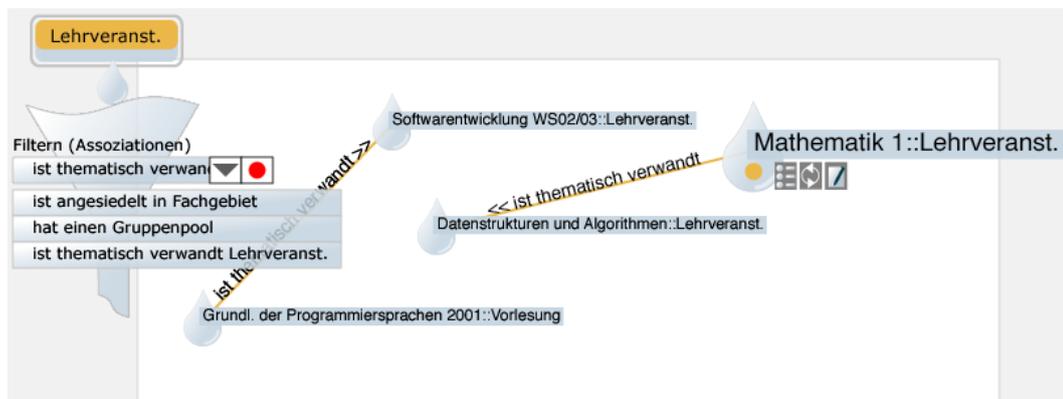


Abbildung 5.6: Filterung der Tropfen nach Assoziationen

der Nutzeranwendung der Ansatz der generellen Filterung angeboten. Basierend auf einem Ausklappmenü kann somit nach den einzelnen Assoziationen aus den jeweiligen Modellen gefiltert werden. Am Beispiel der Abbildungen 5.4 und 5.6 würde die Auswahl der Assoziation *ist thematisch verwandt* die Tropfen *Softwareentwicklung*, *Grundl. d. Programmiersprachen*, *Mathematik 1* und *Datenstrukturen und Algorithmen* als einzig passende Individuen anzeigen bzw. filtern (vgl. Abbildung 5.6). Durch Anklicken des roten Punktes in der Filterliste kann der Nutzer die derzeitige Filterung aufheben.

5.1.3 Awareness

Um dem Nutzer einen Überblick darüber zu geben, wer an dem Prozess der Wissensstrukturierung beteiligt ist und was sich über einen bestimmten Zeitraum verändert hat, wird in der Applikation ein Protokollfenster zur Verfügung gestellt.

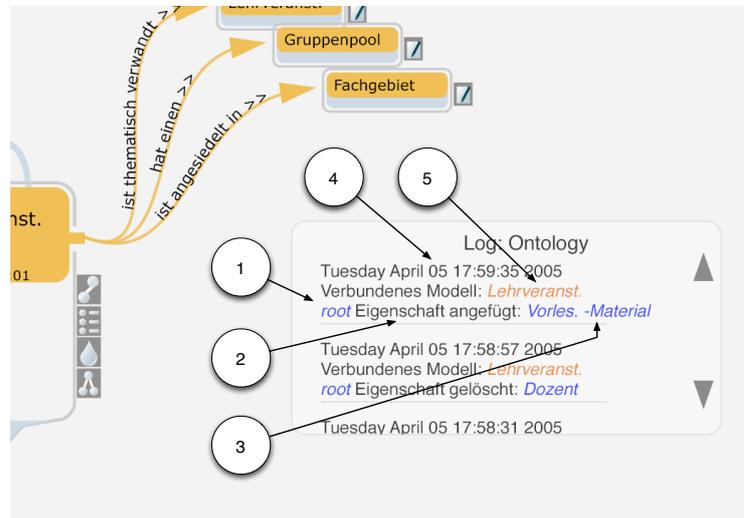


Abbildung 5.7: Protokollfenster mit Liste der ausgeführten Aktionen

Dieses Protokollfenster spezifiziert die in Kapitel 3.1.8 beschriebenen Informationen (vgl. Abbildung 5.7). Das Protokoll gibt darüber Auskunft, wer eine Aktion initiiert hat (1), wann (4) und welche Aktion (2) durchgeführt wurde und auf welche Entität oder Beziehung sich diese Änderungen beziehen (3). Unter zu Hilfenahme von Verweisen (1,3,5) können hiermit sowohl Informationen zu den mit einer Aktion in Verbindung gebrachten Personen abgerufen als auch alle Annotationen und Kommentare einer geänderten Entität bzw. Beziehung angezeigt werden (3). So bewirkt eine Nutzeraktion (Mausklick) auf den Nutzer-Verweis (siehe Verweis *root*,1), dass alle Aktionen der betreffenden Person angezeigt werden und somit auch verdeutlicht werden kann, welches Engagement dieser Nutzer in der Ontologie hat. Eine weitere Verbindung mit dem im ^{open}sTeam System befindlichen Heimatordner des Nutzers, bietet die Möglichkeit direkte Informationen zu der betreffenden Person (z.B. Kontaktmöglichkeiten) zu erhalten. Ferner wird zu jeder Aktion ein entsprechender Modell-Kontext angezeigt (5), auf den sich die entsprechende Aktion auswirkt. Eine Nutzeraktion auf diesen Verweis bewirkt, dass das jeweilige Modell in den sichtbaren Fokus rückt.

Die Verknüpfung auf die von einer Änderung betroffenen Elemente der Ontologie, erlaubt es dem Nutzer einerseits über eine direkte und übergreifende Navigation zum relevanten Modell und andererseits über einen Zugriff auf alle Annotationen des bearbeiteten Elements zu verfügen. Ferner signalisiert dieses Protokoll auch neue Annotationen und Kommentare, die im Kontext einer bestimmten En-

tität oder Assoziation angefügt wurden. Auch hier wird eine direkte Verknüpfung zu den jeweiligen Modellen und den enthaltenen Kommentaren angeboten. Die Diskussion verläuft somit nicht nur beiläufig, sondern ist integraler Bestandteil der kooperativen Zusammenarbeit.

5.1.4 Anlegen einer Ontologie

Eine Ontologie kann im Kontext des `openTeam`-Servers wie eine normale Datei angelegt werden. Über die Listenansicht innerhalb der Benutzungsoberfläche wird für Dateien mit der Endung `*.owl` (zur Bedeutung dieses Suffixes sei an dieser Stelle auf Kapitel 5.2.1 verwiesen) und dem Dateityp `text/xml` eine spezielle Aktion angeboten, mit der die Ontologie in der hier beschriebenen Anwendung editiert und angezeigt werden kann (vgl. Abbildung 5.8). Das Anbieten bzw. Anzeigen

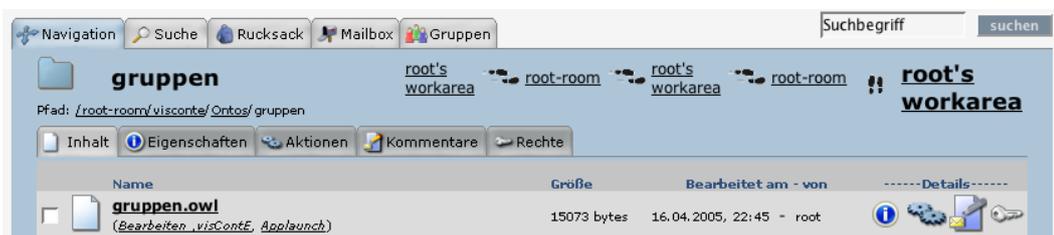


Abbildung 5.8: Aufruf der Ontologie aus der `openTeam`-Weboberfläche

dieser Aktion ist ein zentraler Punkt für die kooperationsgestützte Bearbeitung einer Ontologie. Dies begründet sich zum einen dadurch, dass diese Aktion das Bindeglied zwischen der Weboberfläche des `openTeam`-Servers und der grafischen Anwendung ist, und zum anderen in der Möglichkeit für den Nutzer eine Ontologie formell als Datei bzw. Objekt in dem jeweiligen Container bzw. Verzeichnis zu verwalten. Durch diese Möglichkeit ergibt sich auch der in Kapitel 3.1.3 angesprochene Transport der Ontologie durch den Austausch der Wissensstruktur unter den Nutzern des Systems. Die in Kapitel 3.1.5 angesprochene Formulierung des Schlüsselkonzeptes bzw. die Beschreibung des Inhaltes der Ontologie können über die Annotationsfunktionen der `openTeam`-Weboberfläche erfolgen. Dies ermöglicht den Nutzern bzw. Autoren sich schnell und direkt über den intendierten Inhalt einer Ontologie zu informieren.

5.1.5 Editierung

Basierend auf der in Kapitel 5.1.1 beschriebenen Visualisierungs-Metapher ergibt sich durch die Gruppierung der einzelnen Elemente als grafische Einheit (z.B. Eigenschaften), auch die Möglichkeit bestimmte Aktionen für diese Gruppierungen zu definieren. Diese Gruppierungen sind als auswählbare Aktionsknöpfe bzw. Symbole dem jeweiligen Element zugeordnet (vgl. Abbildung 5.2). Jedes grafische Symbol stellt gewisse zusammengefügte Aktionen bereit, die eine logische Gruppe bil-

5 visContE - Eine Umgebung zur Erstellung kooperativer Ontologien

den. Bei einer Aktivierung der jeweiligen Symbole erhält der Nutzer einen weiteren Auswahldialog, in dem die für diese Gruppe relevanten Aktionen ausgewählt werden können. Für das aktuell selektierte Modell in der drei Ebenen Darstellung sind vier relevante Aktionsgruppen wie folgt definiert.

- *Assoziations-Aktionen*: Anlegen, Ändern und Löschen einzelner ausgehender Assoziationen.
- *Eigenschafts-Aktionen*: Anlegen, Ändern und Löschen einzelner Eigenschaften, sowie die Veränderung des Modellnamens.
- *Individuum-Aktionen*: Anlegen, Ändern und Löschen der auf dem derzeitigen Modell basierenden Individuen.
- *Hierarchie-Aktionen*: Ableiten einzelner Modelle sowie Verschieben und Löschen des derzeitigen Modells (inklusive aller gebundenen Individuen, Eigenschaften und Assoziationen) mit oder ohne die jeweiligen Folgemodelle.

Zusätzlich zu den Aktionen mit Bezug auf das Modell, werden die einzelnen Tropfen ebenfalls mit einem Aktionssymbol dargestellt. Die möglichen Aktionen beschränken sich jedoch auf das Ändern und Löschen eines Individuums. Der ei-

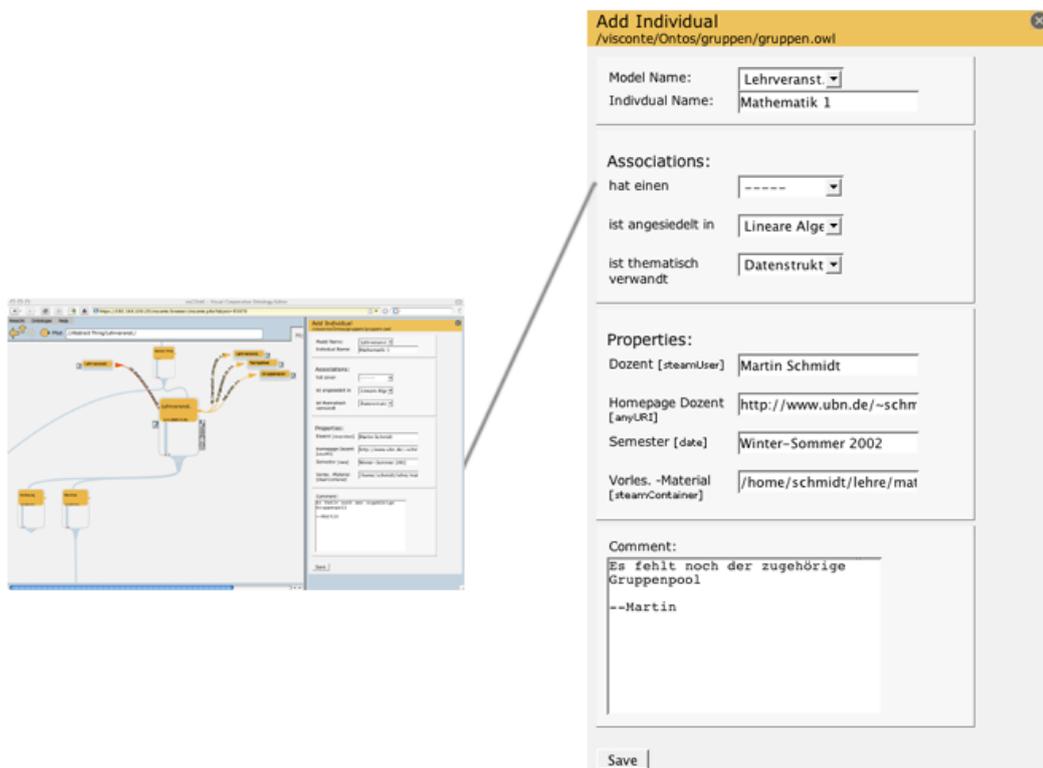


Abbildung 5.9: Bearbeitungsmaske für das Hinzufügen eines Individuums

gentliche Editiervorgang wird durch den Einsatz von HTML-basierten Abfrage-Formularen durchgeführt. Durch die Gruppierung der möglichen Aktionen für einen bestimmten Bereich, sind diese Formulare bezüglich der abgefragten Daten sehr übersichtlich. Für das Anlegen eines Individuums wird beispielsweise ein Formular angeboten, in dem das als Vorlage dienende Modell ausgewählt und ein Name für das Individuum eingegeben werden kann, die jeweiligen Assoziationen und Eigenschaften (basierend auf dem Vorlage-Modell) benannt werden und abschließend ein Kommentar angefügt werden kann. Während Eigenschaften vom Autor direkt (als Text) eingegeben werden können (es erfolgt eine namentliche Nennung des geforderten Datentypes, wie z.B. eine URL), werden die Assoziationen als Auswahllisten dargestellt. Diese Auswahllisten beinhalten jedoch nicht alle in der Ontologie vorhandenen Individuen, sondern nur die, die aufgrund des Assoziationszieles in Frage kommen. Abbildung 5.9 zeigt den Aufbau eines solchen Formulars.

Aufgrund des geführten Dialoges werden nur diejenigen Aktionen bereitgestellt, die auch im jeweiligen Kontext möglich sind. Die Ausführung einer Aktion – wie z.B. das in Abbildung 5.9 dargestellte Anlegen eines Individuums – wird nach Absendung des Formulars durch den Nutzer von dem in Kapitel 5.2 vorgestellten Server Modul übernommen. Nach einer Prüfung der eingegebenen Daten erstellt das Server-Modul entweder eine neue Version der Ontologie, die dann für alle Nutzer die aktuellste Version darstellt, oder signalisiert dem Nutzer einen Fehler. Das Ausführen einer beliebigen Aktion spiegelt sich somit in dem in Abschnitt 5.1.3 beschriebenen Protokollfenster als auch in der automatischen Aktualisierung der Ontologie in den momentan aktiven Anwendungen wider. Über dies hinaus beginnt ab diesem Punkt, aufgrund der Möglichkeit jeder Aktion einen Kommentar bzw. eine Annotation beizufügen, die Phase der Elementbasierten Diskussion. Das Ausführen einer Aktion ist somit der erste Schritt hin zu einer konsens-basierten Ontologie (vgl. 3.1.6), dem Auslösen einer gegenseitigen Wahrnehmung und der Etablierung der Nachvollziehbarkeit einer Ontologie.

5.1.6 Diskussion

Wie oben beschrieben, bildet die Ausführung einer Editierfunktion den ersten Schritt im diskursiven Aufbau der Ontologie. Basierend auf diesem ersten Schritt, bietet sich dem Nutzer für jedes Element in der visuellen Darstellung die Möglichkeit eine Annotation bzw. einen Kommentar zu hinterlegen. Ähnlich der Symbolik für das Editieren hat jedes Element eine eigene Kommentar-Schaltfläche (vgl. z.B. Abbildung 5.2, Punkt 6), über die die jeweiligen elementbasierten Annotationen abgerufen werden können. Neben der Anzeige der zeitlichen Abfolge der hinterlegten Kommentare, kann der Nutzer auch einen eigenen Kommentar für das Element hinzufügen (vgl. Abbildung 5.10) Aufgrund der Zuordnung von Kommentar zu Nutzer lassen sich noch weitere Informationen abfragen. So ist es z.B. möglich gezielt alle Kommentare eines bestimmten Autors nachzuvollziehen um so spezielle Sichtweisen gezielter einordnen zu können.



Abbildung 5.10: Elementbasierte Annotationen

5.1.7 Orientierung und Bedienungselemente

Mit der Integration und Adaption von Funktions- und Medienelementen soll der Zugang im Sinne von Navigation, Orientierung und Bedienung für den Nutzer verbessert werden. Mit Bedacht auf die weite Verbreitung von Internetbrowsern und um dem zukünftigen Nutzer einen leichteren Umgang mit der Applikation als Ganzes zu ermöglichen, wurden viele Interaktionselemente derzeitiger Browser-Anwendungen direkt in die Oberfläche der Nutzer-Applikation integriert. Der gesamte visuelle Rahmen der Struktur beruht auf dem Konzept eines *Webrowsers*. Im Folgenden werden die einzelnen Elemente und deren Funktionen mit Bezug auf Abbildung 5.11 detailliert beschrieben.

Registerkarten

Die Visualisierung verschiedener Sichten mit Hilfe von Registerkarten bzw. engl. *Tabs* (Abb. 5.11,1) bietet den Vorteil, die Anzahl frei schwebender Fenster bzw. Sichten zu minimieren. Des Weiteren ergibt sich für den Nutzer die Möglichkeit schnell und ohne Fensterwechsel zwischen verschiedenen Ansichten zu springen. In Bezug auf die Applikation bedeutet dies, zwischen der Repräsentation der Übersicht (Treemap) und der lokalen Modellhierarchie zu springen.

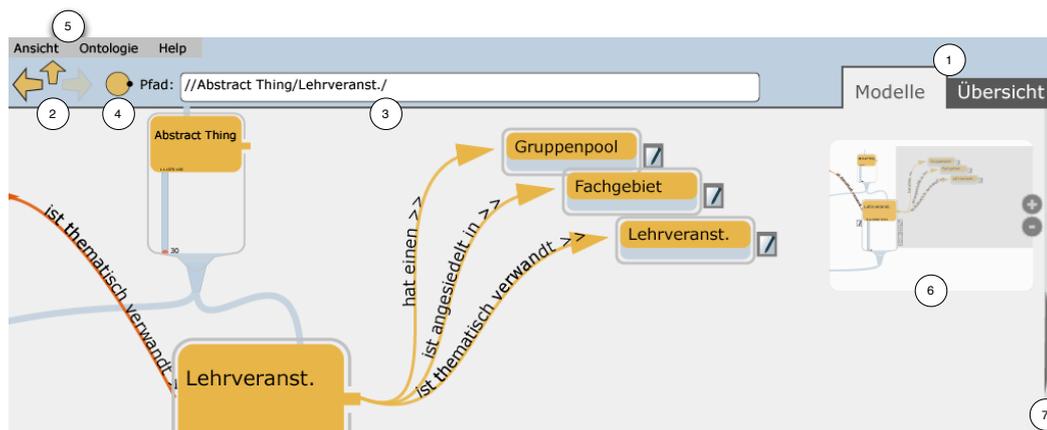


Abbildung 5.11: Navigations- und Bedienelemente der grafischen Anwendung

Chronik

Ähnlich dem Verlaufsspeicher besuchter Webseiten in einem Internetbrowser, bietet die Client Applikation auch die Möglichkeit bereits besuchte Modelle bzw. Konzepte mit Hilfe einer Vor- und Zurück-Navigation (Abb. 5.11,2) ein weiteres Mal anzusteuern (vgl. Utting und Yaneklovich, 1989). Der Einsatz dieser Technik unterstützt einerseits die Orientierung des Nutzers und andererseits eine reversible Navigation. Zusätzlich kann mit dem nach oben zeigenden Pfeil die Modellstruktur um eine Ebene nach oben navigiert werden.

Adressleiste

Um die Orientierung in den verschiedenen Ansichten weiter zu unterstützen, stellt die omnipräsente Adressleiste (Abb. 5.11,3) Informationen über die derzeitige Position innerhalb der Hierarchie bereit. Mit der Darstellung eines virtuellen Pfades zum aktuell betrachteten Modell lässt sich zum einen die derzeitige Tiefe der Position in der Hierarchie erkennen, und zum anderen (dies ist eine Erweiterung zu den bekannten Adressleisten), sind die einzelnen Pfadelemente per Mausaktion aktivierbar und erweitern so die Navigationsmöglichkeiten innerhalb der Hierarchie.

Aktualisierung

Ebenfalls dem Konzept der Webbrowser entliehen ist das Kennzeichnen eines momentan stattfindenden Ladevorgangs. Hiermit wird dem Nutzer über eine Animation (Abb. 5.11,4) verdeutlicht, dass in der Anwendung eine Datenübertragung stattfindet. Des Weiteren kann über eine Nutzeraktion auf diesem Bedienungselement die Ontologie erneut vom Server abgefragt werden. Um eine stetige Abgleichung mit der aktuellen Ontologie zu gewährleisten, ist dieser Prozess aber auch

automatisiert vorhanden, wobei die Ontologie bei einer neuen Version automatisch neu geladen wird.

Menü

Um eine weitere Form der Interaktion zu ermöglichen wurde auch das Konzept der ausklappbaren Menüs (Abb. 5.11,5) in die Benutzungsschnittstelle integriert. Dieses Menü kann mit Hilfe der externen Konfigurationsdatei (vgl. Kapitel 5.1.8) leicht angepasst werden. So ist es möglich zum einen diverse Interaktionselemente ein- bzw. auszuschalten, und zum anderen definierte Funktionen oder externe Referenzen in der Anwendung gezielt aufzurufen. Dies betrifft beispielsweise das Ein- bzw. Ausblenden des oben genannten Protokollfensters oder die nachfolgend beschriebene Vergrößerung.

Vergrößerung

Zusätzlich wird eine verkleinerte Darstellung der aktiven Sicht angeboten (Abb. 5.11,6). In dieser Darstellung ist es sowohl möglich den angezeigten Inhalt zu vergrößern und zu verkleinern, als auch, dem Vergrößerungsgrad entsprechend, mit einer rechteckigen Auswahl innerhalb der Anzeige zu navigieren.

Scrollbalken

Aufgrund der unbeschränkten Anzahl möglicher Assoziationen und abgeleiteter Modelle ist es erforderlich auch außerhalb des sichtbaren Bereiches die Visualisierung fortzusetzen. Um diese nicht sichtbaren Strukturen anzusteuern wird für die einzelnen Sichten eine horizontale Navigation (Abb. 5.11,7) angeboten.

Rahmen

Aufgrund der freien Positionierung der Tropfen in der Individuen-Ansicht, kann sich der Nutzer nie vollkommen sicher sein, einen kompletten Überblick über die Gesamtheit der Individuen zu erhalten. Um diesen Überblick jedoch zu gewährleisten wird ein zum Hintergrund farblich abgesetztes Rechteck (vgl. hierzu das weiße Rechteck in Abbildung 5.4) eingesetzt, das die flächige Ausprägung der Tropfen begrenzt. Hierbei bilden die am weitesten oben-, rechts-, links- und untenstehenden Symbole jeweils die Ränder des Rechtecks. Ferner ermöglicht dieses Rechteck durch eine Verschiebeoperation die Individuen bzw. Tropfen in ihrer Gesamtheit neu zu positionieren.

5.1.8 Konfiguration und Internationalisierung

Die Anwendung kann zusätzlich über eine externe Konfigurationsdatei modifiziert und erweitert werden. In dieser Datei kann – abweichend von der Standardsprache

Englisch – jedes Benutzungselement mit einer neuen Bezeichnung versehen werden. Hiermit wird gewährleistet, dass die Anwendung äquivalent zum `openTeam-Server international` und an die Nutzer angepasst verwendet werden kann. Überdies hinaus können in dieser Konfiguration auch, wie oben erwähnt, angepasste Menüeinträge definiert werden, die den ausführbaren Programmcode in Form von JavaScript² beinhalten. Dies ermöglicht es sowohl anwendungsinterne als auch externe Referenzen (wie z.B. anstuern von URLs) aus dem Menü aufzurufen. Der folgende Ausschnitt zeigt die Menüdefinition der in Abbildung 5.11 dargestellten Menüstruktur *Ansicht*.

```
<menu id="view" label="Ansicht">
  <item id="zoom" label="Vergrößerung" action="magnify();" />
  <item id="log" label="Log" action="toggleView('log.window');" />
</menu>
```

Die Attributreferenzen *label* (Name des Menüeintrags) und *action* (enthält den Programmcode) können in einem *item*-Element in der Reihenfolge beschrieben werden, wie sie im dargestellten Menü abgebildet werden sollen. Eine mehrfache Definition von *menu* Elementen ist möglich. Diese werden – wie in der Darstellung 5.11 zu sehen – nebeneinander positioniert.

5.1.9 Technische Umsetzung - Ein Überblick

Im Folgenden wird ein Überblick über die technische Realisierung der Nutzeranwendung gegeben. Die weite Verbreitung von Internetbrowsern und deren Potenzial mit Hilfe von integrierbaren Hilfsprogrammen bzw. Plug-Ins³ komplette Anwendungen auszuführen ermöglicht es, netzbasierte und systemübergreifende Applikationen einem großen Publikum zugänglich zu machen. Von dieser Möglichkeit macht das hier beschriebene Werkzeug Gebrauch. Primär basiert dieses Werkzeug auf dem vom W3C⁴ definierten Standard für Skalierbare Vektor Grafiken kurz SVG (vgl. SVG Working Group, 2003) zur Visualisierung, der Programmiersprache JavaScript als ausführende Schicht, sowie einem Plug-In zur Gesamtausführung der Nutzerapplikation. Hauptkriterien für die Wahl dieser Komposition sind die relativ einfache Ausführung der Anwendung innerhalb eines Internetbrowsers, sowie die auf XML basierende Struktur der Skalierbaren Vektorgrafiken die von Server und Client gleichermaßen modifiziert werden kann. Hierbei macht die Nutzeranwendung von der Möglichkeit zur Manipulation des SVG-Formats durch Java-Script bzw. dem Document Object Model⁵(vgl. Le Hors u. a., 2000) zur

²JavaScript ist eine objektbasierte Programmiersprache zur dynamischen und clientseitigen Manipulation von Metasprachen wie z.B. HTML oder SVG

³Plug-Ins - von engl. *einstecken* - beschreibt ein Hilfsprogramm für eine andere Anwendung

⁴W3C steht für World Wide Web Consortium - Organisation die über Webstandards wie z.B. XML oder HTML wacht

⁵Das Document Object Model, kurz DOM bildet intern die Baumstruktur einer XML-Datei ab und ermöglicht so die Manipulation der einzelnen Knoten durch eine clientseitige Programmiersprache wie z.B. Java-Script

Laufzeit Gebrauch.

Entwurfsmuster

Der Implementierung der Client-Anwendung wurde das *Model View Controll* Entwurfsmuster, kurz (MVC) (vgl. Balzert, 1999), zu Grunde gelegt. Hierbei werden die einzelnen Elemente der Ontologie in einem definierten Datenmodell (Model) beschrieben. Auf diese Datenmodelle greifen die zur Visualisierung notwendigen Kontrollstrukturen (Controll) zu. Diese Kontrollstrukturen haben die Aufgabe, basierend auf den Informationen aus den Datenmodellen, die Visualisierung dem jeweiligen Kontext anzupassen. Hierbei stellt die SVG-Visualisierung in der Funktion als Präsentationsschicht (View) Vorlagen bereit, die von der Kontrollstruktur manipuliert und für den jeweiligen Kontext erweitert und kopiert werden. Exemplarisch hierfür ist der Verlauf des Prozesses zur Darstellung der *Drei Ebenen Hierarchie*. Ausgelöst durch eine Nutzeraktion verläuft die Neudarstellung der einzelnen Grafiken folgendermaßen: Das Vatermodell sowie das aktuell fokussierte Modell, stehen als statische Grafiken zur Verfügung, die aufgrund der im Datenmodell repräsentierten Informationen wie z.B. Name, Anzahl der Individuen oder Änderungsdatum eine jeweilige Anpassung erfahren. Anders verhält es sich für die Erzeugung der Kindmodelle und Assoziationen. Basierend auf Vorlagen bzw.

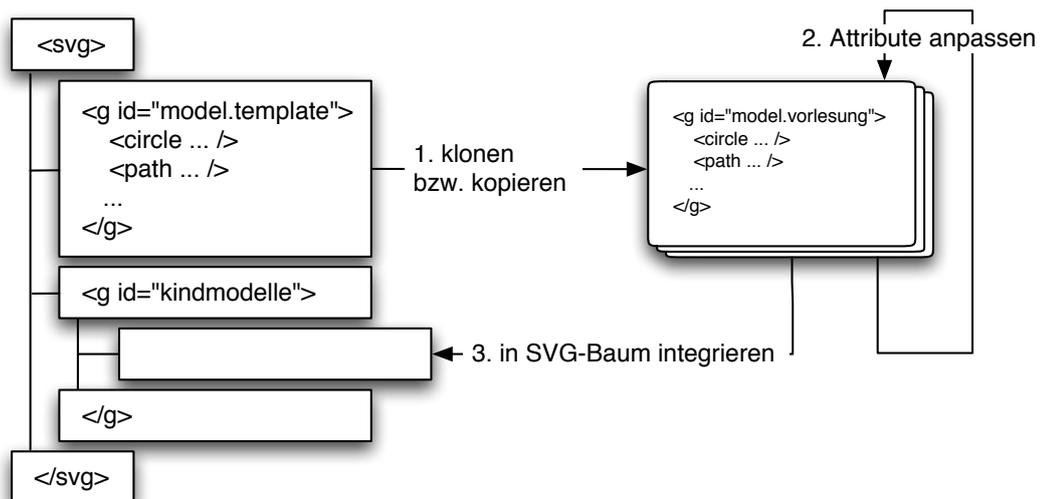


Abbildung 5.12: Genereller Ablauf der dynamischen Grafikerzeugung

Templates werden die benötigten Grafiken dynamisch erstellt. Bei diesen Templates handelt es sich um gruppierte Grafiken (vgl. das Auszeichnungselement `<g>` in Abbildung 5.12) und Elemente wie z.B. Rechtecke, Kreise, Pfade und Text, die zusammen z.B. die Repräsentation eines Modells (inklusive Reservoir, Etikett etc.) bilden. Da es sich bei diesen Vorlagen ausschließlich um XML-spezifizierte Daten handelt, werden von diesen Templates so genannte Klone erstellt, die eine Kopie

des jeweiligen XML-Unterbaums darstellen. Durch die Modifikation der Attribute wie z.B. id, Position, Name oder Verweisziel werden diese Klone an eine definierte Position des internen SVG- bzw. XML-Datenbaums angehängt. Abbildung 5.12 stellt diesen generellen Ablauf dar.

Alternativ könnte die Erzeugung der resultierenden Darstellung auch auf Seiten des Servers stattfinden und mit Hilfe der im nächsten Abschnitt beschriebenen Serververbindung an die Client-Applikation übermittelt werden. Aufgrund der resultierenden höheren Serverbelastung und dem zunehmenden Datenverkehr wurde diese Alternative jedoch nicht berücksichtigt.

Serveranbindung

Mit dem erweiterten Funktionssatz des eingesetzten – von Adobe⁶ entwickelten – SVG-Anzeigeprogramms (Version 3.0)⁷ ist es möglich direkte Anfragen an einen Server zu stellen ohne die gesamte Anwendung im Browser erneut auszuführen bzw. zu laden. Hiermit ist die webgestützte Anwendung in sich geschlossen und kann unabhängig von dem umgebenden Internetbrowser agieren. Diese Funktion kommt unter anderem der manuellen und automatischen Aktualisierung der Ontologie zu Gute. Hierbei wird in einem kurzen Intervall die aktuelle Versionsnummer der Ontologie erfragt. Eine Änderung dieser Versionsnummer hat zur Folge, dass sich zum einen die Ontologie und zum anderen die dargestellten Informationen im Protokollfenster aktualisieren.

Der Austausch zwischen Server und Client wird durch ein speziell angepasstes XML-Format gewährleistet. Dieses entwickelte Format erfüllt zwei Funktionen. Zum einen sind hierbei die Daten und Informationen in soweit aufbereitet, dass – anders als im eigentlichen Ontologieformat – der Client semantisch verbundene Entitäten nicht mehr nachvollziehen, sondern – aufgrund von gruppierten Blöcken – mit minimalem Aufwand darstellen kann. Auch hier wird ein weiteres Mal zwischen Modell- und Individuen-Ebene unterschieden, was sich in der Trennung der spezifischen Daten widerspiegelt. Während die Hierarchie als komplettes Datenpaket vom Server übermittelt wird, werden die Individuen nicht in ihrer Gesamtheit, sondern gruppiert nach Modellzugehörigkeit oder Assoziation vom Client angefragt. Dieses Vorgehen vermindert das zu übertragende Datenvolumen, da die Anzahl der Individuen die der Modelle um ein Vielfaches übersteigen kann. Eine genaue Beschreibung des Formates findet sich in Kapitel 5.2.4.

Anzeigefenster

Für die Anzeige der Editierfunktionen, der Kommentare und sonstiger verbundener Inhalte (vgl. Abbildung 5.9) kommt der Einsatz eines so genannten Framesets

⁶Adobe ist ein auf Grafikprodukte spezialisierter Softwarehersteller

⁷Ein Überblick befindet sich unter <http://www.adobe.com/svg/viewer/install/main.html>

zum Einsatz. Dieses Frameset teilt das vorhandene Browserfenster in zwei Sichtbereiche auf, die unabhängig voneinander agieren können. Diese Eigenschaft ermöglicht es, die Visualisierung der Ontologie in der linken Sicht unverändert zu lassen und beispielsweise die Editierfunktionen auf der rechten Seite als dynamischen Client-Server Dialog anzubieten. Die Einblendung des rechten Sichtbereiches geschieht jeweils nach einer Nutzeraktion auf den entsprechenden Aktionsknöpfen innerhalb der grafischen Darstellung. Wird der Editierbereich vom Nutzer nicht weiter benötigt, z.B. wenn die Aktion beendet ist, kann über das rechts-oben angebrachte *Fenster schließen*-Symbol der Dialog wieder ausgeblendet werden.

Auf den alternativen Einsatz von externen Dialogfenstern wurde hierbei speziell verzichtet. Dies begründet sich hauptsächlich durch die Tatsache, dass die Größe der bereitgestellten Dialoge sehr unterschiedlich sein kann, was zu einem Verdecken eines Großteils der Anwendung führen kann. Dieser visuelle Kontext ist aber nötig, um z.B. beim Anlegen einer Assoziation die entsprechenden Entitäten auch grafisch nachvollziehen zu können.

5.2 Einbettung in ^{open}sTeam

Der organisatorische Teil der Umgebungen bildet das für den ^{open}sTeam-Server entwickelte Modul. Dieses Modul bildet zum einen das Pendant zu den auf der Client-Seite möglichen Aktionen, wie Anlegen, Löschen und Editieren und zum anderen organisiert es die einzelnen Versionen und Annotationen einer Ontologie. Darüber hinaus werden einzelne Abfragestrukturen angeboten, die sowohl einzelne Elemente, als auch spezielle Kontexte an die aufrufende Anwendung zurückliefern können. Im Folgenden werden nun Aufbau und Konzept dieses Moduls eingehend betrachtet.

5.2.1 Datenformat für Ontologien

Bevor der Modulaufbau im Einzelnen beschrieben wird müssen, für ein besseres Verständnis, zunächst das Datenformat der Ontologie, sowie dessen interne Strukturen und Beziehungen erläutert werden. Die Ontologie wird in Form des vom World Wide Web Consortium definierten Standards OWL (vgl. Smith u. a., 2004) aufgebaut. Ausgangspunkt der Web Ontology Language (OWL) ist das für die Beschreibung von Ressourcen definierte und ebenfalls XML-basierte Resource Description Framework (RDF) (vgl. Frank Manola and Eric Miller, 2004). Im Folgenden sollen die einzelnen Konzepte dieses Ontologieformates näher präzisiert werden.

RDF und RDF Schema

Das Resource Description Framework (RDF) ist eine XML-basierte Beschreibungssprache für Informationen und Ressourcen im WWW. Diese Beschreibung basiert auf der Konstruktion von einfachen Assoziationen. Assoziationen werden mit Hilfe

so genannter Tripel aufgestellt. Ähnlich dem Satzbau in einer natürlichen Sprache finden hierbei die grammatikalischen Elemente Subjekt, Prädikat und Objekt als Tripel ihre Anwendung. Hierbei bildet das Prädikat – als verbindendes Element – die Assoziation zwischen zwei Elementen bzw. dem Subjekt und dem Objekt. Basierend auf diesem Konstrukt lassen sich gegenseitige Abhängigkeiten spezifizieren wie das folgende Beispiel zeigt. Die Aussage: *Die Webseite <http://www.example.org/index.html> hat den Autor Hans Schmidt* kann im Sinne des oben genannten Tripels in *<http://www.example.org/index.html> als Subjekt, hat den Autor* als Prädikat und *Hans Schmidt* als Objekt transformiert werden (vgl. Frank Manola and Eric Miller, 2004). Das Hauptkonzept in der Formulierung dieses Tripels bildet der Einsatz des Uniform Resource Identifiers (URI)⁸, der Subjekt, Prädikat und Objekt eindeutig beschreibt. Das Beispiel könnte mit der Technik der URIs folgendermaßen dargestellt werden:

- Subjekt: *<http://www.example.org/index.html>* - die Webseite
- Prädikat: *<http://purl.org/dc/elements/1.1/creator>* - diese eindeutige Adresse beschreibt das verbindende Element: *hat den Autor*
- Objekt: *<http://www.example.org/staffid/85740>* - die URI dient der eindeutigen Identifizierung der Person Hans Schmidt

Überführt in die RDF-Repräsentation beschreibt das Attribut *rdf:about* das Subjekt, das Element *dc:creator* das Prädikat und *rdf:resource* die Person bzw. das Objekt.

```
<rdf:Description rdf:about="http://www.example.org/index.html">  
  <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>  
</rdf:Description>
```

Hierbei gilt es zu beachten, dass diese Webadressen bzw. URIs nicht zwangsläufig auf existierende Dokumente innerhalb des Netzes verweisen müssen, sondern die Funktion einer eindeutigen Identifizierung übernehmen. Durch diese Identifizierung wird es möglich in anderen RDF-Repräsentationen – oft auch (engl.) *Statement* genannt – auf die jeweilige Ressource (in diesem Fall Hans Schmidt) zu verweisen. So ist es vorstellbar, dass z.B. ein weiteres Statement Hans Schmidt als Subjekt unter der Adressierung *<http://www.example.org/staffid/85740>* deklariert und über ein Prädikat mit einem weiteren Objekt verbindet. Eine Anwendung kann aufgrund der gleichen URIs schließen, dass mit dem Bezeichner *<http://www.example.org/staffid/85740>* immer die Ressource bzw. Person Hans Schmidt gemeint ist. Das nachfolgende Statement besagt: *Hans Schmidt kennt Elke Meier*.

```
<rdf:Description rdf:about="http://www.example.org/staffid/85740">  
  <meinVokabular:kennt rdf:resource="http://www.fiktiv.moc/elkeM"/>  
</rdf:Description>
```

⁸URI - engl. Uniform Resource Identifier, dient zur eindeutigen Bezeichnung einer abstrakten oder realen Ressource

Dabei beschreibt die nicht *physisch* existierende Adresse `http://www.fiktio.moc/elkeM` die Person Elke Meier eindeutig. Eine wichtige Stellung nimmt auch das Prädikat ein. Das oben genannte Prädikat `dc:creator` verknüpft die Webseite mit dem Autor und basiert ebenfalls – in einer verkürzten Schreibweise – auf einer URI (`http://purl.org/dc/elements/1.1/creator`) die ein spezielles Vokabular definiert. Durch Einsatz eines solchen Vokabulars – vorausgesetzt die Bedeutung von *creator* ist einer speziellen Anwendung bekannt – ist es möglich beliebige Metainformationen und Beziehungen zu erfassen. Beispiel für ein solches Vokabular ist der so genannte Dublin-Core⁹, der unter anderem auch das oben beschriebene Prädikat `dc:creator` spezifiziert. Dieses Vokabular definiert Eigenschaften für die Darstellung von Metainformationen von Webdokumenten, wie z.B. Autor, Erstellungszeitraum, Sprache etc.

Zur Definition von speziellen Zusammenhängen im Kontext der Ontologie wird das Konzept von RDF durch den Einsatz der *RDF Vocabulary Description Language* bzw. RDF Schema (RDFS) erweitert (vgl. Dan Brickley and R.V. Guha, 2004). RDFS erlaubt es z.B. einem ontologischen Modell mit dem Ausdruck `rdfs:subClassOf` eine Spezialisierung zu einem anderen Modell zu beschreiben. Der Einsatz der beiden Konzepte RDF und RDFS ermöglicht es über einen einheitlichen Standard für das Beschreiben von etwaigen Ressourcen oder Begriffen (jedes Vokabular besteht aus RDF-Statements und wird wiederum in RDF-Statements integriert) zu verfügen und zum anderen Vokabulare (wie z.B. der oben genannte DublinCore) frei zu definieren. Der Vorteil dieses Konzeptes begründet sich hierbei in der – vom W3C als Empfehlung definierten – Beschreibung der Ressource (RDF) und des Vokabulars (RDFS), die jedem weiteren Vokabular (z.B. Dublin-Core) als Ausgangsbasis dienen.

OWL - Web Ontology Language

Der ebenfalls vom World Wide Web Consortium definierte Standard der Web Ontology Language (OWL) (vgl. Smith u. a., 2004) basiert auf dem Konzept von RDF und RDFS. RDF Schema beinhaltet die für taxonomische Strukturen nötige Ableitung, wie z.B. das Anlegen von Unter- bzw. Submodellen (`rdfs:subClassOf`). Diese Eigenschaften dienen als Ausgangsbasis für die Konstruktion bzw. Definition von Ontologien. Generell sind für die in Kapitel 2.2 beschriebenen Entitäten und Beziehungen (Modell, Individuen etc.) folgende Strukturen vorgesehen und sollen am Beispiel der Oper *La Traviata* kurz beschrieben werden:

- Modell:

```
<owl:Class rdf:about="#Oper">
  <rdfs:label>Oper</rdfs:label>
  <rdfs:subClassOf rdf:resource="#Musikauffuehrung"/>
</owl:Class>
```

⁹Dublin-Core bezeichnet ein Vokabular zur Beschreibung von Web-Dokumenten wie z.B. Autor oder Erstellungsdatum

Das Element *owl:Class* beschreibt eine Begrifflichkeit bzw. ein Modell und enthält gegebenenfalls eine Referenz (*subclassOf*) zu dem abstrakteren Modell bzw. – im Bezug auf eine mehrfache Vererbung – den abstrakteren Modellen. Das mit *rdf:about* gekennzeichnete Modell *Oper* ist somit eine Spezialisierung des Modells *Musikauffuehrung*. Ferner ist zu beachten, dass die Schreibweise *#Oper* eine URI auf das aktuelle Dokument widerspiegelt. Ist z.B. die Ontologie unter die Adresse <http://www.example.org/KunstOntologie> abgelegt, bedeutet die Kurzform *#Oper* das gleiche wie <http://www.example.org/KunstOntologie#Oper>. Dies erklärt auch, dass Modell, Individuum, Assoziation und Eigenschaft namentlich nur einmal in einer Ontologie auftreten können. Zusätzlich kann über das Auszeichnungselement *rdfs:label* auch eine für Menschen lesbare Bezeichnung in das Statement integriert werden.

- Assoziation:

```
<owl:ObjectProperty rdf:about="#komponiertVon">
  <rdfs:label>komponiert von</rdfs:label>
  <rdfs:domain rdf:resource="#Musikauffuehrung" />
  <rdfs:range rdf:resource="#Komponist" />
</owl:ObjectProperty>
```

Das Element *owl:ObjectProperty* beschreibt eine Assoziation zweier Modelle bzw. Klassen. Hierbei kommen die in RDFS spezifizierten Elemente *domain* (als Assoziationsquelle) und *range* (als Assoziationsziel) zum Einsatz. Dabei gilt zu beachten, dass die Assoziationen nicht direkt mit dem Modell bzw. mit der Klasse verbunden sind, sondern nur eine Aussage darüber getroffen wird, welche Modelle über diese Assoziation verbunden werden.

- Eigenschaft:

```
<owl:DatatypeProperty rdf:about="#Urauffuehrung">
  <rdfs:label>Uraufführung</rdfs:label>
  <rdfs:domain rdf:resource="#Oper" />
  <rdfs:range rdf:resource="&xsd;Date" />
</owl:DatatypeProperty>
```

Anders als das *owl:ObjectProperty*-Element definiert das *owl:DatatypeProperty*-Element eine Eigenschaft, die sich nicht auf ein anderes Modell bezieht, sondern auf einen bestimmten Wert. Während sich die Intention von *rdfs:domain* zu der Verwendung in *owl:ObjectProperty* nicht unterscheidet, zielt das Prädikat *rdfs:range* auf eine Definition von simplen Datentypen (wie z.B. *Date*). Die Schreibweise *&xsd;Date* ist eine verkürzte Schreibweise und äquivalent zu der URI <http://www.w3.org/2001/XMLSchema#Date>¹⁰. Die *Urauffuehrung* verbindet somit eine Oper mit einem Datum.

¹⁰Die OWL-Spezifikation empfiehlt den Einsatz von XSD (XML Schema Definition) zur Beschreibung von Datentypen. Weitere Informationen zu den unterstützten Datentypen finden sich unter Biron und Malhotra (2004)

- Individuum:

```
<Oper rdf: about ="#LaTraviata" Urauffuehrung="Mar, 6 1853">
  <rdfs:label>La Traviata</rdfs:label>
  <komponiertVon rdf:resource="#Verdi" />
</Oper>
```

Basierend auf der im Modell festgelegten Begrifflichkeit und den definierten Assoziationen und Eigenschaften des Modells *Oper* lässt sich nun das sogenannte Individuum oder auch die Instanz einer Klasse erzeugen. Durch das Auszeichnungselement *Oper* wird definiert, dass dieses Individuum vom Modell *Oper* und somit ebenfalls von *Musikauffuehrung* abgeleitet ist und den Namen *La Traviata* trägt. Ferner kann aufgrund der Definition der Beziehung *Musikauffuehrung* → *komponiert von* → *Komponisten* das Individuum *La Traviata* mit dem an anderer Stelle definierten Individuum *Verdi* verbunden werden. Gleiches gilt für die Definition der Eigenschaft *Urauffuehrung*, die einer *Oper* ein Datum zuweist.

Die angeführte Liste zeigt die generelle Struktur einer Ontologie im OWL-Format. Erweiterungen, wie z.B. die Beschränkungen von Relationen bzw. Beziehungen und Eigenschaften (transitiv, symmetrisch etc.), sind ebenso in diesem Konzept enthalten wie die Möglichkeit Ableitungen von Assoziationen zu spezifizieren (vgl. Smith u. a., 2004) oder Gleichheit bzw. Unterschiede von bestimmten Modellen bzw. Klassen zu definieren. Ein Beispiel für eine symmetrische Relation ist die Assoziation *Peter ist verwandt mit Hans*, wobei Hans ebenfalls mit Peter verwandt ist. Bezüglich der Ableitungen von Assoziationen sei an dieser Stelle erneut auf das in Kapitel 2.2 erläuterte Beispiel, *ist Vater von* als Ableitung von *ist Elter von*, verwiesen. Ähnlich der Ableitung eines Modells, kann durch die Angabe des Attributes *rdfs:subPropertyOf* in der Beschreibung einer Assoziation (*ObjectProperty*) oder Eigenschaft (*DatatypeProperty*) diese Spezialisierung angegeben werden. Ferner können zwei Modelle wie z.B. *Kapelle* und *Musikgruppe* über das Prädikat *equivalentClass* als gleichartig beschrieben werden. In gleicher Weise bestimmt das Prädikat *differentFrom* den Unterschied zwischen zwei Modellen.

Eine weitere Unterscheidung erfährt die Web Ontology Language in der jeweiligen Ausprägung der Komplexität. Es wird zwischen den aufeinander aufbauenden Konstrukten OWL-Lite, OWL-DL (Description Logic) und OWL-Full unterschieden (vgl. McGuinness und van Harmelen, 2004). Während es mit den komplexeren Sprachen OWL-DL oder OWL-Full z.B. möglich ist komplexe Klassen in Form von mengentheoretischen Verhältnissen (Differenz, Vereinigung, etc.) zu formulieren, ist OWL-Lite gezielt für Hierarchien und einfache Beziehungen konzipiert. Aufgrund der verminderten Komplexität und der Möglichkeit zur Erweiterung zu den höher angesiedelten Sprachen OWL-DL oder OWL-Full, bildet OWL-Lite die Ausgangsbasis des im Folgenden beschriebenen Server Moduls.

5.2.2 Modul Aufbau

Wie der `openSTeam`-Server selbst, basiert auch das Modul auf der objektorientierten Programmiersprache Pike¹¹. Durch die Unterstützung von RDF und OWL stellt Pike die nötigen Basisfunktionen für Aufbau und Abfrage einer Ontologie bereit. Das entwickelte Modul erweitert diese Funktionen durch die Integration einer weiteren Schicht. Diese Schicht bildet die in Kapitel 5.2.1 beschriebenen Elemente als Objekt-orientierte Datenstruktur ab. Anders als mit den oben genannten Basisfunktionen wird es somit möglich, die einzelnen Elemente als direkte Objekte in einen Anwendungskontext zu integrieren. Eine dritte Schicht stellt mögliche Aktionen bereit, die mit Hilfe dieser Objekte die Ontologie modifizieren können. Basierend auf dem in Abbildung 5.13 dargestellten Aufbau, wird im Folgenden das Modul im Einzelnen beschrieben.

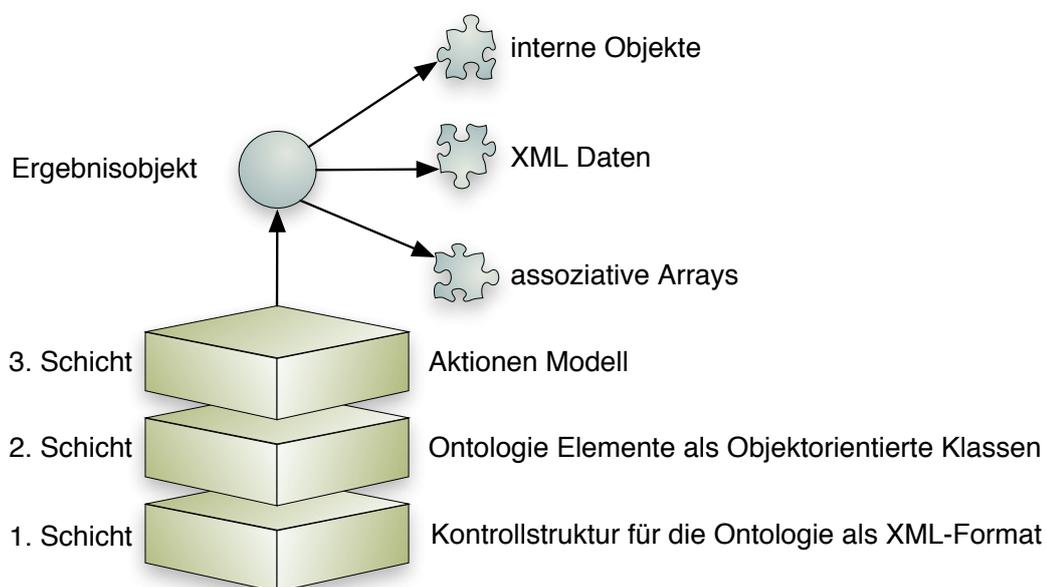


Abbildung 5.13: Die einzelnen Schichten des Servermoduls

Kontrollstruktur - Die erste Schicht

Die Kontrollstruktur ist das Bindeglied zwischen der Ontologie als XML- bzw. OWL-Datenstruktur und der höher angesiedelten zweiten (und vereinzelt auch dritten) Schicht. Basierend auf den von der Pike-API¹² (in der Version 7.6) bereitgestellten Funktionen definiert sich diese Schicht bzw. die Klassenstruktur *Onto-*

¹¹Informationen zu dieser Programmiersprache finden sich unter <http://pike.ida.liu.se/>

¹²Ein API (engl. Application Programming Interface) wird meist von einer Programmiersprache zur Verfügung gestellt und beinhaltet z.B. Schnittstellen, Funktionen oder auch Dienstprogramme

logy als Übersetzungs- und Abfrageinstrument zwischen den als XML repräsentierten Entitäten und Beziehungen auf der einen, und der objekt-orientierten Ent-

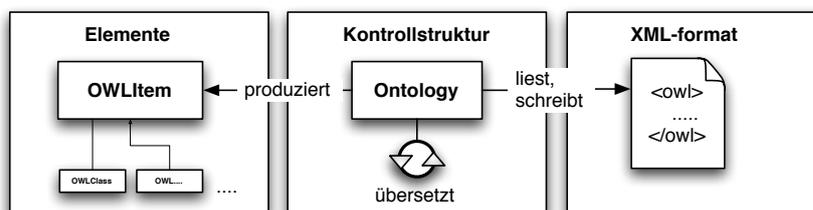


Abbildung 5.14: Übersetzungsfunktion der Kontrollstruktur

sprechung dieser Elemente (in der zweiten Schicht) auf der anderen Seite (vgl. Abbildung 5.14). Da der ^{open}sTeam-Server derzeit auf der Pike Version 7.4 aufbaut, wurde für eine reibungslose Integration der hier beschriebenen Umgebung die RDF- und OWL-Funktionalität aus der Pike Version 7.6 in das Server-Modul integriert und entsprechend angepasst. Die Übersetzung der Ontologie basiert auf der Abfrage von Subjekt, Prädikat und Objekt (vgl. 5.2.1) – über die vorhandenen RDF-Funktionen der Pike-Bibliothek – und der Vor- bzw. Rücktransformation in die entsprechenden Klassen der zweiten Schicht. Um diese Transformation zu verdeutlichen, sei ein weiteres Mal das Beispiel der *Oper* angeführt (vgl. 5.2.1):

```
<owl:Class rdf:about ="#Oper">
  <rdfs:label>Oper</rdfs:label>
  <rdfs:subClassOf rdf:resource ="#Musikauffuehrung" />
</owl:Class>
```

In diesem Auszug bildet *Oper* das Subjekt, *subClassOf* das Prädikat und *Musikauffuehrung* das Objekt. Wird nun das Modell *Oper* angefragt, kann die Kontrollstruktur mit dem vorhandenen Subjekt *Oper* und dem Prädikat *subClassOf* alle Elemente bzw. Statements herausfinden, die als Objekt zu dieser Anfrage in Betracht kommen. In diesem Fall kommt – aufgrund der einfachen Vererbungsstruktur – nur das Element *Musikauffuehrung* als Objekt in Frage. Die Formulierung einer Abfrage lässt sich jedoch beliebig umstellen und auch gezielt einschränken bzw. ausweiten. So würde eine Abfrage die z.B. nur das Objekt *Oper* beinhaltet, alle Sätze bzw. Statements wiedergeben, in dem dieses als Objekt fungiert (z.B. *Urauffuehrung*, *hat-Komponisten* oder alle Modelle die *Oper* weiter spezialisieren). Mit dieser Form der Abfrage bzw. dem gezielten Auslassen eines *Satzteiles* lassen sich somit sukzessive alle Attribute herleiten und z.B. ein Modell mit allen Beziehungen und Eigenschaften in die nachfolgende Datenstruktur überführen.

OWL Elemente - Die zweite Schicht

Modelle, Individuen, Assoziationen und Eigenschaften finden – als einzelne Klassen – ihre Entsprechung in der definierten Klassenstruktur, die in Abbildung 5.13

OWLClass: Diese Klasse entspricht dem *owl:Class*-Element der OWL-Lite Sprache bzw. einem Modell. Der einfachen Modellvererbung entsprechend enthält auch diese Klasse genau eine direkte Referenz – in Form einer *id* – auf das Obermodell das es spezialisiert. Des Weiteren bildet der Aufbau der Klasse die einzelnen Kontexte zu anderen Modellen sowie Individuen, Eigenschaften und Beziehungen ab. Jedes Modellobjekt beinhaltet neben den erwähnten Referenzen zum Obermodell auch jene zu den abgeleiteten Kindmodellen. Basierend auf diesen Referenzen bzw. *ids* können integrierte Funktionen somit rekursiv auf alle im Unterbaum vorhandenen Modelle als auch auf alle direkten Vorfahren des Modells zugreifen (vgl. Punkt 3 in Abb. 5.15). In gleicher Weise hat die Klasse auch Kenntnis über die in Beziehung stehenden Eigenschaften, Assoziationen und Individuen (vgl. Punkt 2,4 in Abb. 5.15). Instanzen von OWLClass haben somit Kenntnis über die Assoziationen (OWLObjectProperty) und Eigenschaften (OWLDatatypeProperty) in denen sie die Rolle der Quelle (*domain*) und/oder die des Zieles (*range*) einnehmen. Ähnliches gilt für die Referenzen auf die Individuen. In diesem Fall beinhaltet die Klasse Informationen bzw. Referenzen über die von sich selbst erzeugten Individuen (OWLIndividual) und kann mit Hilfe der oben genannten rekursiven Funktionen auf alle Individuen der Ober- und Untermodelle zugreifen.

OWLProperty: Diese Klasse bildet die gemeinsame Grundlage der zugehörigen Unterklassen OWLObjectProperty und OWLDatatypeProperty. OWLProperty hält hierbei das von beiden Unterklassen benötigte Modell bereit, das als Quelle (*domain*) einer Assoziation (OWLObjectProperty) oder einer Eigenschaft (OWLDatatypeProperty) dient (vgl. Punkt 4 in Abb. 5.15). Die eigentlichen Unterschiede bilden dabei – äquivalent zu den oben genannten RDF bzw. OWL Statements – die verschiedenen Ausprägungen des Zieles (*range*), das sich bei einem OWLObjectProperty durch ein Modell (vgl. Punkt 4a in Abb. 5.15) und bei OWLDatatypeProperty durch einen Datentyp definiert. Die Klasse OWLDatatypeProperty enthält eine Liste möglicher Datentypen, die unter anderem auch die in Kapitel 5.1.2 (Darstellung der Individuen) beschriebenen Verweise zu den ^{open}sTeam internen Dokumenten wie z.B. Container oder User definieren.

OWLIndividual: Repräsentiert ein abgeleitetes Individuum von einem Modell. Die Klasse OWLIndividual besitzt ähnlich wie OWLProperty eine Referenz zu der *id* des Modells, von dem es abgeleitet ist bzw. das als Vorlage dient (vgl. Punkt 1 in Abb. 5.15). Hieraus resultiert, dass sich jedes Individuum zu einer bestimmten Position innerhalb der Hierarchie selbst einordnen kann und ein Zugriff auf das als Vorlage dienende Modell zu jeder Zeit möglich ist. Individuen haben zudem Zugang zu den Eigenschaften und Assoziationen die vom Modell und dessen Obermodell(en) vorgegeben wurden (vgl. Punkt 5 in Abb. 5.15). Ein OWLIndividual-Objekt erhält somit einen direkten Zugriff auf die potenziell (sowohl eingehenden als auch ausgehenden) assoziierten

Individuen und Eigenschaftswerte. Möglich ist dies durch die gezielte Abfrage des Quell- und/oder Ziel-Modells aus den OWLProperty-Klassen, das wiederum über die einzelnen Individuen verfügt.

Aufgrund der Repräsentation von tatsächlichen Zusammenhängen auf Basis eines Modells, betrachtet die Klasse OWLIndividual Assoziationen nicht als Modelle, sondern als verbundene Individuen (vgl. Punkt 6 in Abb. 5.15). Durch die Referenz auf die jeweiligen OWLProperty Objekte ist die Klasse in der Lage übergebene Eigenschaften (als tatsächliche Werte) und Individuen (aufgrund der Assoziation) zu validieren. Überprüft wird im Fall der Assoziation, ob das Ziel-Modell des OWLObjectProperty-Objekts und das Ableitungsmodell des übergebenen OWLIndividual-Objekts übereinstimmen. Zum besseren Verständnis dieses Zusammenhangs soll ein weiteres Mal das Beispiel der *Oper* bemüht und etwas erweitert werden.

Als Erweiterung wird angenommen, dass das Modell *Komponist* eine Spezialisierung des Modells *Künstler* ist und das *Künstler* ebenfalls vom Modell *Maler* spezialisiert wird. Wird nun versucht dem Individuum *La Traviata* über die Assoziation *komponiert von* das Individuum *Rembrandt* (vom Modell *Maler*) zuzuweisen, kann aufgrund der Referenz auf das entsprechende Objekt der Klasse OWLObjectProperty geprüft werden, ob diese Zuweisung (vom Typ *Maler*) konform zu der Ziel-Assoziation (vom Typ *Komponist*) ist. Obwohl *Komponist* und *Maler* beide *Künstler* sind, wäre die Zuweisung von *Rembrandt* also falsch. Anders wäre der Sachverhalt, wenn das Individuum *Rembrandt* nicht von *Maler*, sondern direkt von *Künstler* abgeleitet wäre. Aufgrund der Vererbungslinie *Künstler* → *Komponist*, ist diese Zuweisung strukturell richtig, da *Rembrandt* nun die Rolle eines generellen *Künstlers* einnimmt.

OWLItemFactory: Diese Klasse stellt eine Hilfsklasse dar, die für die Konstruktion neuer Instanzen der Klasse OWLItem und deren Unterklassen, die in eine Ontologie integriert werden sollen, zuständig ist. Ferner prüft diese Klasse nach, ob die übergebenen Daten, wie z.B. Quelle und Ziel einer Assoziation, in der Ontologie existieren.

OWLContainer: Diese Hilfsklasse fungiert als Datenstruktur für die Aufnahme einer beliebigen Anzahl an OWLItem-Objekten.

Aufgrund der internen Referenzen bzw. Bezüge der einzelnen Klassen untereinander ergibt sich eine hochflexible Struktur, die es jeder Instanz dieser Klassen erlaubt auf direkte oder indirekte Weise auf den umgebenden Kontext zuzugreifen.

Aktionen Modell - Die dritte Schicht

Aufbauend auf diesen Verzweigungen und den denkbaren Editiermöglichkeiten einer Ontologie bietet das Modul eine weitere Schicht an. Diese dritte Schicht spiegelt die einzelnen Aktionen wider, die mit den Entitäten und Beziehungen einer

Ontologie möglich sind. Hierbei sind die möglichen Aktionstypen jeweils nach Elementeigenschaften wie z.B. dem Anlegen eines neuen Untermodells gestaffelt. Zu den oben genannten Klassen gibt es jeweils spezifische Aktionsklassen, die Funktionen für die Editierung einer Ontologie bereitstellen. So bietet die Klasse *ClassAction* beispielsweise Funktionen bzw. Methoden zum Anfügen, Editieren oder Verschieben von Modellen innerhalb der Ontologie an (Abbildung 5.16 benennt die wichtigsten Funktionen). Ferner stellen diese Aktionsklassen auch elementspezifische

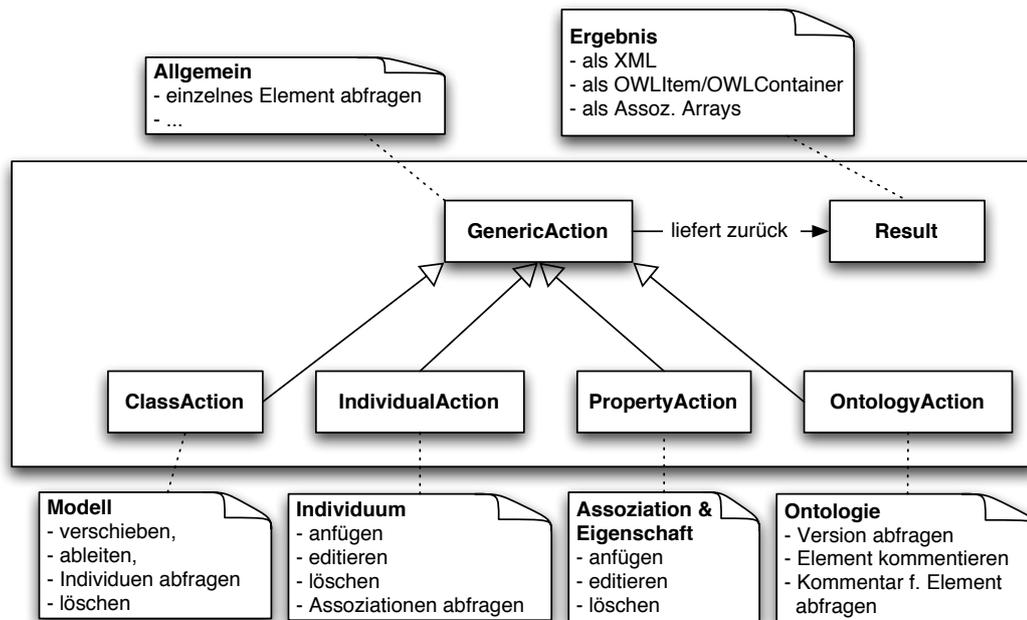


Abbildung 5.16: Aufbau der Klassenstruktur des Aktionen-Modells

sche Abfragestrukturen, wie z.B. für das Abfragen der Modellhierarchie ab einem gewissen Knoten oder das Abfragen von direkt verbundenen Individuen, bereit. Neben den in Kapitel 5.1.5 beschriebenen Aktionen für das Editieren (z.B. anfügen, löschen, ändern, anlegen einer Annotation), wird das Aktionen-Modell um die folgenden Abfragestrukturen erweitert.

- Abfragen einzelner Modell-, Individuum-, Beziehungs- und Eigenschafts-Elemente
- Abrufen der Modellhierarchie ab einem speziellen Modell
- Abfrage von Individuen eines bestimmten Modells, mit oder ohne den direkt assoziierten Individuen
- Abfrage aller Assoziationen (inklusive der geerbten) eines Modells
- Abfrage aller assoziierten Individuen die auf ein spezielles Individuum verweisen (sowohl vorwärts- als auch rückwärtsgerichtete Assoziationen)

Die bereitgestellten Funktionen ermöglichen somit sowohl das Abfragen von Kontexten, der gesamten Ontologie, sowie einzelner Elemente. Jede ausgeführte Aktion liefert ein spezielles Ergebnisobjekt (in der Klassenstruktur als Result bezeichnet) zurück. Dies ermöglicht es auf vielfältige Art und Weise mit den angelegten oder abgefragten Daten umzugehen. Das Ergebnisobjekt enthält die einzelnen Ontologie-Elemente als Objekte, die auf den oben genannten Klassen (OWLClass etc.) basieren. Auf Basis dieser Objekte können nun verschiedene Ergebnisse produziert werden. Neben dem Abruf der Objekte, ist es ferner möglich diese als XML Format abzufragen, das z.B. dem grafischen Client als Dateneingabe dient (vgl. 5.2.4), oder die Objekte in assoziative Arrays¹³ zu transformieren. Des Weiteren gibt jedes Ergebnisobjekt darüber Auskunft, ob die ausgeführte Aktion einen Fehler verursacht hat und was genau dieser Fehler bedeutet. Über definierte Fehlernummern lassen sich somit für die weitere Verarbeitung spezielle Fehler abfangen. Die so gewonnenen Daten lassen sich beliebig in neue Anwendungskontexte integrieren.

Als Alternative wäre es sicherlich möglich die Aktionsklassen direkt mit der ersten Schicht bzw. der Kontrollstruktur zu verbinden und somit die zweite Schicht zu verwerfen. Dies würde aber nicht nur die Weiterverarbeitung der Daten in anderen Anwendungskontexten massiv erschweren, sondern auch die von der zweiten Schicht bereitgestellten Möglichkeiten zur Abfrage einzelner Element-Abhängigkeiten in den Bereich der Aktionsklassen verschieben. Neben den oben angesprochenen Schwierigkeiten würde dies auch gleichzeitig bedeuten, dass die Notwendigkeit einer Kontrollstruktur nicht mehr gegeben ist und die derzeitige Applikationslogik und die Datenhaltung der Ontologie allein in der Aktionsschicht behandelt würde. Die Vorteile der hier vorgestellten Struktur werden somit umso klarer, bietet sie doch aufgrund der zweiten Schicht eine Abstraktionsebene die es ermöglicht auf einfache Art und Weise zu einem späteren Zeitpunkt neue oder erweiterte Aktionen hinzuzufügen.

5.2.3 Die Hilfsklasse ID

Um eine URI bei jeder Erzeugung eines neuen OWL-Elements auf unzulässige Zeichen zu überprüfen, wurde die Hilfsklasse ID in das Modul integriert. Diese Klasse ermöglicht es zudem die übergebenen Kurzformen der URIs zu verändern. Jedes OWLItem-Objekt beinhaltet, wie oben erwähnt, sowohl eine id bzw. abgekürzte URI (z.B. #Oper) als auch einen, für die spätere grafische Darstellung, lesbaren Namen (siehe *rdfs:label*) des Elements. Die URI muss also nicht zwangsläufig dem Namen gleichen, sondern kann einen beliebigen aber dennoch eindeutigen Schlüssel beinhalten. Genau davon macht diese Hilfsklasse Gebrauch. Anstatt des tatsächlichen Namens kann ein Hash-Wert¹⁴ als eindeutiger Schlüssel hinterlegt werden. Zudem kann hiermit auch die Beschränkung der namentlichen Eindeutigkeit (z.B.

¹³Datenstruktur für Mengen, wobei jedes Element einen eindeutigen Namen als Schlüssel erhält

¹⁴Ein Hash-Werte (auch Streuwert genannt) wird mit Hilfe einer Hash-Funktion ermittelt um z.B. eine Zeichenkette oder ein Objekt eindeutig identifizieren zu können

Oper) umgangen werden. Dies wird möglich wenn z.B. der Hash-Wert mit einem Zeitwert (z.B. die aktuelle Zeit zu der eine Aktion ausgeführt wird) verbunden wird. Der Einsatz dieser Klasse erlaubt somit einen freieren Umgang mit den URI's und zum anderen eine stetige Überprüfung der erlaubten Zeichen (z.B. keine Umlaute) eines übergebenen Elementnamens. Das nachfolgende Fragment zeigt, wie sich der Einsatz dieser Umformung auf das Modell *Oper* (oben) und das Individuum *La Traviata* (unten) auswirkt. Obwohl durch diese Umformungen die Bezüge für den Menschen schwer lesbar sind, hat sich der Bedeutungsgehalt zu den oben aufgezeigten Beispielen in keiner Weise geändert.

```
<owl:Class rdf:about='#visconte_item321120458'  
  <rdfs:label>Oper</rdfs:label>  
  <rdfs:subClassOf rdf:resource='#visconte_item838661824' />  
</owl:Class>  
  
<visconte_item321120458 rdf:about="#visconte_item2091844805">  
  <rdfs:label>La Traviata</rdfs:label>  
  <visconte_item239423342 rdf:resource="#visconte_item233423423" />  
</visconte_item321120458>
```

5.2.4 Client-Server Interaktion

Das oben vorgestellte Aktionen-Modell bildet den Anknüpfungspunkt für den web-basierten Client. Diese Anbindung erfolgt über so genannte – ebenfalls auf Pike basierende – Webskripte¹⁵, die die Funktion eines Mediators zwischen der webgestützten Anwendung und dem Server-Modul einnehmen. Basierend auf der Anfrage der Client-Anwendung initiiert das zuständige Webskript die jeweilige Aktion. Am Beispiel der Aktion *Modell ableiten*, wird in Abbildung 5.17 der Prozess dieser Aktion und das Zusammenspiel der einzelnen Schichten verdeutlicht. Für ein besseres Verständnis soll an dieser Stelle das Diagramm kurz kommentiert werden: Das Webskript erzeugt unter zu Hilfenahme des Dateinamens (der Ontologie), der Versionsnummer, eines (in 5.2.6 erklärten) Datei-Steuerung-Objekts (SteamStorageHandler) und dem jeweilig bearbeitenden Autor ein neues Ontologie Objekt. Je nach gewünschter Aktion wird eine spezielle Aktionenklasse instanziiert (in diesem Fall *ClassAction*), der das Ontology-Objekt (*ont*) übergeben wird. Basierend auf den übergebenen Parametern des Methodenaufrufs *subclass* wird mit Hilfe der Klasse *OWLItemFactory* ein neues Objekt vom Typ *OWLClass* konstruiert. Dabei wird der Methode *subclass* nicht die id bzw. die Kurzform der URI direkt übergeben, sondern ein Objekt der oben beschriebenen Hilfsklasse *ID*. Falls dieses neue Modell (in Bezug auf die URI) noch nicht vorhanden ist, wird es der Ontologie mit dem Aufruf *add_Item* hinzugefügt. Sollten etwaige Fehler in dieser Aktion aufgetaucht sein, wie z.B. das Modell existiert schon, oder das Vatermodell ist nicht existent, wird der Fehler in das Result-Objekt integriert. Sollte jedoch kein Fehler aufgetreten sein, wird automatisch die jeweilige Aktion mit dem verbundenen

¹⁵Webskripte übernehmen die direkte Kommunikation mit dem aufrufenden Webbrowser

Kommentar der später in Kapitel 5.2.5 beschriebenen Annotationen-Datei zugeführt. Schließlich wird das neue Modell als OWLClass-Objekt dem Ergebnisobjekt *result* übergeben, das dem aufrufenden Webskript zurückgeliefert wird. Neben dem Zugriff auf das spezifische Objekt und den assoziativen Arrays, kann nun das Webskript von dem Ergebnisobjekt auch als eine spezielle XML-Datenstruktur abgefragt werden.

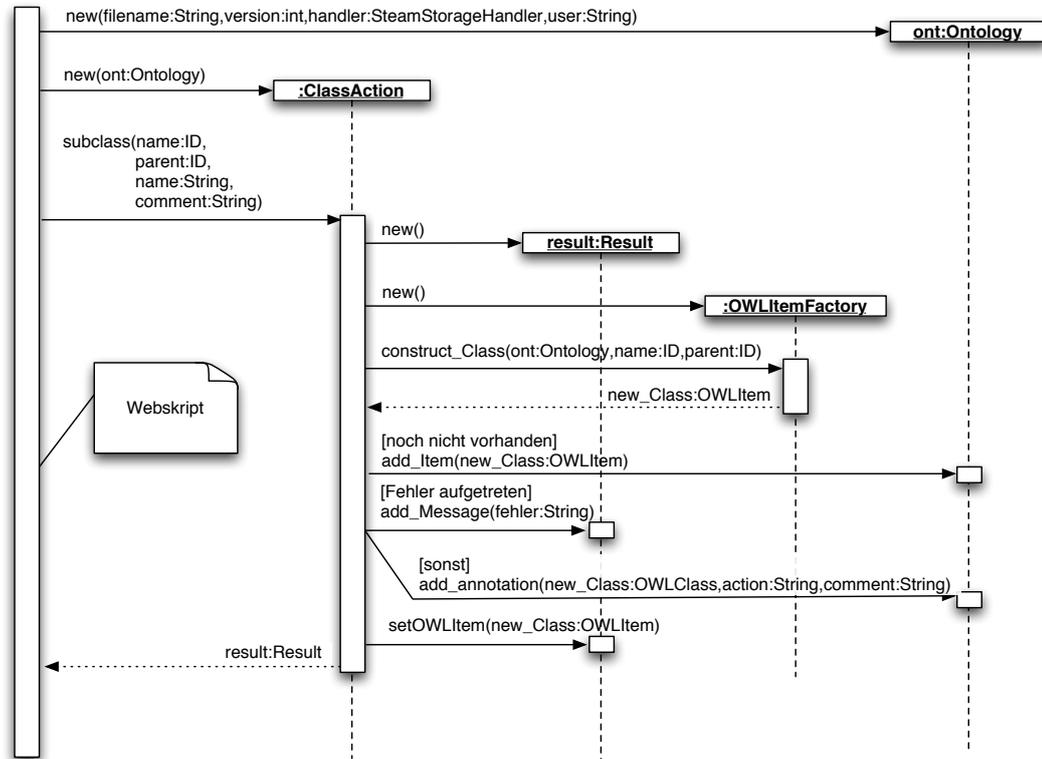


Abbildung 5.17: Sequenzdiagramm der Aktion *Modell ableiten bzw. subclass*

Die XML-Datenstruktur gliedert sich, wie oben erwähnt, in die Beschreibung der Modelle und Individuen. Die Modelle werden mit allen benötigten Eigenschaften versehen, die für eine Darstellung in der Client Anwendung nötig sind. Zudem findet die Modellhierarchie in der geschachtelten Struktur ihre Entsprechung und kann so direkt als Baumstruktur verarbeitet werden. Der folgende Ausschnitt aus dem Datenformat gibt einen Überblick wie die Baumstruktur abgebildet wird.

```
<model lastModified="" drops="" label="" weight="" id="">
  <model ... >
    <property ... type="" label="" id="" />
    <association ... domain="" label="" with="" id="" />
  </model>
</model>
```

Das Auszeichnungselement *model* beschreibt die Rahmendaten eines Modells. Im Einzelnen sind dies der Modellname (*label*), die eindeutige Identifikation (*id*) aus der Ontologie, das Änderungsdatum (*lastModified*) oder auch die Anzahl der abgeleiteten Individuen (*drops*) und die Gewichtung des gesamten Unterbaums (*weight*) des Modells.

Die von *model* eingeschlossenen Elemente *property* und *association* beschreiben die jeweils definierten Eigenschaften und Assoziationen des umgebenden Modells. Für das Element *association* ist sowohl ein Attribut *with* definiert, das auf die *id* des assoziierten Modells verweist, als auch ein Attribut *domain*, das nochmals die Quelle der Assoziation (als *id*) beinhaltet.

Die Repräsentation eines Individuums wird durch das Auszeichnungselement *individual* und den eingegliederten Elementen *attribute* beschrieben.

```
<individual lastModified="" domain="" label="" id="">
  <attribute range="" type="" label="" propertyId="" value="" />
</individual>
```

Die Attribute *lastModified*, *label* und *id* entsprechen den oben genannten Beschreibungen für ein Modell. Das Attribut *domain* verweist auf die eindeutige Identifikation (*id*) des als Vorlage dienenden Modells. Innerhalb des *individual Elements* werden die zugewiesenen Eigenschaften und Assoziationen des Individuums beschrieben. Das Element *attribute* dient hierbei sowohl der Beschreibung einer Assoziation, als auch der einer Eigenschaft. Unterschieden werden diese durch das Elementattribut *type*, das entweder den Wert *object* für eine Assoziation oder den jeweiligen Datentyp wie z.B. Text (*string*), oder eine URI (*anyURI*) etc. annehmen kann (vgl. Abschnitt: OWL - Web Ontology Language). Assoziierte Individuen oder eingesetzte Eigenschaften werden durch das Attribute *value* definiert. Die Attribute *type* und *value* beschreiben somit sowohl Art als auch Assoziationsziel bzw. Eigenschaftswert eines Individuums. Das Element Attribut *propertyId* verweist zudem auf die im Kontext des Modells definierten Eigenschaften oder Assoziationen.

Der Aufbau der Datenstruktur fasst – anders als das OWL Format – Elemente zu einer Gruppe zusammen, die einen gemeinsamen Kontext bilden. Die strukturelle Untergliederung in Modell und Individuum und die gegenseitigen Referenzen ermöglichen es, die einzelnen Ebenen sowohl getrennt als auch im Zusammenhang zu verarbeiten. Basierend auf diesem Datenformat ist es ferner möglich weitere webbasierte Anwendungen einerseits mit den Daten der Ontologie zu versorgen und andererseits – mit Hilfe der existierenden Schnittstelle der Webskripte – zu bearbeiten. Basierend auf diesem Format und der vom ^{open}sTeam-Server bereitgestellten XSLT¹⁶-Bibliothek ist es denkbar, die Ontologie z.B. in das HTML-Format zu transformieren und somit eine weitere Repräsentation bereitzustellen. Zum Teil wird dies bereits in der derzeitigen Anwendung realisiert. Die oben angesprochenen Webskripte werden – für die Darstellung der Editier-Dialoge – bereits mit dem

¹⁶XSLT engl. Extensible Stylesheet Language Transformation - wird benutzt für die Transformation bzw. Übersetzung zwischen verschiedenen XML-Formaten

beschriebenen XML-Format versorgt und generieren mit Hilfe der XSLT-Bibliothek z.B. die Formularelemente zur Erstellung eines Individuums.

5.2.5 Annotationen

Die Speicherung und Abfrage der Annotationen wird ebenfalls von der Klasse *Ontology* übernommen. Die Annotationen selbst werden in einer separat geführten XML-Datei aufgezeichnet, die z.B. Auskunft über Autor, geändertes Modell oder das Modifikationsdatum gibt. Basierend auf der jeweils ausgeführten Aktion, wird der Annotationsfunktion (*add_annotation*) auch der entsprechende Aktionstyp, wie z.B. *verschieben* oder *anfügen* (als Text bzw. String) mitgeteilt. Hieraus ergibt sich, dass die einzelnen Aktionen mit den angefügten Kommentaren eine Einheit bilden. Dies ermöglicht es, die Annotationen-Datei nach bestimmten Kriterien auszuwerten. Neben der Filterung nach Nutzern und Elementen können so auch die einzelnen Aktionen im grafischen Client als Änderungsprotokoll aufgeführt werden. Ferner ist es mit der oben bereits erwähnten XSLT-Bibliothek möglich bzw. denkbar, auch die Annotationen in ein anderes Format zu transformieren wie z.B. RSS. Hieraus ergibt sich die Möglichkeit außerhalb der grafischen Anwendung – z.B. im Emailprogramm – stetig über die Änderungen einer speziellen Ontologie unterrichtet zu werden.

5.2.6 Speicherung und Archivierung

Die Archivierung und Speicherung von Ontologie und Annotationen erfolgt durch spezielle Datei-Steuerungs Objekte. Um die Wiederverwendbarkeit des Moduls auch für Anwendungen außerhalb des *open'sTeam*-Servers zu gewährleisten, bilden diese Objekte die Schnittstelle zwischen Modul und Dateisystem der verwendeten Plattform. So ist es möglich, ohne das Modul zu ändern, eine jeweils angepasste Datei-Steuerungs Klasse zu verwenden. Diese Klasse muss die nötigen Funktionen zur Öffnung, Speicherung, Prüfung der Existenz und zur Archivierung einer Datei bereitstellen. Das Steuerungsobjekt wird bei Aufruf der Konstruktor-Funktion der *Ontology* Klasse übergeben, die dann lediglich die genannten Funktionen aufruft um die Daten der Ontologie einzulesen oder zu speichern.

Die Archivierung erfolgt in einer redundanten Speicherung der Ontologie. Hierbei wird die aktuellste Version bei der Konstruktion eines neuen *Ontology*-Objekts intern zwischengespeichert und nach einem expliziten Aufruf zur Speicherung von z.B. einem Webskript mit der jeweiligen Versions-Nummer als Datei-Suffix gespeichert. Die aktuelle Versionsnummer wird direkt in die Ontologie Struktur eingebunden und bei einem erneuten Aufruf der Ontologie wiederum von dort aus bestimmt.

6 Ausblick und Zusammenfassung

Diese Arbeit zeigt, dass das Konzept der Ontologie nicht zwangsläufig auf den Gebrauch einer maschinellen Erschließung – von der eigentlichen Intention des Semantic Webs abweichend – von Semantik beschränkt sein muss. Vielmehr wurde das Konzept der Ontologie diesem Wirkungskreis entzogen und unabhängig von der Frage inwieweit Ontologien in den Bereich der künstlichen Intelligenz hineinreichen, als eine Form der Wissensrepräsentation für den Austausch zwischen Menschen betrachtet. Wichtig – und dies wurde als eine der Grundthesen dieser Arbeit herausgestellt – ist der Mensch, oder im engeren Kontext der Wissensträger, der Wissen bzw. Sinnzusammenhänge als primären Schritt in den Computer überführt. Ferner wurde in dieser Arbeit ergründet, dass die Erschließung von neuem Wissen nicht ausschließlich auf einem Akt bloßer Rezeption, sondern vielmehr auf Kommunikation und Zusammenarbeit basiert. Folglich wurde die oben genannte Grundthese um eine kollaborative bzw. kooperative Komponente erweitert. Ausgehend von diesen Thesen wurde das Konzept der Ontologie sowohl in seiner Struktur als mögliche Form der Wissensrepräsentation untersucht, als auch notwendige Eigenschaften wie die eines Konsenses und der Nachvollziehbarkeit herausgearbeitet.

Mit der primären Annahme, Ontologien als semantische Wissensspeicher für eine kooperative Herangehensweise zu benutzen, wurden im weiteren Ansätze für eine mögliche Visualisierung dargelegt. Hierbei wurde die interne Struktur einer Ontologie eingehend analysiert und den einzelnen Zusammenhängen schrittweise eine potenzielle Darstellungsform zugewiesen. Die Visualisierung der Ontologie ist für das Verständnis der Wissenszusammenhänge und den Prozess einer kooperativen Erstellung von außerordentlicher Wichtigkeit. Als einheitliche Projektionsfläche der Wissensstruktur bildet die Darstellung den eigentlichen Kern des abstrakten Konzeptes der Ontologie. Neben dieser konzeptionellen Betrachtungsweise über Darstellung, Aufbau und Prozesse für den gemeinschaftlichen Aufbau (z.B. Findung eines Konsenses) wurde im Kapitel 4 auf Einsatzmöglichkeiten des ontologischen Konstrukts eingegangen. An dieser Stelle wurde deutlich, dass die abstrakte Struktur ausreichend Freiraum bietet verschiedenartige Konzepte und Sinnzusammenhänge zu konstruieren bzw. abzubilden. Anhand von Szenarien wurden verschiedene thematische und strukturelle Ausprägungen (z.B. Organigramme, Übungen, Systemstruktur) für den Einsatz in Lehr-Lernprozessen identifiziert, und damit auch die Stärken der Ontologie herausgearbeitet. Besonders der Verbund aus definierten Begrifflichkeiten, deren Zusammenhängen und der Verknüpfung mit externen Ressourcen erlauben es, z.B. Dokumente in beliebige semantische Zusammenhänge zu setzen. Ferner konnte auch verdeutlicht werden,

dass basierend auf den expliziten und von den Autoren definierten Zusammenhängen, auch implizite Verbindungen geschaffen werden, die ein nicht direkt sichtbares verwandtschaftliches Verhältnis zweier Individuen oder Modelle herstellen. Auch in diesem Zusammenhang spiegelt sich die Notwendigkeit einer grafischen Darstellung wider, die es einem zukünftigen Nutzer erlaubt, diese verdeckte Assoziation zu explorieren und über deren Sinn selbst zu entscheiden.

Es soll an dieser Stelle aber auch um die möglichen Schwächen dieses Konzeptes gehen. In erster Linie – und dies sollte eher als eine Herausforderung angesehen werden – geht es darum, dem zukünftigen Nutzer Funktion und Struktur dieses Konzeptes zu vermitteln. Nicht repräsentative Tests haben hierbei gezeigt, dass dies eine der Hauptschwierigkeiten darstellt. Während die Verknüpfung zweier Begrifflichkeiten bzw. Modelle noch als nachvollziehbar galt, wurde die Unterscheidung zwischen Modell und Individuum nicht sofort deutlich. Genau an dieser Schnittstelle betritt die Ontologie als informatorisches Konzept den Raum der Philosophie. Eine Unterscheidung zwischen Universalie bzw. Modell und Einzelding bzw. Individuum kann nicht immer zweifelsfrei erfolgen. Eine Abgrenzung definiert sich immer nur aufgrund persönlicher Sichtweisen auf das jeweilige Konzept. Diese unterschiedlichen Sichtweisen gilt es – ist das Konzept strukturell von den beteiligten Personen verstanden – mit Hilfe eines Konsenses auf eine gemeinsame Basis zu bringen. In diesem Zusammenhang muss auch die Visualisierung abgegrenzt werden. Primäre Funktion ist hier die Verdeutlichung der ontologischen Struktur in einer sinnvollen Abgrenzung der einzelnen Elemente. Zur Findung des Konsenses kann die jeweilige Visualisierung oder auch Anwendung nur als Mediator zwischen den einzelnen Autoren fungieren und muss folglich für diese Vermittlung funktionell ausgelegt sein.

Aufgrund des selbstorganisierten Aufbaus der Ontologie auf der Nutzerseite muss dieses Konzept auch noch weiteren Kriterien standhalten. Zunächst gilt es dem zukünftigen Nutzer zu vermitteln, welchen Mehrwert die Strukturierung in Form einer Ontologie für ihn persönlich birgt. Die Vorteile für z.B. kleine bis mittelgroße Gruppen¹ zur Strukturierung der erzeugten Dokumente und Konzepte bzw. Begriffe sind sicherlich am einfachsten zu vermitteln, da z.B. die Struktur des vermittelten Wissens einer Vorlesung in der Ontologie seine Entsprechung finden kann. Um die Arbeitsintensität für den Aufbau einer Ontologie zu minimieren ist es überdies vorstellbar gewisse Ontologie-Vorlagen (z.B. eine Taxonomie von Dokumentenarten) bereit zu halten, die vom Server automatisch mit den tatsächlich vorhandenen Dokumenten einer Gruppe oder eines Containers gefüllt werden.

Schwerer hingegen wird die Verdeutlichung des Mehrwertes für einen Nutzer in Bezug auf eine globale bzw. systemweite Ontologie. Hier wird ein Großteil der Nutzer sicherlich nur ein begrenztes Engagement zeigen. Einen vorstellbaren Anreiz stellt hier – unabhängig von der gewählten Thematik der Ontologie – die Vorgehensweise dar, einen Bezug zu den im System beheimateten Nutzern in die

¹Im universitären Einsatz sind hierbei 3-4 Personen für eine Kleingruppe (z.B. Übungsgruppe einer Vorlesung) oder 10-15 Personen für eine mittelgroße Gruppe (z.B. Blockseminar) denkbar

Struktur zu integrieren. Damit wird jedem Nutzer bzw. Autor die Möglichkeit gegeben sich selbst in der Ontologie zu positionieren und so Gemeinsamkeiten oder Unterschiede zu anderen Nutzern zu ersehen. Dies bedeutet aber auch, dass die Ontologie entweder von vornherein diese Nutzerkomponente bietet oder im Verlauf der Konstruktion dahingehend angepasst wird. Ein weiterer Aspekt, der an dieser Stelle besonders betont werden sollte, ist der Umgang mit den persönlichen Daten eines Nutzers. Besonders im Falle einer globalen Ontologie muss es jedem Nutzer gestattet sein persönliche Daten selbstbestimmt in die Ontologie einzusetzen. Die oben genannte Integration der Nutzerkomponente sollte also lediglich das abstrakte Modell definieren und – obwohl dies automatisiert machbar erscheint – die einzelnen Nutzer nicht von vornherein als Individuen bzw. Instanzen integrieren. Ferner, und dies leitet zu dem technischen Ausblick über, ist auch zu prüfen, wie überschaubar eine Ontologie ab einem bestimmten Größenverhältnis (Anzahl der Elemente) sein wird.

Technik

Um dies zu beurteilen bedarf es zunächst einer Evaluation, die insbesondere die Client-Anwendung bewertet. Neben der Bewertung der Übersichtlichkeit der dargestellten Struktur und der Navigation ist es auch wichtig, die gewählte Visualisierungs-Metapher auf ihre Nachvollziehbarkeit hin zu überprüfen. In diesem Zusammenhang muss dem Nutzer insbesondere die Vererbungsbeziehung der Modelle und die Abhängigkeit bzw. Zusammengehörigkeit von Individuum und Modell klar werden. Ebenso interessant ist es zu erfahren, inwiefern der Treemap-Ansatz als Navigationsinstrument dem Nutzer dabei behilflich sein kann die Hierarchie zu ergründen.

Die Nutzeranwendung bietet derzeit alle Grundfunktionen zum Anlegen, Ändern und Löschen bestimmter Elemente und Assoziationen. Hauptdefizit im Bezug auf das Anlegen von Individuen ist die Beschränkung der Kardinalität von entweder keiner oder einer Assoziation. Diese bedeutet, dass ein Individuum eine Assoziation nicht mehrmals, sondern nur einmal benutzen kann. Diese Beschränkung ist auf die derzeitige Unterstützung von OWL-Lite bzw. der so genannten *lightweight Ontologien*² des in 5.2 beschriebenen Server-Moduls zurückzuführen. Auch hier müssen Einsatztests zeigen, ob diese Einschränkung den Aufbau einer Wissensstruktur massiv behindert und ob daraufhin das Server-Modul gegebenenfalls erweitert werden muss. Weitere Entwicklungen könnten überdies die direkte Manipulation der grafischen Objekte erweitern. Vorstellbar ist, von der bestehenden Anwendung ausgehend, eine Adaption des in *open'sTeam* verwendeten Ansatzes des *Rucksacks*, der Referenzen zu selektierten Elementen aufnimmt. Diese Funktionalität ließe zahlreiche *Drag and Drop*³ Aktionen zur weiteren grafischen Gestal-

²lightweight Ontologien sind in der Komplexität beschränkte Ontologien

³Drag and Drop beschreibt das ziehen bzw. verschieben und fallenlassen von grafischen Elementen mit Hilfe einer Mausektion

tung der Ontologie zu und könnte auch die Editier-Dialoge weiter vereinfachen.

Eine weitere Einschränkung der direkten Editierung begründet sich in der fehlenden Möglichkeit zur Texteingabe innerhalb einer SVG gestützten Anwendung. Zwar bietet das SVG-Plugin von Adobe – vom derzeitigen SVG-Standard abweichend – die Möglichkeit auf Tastatureingaben zu reagieren, erfordert aber auch einen kompletten Nachbau von beispielsweise Textfeldern oder der Funktionalität einer Rücktaste. Die für Mai 2005 geplante Verabschiedung eines neuen SVG-Standards (vgl. SVG Working Group, 2004) wird unter anderem auch diese Lücke schließen. Es muss also abgewartet werden, inwiefern und wie schnell dieser neue Standard durch geeignete Anzeigeprogramme nutzbar wird. Die jüngsten Entwicklungen der Browser-Technologie zeigen, dass es zunehmend Bestrebungen gibt SVG direkt und ohne ein entsprechendes Plug-In anzeigen zu können (vgl. Mozilla Organization), was eine Mischung verschiedener XML-Formate wie z.B. HTML und SVG ermöglichen wird. Aus heutiger Sicht ist das für die Client-Anwendung nötige Plug-In von Adobe die beste Alternative zur Anzeige von SVG. Dies begründet sich zum einen in der fast kompletten Unterstützung des definierten SVG-Standards und zum anderen durch die Verfügbarkeit für die meisten Betriebssysteme.

Das Server-Modul entspricht dem Entwicklungsstand der Benutzer-Applikation und bietet somit alle Funktionen zur Editierung der einzelnen Elemente. Aufgrund der differenzierten Abfragemöglichkeiten und dem vorgestellten XML-Format lassen sich die Daten auch in andere Programmkontexte einbetten. In diesem Zusammenhang wäre es durchaus vorstellbar die definierten Sinnzusammenhänge auch in die Webschnittstelle des ^{open}sTeam-Systems zu integrieren. Konkret könnten Dokumente, die in einer bestimmten Ontologie eine definierte Rolle einnehmen und aufgrund einer Assoziation mit anderen Dokumenten einen Verbund bilden, in der Weboberfläche verknüpft werden und somit einen Bezug zur Ontologie bilden. In gleicher Weise ist es denkbar, die in der Ontologie definierten Zusammenhänge für eine stichwortbasierte Suche zu nutzen. In diesem Fall bietet die Ontologie einen Vorteil gegenüber einer normalen Suchfunktion, da es möglich ist den umgebenden Kontext in der Ergebnisliste mit aufzuführen. Diese Funktion muss zukünftig gleichermaßen in der webbasierten Anwendung und im Server-Modul integriert werden. Dies wird ebenfalls, wie oben beschrieben, von den zukünftigen Entwicklungen im Bereich der Skalierbaren Vektorgrafiken (SVG) abhängen.

Das Server-Modul deckt zu einem großen Teil die durch das OWL-Lite-Format vorgegebenen Strukturen ab. Die im Kapitel 5.2.1 aufgeführten Erweiterungen zur Beschränkung und Spezialisierung von Assoziation und Eigenschaft und dem spezifizieren von Gleich- bzw. Ungleichheit zweier Modelle sind in dem Modul bisher nur teilweise unterstützt und wurden zu Gunsten der einfacheren Editierung auch nicht in den grafischen Client integriert. Eine Implementierung dieser Merkmale hängt zu einem Großteil davon ab, inwiefern die Ontologien einem automatisierten Prozess zugeführt werden sollen. Ferner muss die oben angesprochene Evaluierung prüfen, ob sich die bewusste Reduktion auf eine einfache Vererbungsstruktur für die Komposition von Begrifflichkeiten bzw. Modellen als hinderlich erweist,

6 Ausblick und Zusammenfassung

oder ob diese Komposition mit Hilfe von Assoziationen besser gelingt.

Denkbare Erweiterungen könnten z.B. auch im Bereich der Aufzeichnung von weiteren Meta-Informationen liegen. Hierbei könnte z.B. die Beliebtheit von Modellen oder Individuen aufgrund der Aufrufe in der Applikation erfasst werden, oder die Relevanz von Dokumenten im System – aufgrund der Anzahl der auf sie zielenden Verweise – bestimmt werden. Möglich ist dies aufgrund des gewählten Ontologie-Formats (OWL) das mit Hilfe von URI's die jeweilige Ressource eindeutig identifiziert. Ferner ermöglicht dieses Format auch den Austausch der definierten Wissensstruktur über Systemgrenzen hinweg.

Trotz der Anzahl möglicher Verbesserungen bietet die Umgebung in Kombination mit dem `openTeamServer` sowohl die benötigten individuellen als auch kooperativen primären Medienfunktionen. Die Anwendung bzw. Umgebung ermöglicht somit ein gemeinschaftliches und selbstorganisiertes Erarbeiten einer Wissensstruktur und erweitert die Möglichkeit der kooperativen Zusammenarbeit im `openTeam-Server` um eine weitere Facette.

Literatur

- [Aschoff u. a.] ASCHOFF, F.-R. ; SCHMALHOFER, F. ; ELST, L. van: *Knowledge Mediation: A Procedure for the Cooperative Construction of Domain Ontologies*. – URL http://www.dfki.uni-kl.de/~elst/papers/Aschoff_Schmalhofer_van_Elst_AMKM2004.pdf. – Zugriffsdatum: 2. März 2005
- [Balzert 1999] BALZERT, Helmut: *Lehrbuch Grundlagen der Informatik*. Heidelberg Berlin : Spektrum Akademischer Verlag, 1999
- [Berners-Lee u. a. 2001] BERNERS-LEE, Tim ; HENDLER, James ; LASSILA, Ora: *The Semantic Web*. 2001. – URL http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21. – Zugriffsdatum: 11. Nov 2004
- [Biron und Malhotra 2004] BIRON, Paul V. ; MALHOTRA, Ashok: *XML Schema Part 2: Datatypes Second Edition*. 2004. – URL <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/#built-in-datatypes>. – Zugriffsdatum: 6. Mai 2005
- [Daconta u. a. 2003] DACONTA, M. ; OBRST, L. J. ; SMITH, K. T.: *The Semantic Web - A guide to the Future of XML, Webservices and Knowledge Management*. Indianapolis, Indiana : Wiley Publishing, Inc., 2003
- [Dan Brickley and R.V. Guha 2004] DAN BRICKLEY AND R.V. GUHA: *RDF Vocabulary Description Language 1.0: RDF Schema*. 2004. – URL <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>. – Zugriffsdatum: 19. April 2005
- [Degele 2000] DEGELE, Nina: *Informiertes Wissen - Eine Wissenssoziologie der computerisierten Gesellschaft*. Frankfurt/Main : Campus Verlag, 2000
- [Duden 2000] DUDEN: *Duden - Das Fremdwörterbuch, 7. Auflage*. Mannheim : Bibliographisches Institut & F.A. Brockhaus AG, 2000
- [Fensel u. a. 2003] FENSEL, Dieter (Hrsg.) ; SYCARA, Katia (Hrsg.) ; MYLOPOULOS, John (Hrsg.): *The Semantic Web - ISWC 2003 Second International Semantic Web Conference, Sanibel Island, FL, USA, October 20-23*. Bd. 2870. Springer-Verlag GmbH, 2003. (Lecture Notes in Computer Science)

Literatur

- [Fluit u. a. 2002] FLUIT, Christiaan ; SABOU, Marta ; HARMELEN, Frank van: Ontology-based Information Visualization. In: GEROIMENKO, Vladimir (Hrsg.) ; CHEN, Chaomei (Hrsg.): *Visualizing the Semantic Web*. Springer, 2002, S. 36–48
- [Frank Manola and Eric Miller 2004] FRANK MANOLA AND ERIC MILLER: *RDF Primer*. 2004. – URL <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#basicconcepts>. – Zugriffsdatum: 7. April 2005
- [Gruber 1995] GRUBER, Thomas R.: Toward principles for the design of ontologies used for knowledge sharing. In: *International Journal of Human-Computer Studies*, Vol. 43 , Issue 5-6 (1995), S. 907–928
- [Hampel 2002] HAMPEL, Thorsten: *Virtuelle Wissensräume - Ein Ansatz für die kooperative Wissensorganisation*, Universität Paderborn, Dissertation, 2002. – URL <http://iug.uni-paderborn.de/hempel/Publikationen/Dissertation.pdf>. – Zugriffsdatum: 8. März 2005
- [Hampel und Keil-Slawik 2002] HAMPEL, Thorsten ; KEIL-SLAWIK, Reinhard: sTeam: Structuring Information in a Team - Distributed Knowledge Management in Cooperative Learning Environments. In: *ACM Journal of Educational Resources in Computing Vol. 1, No. 2* (2002), S. 1–27
- [Hampel u. a. 2004] HAMPEL, Thorsten ; KEIL-SLAWIK, Reinhard ; SELKE, Harald: Semantische Räume – Von der Navigation zur kooperativen Wissensstrukturierung. In: R. KEIL-SLAWIK, G. Szwillus H. (Hrsg.): *Mensch & Computer 2004, Allgegenwärtige Interaktion*. München : Oldenbourg Verlag, 2004, S. 221–230
- [Hesse und von Braun 2001] HESSE, Wolfgang ; BRAUN, Hubert von: Wo kommen die Objekte Her? Ontologisch-erkenntnistheoretische Zugänge zum Objektbegriff. In: *GI Jahrestagung* (2), 2001, S. 776–781
- [Horrocks und Hendler 2002] HORROCKS, I (Hrsg.) ; HENDLER, J (Hrsg.): *The Semantic Web - ISWC 2002: First International Semantic Web Conference, Sardinia, Italy, June 9-12*. Bd. 2342. Springer-Verlag GmbH, 2002. (Lecture Notes in Computer Science)
- [Kuhlen 2003] KUHLEN, Rainer: Informationskompetenz und Vertrauen als Grundlage informationeller Autonomie und Bildung. Was bedeutet die fortschreitende Delegation von Informationsarbeit an Informationsassistenten? In: T. CHRISTALLER, J. W. (Hrsg.): *Autonome Maschinen*. Wiesbaden : Westdeutscher Verlag, 2003, S. 186–206
- [Le Hors u. a. 2000] LE HORS, Arnaud ; LE HÉGARET, Philippe ; WOOD, Lauren u. a.: *Document Object Model (DOM) Level 2 Core Specification - Version 1*. 2000. – URL <http://www.w3.org/TR/2000/REC-DOM-Level-2-Core-20001113/>. – Zugriffsdatum: 22. April 2005

Literatur

- [Liechti 2000] LIECHTI, Olivier: Awareness and the WWW: an overview. In: *ACM SIGGROUP Bulletin, Vol. 21, Issue 3* (2000), S. 3 – 12
- [McGuinness und van Harmelen 2004] MCGUINNESS, Deborah L. ; HARMELEN, Frank van: *OWL Web Ontology Language Overview*. 2004. – URL <http://www.w3.org/TR/2004/REC-owl-features-20040210/>. – Zugriffsdatum: 7. April 2005
- [Mozilla Organization] MOZILLA ORGANIZATION: *Mozilla SVG Project*. – URL <http://www.mozilla.org/projects/svg/>. – Zugriffsdatum: 2. Mai 2005
- [Noy und McGuinness] NOY, Natalya ; MCGUINNESS, Deborah: *A Guide to Creating Your First Ontology*. – URL <http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf>. – Zugriffsdatum: 6. März 2005
- [Noy u. a. 2001] NOY, Natalya F. ; SINTEK, Michael ; DECKER, Stefan ; CRUBÉZY, Monica ; FERGERSON, Ray W. ; MUSEN, Mark A.: Creating Semantic Web Contents with Protégé-2000. In: *IEEE Intelligent Systems, Vol. 16 , Issue 2*, 2001, S. 60–71
- [Robertson u. a. 1991] ROBERTSON, George G. ; MACKINLAY, Jock D. ; CARD, Stuart K.: Cone Trees: animated 3D visualizations of hierarchical information. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, 1991, S. 189 – 194
- [Rubart und Hampel 2003] RUBART, Jessica ; HAMPEL, Thorsten: Structuring Cooperative Spaces: From Static Templates to Self-Organization. In: HICKS, David L. (Hrsg.): *Metainformatics: International Symposium, MIS 2003, Graz, Austria, September 17-20, 2003*, S. 119–125
- [Scheffe 2001] SCHEFFE, Peter: Die Rolle der Ontologie in der Softwaretechnik am Beispiel der Visualisierung. In: *GI Jahrestagung (2)*, 2001, S. 788–793
- [Shneiderman 1991] SHNEIDERMAN, Ben: The eyes have it: a task by data type taxonomy for information visualisation. In: *Proceedings of the 1996 IEEE Symposium on Visual Language*, 1991, S. 336–343
- [Shneiderman 1992] SHNEIDERMAN, Ben: Tree visualization with tree-maps: a 2-d space-filling approach. In: *ACM Transaction on Graphics , Vol. 11, No. 1* (1992), S. 92–99
- [Shneiderman 1998] SHNEIDERMAN, Ben: *Treemaps for space-constrained visualization of hierarchies*. 1998. – URL <http://www.cs.umd.edu/hcil/treemap-history/>. – Zugriffsdatum: 15. März 2005
- [Smith u. a. 2004] SMITH, Michael K. ; WELTY, Chris ; MCGUINNESS, Deborah L.: *OWL Web Ontology Language Guide*. 2004. – URL <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>. – Zugriffsdatum: 7. April 2005

Literatur

- [Stephens und Huhns 2001] STEPHENS, Larry M. ; HUHNS, Michael N.: Consensus ontologies. Reconciling the semantics of Web pages and agents. In: *Internet Computing, IEEE, Vol. 5* (2001), S. 104–106
- [Studer u. a.] STUDER, R. ; OPPERMAN, H. ; SCHNURR, H.-P.: *Die Bedeutung von Ontologien für das Wissensmanagement*. – URL http://www.ontoprise.de/documents/Bedeutung_von_Ontologien_fuer_WM.pdf. – Zugriffsdatum: 4. März 2005
- [Sure u. a. 2002] SURE, York ; ERDMANN, Michael ; ANGELE, J. u. a.: OntoEdit: Collaborative Ontology Development for the Semantic Web. In: *The Semantic Web - ISWC 2002: First International Semantic Web Conference, Sardinia, Italy, June 9-12, Vol. 2342, 2002*, S. 221–235
- [SVG Working Group 2003] SVG WORKING GROUP: *Scalable Vector Graphics (SVG) 1.1 Specification*. 2003. – URL <http://www.w3.org/TR/SVG11/>. – Zugriffsdatum: 15. Dez 2004
- [SVG Working Group 2004] SVG WORKING GROUP: *Scalable Vector Graphics (SVG) 1.2 - Working Draft*. 2004. – URL <http://www.w3.org/TR/2004/WD-SVG12-20041027/>. – Zugriffsdatum: 15. Dez 2004
- [Uschold und Gruninger 1996] USCHOLD, Mike ; GRUNNINGER, Michael: Ontologies: Principles, Methods and Applications. In: *Knowledge Engineering Review, Vol. 11, No. 2* (1996), S. 93–155
- [Utting und Yaneklovich 1989] UTTING, K. ; YANEKLOVICH, N.: Context and orientation in hypermedia networks. In: *ACM Transactions on Information Systems, Vol. 7*, 1989, S. 58–84
- [Weng und Gennari 2004] WENG, C. ; GENNARI, J.: Asynchronous collaborative writing through annotations. In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, 2004, S. 578 – 581
- [Ziegler 2003] ZIEGLER, Cai: Web Ontology Language (OWL): Vokabulare fürs Web: Surfende Maschinen. In: *iX - Magazin für professionelle Informationstechnik, Vol. 12* (2003), S. 108–113

Versicherung

Ich versichere, dass ich diese Arbeit ohne fremde Hilfe und ohne Benutzung anderer als den angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, welche wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Paderborn, den 11.05.2005