Sekundäre Informationsstrukturierung

Eine Methodologie zur Verbindung XML- und RDF- basierter Informationsmodellierung sowie ihre Anwendung auf linguistische Korpora

Dissertation
zur Erlangung des akademischen Grades
Doctor philosophia (Dr. phil.)
eingereicht an der Fakultät für Linguistik und Literaturwissenschaft
Universität Bielefeld

von
Felix Sasaki

Verteidigt am 15. November 2004

Gutachter:
Dr. Andreas Witt (Universität Bielefeld)
Prof. Dr. Henning Lobin (Justus-Liebig-Universität Gießen)

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbständig verfasst, keine anderen als die angegebenen Hilfsmittel verwendet und wörtlich oder inhaltlich übernommene Quellen als solche gekennzeichnet habe.

Gedruckt auf altersbeständigem Papier nach ISO 9706

Bielefeld, den 22. November 2004

Für Aki, Taiki und Clara, ohne die ich diese Arbeit nie zu Stande gebracht hätte.

Inhaltsverzeichnis

Abbildungen

Tabellen

Beispiele

1. Einleitung

1.1. Gegenstand und Ziele der Arbeit

Die vorliegende Arbeit thematisiert die Verbindung informationeller Ressourcen durch eine Methodologie **texttechnologischer Informationsmodellierung**, und ihre Anwendung in der Modellierung linguistischer Korpora und konzeptueller Modelle. Der Begriff **informationelle Ressourcen** umfasst:

- Dokumente mit textuellen Daten, d.h. einzelne Zeichen und Zeichenketten in eventuell verschiedenen Schriftsystemen, vgl. die lateinbasierte Alphabetschrift und die piktographische, syllabische japanische Schrift;
- die Segmentierung der Daten, z.B. in Morpheme und Wörter oder Absätze und Kapitel;
- die Anreicherung der Segmente mit Zusatzinformationen, z.B. zur Klassifizierung von Wortarten;
- die Hierarchisierung der Segmente: Wörter sind in Sätzen enthalten, Absätze in Kapiteln etc.;
- die Beschreibung von Regeln hinsichtlich der Benennung der Segmente und Zusatzinformationen sowie hinsichtlich erlaubter Hierarchisierungen und Sequenzierungen: Ein Absatz darf z.B. in einem Kapitel vorkommen, nicht jedoch umgekehrt, und Überschriften stehen vor Paragraphen;
- die Modularisierung der Regelbeschreibungen, z.B. in textstrukturelle Regeln hinsichtlich Kapiteln, Paragraphen etc. versus linguistische Regeln hinsichtlich Sätzen, Wörtern etc.;
- die Relationierung der modularisierten Regeln zueinander in einer hierarchischen oder netzartigen Struktur;

- die Beschreibung von abstrakten, konzeptuellen Modellen, z.B. zur Strukturierung von Texten aus typographischer Sicht oder aus linguistischer Sicht.

Eine hervorstechende Eigenschaft texttechnologischer Informationsmodellierung besteht in der Tatsache, dass für viele dieser informationellen Ressourcen formale, standardisierte Repräsentationsformate bestehen: Standardisierungen zur Kodierung von Zeichen wie z.B. **Unicode** (Graham, 2000), zur Informationsanreicherung von Dokumenten (Segmentierung, Hierarchisierung) mittels **Auszeichnungssprachen** wie **XML** (eXtensible Markup Language, Bray et al. (2000)), zur Beschreibung von Regeln der Informationsanreicherung in **Dokumentgrammatiken** bzw. **Schemasprachen** wie **XML Schema** (Thompson et al., 2001) oder **RELAX NG** (Clark und Murata, 2001), zur Modularisierung und Relationierung der Regeln z.B. in Form von **Architekturen** (ISO/IEC10744, 1997), für konzeptuelle Modellierung mittels der auf **RDF** (Resource Description Framework, Lassila und Swick (1999)) aufbauenden Standards, etc. Obwohl die Standards für eine implementationsnahe, physikalische Ebene der Modellierung, d.h. zur maschinellen Repräsentation von Daten konzipiert sind, gewährt ihre Anwendung auf textuelle Daten und konzeptuelle Modelle einen oft unmittelbaren Bezug zur logischen, algorithmisierbaren Modellierung. Unter Rückgriff auf Mehler und Lobin (2004) wird deshalb der Begriff „informationelle Ressource“ in dieser Arbeit synonym gebraucht für die physikalische, d.h. die maschinelle Repräsentation fokussierende Sicht auf Modellierungsobjekte und die logisch algorithmisierbare Sicht, die im Zentrum des eigentlichen Erkentnissinteresses steht.

Die Forschung im Bereich der noch jungen Disziplin **Texttechnologie**, vgl. Lobin (2004b), nimmt häufig eine horizontale Perspektive ein, d.h. Modellierungsaspekte von standardisierten Formaten für einzelne informationelle Ressourcen werden untersucht. So lassen sich Schemasprachen in eine Taxonomie formaler Sprachen einordnen, vgl. Murata et al. (2001). Die vorliegende Arbeit geht hingegen von einer vertikalen Perspektive aus: Welche standardisierten Formate sind geeignet, um verschiedene informationellen Ressourcen miteinander zu verbinden? Müssen die Formate, und wenn ja auf welche Weise erweitert werden, um die Verbindung zu realisieren? Welche Operationen zwischen informationellen Ressourcen sind formal definierbar?

Diese vertikale Sicht trägt zur Lösung von Forschungsfragen bei, welche bei einer horizontalen Sicht auf informationelle Ressourcen auftreten. Die vorliegende Arbeit fokussiert hierbei die Themenbereiche der **Bedeutungsbeschreibung für Auszeichnun-**

gen (markup semantics) und der **semantischen Auszeichnung** (semantic markup) von Dokumenten. Diese Terminologie wurde von Renear et al. (2002, S. 120) geprägt:

- Regeln zur Segmentierung und Hierarchisierung textueller Dokumente lassen sich als formale Grammatiken auffassen. Für viele Restriktionen über Auszeichnungen, die bei der Dokumenterstellung, -verarbeitung und -analyse vorteilhaft wären, sind diese Regeln zu ausdrucksschwach (Ramalho et al., 1999). Deshalb erarbeiten z.B. Sperberg-McQueen et al. (2000) eine formale Semantik, d.h. eine Bedeutungsbeschreibung für Auszeichnungen.

- Repräsentationsformate für konzeptuelle Modelle wie **RDF Schema**, vgl. Brickley und Guha (2004), ermöglichen die Beschreibung von Konzepten und ihren Eigenschaften. Werden diese Beschreibungen auf Informationen in Dokumenten bezogen, sprechen Renear et al. (2002) von einer semantischen Auszeichnung der Dokumente.

Die Forschungen zur Bedeutungsbeschreibung für Auszeichnungen, z.B. Sperberg-McQueen et al. (2000), Simons (2003), Welty und Ide (1999) sowie Lobin (2001), sind oft nicht auf Forschungen zur semantischen Auszeichnung, vgl. z.B. Erdmann und Studer (1999), bezogen. Ansätze aus beiden Bereichen sind zumeist unidirektional ausgerichtet und prozedural formuliert. In einem Bottom-Up Verfahren erhalten bestehende Dokumentgrammatiken bzw. ausgezeichnete Dokumente sukzessive eine Bedeutungsbeschreibung, oder für bestehende Repräsentationen konzeptueller Modelle werden Top-Down Dokumente als Instanziierungen der Konzeptbeschreibungen erzeugt. Im Gegensatz dazu stellt die vorliegende Arbeit eine Methodologie vor, die bidirektional ausgerichtet ist und es erlaubt, Beziehungen zwischen informationellen Ressourcen deklarativ zu formulieren. Dokumente und konzeptuelle Modelle müssen nicht verändert werden. Auf diese Weise leistet die vorliegende Arbeit einen Beitrag zum Anliegen von Melnik und Decker (2000), Interoperabilität zwischen heterogenen informationellen Ressourcen zu gewährleisten. *Bestehende* abstrakte, konzeptuelle informationelle Ressourcen werden zu *bestehenden* konkreten, ausgezeichneten, textuellen Dokumenten relationiert.

In den letzten Jahren ist eine Vielzahl von Dokumenten auf der Basis standardisierter, umfangreicher Dokumentgrammatiken erstellt worden. Die Dokumentgrammatik der **TEI** (Text Encoding Initiative, Sperberg-McQueen und Burnard (1994)) dient z.B. der Auszeichnung hauptsächlich geisteswissenschaftlicher Texte. Der **CES** (Corpus Encoding

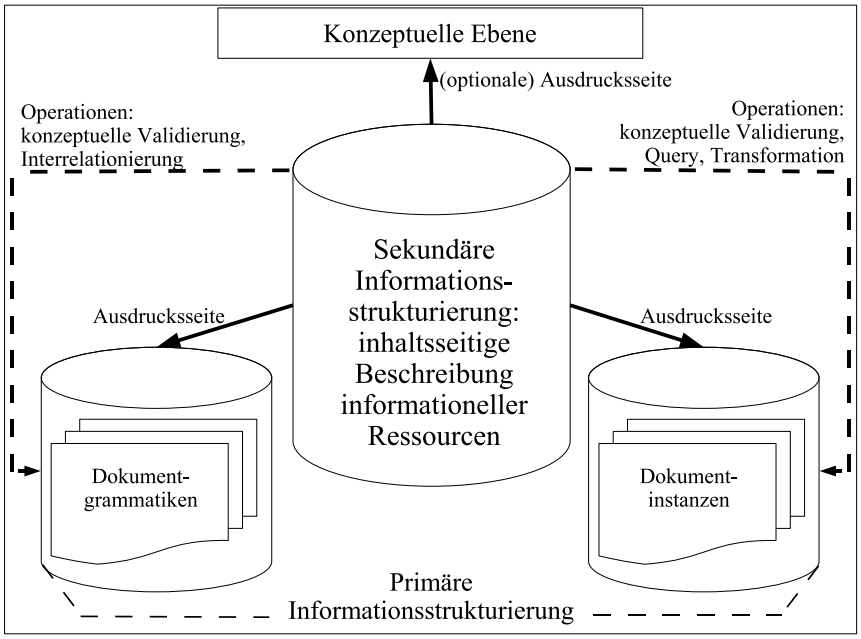

Abbildung 1.1.: Überblick über die Methodologie

Standard, Ide (1998)) wurde zur Auszeichnung linguistischer Informationen konzipiert. Diesen dokumentbezogenen informationellen Ressourcen steht eine Fülle konzeptueller informationeller Ressourcen gegenüber, etwa lexikalische Ressourcen wie **WordNet** (Fellbaum, 1998) oder **SUMO** (Suggested Upper Merged Ontology, Niles und Pease (2001)). Die in der vorliegenden Arbeit entwickelte Methodologie eröffnet einen Weg, derartige heterogene Ressourcen aufeinander zu beziehen.

1.2. Die Methodologie

Die Methodologie ist in Abbildung 1.1 visualisiert. Sie fußt auf einer dreifachen Unterteilung informationeller Ressourcen:

1. eine **primäre Informationsstrukturierung**, welche ausgezeichnete **Dokumentinstanzen** und **Dokumentgrammatiken**, d.h. Regelbeschreibungen für **Dokumentklassen** umfasst;

2. eine **konzeptuelle Ebene**, d.h. **konzeptuelle Modelle** für verschiedene

Domänen;

3. eine **sekundäre Informationsstrukturierung**, welche informationelle Ressourcen aus der primären Informationsstrukturierung und – optional – aus der konzeptuellen Ebene selektiert, interrelationiert und operationalisiert.

Die Begriffe der primären und sekundären Informationsstrukturierung wurden von Lobin (2000) geprägt. Er beschreibt als Aufgabe sekundärer Informationsstrukturierung den Ausdruck von Beziehungen zwischen dokumentgrammatischen Konstrukten. In Ergänzung zu dieser Form dokumentgrammatikbezogener sekundärer Informationsstrukturierung stellt die Arbeit zum einen die Aufgabe einer dokumentinstanzbezogenen sekundären Informationsstrukturierung, bei der Dokumentinstanzen relationiert und transformiert werden. Zum anderen gilt es, informationelle Ressourcen der primären Informationsstrukturierung und der konzeptuellen Ebene zu verknüpfen.

Hauptcharakteristikum der sekundären Informationsstrukturierung, wie sie in der vorliegenden Arbeit verstanden wird, ist die Komplementarität einer inhaltsseitigen, d.h. intensionalen und einer ausdrucksseitigen, d.h. extensionalen Darstellung informationeller Ressourcen. Formal begründen lässt sich diese Komplementarität in der Beschreibung der sekundären Informationsstrukturierung als eine **terminologische Ontologie** im Sinne von Fischer (1998). Die **inhaltsseitige Beschreibung** ist eine Menge von **Aussagen**, die informationelle Ressourcen unter Berücksichtigung für sie spezifischer, zentraler Aspekte aufeinander bezieht. Die **ausdrucksseitigen Beschreibungen** sind die informationellen Ressourcen selbst, d.h. die primäre Informationsstrukturierung und die konzeptuelle Ebene. Zentrale Bestandteile der primären Informationsstrukturierung sind in Dokumentgrammatiken deklarierte **Regeln** sowie **Bedingungen**, die sich in Dokumentinstanzen überprüfen lassen. Zentraler Bestandteil einer konzeptuellen Modellierung sind **Konzepte** sowie die **inferentielle Kraft einer Konzepthierarchie und interkonzeptueller Beziehungen**. Die sekundäre Informationsstrukturierung verknüpft diese Modellierungsinstrumentarien. Sie selektiert in Aussagen mittels einer Reihe vordefinierter Prädikate Regeln und Bedingungen aus der primären Informationsstrukturierung, integriert sie in eine Konzepthierarchie und nutzt die inferentielle Kraft, welche die terminologische Ontologie bietet. Da diese Methodologie die informationellen Ressourcen zwar aufeinander bezieht, aber getrennt repräsentiert, ist eine Veränderung der verbundenen informationellen Ressourcen nicht nötig. Möglich ist aber ein **Zugriff auf informationelle Ressourcen**, der von den Aussagen in der sekundären Informa-

tionsstrukturierung ausgeht. In dieser Arbeit definierte Operationen betreffen u.a. die **konzeptbezogene Validierung**, **Abfrage** und **Transformation** von informationellen Ressourcen der primären Informationsstrukturierung.

Der Kern der ausdrucksseitigen Beschreibung sekundärer Informationsstrukturierung sind die **Selektionsmechanismen**, welche für die jeweiligen informationellen Ressourcen zum Einsatz kommen. In Dokumentgrammatiken werden **dokumentgrammatische Konstrukte**, d.h. Regelbeschreibungen selektiert. Die Subordination von Konzepten in der Konzepthierarchie, beschrieben durch Aussagen in der inhaltsseitigen Beschreibung, wird ausdrucksseitig interpretiert als die Spezialisierung dokumentgrammatischer Konstrukte: Ein übergeordnetes Konzept in der sekundären Informationsstrukturierung selektiert eine weniger restriktive Regel der Dokumentgrammatik als ein untergeordnetes Konzept.

In Dokumentinstanzen werden **Informationseinheiten** im Sinne von Cowan und Tobin (2004) selektiert. Sie machen den informationellen Gehalt einer Dokumentinstanz aus. Die Beschreibung von Bedingungen, d.h. Selektionskriterien für Informationseinheiten hat diejenigen Informationseinheiten zum Gegenstand, welche durch Aussagen in einem superordinierten Konzept selektiert sind. Zur Selektion in einer Dokumentinstanz kommt die von Brüggemann-Klein und Wood (2000) entwickelte **Pfadbeschreibungssprache der Caterpillar-Ausdrücke** zur Anwendung. Neben der inferentiellen Kraft der Konzepthierarchie lässt sich in Caterpillar-Ausdrücken eine weitere inferentielle Kraft einsetzen, d.h. durch die Berücksichtigung interkonzeptueller Beziehungen. Aus diesen können im Pfadausdruck zu testende Knoteneigenschaften inferiert werden, unter Rückgriff auf die in den Konzepten selektierten dokumentgrammatischen Konstrukte bzw. Informationseinheiten. Deshalb ist in der sekundären Informationsstrukturierung ein unmittelbarer, expliziter Bezug auf informationelle Ressourcen der primären Informationsstrukturierung nur bei Selektionen in übergeordneten Konzepten notwendig. Untergeordnete Konzepte nutzen die inferentielle Kraft der Konzepthierarchie und interkonzeptueller Beziehungen zur Selektion von Informationseinheiten in der beschriebenen Weise.

Auch **Informationseinheiten in mehreren Dokumentinstanzen** können selektiert werden. Die Notwendigkeit, auf mehrere Dokumentinstanzen gleichzeitig zuzugreifen, ergibt sich auf Grund der Einschränkung, in einer Dokumentinstanz nur eine singuläre Hierarchie von Auszeichnungen erzeugen zu können. Adäquate Auszeichnungen, im Sinne der engen Verzahnung physikalischer *und* logischer Modellierungen, sind des-

halb nicht immer realisierbar, z.B. bei der Auszeichnung typographischer versus linguistischer Textstrukturen. Als ein Ansatz zur Lösung dieses Problems wird in der vorliegenden Arbeit auf die von Witt (2004a) entwickelte Methodologie der **multiplen Auszeichnung primärdatenidentischer Dokumente** zurückgegriffen. Die eindeutige, in jeder Dokumentinstanz identische Ordnung der textuellen Zeichen ist der Ankerpunkt für die dokumentinstanzübergreifende Relationierung von Auszeichnungen. Eine Reihe vordefinierter Prädikate dient der inhaltsseitigen Beschreibung von Beziehungen zwischen den Auszeichnungen. Selektiert werden dabei jene Informationseinheiten, welche die beschriebenen Beziehungen aufweisen.

Inhaltsseitige Beschreibungen in der sekundären Informationsstrukturierung kombinieren die vorgestellten Selektionen in mehrfach ausgezeichneten Dokumentinstanzen mit Selektionen dokumentgrammatischer Konstrukten bzw. von Informationseinheiten in einzelnen Dokumentinstanzen, unter Rückgriff auf die inferentielle Kraft der Konzepthierarchie sowie interkonzeptueller Beziehungen. Die selektierten, ausdrucksseitigen Beschreibungen der primären Informationsstrukturierung können zum einen durch weitere Aussagen zueinander in Beziehung gesetzt werden. Wird bei den Aussagen nur auf Konzepte zurückgegriffen, die dokumentgrammatische Konstrukte selektieren, so handelt es sich um **dokumentgrammatikbezogene sekundäre Informationsstrukturierung**, sonst um **dokumentinstanzbezogene sekundäre Informationsstrukturierung**. Zum anderen dienen Aussagen, die informationelle Ressourcen der konzeptuellen Ebene selektieren, der Verknüpfung der primären Informationsstrukturierung und konzeptueller Ressourcen. In Bezug auf informationelle Ressourcen der konzeptuellen Ebene gibt es dabei keine Selektionsbeschränkungen. Ob ein Konzept, eine Menge von Konzepten oder interkonzeptuelle Beziehungen selektiert werden, ist irrelevant. Wenn die ausgewählten konzeptuellen Modelle in der konzeptuellen Ebene es erlauben, kann die sekundäre Informationsstrukturierung bestehende Konzepthierarchien und interkonzeptuelle Beziehungen kopieren. Inwiefern dies möglich ist, hängt von den zur Verfügung stehenden konzeptuellen Modellen bzw. Dokumentgrammatiken und -instanzen ab, die es zu verknüpfen gilt.

Zusammenfassend lässt sich sekundäre Informationsstrukturierung als ein **Ansatz zur bidirektional operationalisierbaren, deklarativen, wissensbasierten Bedeutungsbeschreibung für Auszeichnung** charakterisieren, der zugleich zur automatischen Erzeugung von semantischer Auszeichnung verwendet werden kann, und eine Verbindung informationeller Ressourcen erlaubt, die keine Notwendigkeiten zu ihrer

Veränderung aufzwingt. Der Ansatz wird als wissensbasiert bezeichnet, weil er auf einer Menge von Aussagen beruht, die regel- und bedingungskonforme, informationelle Ressourcen der primären Informationsstrukturierung in einer terminologischen Ontologie strukturieren, welche interkonzeptuelle Beziehungen und die Konzepthierarchie als grundlegende Eigenschaften der konzeptuellen Ebene realisiert. Im Gegensatz dazu stehen Ansätze zur Bedeutungsbeschreibung für Auszeichnung, die auf einer **objektorientierten Modellierung** beruhen. Beispiele sind die Arbeiten von Sperberg-McQueen et al. (2000), die erwähnten Architekturen oder das Typensystem von XML Schema (Biron und Malhotra, 2001). Die unverzichtbare Basis dieser Ansätze ist die Eindeutigkeit der **Typisierung von Informationseinheiten** in Dokumentinstanzen. Bei der sekundären Informationsstrukturierung ist Ambiguität bzw. **Polysemie** im Sinne von Sperberg-McQueen et al. (2002) erwünscht und notwendig, um so den Bezug der gleichen Dokumentgramatiken bzw. -instanzen auf verschiedene Modelle in der konzeptuellen Ebene zu ermöglichen.

1.3. Eine Anwendungsdomäne

Eine Domäne, in der textuelle, ausgezeichnete Daten große Bedeutung besitzen, sind Auszeichnungen **linguistischer Korpora**. Das Modellierungsinventar derartiger Auszeichnungen, die im linguistischen Bereich auch als **Annotationen** bezeichnet werden, erfährt durch die Methodologie der sekundären Informationsstrukturierung eine erhebliche Bereicherung. Verschiedene Probleme in derartigen Korpora erfahren hier neue Lösungsansätze, insbesondere die Multidimensionalität der Sprache und die Annotation tiefergehender Strukturen. Die von Simons (1998) beschriebene **Multidimensionalität der Sprache** erschwert eine hierarchisch-kompositionale Anordnung von linguistischen Annotationen unterschiedlicher Beschreibungsebenen. Die Methodologie der sekundären Informationsstrukturierung, in Verbindung mit einer Annotation in multiplen, primärdatenidentischen Dokumentinstanzen, öffnet den Zugang zu einer deklarativen Beschreibung nicht-hierarchischer Beziehungen zwischen den Annotationen. Die inhaltsseitige Beschreibung kann zudem ausdrucksseitig operationaliert werden z.B. in Form konzeptbezogener Suchanfragen. **Tiefergehende Strukturen**, d.h. in linguistischen Daten nicht eindeutig identifizierbare Segmente, sind mit gegenwärtig gebräuchlichen Verfahren der Korpusanalyse nur schwer erfassbar. Die vorliegende Arbeit zeigt, welchen Beitrag sekundäre Informationsstrukturierung zur Beschreibung von Beziehungen

derartiger Strukturen zu explizit repräsentierbaren Annotationen leisten kann.

Das Stichwort **Explikation statt Generalisierung** fasst den Gewinn der sekundären Informationsstrukturierung für eine datenorientierte Linguistik zusammen. In Projekten wie **MATE**, vgl. Mengel et al. (2000), wurde versucht, übergreifende Annotationsvokabulare zu schaffen. Theorie-, Sprach- und Domänenspezifik linguistischer Daten sollte auf diese Weise überwunden werden. Diese Herangehensweise führt jedoch zum **Informationsverlust durch Generalisierung**. So lassen sich z.B. für Wortarten nur sprachübergreifende Kategorien definieren, deren kleinster gemeinsamer Nenner Wort lautet. Neuere Ansätze wie Ide und Romary (2003) oder Simons (2003) versuchen dem Dilemma zu begegnen, indem sie Abbildungen spezifischer, bestehender Annotationsvokabulare auf ein generelles, abstraktes konzeptuelles Modell formulieren. Dabei gehen sie jedoch, ähnlich wie viele Ansätze zur Bedeutungsbeschreibung für Auszeichnungen, unidirektional vor, Bottom-up von konkreten Auszeichnungen zu konzeptuellen Modellen oder Top-Down. Die verlustbehaftete Generalisierung der linguistischen Kategorien, die den Annotationen zu Grunde liegen, bleibt bestehen.

Im Gegensatz zu diesen Verfahren zielt der Ansatz der vorliegenden Arbeit darauf ab, Eigenschaften von Annotationsvokabulare nicht zu generalisieren, sondern ihre Bedeutung mittels der sekundären Informationsstrukturierung in konzeptuellen Modellen zu explizieren. Die Modellierung ist **bidirektional**: Theorie-, sprach- und domänenspezifische Annahmen sind sowohl deduktiv, d.h. aus Sicht des konzeptuellen Modells validierbar, als auch induktiv, d.h. aus Sicht der primären Informationsstrukturierung erweiterbar. Eine derartige Methode verspricht in zweierlei Hinsicht Gewinn. Erstens werden Modellierungen linguistischer Phänomene trotz ihrer Spezifik vergleichbar, wobei sowohl die Annotationseinheiten als auch die zu Grunde liegenden Konzepte in Vergleiche mit einfließen. Dies entspricht der datenbasierten Ausführung des von Shieber (1987) beschriebenen Verfahrens, die Unterschiede linguistischer Theorien formal zu spezifizieren. Zweitens stellt die Explikation der Modellierungsunterschiede eine semiautomatische, informationserhaltende Überführung von Daten verschiedener Spezifika in Aussicht. Bisher werden solche Überführungen prozedural beschrieben, vgl. Hajicova und Kucerova (2002) oder Xia und Palmer (2001). Am Beispiel der deklarativ formulierten Konversion von theorie- und sprachspezifischen **Treebanks** (Baumbanken, vgl. Sasaki et al. (2003)) wird in dieser Arbeit der Beitrag exemplifiziert, den sekundäre Informationsstrukturierung zur korpusbasierten, multidimensionalen, tiefergehende Strukturen erfassenden Modellierung linguistischer Phänomene leisten kann. Die Beschreibung von

Koreferenz aus sprachspezifischer und sprachübergreifender Sicht, vgl. Sasaki und Witt (2004a), ist der zweite lingistische Phänomenbereich, der ausführlicher behandelt wird. Die beiden Phänomenbereiche sind ausgewählt hinsichtlich ihrer extremen Unterschiedlichkeit: Im linguistischen Sinne syntaktische versus semantische Beschreibungen bzw. ausgezeichnete Dokumente sollen mit der Methodologie der sekundären Informationsstrukturierung modelliert und aufeinander bezogen werden.

1.4. Aufbau der Arbeit

Der erste, zentrale Teil der Arbeit entwickelt die Methodologie der sekundären Informationsstrukturierung. Ausgehend von der in Kapitel 2 gestellten Frage, welche bestehenden Standards für eine vertikale Verbindung informationeller Ressourcen geeignet sind, werden in Kapitel 3 über Standardisierungen hinausgehende Forschungsansätze diskutiert. Daraus folgen eine Reihe von Desiderate an eine sekundäre Informationsstrukturierung, welche bei der Beschreibung der Methodologie in Kapitel 4 umgesetzt werden.

Der zweite Teil widmet sich der Anwendungsdomäne linguistischer Korpora. Die Fragestellungen dieser Domäne – Repräsentation von Multidimensionalität und Annotation tiefergehender Strukturen – werden in Kapitel 5 aus einer Grundsatzperspektive heraus angegangen, in Kapitel 6 hingegen stehen die Themenbereiche der Transformation von theoriespezifischen Treebanks und der Beschreibung von Koreferenz im Zentrum. Die Arbeit schließt mit einem Resümee und einem Ausblick auf zukünftige Forschungen sowie einem Anhang, der typographische Kennzeichnungen informationeller Ressourcen in dieser Arbeit beschreibt, in einem Glossar die wichtigsten Begriffe zusammenfasst und Formate zur Repräsentation der sekundären Informationsstrukturierung vorstellt.

Teil I.

Die Methodologie

2. Ausgangspunkt: Texttechnologische Informationsmodellierung

2.1. Vorbemerkung

2.1.1. Der Begriff der textuellen Informationsmodellierung

Grundlage der in dieser Arbeit entwickelten Methodologie ist der Begriff der **textuellen Informationsmodellierung**. Texte haben laut Lobin (2000, Einleitung) für diese Modellierung drei hervorstechende Eigenschaften. Erstens können unterschiedliche Ebenen voneinander unterschieden werden. Diese Eigenschaft verdeutlicht Beispiel 2.1[1].

(2.1)

nagai	*burokku*	*wo*	*shita*	*ni*	*oite*	*kudasai*
lang	Block	-	unten	-	legen	bitte
ADJ	N	ACC	N	LOC	V-TE	V-IMP

‚Leg bitte den langen Block nach unten.'

Beispiel 2.1: Beispiel eines Textes

Der Satz „nagai burokku wo shita ni oite kudasai" ist die Basisebene, die aus einer Folge von Zeichen besteht (erste Zeile des Beispiels). Einzelne Wörter lassen sich unterscheiden, die mit Zusatzinformationen angereichert sind (dritte Zeile). Die Wörter bilden einen Satz, der als ganzes ebenfalls mit Zusatzinformationen versehen werden kann, hier eine Übersetzung (dritte Zeile). Diese Zusatzinformation ist abstrakter als die wortbezogenen Informationen, was die Vielfalt von Übersetzungsvarianten verdeutlicht

[1] Das Beispiel bildet die Grundlage für die Diskussion in vielen Kapiteln der vorliegenden Arbeit. In den meisten Fällen stehen dabei Fragen der Informationsmodellierung im Vordergrund. Ein japanisches, also deutschen Lesern für gewöhnlich inhaltlich nicht zugängliches Beispiel wurde gewählt, um deutlich zu machen, dass Aspekte der Informationsmodellierung behandelt werden, die unabhängig von der zu modellierenden Sprache bzw. Domäne sind.

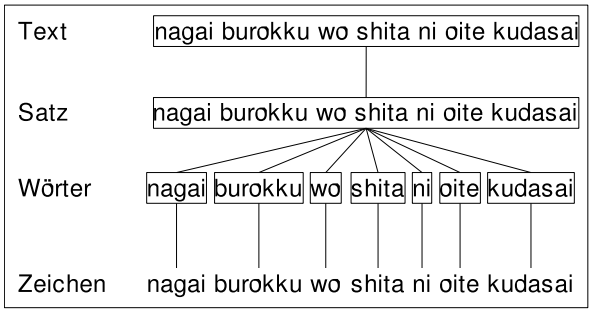

Abbildung 2.1.: Hierarchische Strukturierung von Informationen

(vgl. ‚Leg bitte den langen Block nach unten' versus ‚Bitte legen Sie den langen Block nach unten').

Zweitens stehen die Ebenen in einer regelhaften Beziehung zueinander. Die Einheiten der zweiten Zeile des Beispiels sind den Einheiten der dritten Zeile untergeordnet – Wörter stehen in Sätzen, nicht umgekehrt. Die Einheiten der ersten Zeile sind der zweiten Zeile untergeordnet – Wörter bestehen aus Buchstaben. Lobin leitet aus dieser hierarchischen Beziehung eine dritte Eigenschaft von Texten ab: Die Regeln sind immer auf eine hierarchische, baumstrukturierte Form zurückführbar. Abbildung 2.1 visualisiert die Baumstruktur von Beispiel 2.1. Die konkreten Einheiten, d.h. die textuellen Zeichen, sind in jeder Strukturierungsebene aufgeführt, um die unterschiedliche Segmentierung zu verdeutlichen.

Diese drei Aspekte der textuellen Informationsmodellierung spielen eine zentrale Rolle in der vorliegenden Arbeit. **Primäre Informationsstrukturierung**, vgl. Lobin (2000), nutzt texttechnologische Standards wie **Auszeichnungssprachen**, um die Aspekte bei der Erstellung, Verarbeitung und Analyse von Dokumenten umzusetzen. Die Verfahren der primären Informationsstrukturierung werden in der vorliegenden Arbeit so umgesetzt, wie in Lobin (2000) geschildert. Ein wichtiger Unterschied zu dem dort vorstellten Strukturierungsverfahren ist die Annahme, dass informationelle Ressourcen prinzipiell *gleichzeitig* vorliegen:

- verschiedene Realisierungen der primären Informationsstrukturierung, d.h. Segmentierungen, Hierarchisierungen und Regelbeschreibungen, angewandt auf die gleichen textuellen Daten;

- verschiedene **konzeptuelle Modelle** auf einer **konzeptuellen Ebene**.

Die Informationen zu Wörtern, Satz und Text aus Abbildung 2.1 können als Instanziierung eines Modells für die – im linguistischen Sinne – grammatische Struktur der Sprache verstanden werden. Andere Informationen, z.B. hinsichtlich der Unterteilung in Zeilen, sind Instanziierungen eines anderen Modells, z.B. für Layout. Die vorliegende Arbeit entwickelt eine Methodologie, um eine **vertikale Verbindung informationeller Ressourcen** zu erreichen. Zum einen sollen formale Repräsentationen von Modellen mit den Einheiten der primären Informationsstrukturierung verknüpft, zum anderen Einheiten der primären Informationsstrukturierung untereinander relationiert werden.

Für die zweite Aufgabe prägt Lobin den Begriff der **sekundären Informationsstrukturierung**. Auf einer Meta-Ebene, die den Konstrukten zur Regelformulierung für Dokumente übergeordnet ist, werden diese Konstrukte zueinander in Beziehung gesetzt. Die vorliegende Arbeit weist der sekundären Informationsstrukturierung eine weitreichendere Rolle zu: Sie soll beide gestellten Aufgaben erfüllen bzw. miteinander kombinieren. Die Relationierung von (u.a.) Regelbeschreibungen dient der vertikalen Verbindung informationeller Ressourcen. Aus dieser Sicht müssen auch die potentiell anwendbaren Standards zur Repräsentation primärer Informationsstrukturierung und der konzeptuellen Ebene bewertet werden: Die zentrale Frage in diesem Kapitel bei der Bewertung texttechnologischer Standards lautet, wie sie für die Kombinierbarkeit mit anderen Standards und den übergreifenden Zugriff auf informationelle Ressourcen einsetzbar sind.

Ein weiterer Unterschied zur primären Informationsstrukturierung, wie sie Lobin (2000) beschreibt, ist der Einsatz von Bedingungen, als Ergänzung zu Regelbeschreibungen. Für eine vertikale Sicht auf informationelle Ressourcen sind Bedingungen unerlässlich. Der Unterschied zwischen beiden wird in Abschnitt 2.2.1.3 diskutiert.

2.1.2. Was sind informationelle Ressourcen?

Offen bleibt die Frage, worum es sich bei informationellen Ressourcen eigentlich handelt. Sie kann beantwortet werden unter Rückgriff auf eine Unterscheidung von drei interdependenten Ebenen, die von Mehler und Lobin (2004) getroffen wird. Sie ist zentral für die texttechnologische Modellierung, wie sie in der vorliegenden Arbeit verstanden wird. Das **konzeptionelle Modell** umfasst möglichst anschauliche Beschreibungen der Eigenschaften, die ein Untersuchungsgegenstand hat. In Bezug auf Texte als Untersuchungs-

gegenstand kann das konzeptionelle Modell eine natürlichsprachliche Beschreibung von Texteigenschaften darstellen. Das **logische Modell** beschreibt den Untersuchungsgegenstand in Hinblick auf formale, logische Gesetzmäßigkeiten, die u.a. für einer maschinelle Repräsentation von Bedeutung sind. Bei Texten müssen z.B. die Gesetzmäßigkeiten für Segmentierung, Hierarchisierung und Regelbeschreibung hinreichend formal beschrieben werden, um ihre maschinelle Verarbeitung gewährleisten zu können. Die eigentliche Implementation von Programmen, welche die Gesetzmäßigkeiten des logischen Modells umsetzen, beschreibt das **physikalische Modell**.

Der Kernpunkt texttechnologischer Modellierung liegt darin begründet, das logische und physikalische Modellierung sich eng aufeinander beziehen lassen und oft zusammen fallen. Texttechnologische Standards spezifizieren aus Sicht der physikalischen Modellierung Implementationsrichtlinien für Programme und sichern so ihre Interoperabilität und die Austauschbarkeit von Daten. Aus Sicht der logischen Modellierung erlauben die Standards die formale Beschreibung der Eigenschaften von Untersuchungsgegenständen. Eine Aufgabe der logischen Modellierung, wie z.B. die Beschreibung von Gesetzmäßigkeiten der Segmentierung und Hierarchisierung von Texten, kann demnach nahezu unmittelbar in den Standards, welche das physikalische Modell ausmachen, realisiert werden. Abbildung 2.2 visualisiert dieses Verhältnis und macht noch einmal deutlich, wie die vertikale Verbindung informationeller Ressourcen zu verstehen ist: Informationelle Ressourcen der primären Informationsstrukturierung werden auf andere informationelle Ressourcen der primären Informationsstrukturierung oder auf solche der konzeptuellen Ebene bezogen. Zu beachten ist dabei, dass zwischen konzeptioneller und konzeptueller Modellierung unterschieden wird. Konzeptuelle Modellierung geschieht auf der konzeptuellen Ebene und ist eine Modellierungsform, die – wie die primäre Informationsstrukturierung – eng an bestimmte Standards geknüpft ist und der formalen Beschreibung von Konzepten und ihren Eigenschaften in einer Domäne dient.

Bei einer texttechnologische Modellierung informationeller Ressourcen ist es also nicht unbedingt nötig, zwischen den informationellen Ressourcen an sich – ihrem logischen Modell –, z.B. textuellen Daten oder konzeptuellen Modellen, und den Standards zu ihrer maschinellen Repräsentation – dem physikalischen Modell –, z.B. Auszeichnungssprachen oder Sprachen zur Beschreibung konzeptueller Modelle zu unterscheiden. Deshalb wird der Begriff **informationelle Ressourcen** auch im doppelten Sinne verwandt, d.h. ohne zwischen den zu modellierenden Informationen und den Formaten ihrer maschinellen Repräsentation zu unterscheiden. Viele Standards, die für die primäre Informa-

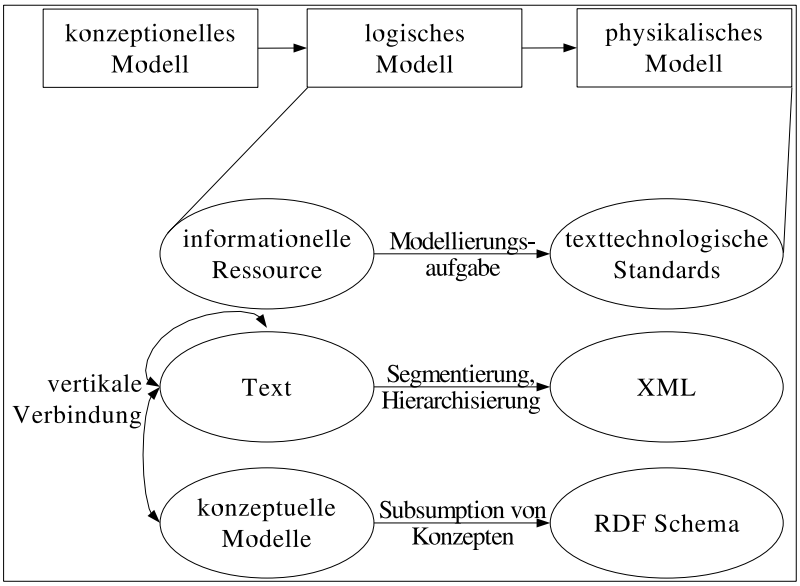

Abbildung 2.2.: Das Verhältnis informationeller Ressourcen zu standardisierten Formaten ihrer Repräsentation

tionsstrukturierung bzw. die Beschreibung konzeptueller Modelle von geringer Bedeutung sind, werden allerdings nur angerissen. Beispiele sind die von Clark (1999) spezifizierte Transformationssprache XSLT oder die von deRose et al. (1999) beschriebene Verlinkungssprache XLink.

Auszeichnungssprachen werden in dieser Arbeit zur Repräsentation von Segmentierungen und Hierarchisierungen textueller Daten bzw. von textuellen Dokumenten benutzt. Oft werden sie aber auch eingesetzt, um Daten jeglicher Art zu repräsentieren. Dabei spielt die Segmentierung und Hierarchisierung der Daten keine Rolle mehr. Diese beiden Anwendungen von Auszeichnungssprachen werden als **dokumentzentriert** versus **datenzentriert** bezeichnet. Bei einer dokumentzentrierten Anwendung von Auszeichnungssprachen ist die Verbindung des physikalischen Modells zum logischen Modell, welches hierarchisch relationierte Ebenen der Textbeschreibung annimmt, unvermittelt. Die vorliegende Arbeit fokussiert deshalb dokumentzentrierte Anwendung von Auszeichnungssprachen. Dies erscheint sinnvoll auf Grund der fokussierten informationellen Ressource „Text“. Bei informationellen Ressourcen hingegen, die sich einer regelhaften, hierarchischen Beschreibung entziehen, ist die dokumentzentrierte Anwendung

von XML nicht möglich. Entsprechende Ansätze, vgl. z.B. Schmidt, i.V., gehen datenzentriert vor und beschreiben deshalb auch nur eine schwache Verbindung von logischem und physikalischem Modell.

2.2. Eine vertikale Sicht auf texttechnologische Standards

2.2.1. Primäre Informationsstrukturierung

2.2.1.1. Auszeichnungssprachen

Zentrales Mittel zur primären Informationsstrukturierung sind die bereits in der Einleitung zu dieser Arbeit erwähnten **Auszeichnungssprachen**. Sie dienen zum einen der Beschreibung des Aufbaus von **Dokumentinstanzen**, zum anderen von formalen Regeln, die für **Dokumentklassen** gelten. Diese beiden Funktionen werden anhand einer Dokumentinstanz (Beispiel 2.2) erklärt, welche den Satz aus Beispiel 2.1 enthält.

(2.2)
```
<corpus>
  <s trans="Leg bitte den langen Block nach unten.">
    <w trans="lang" cat="ADJ">nagai</w>
    <w trans="Block" cat="N">burokku</w>
    <w cat="ACC">wo</w>
    <w trans="unten" cat="N">shita</w>
    <w cat="LOK">ni</w>
    <w trans="legen" cat="V-TE">oite</w>
    <w trans="bitte" cat="V-IMP">kudasai</w>
  </s>
</corpus>
```

Beispiel 2.2: Dokumentinstanz

Die Dokumentinstanz umfasst neben den konkreten, textuellen Daten „nagai burokku wo shita ni oite kudasai“ verschiedene Einheiten: Segmentierungen und Hierarchisierungen hinsichtlich Wörtern und Sätzen, wie sie auch in Abbildung 2.1 visualisiert wurden, und verschiedene Zusatzinformationen über Wortartenkategorien sowie Einzelwortübersetzungen. Auszeichnungssprachen definieren für Dokumentinstanzen sogenanntes **Markup**, welches der Identifikation und Differenzierung dieser Einheiten dient.

Die linksseitige Markierung eines Segments wird im Beispiel durch spitze Klammern angezeigt, die Benennung steht unmittelbar nach der nach links weisenden spitzen Klammer, z.B. `<s>`. Die rechtsseitige Markierung wird durch eine spitze Klammer mit einem Rückstrich angezeigt. Die Zusatzinformationen stehen in der linksseitigen Markierung.

Nicht für den Aufbau von Dokumentinstanzen, sondern für Klassen von Dokumenten lassen sich nun – als zweite Aufgabe von Auszeichnungssprachen – Regeln angeben. Z.B. sollen die `<s>` Segmente den `<w>` Segmenten übergeordnet sein; Zusatzinformationen hinsichtlich Kategorien sollten bei jedem `<w>` Segment gegeben sein, Übersetzungen hingegen sind bei grammatischen Markierungen wie `wo` fakultativ.

Formal gesehen lassen sich die vorgestellten Regeln als eine **kontextfreie Grammatik** mit Nichtterminalen (`corpus s w`), Terminalen (z.B. `burokku`), Phrasenstrukturregeln (z.B. $s \rightarrow w+$) und einem Startsymbol `corpus` beschreiben. Sie ist in Beispiel 2.3[2] wiedergegeben.

(2.3) *corpus* $\rightarrow$ *s+*

s $\rightarrow$ *w+*

w $\rightarrow$ *T+*

T $\rightarrow$ *burokku, kudasai, nagai, ni, oite, shita, wo*

Beispiel 2.3: Regeln einer kontextfreien Grammatik

Der Vergleich zu in der formalen Linguistik weit verbreiteten, kontextfreien Grammatiken, hier ausgeführt in Anlehnung an Witt (1999, S. 123), verdeutlicht, wie nahe der Einsatz von Auszeichnungssprachen für die Modellierung linguistischer Phänomene liegt. Dies ist ein Grund für die Wahl der Domäne, in der die Methodologie der sekundären Informationsstrukturierung im zweiten Teil der vorliegenden Arbeit exemplifiziert wird.

Die 1986 von der ISO standardisierte **SGML** (Standard Generalized Markup Language, vgl. ISO/IEC8859 (1986)), war lange Zeit die verbreiteste Auszeichnungssprache. Die auf SGML basierende, im Jahre 2000 verabschiedete **XML** (eXtensible Markup Language, vgl. Bray et al. (2000)), hat in den letzten Jahren einen phänomenalen Erfolg verbuchen können, ja man kann behaupten, dass SGML inzwischen von seiner Nachfolgerin abgelöst und hinsichtlich des Verbreitungsgrades übertroffen wurde. Die vorlie-

[2] Die Attributionen, d.h. die Zusatzinformationen, sind aus Gründen der Übersichtlichkeit nicht in der Grammatik enthalten.

gende Arbeit verwendet XML zur Repräsentation von Dokumenten. Diese Entscheidung ist hauptsächlich durch zwei Unterschiede zwischen XML und SGML begründet. Zum einen ist es mit XML möglich, Dokumentinstanzen ohne eine Beschreibung von formalen Regeln zu erzeugen. Derartige Dokumentinstanzen werden als **wohlgeformt** bezeichnet, wenn sie Markup in der angesprochenen Weise verwenden. Liegt eine Regelbeschreibung vor, der diese Dokumentinstanz genügt, ist sie gleichzeitig gegenüber dieser Beschreibung **valide**. Validität setzt also Wohlgeformtheit voraus, nicht jedoch umgekehrt. Informationelle Ressourcen wie textuelle Daten, ihre Segmentierung und ihre Hierarchisierung können *unabhängig* von Regelbeschreibungen existieren. Deshalb ist die Möglichkeit, sie unabhängig voneinander zu repräsentieren und variabel[3] aufeinander zu beziehen, von entscheidender Bedeutung.

Zum anderen unterscheiden sich die Strukturierungsmöglichkeiten der beiden Auszeichnungssprachen. Während bei XML-Dokumentinstanzen die Baumstrukturierung durch die Markierung der Segmentgrenzen explizit sein muss, können die rechtsseitigen Segmentgrenzen bei SGML-Dokumentinstanzen wegfallen. Ein SGML-Prozessor kann die Grenzen anhand der gegebenen Regelbeschreibung automatisch inferieren. Für die automatische Sprachverarbeitung kann dies von hoher Relevanz sein, vgl. Witt (2002, S. 30ff.). So können z.B. die `<s>` Segmente in Beispiel 2.2 automatisch in eine SGML-Dokumentinstanz integriert werden. Wenn jedoch – wie in dieser Arbeit – das Ziel darin besteht, unterschiedliche informationelle Ressourcen zueinander in Beziehung zu setzen, dürfen diese Beziehungen a priori nicht bestehen. Die Auszeichnungen müssen also explizit sein und dürfen keine impliziten Segmentierungsinformationen voraussetzen, die sich nur durch eine separate Regelbeschreibung ableiten lassen.

2.2.1.2. Struktur und Informationsgehalt von Dokumentinstanzen

Die folgenden Ausführungen beschreiben Voraussetzungen für die Validität einer XML-Dokumentinstanz und beruhen auf der XML-Spezifikation, vgl. Bray et al. (2000). Sie werden verdeutlicht anhand des Beispiels 2.2. Segmente in XML-Dokumentinstanzen werden **Elemente** genannt. Ein Element besteht aus einem **Start-Tag**, welches die linke Segmentgrenze markiert, und einem **End-Tag**, welches die rechte Segmentgrenze markiert. Start- und End-Tags, die den Namen des Elements beinhalten, sind jeweils

[3] Zwar bietet SGML wie auch XML die Möglichkeit, einer Dokumentinstanz verschiedene Dokumentgrammatiken zuzuordnen. Allerdings muss bei SGML am Anfang des Dokumenterstellungsprozesses immer die Dokumentgrammatik stehen.

durch eine spitze, nach links und nach rechts weisende Klammer gekennzeichnet, wobei der End-Tag zusätzlich einen Schrägstrich enthält. Ein Element, welches keine konkreten textuellen Daten beinhaltet, nennt man **leeres Element**. Es wird in der XML-Dokumentinstanz durch ein einzelnes Tag expliziert, welches einen Schrägstrich vor der nach rechts weisenden Klammer enthält, nach dem Muster `<element-name/>`.

Die Zusatzinformationen werden in so genannten **Attributen** als **Attribut-Wert-Paare** wiedergegeben. Diese stehen nach den Elementnamen innerhalb des Start-Tags. Einem Attributnamen ist jeweils ein in Anführungsstrichen stehender Wert zugeordnet, getrennt durch ein Gleichheitszeichen, vgl. `trans="Block"`. Ein Element kann mehrere Attribute unterschiedlichen Namens haben, wobei keine bestimmte Reihenfolge der Attribute innerhalb des Start-Tags gegeben sein muss.

Der Anspruch an Dokumente, dass sie hierarchisch strukturiert sein müssen, vgl. Abschnitt 2.1, führt zu zwei weiteren Bedingungen für die Validität eines XML-Dokuments. Erstens muss es genau ein **Wurzelelement** geben, dem alle anderen Elemente untergeordnet sind, z.B. das `<corpus>` Element. Zweitens müssen alle Elemente bzw. die entsprechenden Start- und End-Tags ineinander verschachtelt sein. Deshalb wäre die XML-Dokumentinstanz in Beispiel 2.4 nicht valide.

(2.4)
```
<corpus>
 ...
  <syll><m>le</syll>
  <syll>g</m><m>en</m></syll>
 ...
</corpus>
```

Beispiel 2.4: Nicht valide Dokumentinstanz

Die morphologische und die syllabische Segmentierung des Wortes ‚legen' lassen sich nicht in eine hierarchische Beziehung zueinander bringen.

Die XML-Spezifikation beschreibt den Aufbau einer XML-Dokumentinstanz aus einer strukturbezogenen Sicht. Die Funktion bestimmter Konstrukte z.B. zur Differenzierung von Markup und textuellen Daten mittels spitzer Klammern etc. wird definiert. In der vorliegenden Arbeit sind hingegen XML-Dokumentinstanzen in erster Linie als informationelle Ressourcen interessant. Für eine derartige, von der physikalischen Repräsentation abstrahierende Sicht ist also die Beschreibung des informationellen Ge-

haltes einer XML-Dokumentinstanz bedeutsam, wie sie in der Spezifikation des **XML Information Set** Cowan und Tobin (2004) zu finden ist. Zudem thematisiert das XML Information Set nur Informationseinheiten in einer XML-Dokumentinstanz, nicht jedoch – wie die XML-Spezifikation – gleichzeitig die Eigenschaft einer Regelbeschreibung für Dokumentklassen. Diese Beschränkung des Geltungsbereichs der Spezifikation des XML Information Set erleichtert die angestrebte Trennung zwischen der informationellen Ressource „Dokumentinstanz“, vgl. diesen Abschnitt, versus der informationellen Ressource „Regel“, vgl. Abschnitt 2.2.1.3.

Das Information Set einer XML-Dokumentinstanz kann elf verschiedene Arten von **Informationseinheiten** (Information Items) enthalten, von denen einige, für die in der vorliegenden Arbeit zu entwickelnde Methodologie bedeutsame, hier aufgelistet werden[4]:

- Informationseinheiten für Zeichen;
- Informationseinheiten für Elemente;
- Informationseinheiten für Attribute;
- Informationseinheiten für Namensräume;
- Die Informationseinheit für das Dokument;
- Die Informationseinheiten für die Dokumenttypdeklaration.

Die modellierungsrelevanten Eigenschaften werden im Folgenden vorgestellt. Dabei wird auf die Dokumentinstanz in Beispiel 2.5 zurückgegriffen:

(2.5)
```
<?xml version="1.0" encoding="iso-8859-1"?>
<sekimo:corpus meta:corpus-id="d-3-44-5"
               xmlns:sekimo="http://example.org/sekimo"
               xmlns:meta="http://example.org/meta"
>
AbbA</sekimo:corpus>
```

Beispiel 2.5: Dokumentinstanz und darin enthaltene Informationseinheiten

[4] Nicht behandelt werden Informationseinheiten für Verarbeitungsanweisungen, nicht expandiert geparste Entitäten, Kommentare, nicht geparste Entitäten sowie Notationen.

Das Information Set für dieses Dokument enthält vier Informationseinheiten für **Zeichen**, d.h. Einheiten der konkreten, textuellen Ebene. Die zeichenbezogenen Informationseinheiten umfassen u.a. die Information, in welchem Element die Zeichen vorkommen. Das Information Set für dieses Dokument enthält eine Informationseinheit für Elemente, namentlich das `<sekimo:corpus>` Element. Die Informationseinheiten für Elemente enthalten immer einen **lokalen Teil**, hier `corpus`. Optional enthalten sie ein **Namensraumpräfix**, hier `sekimo` bzw. `meta`. Namensraumpräfixe sind ebenfalls ein fakultativer Bestandteil der Informationseinheiten für Attribute. Die Informationseinheiten für Attribute bestehen obligatorisch aus einem lokalen Namen, z.B. `corpus-id` und einem Wert, z.B. `d-3-44-5`. Zusätzlich ist zur Informationseinheit für das Dokument angegeben, um welche Version von XML – 1.0 – es sich handelt und welche Zeichenkodierung – iso-8859-1 – besteht. Die Zeichenkodierung wird detailliert in Abschnitt 2.2.2.1 behandelt.

Namensräume dienen dazu, die **Auszeichnungsvokabulare** (Elementtypen[5] und Attributnamen) zu differenzieren, vgl. Bray et al. (1999). Sie bestehen aus einem Namen und einem Präfix. Die Informationseinheiten der zwei Namensräume in Beispiel 2.5 haben die Namen `http://example.org/sekimo` und `http://example.org/meta`, sowie die Präfixe `sekimo` und `meta`. Der Mechanismus der Namensräume und die damit verbundenen Informationseinheiten erfüllen eine wichtige Aufgabe bei der Differenzierung informationeller Resssourcen.

Diese Aufgabe erfüllen sie nicht nur in der primären Informationsstrukturierung, sondern auch in der konzeptuellen Modellierung. Der Name eines Namensraums ist nämlich nach dem Muster von **URI** (Uniform Resource Identifiers) aufgebaut. Ein URI ist eine Zeichenkette mit dem Zweck, eine abstrakte oder konkrete Ressource zu identifizieren, vgl. Berners-Lee et al. (1998). URI sind Bestandteil vieler Standardisierungen und -standardisierungsbestrebungen[6], so etwa auch bei der Entwicklung von Standards zur Repräsentation konzeptueller Ressourcen (vgl. Abschnitt 2.2.3.3). URI erlauben durch **Fragment Identifier** nicht nur die Identifikation von Dokumenten, sondern auch von

[5] Zum Begriff der Typen vgl. Abschnitt 2.2.1.3.

[6] Im Rahmen der Internationalisierung von XML verwenden neuere Standardisierungsbestrebungen sogenannte **IRI** (Internationalized Resource Identifiers), vgl. http://www.w3.org/TR/xml-names11/#dt-IRI. URI mit Zeichen, die in der Version 3.2 oder neueren Fassungen des Unicode-Standards definiert sind, können automatisch in IRI transformiert werden, weshalb in dieser Arbeit auf URI zurückgegriffen wird.

einzelnen Informationseinheiten innerhalb von Dokumenten.[7] Auf diese Weise lassen sich z.B. Elements- oder Attributsdeklarationen in Dokumentgrammatiken als auch die entsprechenden Informationseinheiten in Dokumentinstanzen eindeutig identifizieren. Die Verwendung von URI erleichtert somit die in dieser Arbeit angestrebte Verbindung der unterschiedlichen informationellen Ressourcen.

Die Informationseinheiten für Elemente enthalten zusätzlich zu den beschriebenen Informationen eine Liste von Informationseinheiten über die unmittelbar untergeordneten **Children**, d.h. untergeordnete Elemente, textueller Inhalt sowie Attribute. Zudem gibt es eine Information zu einem übergeordneten Element **Parent**. In Beispiel 2.5 besitzt die Informationseinheit zum `<sekimo:corpus>` Element Informationen zu den Zeichen `AbbA`, sowie die Attribute `meta:corpus-id`, `xmlns:sekimo` und `xmlns:meta`. Übergeordnet ist dem `<sekimo:corpus>` Element eine Informationseinheit für das Dokument. Die Attribute sind eine ungeordnete Menge. Elemente und textueller Inhalt hingegen stehen in Listen. Die hierarchische Strukturierung einer XML-Dokumentinstanz spiegelt sich also durch die Repräsentation von Elementen und Text als geordnete Mengen wieder.

Die letzte zu behandelnde Informationseinheit steht für die **Dokumenttypdeklaration** (Document type declaration), welche die Verbindung der Dokumentinstanz zu Regelbeschreibungen für eine Klasse von Dokumenten, einer **Dokumentgrammatik** schafft, vgl. Abschnitt 2.2.1.3. Eine Dokumenttypdeklaration kann nur auf eine bestimmte, im XML-Standard definierte Form von Dokumentgrammatiken verweisen, auf eine sogenannte **DTD** (Document type definition, Dokumenttypdefinition). DTDs, die bereits für SGML verwendet wurden – dort allerdings in einem erweiterten Umfang –, sind das derzeit verbreiteste Mittel zum Ausdruck von Regeln für Dokumentklassen.

Das XML Information Set beschreibt den Informationsgehalt einer Dokumentinstanz auf eine Weise, die eine **aussagenlogische Interpretation** ermöglicht. Informationseinheiten lassen sich als eine Menge von Aussagen verstehen, die in einer Graphenstruktur miteinander verknüpft sind. Informationseinheiten bilden Knoten, d.h. die Argumente der Aussagen; die Eigenschaftsbeschreibungen bilden Kanten, d.h. die Prädikate. Die Graphenstruktur der diskutierten Dokumentinstanz ist in Abbildung 2.3 visualisiert. Informationseinheiten bestehen aus einem Namen, z.B. Element, Attribut, Zeichen, und verschiedenen Eigenschaften, beispielsweise *has attribute* oder *has children*. In der Abbildung sind die Eigenschaften durch gerichtete, benannte Verbindungen zwischen den

[7] Gegenstand gegenwärtiger Standardisierungsbemühungen ist die Frage, wie das Verhältnis von `ID` Attributen und Fragment Identifier ist, vgl. http://www.w3.org/TR/webarch/#xml-fragids

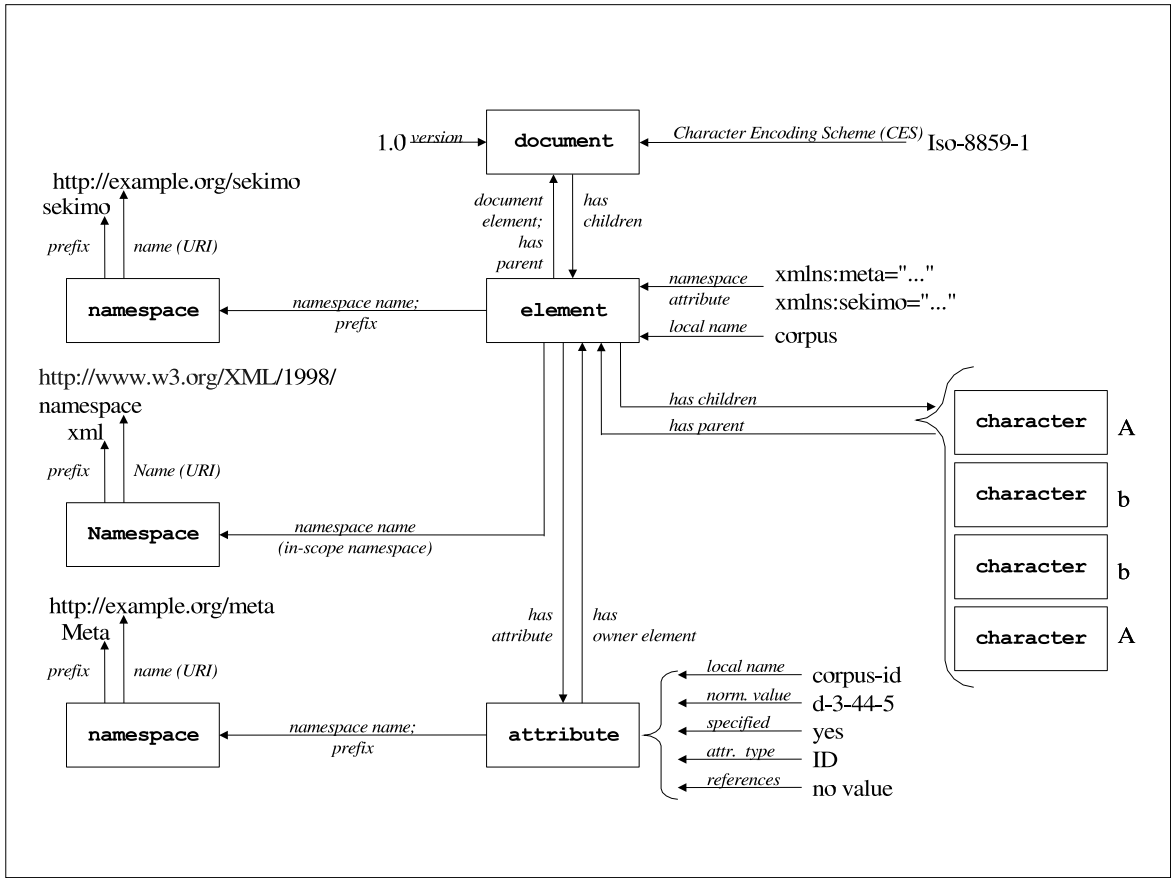

Abbildung 2.3.: Visualisierung eines Information Set

Informationseinheiten dargestellt.

Diese Sicht hat zum einen eine Auswirkung auf die Bestimmung der informationellen Ressourcen, die in Dokumentinstanzen als Teil der primären Informationsstrukturierung relevant sind. Nicht Elemente als durch spitze Klammern markierte Zeichenfolgen oder Attribute mit dem Repräsentationsmuster `Attributname = Attributwert` etc. sind relevant, sondern **Mengen von Informationseinheiten**. Einige Teilmengen sind geordnet, vgl. die Ordnung von Elementen und textuellem Inhalt, andere stellen eine ungeordnete Menge dar, z.B. Attribute. Die Beschreibung des Informationsgehalts einer Dokumentinstanz als Menge ist eine wichtige Voraussetzung für ihre vertikale Verbindung zu anderen informationellen Ressourcen, weil der Informationsgehalt so von seiner physikalischen Repräsentation losgelöst wird. Andere informationelle Ressourcen können ebenfalls als Mengen betrachtet werden. Es gilt, die Mengen aufeinander zu beziehen.

Zum anderen lässt sich die aussagenlogische Interpretation von Informationseinheiten auch durch Standards zur Repräsentation konzeptueller Modelle wiedergeben, z.B. im Format von RDF Schema, vgl. Abschnitt 2.2.3.3 sowie Cowan und Tobin (2004, Appendix E). Dies ist nach der Beschreibung von Dokumentinstanzen als Mengen ein weiterer Schritt in Richtung einer vertikalen Verbindung informationeller Ressourcen: Es gilt Mengen von Aussagen über Dokumentinstanzen mit Mengen von Aussagen über konzeptuelle Modelle zu verknüpfen.

2.2.1.3. Regeln und Bedingungen

Die Informationseinheit „Dokumenttypdeklaration“ verweist auf Regelbeschreibungen für eine Klasse von Dokumenten im bereits erwähnten Format der DTDs. Die Regeln, die sich mittels DTDs ausdrücken lassen, haben immer einen formalgrammatischen Charakter, welcher der Ausdruckskraft einer kontextfreien Grammatik entspricht, vgl. Beispiel 2.3 in Abschnitt 2.2.1.1. Dies verdeutlicht die exemplarische DTD in Beispiel 2.6.

Die für die informationelle Ressource „Regel“ relevanten Konstrukte in einer DTD sind **Deklarationen** für Elemente und Attribute. Andere Konstrukte wie Deklarationen von Entitäten, Parameterentitäten[8] und Notationen werden hier nicht behandelt, da sie für die Regeln nicht bedeutsam sind. Elemente besitzen einen Namen, z.B. `<corpus>`, und ein **Inhaltsmodell**, z.B. `(w+)`. Der Name lässt sich als die linke Seite einer Produktionsregel in einer kontextfreien Grammatik auffassen, das Inhaltsmodell als die rechte

[8] Sie bieten einen Mechanismus zur Modularisierung der Regeln, vgl. Abschnitt 2.2.4.2.

```
(2.6) <!ELEMENT corpus (s+)>
      <!ELEMENT s (w+)>
      <!ELEMENT w (#PCDATA)>
      <!ATTLIST s trans CDATA #IMPLIED>
      <!ATTLIST w trans CDATA #IMPLIED
                  cat CDATA   #REQUIRED>
```

Beispiel 2.6: Dokumenttypdefinition

Seite. Im Gegensatz zu einer kontextfreien Grammatik ist das Startsymbol, d.h. das Wurzelelement, in der DTD nicht festgelegt. Dies geschieht in der angesprochenen Dokumenttypdeklaration, also innerhalb der Dokumentinstanz. Mit der vorliegenden DTD können Dokumentinstanzen validiert werden, die in der Dokumenttypdeklaration als Wurzelelement eines der drei Elemente `<corpus>`, `<s>` oder `<w>` festlegen.

Es gibt drei Varianten von Inhaltsmodellen. Erstens kann das Inhaltsmodell andere Elemente enhalten, z.B. `<w>` als Inhalt von `<s>`. Zweitens kann es leer sein, d.h. in der Dokumentinstanz muss ein leeres Element stehen. Drittens ist nur Text als Elementinhalt vorsehen. Im letzten Fall wird das **Schlüsselwort** `(#PCDATA)` bei der Elementdeklaration verwendet, wie z.B. bei dem Element `<w>`. Ein zusätzlicher Fall ist **Mixed Content** (gemischter Inhalt): Das Inhaltsmodell enthält sowohl textuelle Daten als auch andere Elemente. Die Abfolge von Text und Elementen ist allerdings frei.

Attribute werden innerhalb von **Attributlisten** deklariert, die jeweils einem Element zugeordnet sind. Die Attribute haben einen Namen, z.B. `trans`, einen im Folgenden beschriebenen **Datentyp**, sowie ein Schlüsselwort, welches ihren Status festlegt – z.B. fakultativ `#IMPLIED` versus obligatorisch `#REQUIRED`. Default-Werte werden durch das Schlüsselwort `#FIXED` angezeigt.

Datentypen[9] für Attribute in DTDs geben an, welche Menge von Werten, der so genannte **Wertebereich**, für ein Attribut zulässig ist. Im Beispiel 2.6 gibt die Typbezeichnung **CDATA** an, dass der Attributwert eine Zeichenkette sein muss. In DTDs gibt es z.B. auch Typen, die eine Auswahlliste von Zeichenketten ohne Leerzeichen als mögliche Werte haben, eine sogenannte **NMTOKEN-Gruppe**, oder den Typ **ID**, der

[9] Das Konzept der Datentypen wird ausführlich in Abschnitt 3.2 behandelt.

angibt, dass der Wert des Attributes nur einmal in der Dokumentinstanz vorkommen darf.

DTDs weisen einige Nachteile auf, die gegen ihre Anwendung zur primären Informationsstrukturierung sprechen. Zum einen sind DTDs keine XML-Dokumente, d.h. die Regeln in DTDs und die Informationseinheiten in Dokumentinststanzen, auf welche die Regeln angewendet werden, verwenden unterschiedliche Formate. Dies erschwert den übergreifenden Zugriff auf die informationellen Ressourcen. Zum anderen hat das vorgestellte Typenkonzept in DTDs den Nachteil, nicht auf den Inhalt von Elementen anwendbar zu sein, und es erlaubt dem Benutzer nur im geringen Maße, eigene Datentypen zu definieren. Des weiteren wurde das Konzept der Namensräume, vgl. Abschnitt 2.2.1.2, erst nach der Verabschiedung der DTD-Spezifikation entwickelt. Die Identifikation und Separierung von Regeln ist deshalb mit DTDs nicht unter Rückgriff auf Namensräume realisierbar.

Die Nachteile von DTDs haben in den letzten Jahren zu einer Veränderung des Begriffs von Strukturbeschreibungen für Dokumente geführt. Während DTDs zunächst als eine Form von Dokumentgrammatik angesehen wurden, hat sich inzwischen der Begriff der **Schemasprache** etabliert. Hauptzweck der meisten Schemasprachen ist die Beschreibung von formalen Regeln für Dokumentklassen; die Schemasprache DTD erfüllt diesen Zweck in für viele Anwendungen hinreichender Weise. Zusätzlich dienen einige Schemasprachen der Formulierung von **Bedingungen**, die sich nur anhand von gegebenen XML-Dokumentinstanzen überprüfen lassen und die nicht die baumstrukturierte, an einer kontextfreien Grammatik orientierte Sicht auf Dokumentinstanzen fokussieren. Dies führte zur Differenzierung in **grammatikbasierte und constraintbasierte Schemasprachen**, bzw. Schemasprachen zur Definition von Dokumentklassen versus Schemasprachen zur Anwendung von **Bedingungen** (Constraints) auf Dokumentinstanzen, vgl. Lee und Chu (2000, Abbildung 1)[10]. DTDs stellen den prototypischen Vertreter der ersten Kategorie dar. Der prominenteste Vertreter der bedingungsbasierten Schemasprachen ist **Schematron**, vgl. Jelliffe (2000). Beispiel 2.7 zeigt in einem Auschnitt eines Schematron-Dokuments eine Bedingung, die sich nicht mit DTDs ausdrücken lässt.

Zunächst wird im `context` Attribut des `<rule>` Elements die Menge der Informa-

[10] Als weitere Möglichkeit zur Strukturbeschreibung von Dokumenten führt Lobin (2004a) den Einsatz von Beispieldokumenten auf. Ein solches Verfahren kommt in der vorliegenden Arbeit jedoch nicht zum Einsatz, da die informationellen Ressourcen „Dokument“ und „Strukturbeschreibung für Dokumente“ nicht vermischt werden sollen.

```
(2.7) <rule context="w[@cat='V-IMP']">
        <assert
         test="preceding-sibling::w[1]
               [@cat='V-TE' or @cat='ACC']">
         Das Verb im Imperativ muss unmittelbar
         nach einem Verb in der TE-Form oder
         nach einem Akkusativmarker stehen.
        </assert>
      </rule>
```

Beispiel 2.7: Schematron-Dokument

tionseinheiten bestimmt, für die eine Bedingung erfüllt sein soll. Dies geschieht durch eine **Pfadbeschreibungssprache**, in diesem Fall **XPath**, vgl. Clark und deRose (1999). Der Pfadausdruck selektiert in der Dokumentinstanz alle `<w>` Elemente, die ein `cat` Attribut mit dem Wert `V-IMP` haben. Für diese Informationseinheiten wird anschließend eine Bedingung innerhalb des `<assert>` Elements formuliert, ebenfalls unter Verwendung von XPath: unmittelbar links neben den selektierten Elementen muss ein `<w>` Element mit einem `cat` Attribut stehen, das den Wert `V-TE` oder `ACC` besitzt. Diese Bedingung wird als Inhalt des `<assert>` Elements natürlichsprachlich beschrieben. Für eine Dokumentinstanz wie in Beispiel 2.2 ist diese Bedingung erfüllt.

Dieses Schematron-Dokument beschreibt also Bedingungen hinsichtlich der linearen Abfolge von Informationseinheiten, unabhängig oder in Ergänzung zu einem Inhaltsmodell. Zudem sind Schematron-Dokumente zugleich XML-Dokumente. Die Generierung von URI für bedingungskonforme Informationseinheiten, vgl. Abschnitt 2.2.1.2, in Dokumentinstanzen ist deshalb mit Schematron leicht realisierbar, ebenso wie die Verbindung zu konzeptuellen informationellen Ressourcen.

Andere der Vielzahl von Schemasprachen, welche in den letzten Jahren entwickelt wurden[11], sind grammatikbasiert und konzentrieren sich auf die angesprochene Typenproblematik oder die Erweiterung der formalgrammatischen Ausdruckskraft von Schemasprachen. Der vom **W3C** (World Wide Web Consortium, vgl. `http://www.w3.org`)

[11] Einen Überblick gibt z.B. der erwähnte Aufsatz von Lee und Chu (2000).

als Nachfolger für DTDs propagierte Standard **XML Schema** spezifiziert u.a. ein umfangreiches Repertoire an Datentypen, vgl. Biron und Malhotra (2001)[12], das auch vom Benutzer selbst erweitert werden kann und – im Gegensatz zu DTDs – sowohl für die Typisierung des Inhalts von Attributen als auch von Elementen verwendbar ist. Benutzerdefinierte Datentypen stellen Erweiterungen oder Einschränkungen der vordefinierten Datentypen dar.

(2.8)
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:simpleType name="japaneseCat">
  <xs:restriction base="xs:string">
   <xs:pattern value="ADJ"/>
   <xs:pattern value="ACC"/>
   <xs:pattern value="LOC"/>
   <xs:pattern value="N"/>
   <xs:pattern value="V-TE"/>
   <xs:pattern value="V-IMP"/>
  </xs:restriction>
 </xs:simpleType>
</xs:schema>
```

Beispiel 2.8: Datentypen in XML Schema

In Beispiel 2.8 wird ein Datentyp namens `japaneseCat` definiert, der auf dem eingebauten Datentyp `xs:string` basiert. Hier wird eine weitere Funktion von Namensräumen deutlich. Sie dienen nicht nur der Abgrenzung von Elementtypen und Attributsnamen, sondern auch von Datentypen. Im vorliegenden Fall drückt das Namensraumpräfix `xs` aus, dass es sich um den entsprechenden Namensraum `http://www.w3.org/2001/XMLSchema` handelt. Die möglichen Werte des benutzerdefinierten Datentyps werden in Form einer Auswahlliste in den `value` Attributen der `<pattern>` Elemente wiedergegeben. Dieser Datentyp kann z.B. eine Grundlage für die Deklaration von `cat` Attributen sein, wie sie in der Dokumentinstanz aus Beispiel 2.2 zur Anwendung kommen.

XML Schema erlaubt auch die benutzerdefinierte Definition und Ableitung von Typen

[12] Beispiele für vordefinierte Datentypen sind `positiveInteger` für positive, ganzzahlige numerische Werte oder `anyURI` für URI.

für Inhaltsmodelle von Elementen mit Kindelementen, und verwendet dafür eine Vielzahl von Konstrukten. Diese Möglichkeiten führen bereits zu einer weiteren informationellen Ressource, der sekundären Informationsstrukturierung (vgl. Abschnitt 2.2.4).

Trotz der beschriebenen Erweiterungen gegenüber DTDs ist dieser Standard z.B. von Murata et al. (2001) kritisiert worden. Die Autoren untersuchen die Ausdruckskraft von Schemasprachen unter Rückgriff auf das Konzept der **regulären Baumgrammatiken**, die – in absteigender Ausdruckskraft sortiert – vier Varianten aufweisen: **regular**, **restrained-competition**, **single-type** und **local**. Die Ausdruckskraft von DTDs und XML Schema wird in Beispiel 2.9 deutlich, dass auf Lobin (2003, Abschnitt 2.1) beruht.

(2.9)

	Beispiel	*Ausdruckskraft*
(1)	*word* → *morpheme*	local: nur linke Seite legt rechte Seite fest.
(2a)	*word-in-text-element* → *T*	single-type: linke Seite und Kontext legt rechte Seite fest.
(2b)	*word-in-sentence-element* → *morpheme*	single-type
(3a)	*corpus* → *word-variant1*	regular: Konkurrenz von Inhaltsmodellen gleichnamiger Elemente ist erlaubt.
(3b)	*corpus* → *word-variant2*	regular

Beispiel 2.9: Dokumentgrammatische Regeln unterschiedlicher Ausdruckskraft

Die Regel (1) lässt sich in einer XML-DTD formulieren. Der Aufbau des Inhaltsmodells des `<word>` Elements wird nur durch dieses Element selbst bzw. seinen Namen bestimmt. Anders formuliert: Das Symbol auf der linken Seite der Regel legt die Symbole auf der rechten Seite fest. Deshalb haben XML-DTDs die Ausdruckskraft **local**. Regel (2) lässt sich nicht mit XML-DTDs formulieren. In dieser Regel gibt es zwei Varianten des `<word>` Elements: (2a) innerhalb von `<text>` Elementen, oder (2b) in `<sentence>` Elementen. Die Varianten weisen, in Abhängigkeit vom übergeordneten Kontext, unterschiedliche Inhaltsmodelle auf. Mit anderen Worten, der Kontext, in dem das Symbol der linken Seite der Regel steht, legt die Symbole der rechten Seite fest. XML Schema erlauben diese Regeln, verbietet jedoch Regeln wie (3a) bzw (3b). Hier beinhaltet das Inhaltsmo-

dell des `<corpus>` Elements zwei konkurrierende Deklarationen des `<word>` Elements, `<word-variant1>` versus `<word-variant2>`. Der Typ von `<corpus>` ist also ambigue. Derartige Ambiguitäten sind in XML Schema nicht erlaubt, ein Element muss eindeutig typisierbar sein. Deshalb lässt sich XML Schema der Ausdruckskraft **single-type** zuordnen. Ambige Inhaltsmodelle wie (3a) versus (3b) lassen sich mittels der Schemasprache **RELAX NG** Clark und Murata (2001) ausdrücken. Sie wird als **regular** bezeichnet.

Die von Murata et al. (2001) formulierte Kritik an XML Schema bezieht sich offensichtlich auf die Regeltypen, welche XML Schema bieten. Die Beschränkung von XML Schema auf single-type hat jedoch eine Motivation. Die Typisierbarkeit von Informationseinheiten soll jederzeit gegeben sein, um z.B. den Zugriff auf XML-Dokumentinstanzen aus Datenbanken zu erleichtern. Ambiguitäten in den Dokumentinstanzen würden eine effektive Implementation von Zugriffssoftware erschweren. Die Typisierung ist zudem ein unverzichtbares Instrumentarium für XML Schema, wenn man es zur sekundären Informationsstrukturierung einsetzen möchte. Sie wird in Abschnitt 2.2.4.2 deshalb noch einmal aufgegriffen. Für die vertikale Verbindung informationeller Ressourcen stellt sich die Frage, was von größerer Bedeutung ist. Soll die primäre Informationsstrukturierung eine hohe dokumentgrammatische Ausdruckskraft bereitstellen, oder hat die Typisierbarkeit von Informationseinheiten größere Bedeutung? Da die vorliegende Arbeit Segmentierungen und Hierarchisierungen als informationelle Ressourcen auffasst, erscheint eine Schemasprache mit starker dokumentgrammatischer Ausdruckskraft wichtiger. Die eindeutige Typisierbarkeit wäre von Bedeutung, wenn sekundäre Informationsstrukturierung und primäre Informationsstrukturierung in einem Format zu leisten wären und wenn datenzentrierte Auszeichnungen Gegenstand des Interesses wären, die in Abschnitt 2.1 angesprochen wurden. Die vorliegende Arbeit zielt jedoch auf eine klare Trennung der informationellen Ressourcen ab. Sekundäre und primäre Informationsstrukturierung sollen nicht miteinander vermischt werden. Deshalb wird für die Repräsentation der informationellen Ressource „Regel“ RELAX NG der Vorzug gegenüber XML Schema gegeben.

2.2.1.4. Vordefinierte Auszeichnungsvokabulare

Die meisten bestehenden Auszeichnungsvokabulare werden mittels DTDs definiert. XML Schema und RELAX NG kommen noch verhältnismäßig selten zum Einsatz. Das verbreiteste Vokabular ist **HTML** (Hypertext Markup Language, vgl. Pemperton et al. (2002)).

Die Vielfältigkeit der Domäne von über das Internet verfügbaren Hypertextdokumenten führt dazu, dass HTML ein sehr vielschichtiges Vokabular ist. Element- und Attributdeklarationen sind nur schwer thematisch zu trennen. Es gibt Elemente zur Erzeugung von **Hyperlinks** wie das `<a>` Element, oder zur Segmentierung der Textstruktur wie `<p>` oder `<div>`. Um die Kombination von HTML mit anderen Auszeichnungsvokabularen wie **SVG** für Vektorgraphiken oder **MathML** für die Darstellung mathematischer Formeln zu erleichtern, wurde der Standard in einer modularen Version **XHTML** reformuliert, vgl. Altheim et al. (2001).

Das Prinzip der Modularisierbarkeit ist eine wesentliche Eigenschaft der Dokumentgrammatik der **TEI** (Text Encoding Initiative, vgl. Sperberg-McQueen und Burnard (1994)). Die TEI wurde zunächst in SGML realisiert und liegt inzwischen als Version P4 in XML vor. Die nächste Version P5 ist derzeit in Bearbeitung. Vor allem für verschiedene geisteswissenschaftliche Anwendungen wie die Auszeichnung von literarischen Texten oder Lexika, aber auch für linguistisch relevante Bereiche wie Sprachkorpora, Merkmalsstrukturen oder Schriftsysteme (vgl. Abschnitt 2.2.2.1), definiert die TEI verschiedene Module. Für konkrete Anwendungen gilt es, die Module in adäquater Weise zu kombinieren, wofür inzwischen semi-automatische Tools[13] zur Verfügung stehen.

Am Beispiel verschiedener Vokabulare, die unter anderem auf der TEI beruhen, soll im Folgenden der Nutzen von Modularisierung und Spezialisierung demonstriert werden. Eine Spezialisierung der TEI für Auszeichnungen linguistischer Daten stellt das Vokabular von **CES** (Corpus Encoding Standard, vgl. Ide (1998)) dar, das inzwischen in einer XML-Version **XCES** vorliegt, vgl. Ide und Romary (2003). CES und XCES beinhalten vor allem Spezialisierungen des Auszeichnungsvokabulars für lexikalische, wortbezogene Kategorien wie Wortarten, Wortformen, Morpheme etc. Für die Auszeichnung sprachspezifischer, deutscher Kategorien wurde im Rahmen des Projekts **DeReKo** (Deutsches Referenzkorpus) ein von Ule (2002) vorgestelltes Auszeichnungsvokabular für lingistische Kategorien des Deutschen entwickelt, welches CES und das von Schiller und Teufel (1995) beschriebene **STTS** (Stuttgart-Tübingen Tagset) integriert. Ein Grund, bei DeReKo projektspezifische Namen für Elemente und Attribute zu verwenden, liegt in der Größe des annotierten Korpus: Es umfasst mehrere Millionen ausgezeichnete Wortformen. Durch kurze Namen und Default-Werte in der DeReKo-spezifischen Dokumentgrammatik kann die Verarbeitungsgeschwindigkeit wesentlich erhöht werden.

[13] Vgl. http://www.tei-c.org/pizza.html

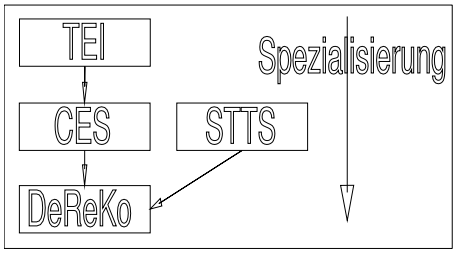

Abbildung 2.4.: Das Verhältnis verschiedener Auszeichnungsvokabulare

Das Verhältnis von TEI, CES, DeReKo und STTS visualisiert die Abbildung 2.4. Die Bildung des Auszeichnungsvokabulars von DeReKo stellt eine manuelle Spezialisierung gegebener Vokabularien dar.

Für den Einsatz von Auszeichnungsvokabularen als informationelle Ressource der primären Informationsstrukturierung müssen mehrere Bedingungen erfüllt sein. Zum einen müssen die Beschreibungen der Kategorien unter Rückgriff auf die entsprechenden texttechnologischen Standards repräsentiert werden, d.h. als Dokumentgrammatik im Format einer DTD, XML Schema etc. Das Projekt **MATE** (Multilevel Annotation, Tools Engineering, vgl. Klein et al. (1998)) verfolgte unter anderem das Ziel, verschiedene Kategorieninventare in XML-Dokumentgrammatiken zu überführen. Das Ergebnis waren spezifische Dokumentgrammatiken für die jeweiligen Inventare. Eine weitere Bedingung liegt in der Überführung vorhandener Daten nach XML, da viele Daten bereits vor der Entwicklung von XML erstellt wurden. Exemplarisch können hier die Korpora des Verbmobil-Projektes genannt werden, vgl. Wahlster (2000). Witt et al. (2000) haben gezeigt, wie die Überführung dieser Daten und ihre Anreicherung mit lexikalischen Informationen automatisch realisiert werden kann.

Die Modularisierung und Relationierung unterschiedlicher Auszeichnungsvokabulare verfolgt u.a. den Zweck, Vokabulare aufeinander beziehen zu können und dabei ein übergreifendes Vokabular für den Austausch zwischen den spezifischen Vokabularen zur Verfügung zu haben. Das Beispiel 2.10 verdeutlicht die dahinterliegende Motivation. Die drei Auszeichnungen des Wortes „kudasai“ entstammen drei fiktiven, verschiedenen Auszeichnungsvokabularen und enthalten die gleichen Informationen, d.h. eine wortbezogene Segmentierung und Informationen über die Wortform Imperativ. Die Informationen werden jedoch durch unterschiedliche Kombinationen von Informationseinheiten realisiert:

Ein <w> Element mit einem Attribut, ein <w> Element mit zwei Attributen sowie ein <v> Element mit einem Attribut. Eine formale Beschreibung der Beziehungen zwischen den drei Auszeichnungsvokabularen würde die automatische Transformation ihrer Dokumentinstanzen erleichtern.

(2.10)
```
<w type="V-IMP">kudasai</w>
<w cat="V" form="IMP">kudasai</w>
<v form="IMP">kudasai</>
```

Beispiel 2.10: Unterschiedliche Auszeichnungen eines identischen Textes

Im erwähnten Projekt MATE wurden manuelle, umfangreiche Analysen bestehender Auszeichnungsvokabulare gemacht, unabhängig davon, ob diese in XML-Dokumentgrammatiken vorlagen oder nicht. Experten für verschiedene linguistische Beschreibungsebenen, z.B. Morphosyntax, Prosodie, Koreferenz[14], erarbeiteten Vorschläge für übergreifende Vokabulare, die schließlich als XML-Dokumentgrammatik repräsentiert wurden. Dieses Vorgehen ist komplementär zu dem Verfahren bei DeReKo, wo aus einem allgemeinem Auszeichnungsvokabular (TEI) anhand verschiedener sprach- und domänenspezifischer Parameter ein spezielleres gebildet wurde. Beide Verfahren ordnen jedoch die speziellen und generellen Auszeichnungsvokabulare manuell einander zu. D.h. es ist nicht möglich, automatisch zwischen den verschiedenen Granularitätsebenen zu wechseln. Diese Möglichkeit ist jedoch zur Relationierung unterschiedlicher informationeller Ressourcen unabdingbar. Semi-automatische Verfahren, die diesen Wechsel realisieren, werden in Abschnitt 2.2.4 vorgestelllt.

2.2.1.5. Grenzen primärer Informationsstrukturierung

Die Diskussion der primären Informationsstrukturierung hat drei Themenbereiche deutlich gemacht, denen man sich bei einer vertikalen Sicht auf die primäre Informationsstrukturierung widmen muss:

1. Die Typisierung von Informationen in Dokumentinstanzen, vgl. Abschnitt 2.2.1.3. XML-DTDs bieten nur wenig Möglichkeiten zur Typisierung. Im Gegensatz dazu stellt XML Schema ein umfangreiches vordefiniertes Typensystem bereit und erlaubt die Definition eigener Typen.

[14]Der Begriff **Koreferenz** wird in Abschnitt 6.3 näher erläutert.

2. Die Ausdruckskraft von Dokumentgrammatiken, vgl. ebenfalls Abschnitt 2.2.1.3. Die Typisierung, welche XML Schema erlaubt, wird mit einer Beschränkung hinsichtlich der Ambiguität von Dokumentgrammatiken erkauft. RELAX NG hat diese Beschränkung nicht, diese Schemasprache erlaubt allerdings keine Typisierung.

3. Der Zwang zu einer singulären Hierarchie, vgl. Abschnitt 2.2.1.2. XML erlaubt keine Auszeichnung von nicht hierarchischen Beziehungen zwischen unterschiedlichen Ebenen. In einem XML Dokument müssen alle Ebenen in eine Hierarchie integriert werden.

Das erste Problem lässt sich lösen, indem die Typisierung auf die Ebene der sekundären Informationsstrukturierung verschoben wird. D.h. die primäre Informationsstrukturierung muss nur eine dokumentgrammatische Beschreibung und die Dokumentinstanzen umfassen. Dies führt zur Lösung des zweiten Problems: Die Schemasprache RELAX NG kommt zur Beschreibung von Regeln für Dokumentklassen zum Einsatz. In Ergänzung zu den in Dokumentgrammatiken formulierten Regeln treten Bedingungen, deren Einhaltung innerhalb von Dokumentinstanzen z.B. mittels Schematron geprüft werden kann. Das dritte Problem ist jedoch mit texttechnologischen Standards momentan nicht lösbar.

Eine hierarchische Sicht auf Texte bildet die Grundlage der **OHCO-Hypothese**. Diese Hypothese, die z.B. von Renear et al. (1996) diskutiert wird, fasst Texte als eine geordnete Hierarchie von Inhaltsobjekten (Ordered Hierarchy of Content Objects) auf. Würden Texte diese Eigenschaft besitzen, ließen sie sich problemlos mit den Ausdrucksmöglichkeiten von Auszeichnungssprachen modellieren. Das Beispiel 2.4 – die Auszeichnung von Morphemen und Silben – hat jedoch gezeigt, dass dies nicht der Fall ist. Verschiedene konzeptuelle Modelle rechtfertigen unterschiedliche Sequenzierungen und Hierarchisierungen bei gleichen, textuellen Daten. Der Standpunkt von Caton (2002, S. 11f.) geht in dieser Hinsicht am weitesten: Eine adäquate Beschreibung der Eigenschaften von Texten ist seiner Ansicht nach mit den auf Hierarchien fokussierten Möglichkeiten von Auszeichnungssprachen unmöglich.

Zur Lösung des dritten Problems greift die vorliegende Arbeit auf den Ansatz der multiplen Informationsstrukturierung zurück, vgl. Witt (2004b). Er wird in Abschnitt 4.4.1.3 ausführlich vorgestellt.

2.2.2. Zeichen

2.2.2.1. Zeichenkodierung

Es mag verwundern, dass die Diskussion der informationellen Ressource „Zeichen“ auf die Behandlung der primären Informationsstrukturierung folgt, also von einer abstrakten zu einer konkreten Ressource übergegangen wird. Dieser „Rückschritt“ liegt in der Verbindung der informationellen Ressourcen begründet, welche die in der vorliegenden Arbeit entwickelte Methodologie ermöglichen will, vgl. Abschnitt 2.2.2.2.

Die konkrete textuelle Ebene besteht aus einer Folge von Zeichen. Im Beispiel 2.1 sind dies dreiundreißig Zeichen plus sechs Leerzeichen. Das Beispiel stellt jedoch eine Lateintranskription des Japanischen dar. Das originäre, japanische Schriftsystem mit einem Umfang von mehreren tausend piktographischen Zeichen (**Kanji**) und zwei syllabischen Schriftsystemen (**Kana**) verdeutlicht die Notwendigkeit, die maschinelle Repräsentation von Zeichen aus unterschiedlichen Kulturräumen zu standardisieren. Ansonsten wäre es nicht möglich, auf diese Zeichen als informationelle Ressource zu zugreifen: In der Lateintranskription gehen differenzierende Informationen verloren, da das Japanische viele Homophone aufweist. Abbildung 2.5 visualisiert eine Reihe von Homophonen, die in Lateintranskriptionen alle durch die gleiche Buchstabenfolge ‚kou‘ wiedergeben werden.

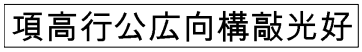

Abbildung 2.5.: Beispiele für homophone, piktographische Zeichen im Japanischen

Zeicheninventare werden in maschinenlesbarer Form als so genannte **Zeichensätze** (engl. character set) repräsentiert. Sie sind oft sprach- und kulturraumspezifisch. Zeichensätze der **iso-8859** Familie wie ISO/IEC8859-1 (1998) werden z. B. für die meisten Sprachen Westeuropas und Amerikas verwendet. Um wie in Abbildung 2.5 sowohl japanische Zeichen als auch z.B. deutsche Umlaute in einem Text maschinell verarbeiten zu können, wurde der **Unicode-Standard** (Graham, 2000) geschaffen, der zur Zeit in der Version 4.0 vorliegt. In Unicode sind Zeichen aus einer Vielzahl von weltweit gebräuchlichen Schriftsystemen enthalten. Die Zeichen sind in sogenannte **Skripts** eingeordnet, wobei über Sprach- bzw. Kulturräume generalisiert wird. Dieser Standard findet auch in der Auszeichnungssprache XML Anwendung.

Die Kodierung von Zeichen ist ein prinzipiell nicht abschließbarer Prozess, der inzwischen selbst standardisiert wird, vgl. Dürst et al. (2002). Dabei sind folgende Schritte

der Zeichenkodierung zu unterscheiden:

1. Auswahl einer Menge von zu kodierenden Zeichen, z.B. Alphabetzeichen inklusive Umlauten und japanische Kanji und Kana.
2. Zuordnung eines eindeutigen, ganzzahlig positiven Identifikatoren zu jedem Zeichen, z.B. `65` für den Buchstaben `A`.
3. Auswahl einer basalen Dateneinheit zur maschinellen Repräsentation von Zeichen, z.B. 8- oder 16-Bit Sequenzen.
4. Auswahl eines Serialisierungsschemas für die Repräsentation. Es muss bei 8-Bit Sequenzen festgelegt werden, welche 8-Bit Einheit höher geordnete oder tiefer geordnete Werte repräsentiert.
5. Die Schritte zwei bis vier werden durch eine eindeutig benannte **Kodierung** spezifiziert. Die Kodierung **UTF-8** steht für ein Serialisierungsschema des **universal character set** (UCS), das den vom Unicode-Konsortium standardisierte Zeichenvorrat umfasst. Der numerische Identifikator eines Zeichens wird in UTF-8 durch 8-Bit Sequenzen serialisiert. **UTF-16** hingegen serialisiert das UCS durch 16-Bit Sequenzen.

Die Problematik der Vielzahl sprach- und kulturraumspezifischer Zeichen kann auch mit Unicode nur bis zu einem gewissen Grad gelöst werden. Zum einen treten immer neue Zeichen auf, man denke z.B. an das Euro-Symbol, zum anderen ist nicht unstrittig, welche textuellen Einheiten den Status eines Zeichens haben sollen. Hier kommt die Unterscheidung zwischen **Glyphen**, vgl. ISO/IEC9541-1 (1991), und Zeichen ins Spiel. Glyphen stellen abstrakte, graphische Formen dar, die unabhängig von einem bestimmten Design sind. Die Visualisierung von Texten im Druck oder auf dem Bildschirm greift auf **Fonts** zurück. Zeichen sind also die „abstrakteste“ Einheit der textuellen Ebene, konkretisiert in Glyphvarianten, die schließlich mit Fonts dargestellt werden. Glyphen und Zeichen stehen nicht immer in einem eindeutigen Verhältnis. Ein Zeichen kann durch mehrere Glyphen repräsentiert werden, z.B. der Umlaut „Ä“ durch ein einzelnes Glyph oder durch zwei Glyphen, d. h. das Glyph „A“ und eine Umlautmarkierung. Ein einzelnes Glyph kann aber auch eine Zeichensequenz darstellen, wie z.B. Ligaturen, oder verschiedene Zeichen. Beispielsweise stellt das Glyph „A“ ein Zeichen im lateinischen, grichieschen oder kyrillischen Alphabet dar. Gegenstand der Kodierung in Unicode sind

Zeichen, nicht Glyphen. Deshalb werden z.B. die chinesischen, japanischen und koreanischen Kanji in Unicode als regional bedingte Glyphvarianten aufgefasst und zu einem Zeichenvorrat zusammengefasst.

Gippert (1999) hat auf die Problematik hingewiesen, die sich aus einer zu generellen Zusammenfassung von Glyphen zu bestimmten Zeichen ergibt. Werden z.B. historische oder regionale Varianten eines Schriftsystems zu stark generalisiert, können die entsprechenden Unterschiede nicht mehr auf der Zeichenebene repräsentiert werden. Eine maschinengestützte Analyse z.B. von historischen Kanji-Varianten des Japanischen wird so erschwert. Eine Lösung bietet das Verfahren der **WSD** (Writing System Declaration), wie es im Rahmen der bereits erwähnten TEI (Version P4, Kapitel 25) beschrieben wird. Eine WSD verschiebt die Differenzierung von Glyphvarianten oder die Zuordnung unterschiedlicher Schriftsysteme zueinander auf die Ebene der Textauszeichnung. Für die Repräsentation historischer Varianten von Kanji kann eine WSD nach dem Muster in Beispiel 2.11 verwendet werden.

(2.11)
```
<character class="lexical">
   <form string="" entityLoc="watashiAncient">
      <desc>Eine historische Variante für das Kanji
            "watashi"</desc>
      <note>Eine Abbildung des Zeichens findet sich in
            Nelson 1974.</note>
   </form>
</character>
```

Beispiel 2.11: Writing System Declaration der TEI

In Beispiel 2.11 wird eine historische, japanische Glyphvariante des Kanji „watashi“ in Beziehung gesetzt zu der Glyphvariante im heutigen Schriftjapanisch. Diese Deklaration der Beziehungen zwischen Glyphvarianten macht eine Notwendigkeit für die textuelle Informationsmodellierung deutlich, wie sie in Abschnitt 2.1 vorgestellt wurde: Verschiedene informationelle Ressourcen, hier Zeichen und Auszeichnungseinheiten, und Standards, hier Unicode und die Auszeichnungssprache, interagieren bei der Formulierung von Regeln und Bedingungen in textuellen Daten. Die Baumstrukturierung, also die primäre Informationsmodellierung, interagiert mit der Zeichenebene. Um z.B. die Abhängigkeiten zwischen den Glyphvarianten festhalten zu können, reicht die Baumstrukturierung allein

nicht aus.

2.2.2.2. Die Beziehung der Zeichenebene zur primären Informationsstrukturierung

Eine Möglichkeit, Zeichen in die primäre Informationsstrukturierung zu integrieren, wurde bereits in Beispiel 2.8 in Abschnitt 2.2.1.3 vorgestellt. Dabei wurde in XML Schema ein Datentyp `japaneseCat` definiert, der mögliche Werte für japanische Kategorien zusammenstellte. Auch mittels XML-DTDs ließen sich die Werte definieren, allerdings mit der eingeschränkten Anwendbarkeit auf Attributinhalte.

Problematisch ist bei beiden Verfahren, dass sie die Dokumentgrammatik durch die Typisierung von Daten spezialisieren. Es kann aber auch sinnvoll sein, nicht von der Dokumentgrammatik auszugehen, sondern die informationelle Ressource „textuelles Datum“ durch Regel- oder Bedingungsbeschreibungen zu charakterisieren. Dies wird an Beispiel 2.12 verdeutlicht.

(2.12)
$$\begin{bmatrix} LEMMA \\ kudasai \\ CONTEXT \\ \left[w[@cat =' V - IMP'] \right] \\ ASSERT \\ \left[\begin{bmatrix} preceding - sibling :: w[1] \\ \left[@cat =' V - TE' or @cat =' ACC' \right] \end{bmatrix} \right] \end{bmatrix}$$

Beispiel 2.12: Lexikalische Sicht auf Konfigurationen von Informationseinheiten

In diesem Beispiel wird die Bedingung wieder aufgenommen, die in Beispiel 2.7 auf Seite 29 mittels der Schemasprache Schematron ausgedrückt wurde. Sie ist nun Bestandteil einer Charakterisierung des textuellen Inhalts, der sich im betreffenden `<w>` Element in der Dokumentinstanz befindet, nämlich „kudasai“. Für dieses Wort wird eine **Merkmalsstrukturbeschreibung** in Form einer **AWM** (Attribut-Wert-Matrize) aufgebaut. Die AWM enthält Merkmalsbeschreibungen mit den entsprechenden Pfadbeschreibungen, um die das `<w>` Element mit den sieben Informationseinheiten „k u d a s a i“ zu selektieren – vgl. `CONTEXT` – und zu validieren – vgl. `ASSERT`. Diese Merkmalsstrukturbeschreibung kann als Grundlage für einen Lexikoneintrag von „kudasai“ dienen. Im Falle von „kudasai“ ist dieser Lexikoneintrag auch in der Lateintranskription aussagekräftig; Im Falle eines Lexems „kou“, vgl. Abbildung 2.5, wäre dies nicht der Fall: Die infor-

mationelle Ressource „japanische Originalverschriftlichung“ leistet also einen wichtigen Beitrag zur Differenzierung der Lexikoneinträge.

Das skizzierte Verfahren stellt einen ersten Schritt dar in Richtung einer Verbindung lexikalischer, d.h. vom textuellen Datum ausgehender Beschreibungen, mit anderen informationellen Ressourcen. Folgen von Zeichen, welche mit Unicode repräsentiert sind, werden verknüpft mit Beschreibungen von Bedingungen, die auf Schematron zurückgreifen. Eine Ausführung dieses Verfahrens mit den Möglichkeiten der sekundären Informationsstrukturierung wird für linguistische Daten in Abschnitt 5.2.1.2 exemplifiziert. Voraussetzung für die Anwendbarkeit des Verfahrens ist jedoch, dass eine Beschreibung der Regeln für die Dokumentklasse, d.h. eine Dokumentgrammatik vorliegt. Dies wird bedingt durch die Rolle von **Whitespace**, vgl. Beispiel 2.13.

(2.13)
```
<corpus>
...
<w> <stem>kudasa</stem> <suffix>i</suffix></w>
...
</corpus>
```

Beispiel 2.13: Probleme von Whitespace in der primären Informationsstrukturierung

Hier wird die morphologische und die wortbezogene Segmentierung von „kudasai“ ausgezeichnet. Vor dem Beginn des `<stem>` Elements und vor dem Beginn des `<suffix>` Elements befinden sich jedoch Leerzeichen, die z.B. durch manuelle Editierung des Dokuments zustande kommen und nicht Bestandteil der informationellen Ressource „Text“ sind. Bei einer Verbindung von Lexikon mit ausgezeichneten Dokumenten könnte so jedoch nicht differenziert werden zwischen unbeabsichtigten und beabsichtigen Leerräumen, so dass als Ergebnis ein Lexikoneintrag für „ kudasa i“ generiert werden würde. Um derartige Fehler zu vermeiden, kann eine grundlegende Beschreibung der Dokumentklasse wie in Beispiel 2.14 verwendet werden.

(2.14)
```
<!ELEMENT w (stem, suffix)>
<!ELEMENT stem (#PCDATA)>
<!ELEMENT suffix (#PCDATA)>
```

Beispiel 2.14: Mögliche Problemlösung für Whitespace

Diese XML-DTD deklariert ein `<w>` Element, welches keine textuellen Zeichen ent-

halten darf, sondern nur zwei Tochterelemente, `<stem>` und `<suffix>`. Bei der Validierung einer XML-Dokumentinstanz wie in Beispiel 2.13 werden durch den Prozess der **Normalisierung** die Leerzeichen im `<w>` Element entfernt, da sie nicht durch die Dokumentgrammatik gerechtfertigt sind. Die darauf aufbauende Verbindung von Lexikon und Dokument führt zu dem erwünschten Ergebnis.

2.2.3. Konzeptuelle Modellierung

2.2.3.1. Exkurs: Was ist eine Ontologie?

Der Begriff „Ontologie“ spielt in den Disziplinen der Philosopie, kognitiven Wissenschaften und künstlicher Intelligenz sowie linguistischen Teildisziplinen wie der lexikalischen Semantik eine große Rolle. Die allgemeinste Definition einer Ontologie ist die einer Wissensbasis[15]. Das Beispiel 2.15, angelehnt an Vossen (2003, S. 464), zeigt eine wichtige Funktion von Wissen.

(2.15) `Ich sah einen Mann / einen Stern / ein Molekül`
`mit einem Mikroskop / einem Teleskop / einem Binokular.`

Beispiel 2.15: Ambiguität sprachlicher Strukturen

Auf Grund des Wissens über die Größe von Gegenständen – vgl. Molekül versus Mann versus Stern – und die Funktion optischer Geräten – vgl. Mikroskop versus Binokular versus Teleskop – kann man beurteilen, dass ein Satz wie `Ich sah einen Mann mit dem Mikroskop.` nicht akzeptabel erscheint. Das Beispiel verdeutlicht außerdem die Nähe von Ontologien und Lexika: Man kann das beschriebene Wissen als Wissen über Wörter auffassen und in lexikalische Einträge integrieren.

Die unscharfe Grenze zwischen Ontologie und Lexion verdeutlicht die Problematik, den Begriff Ontologie erschöpfend zu definieren. Erdmann (2001) führt formale Eigenschaften auf, die seiner Ansicht nach eine Ontologie konstituieren und sie zugleich z.B. von Thesauri abgrenzen. Dazu gehören z.B. die Möglichkeit, Inferenzen ziehen zu können. Noy und McGuinness (2001) beschreiben Motivationen für die Verwendung von Ontologien. Von Experten geteiltes, gemeinsames Wissen über Aufbau und Informationsgehalt einer Domäne soll beschrieben und formalisiert werden, zum Zweck der Explikation und

[15] Oft wird auch zwischen Ontologie als Repräsentation von Konzepten und Wissensbasis als Repräsentation der Instanzen von Konzepten unterschieden, vgl. Erdmann (2001).

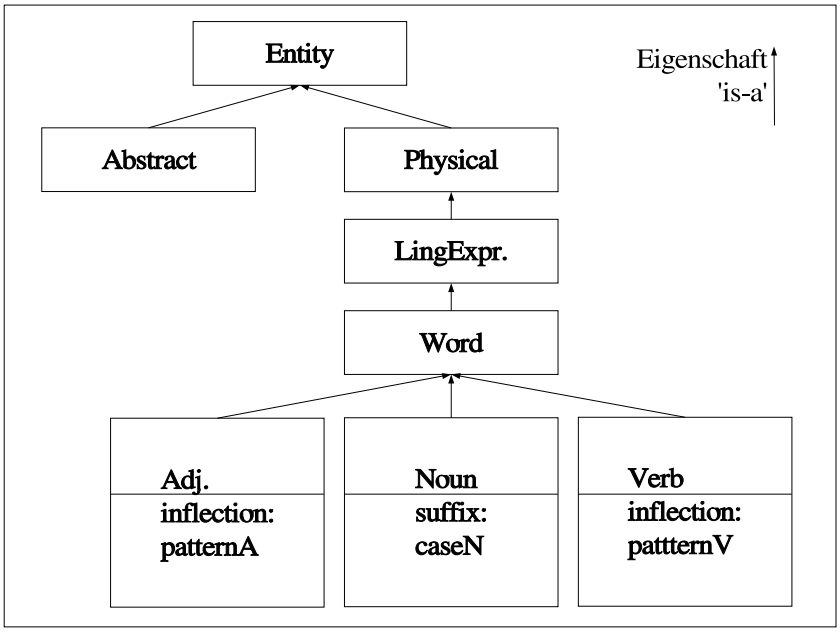

Abbildung 2.6.: Beispielontologie für die semantische Auszeichnung linguistischer Kategorien

Wiederverwendbarkeit. Die Formulierung einer Ontologie ist ein iterativer Prozess, d.h. Ontologie und „physikalische“ Daten der Domäne stehen in einem ständigen Austausch. Für eine linguistische Ontologie sind z.B. Konzepte wie word und morpheme und ihre Beziehung *partOf* grundlegend und unter Experten der Domäne unstrittig. Problematisch wird es jedoch bei theorie- oder sprachspezifischen Konzepten. Ihre Modellierung ist im ständigen Wechsel begriffen und nicht zuletzt abhängig von den Daten, auf die sie bezogen werden sollen.

Wichtigste Bestandteile einer Ontologie sind nach Noy und McGuinness (2001):

- die **Konzepte**, oft als **Klassen** (class) bezeichnet;
- die **Eigenschaften** der Konzepte, oft als **slot**, **role** oder **property** bezeichnet;
- Einschränkungen des Wertebereichs von Eigenschaften, sogenannte **Fassetten** bzw. **facets**;
- Subordination von Konzepten, die im grundlegendsten Fall zu einer **Konzepthierarchie** führt; sie wird durch die Eigenschaft *is-a* ausgedrückt.

- die **Instanzen** (instances), die sich auch separat zur eigentlichen Ontologie als Inhalt der **Wissensbasis** (knowledge base) auffassen lassen.

Die Funktion dieser Bestandteile wird in Abbildung 2.6 erklärt. Die Ontologie in Abbildung 2.6 beruht auf einem Fragment der Ontologie **SUMO** (Suggested Upper Merged Ontology, Niles und Pease (2001)), die als Basisontologie für die Bildung domänenspezifischer Ontologien konzipiert ist. Dem Konzept Entity sind die Konzepte Abstract und Physical untergeordnet. Sie haben also die Eigenschaft (Physical *is-a* Entity.), bzw. (Abstract *is-a* Entity.). Eine physikalische Entität sind z.B. sprachliche Ausdrücke. Sie werden durch das Konzept LingExpr. wiedergegeben. Ein Konzept für Wörter, Word, beinhaltet wiederum drei untergeordnete Konzepte Adj., Noun und Verb. Diese drei Konzepte haben bestimmte Eigenschaften, im vorliegenden Fall *inflection* für Adj. und V sowie *suffix* für N. Die Wertebereiche für die drei Eigenschaften sind auf jeweils eine bestimmte Auswahl eingeschränkt: patternA, caseN bzw. patternV.

Als Instanzen zu dieser Ontologie können die in diesem Kapitel vorgestellten, japanischen Beispiele betrachtet werden. Die Ontologie enthält einen sprachunspezifischen Teil, der das Konzept sprachlicher Ausdrücke lingExpr. und Wörter word umfasst, und einen sprachspezifischen Teil, der die Wortartenkategorien Adjektiv, Nomen und Verb als Konzepte definiert. Hier wird eine zentrale Aufgabe beim Erstellen von Ontologien deutlich: Die Zielsetzung und die fokussierten Konzepte müssen so genau wie möglich definiert werden. Im vorliegenden Fall sind Wortarten nach morphosyntaktischen Kriterien differenziert und als Konzepte in die Ontologie eingeführt worden.

Ein wichtiger Bestandteil der Ontologie ist die Konzepthierarchie. Sie legt die **Vererbung** von Eigenschaften fest. Wenn zum Beispiel für das Konzept Word die Eigenschaft *uses-alphabet-characters* festgelegt wird, so gilt diese auch für die subordinierten Konzepte Adj., Noun und Verb.

Das obige Beispiel demonstriert folgende Schritte, die laut Noy und McGuinness (2001) bei der Erstellung einer Ontologie von Bedeutung sind:

1. Definition des Skopus, d.h. der zu modellierenden Domäne, z.B. Kategorien zur Beschreibung japanischer, linguistischer Phänomene;

2. Wiederverwendung existierender Ontologien, z.B. die erwähnte SUMO-Ontologie;

3. Sammlung wichtiger Terme der Domäne, z.B. **Adjective** oder **inflection**;

4. Definition der Klassenhierarchie unter Verwendung bestimmter Terme, z.B. Adj., Noun, Verb;

5. Definition der Klasseneigenschaften unter Verwendung der übrig gebliebenen Terme, z.B. *inflection*;

6. Definition der Wertebereiche, z.B. patternA;

7. Erzeugung von Instanzen.

2.2.3.2. Terminologie: Ontologie versus konzeptuelles Modell

Der Begriff „Ontologie“ findet offensichtlich in vielen Wissenschaftsbereichen Anwendung. Für eine Beschreibung als informationelle Ressource ist er zu unklar definiert. Eine formale Basis ist nötig, um informationelle Ressourcen vertikal miteinander verbinden zu können.

Aus diesem Grund wird in der vorliegenden Arbeit der Begriff **Ontologie** durch den Begriff **konzeptuelle Ebene** ersetzt. Die konzeptuelle Ebene beinhaltet **konzeptuelle Modelle**. Sie umfassen **Konzepte** und deren **Eigenschaften**. Diese Charakterisierung konzeptueller Modelle lässt sich auf relationale oder objektorientierte Datenbanken, semantische Netzwerke oder Frame-Systeme der künstlichen Intelligenz oder Ontologien im vorab geschilderten Sinne anwenden. Sie hat zudem den Vorteil, auf einem generellen, aussagenlogischen Fundament zu basieren. Dies demonstriert Beispiel 2.16.

(2.16)
```
Entity.
    Abstract subClassOf Entity.
    Physical subClassOf Entity.
    LingExpr. subClassOf Physical.
    Word subClassOf LingExpr.
    Adj. subClassOf Word.
    Noun subClassOf Word.
    Verb subClassOf Word.
    inflection propertyOf Adj.
    suffix propertyOf Noun.
    inflection propertyOf Verb.
```

Beispiel 2.16: Ein konzeptuelles Modell im aussagenlogischen Format

Es reformuliert Teile der in Abbildung 2.6 visualisierten Ontologie in der so genannten **Tripel-Notation**. Die Tripel-Notation gibt Aussagen wieder, die aus einem **Argument**, d.h. dem **Subjekt**, einem **Prädikat**, und einem anderen Argument, namentlich dem **Objekt** bestehen. Sie sind in der Abfolge (Subjekt *Prädikat* Objekt.) dargestellt, z.B. (Abstract *subClassOf* Entity.). Das Prädikat *subClassOf*, eine andere Nomination für die vorgestellte Eigenschaft *is-a*, dient der Eingliederung von Konzepten in die Konzepthierarchie. Weitere Eigenschaften der Konzepte werden durch das Prädikat *propertyOf* beschrieben, z.B. (inflection *propertyOf* Adj..).

Dieses einfache aussagenlogische Inventar ist für viele Aufgaben zu schwach, um z.B. Wertebereiche von Eigenschaften einzuschränken. Es bietet aber eine formale Basis, um komplexere Formalismen zu definieren. Dies zeigt sich daran, dass die durch das W3C definierten Formate zur Repräsentation von konzeptuellen Modellen bzw. Ontologien, RDF Schema und OWL (vgl. Abschnitt 2.2.3.3), sich letztlich alle auf das aussagenlogische Inventar zurückführen lassen. Der Gehalt eines konzeptuellen Modells ist eine Menge von Aussagen. Sie kann als eine Graphenstruktur verstanden werden: Konzepte sind die Knoten, Prädikate die Kanten. Da Konzepte zugleich Subjekt als auch Objekt einer Aussage sein können, entsteht ein nicht gerichteter, potentiell zyklischer Graph. Wie in Abschnitt 2.2.1.2 gezeigt wurde, lassen sich auch Dokumentinstanzen unter Rückgriff auf das XML Information Set als Mengen von Aussagen repräsentieren. Eine vertikale Verbindung informationeller Ressourcen bedeutet deshalb, die Relationierung dieser Mengen zu beschreiben.

2.2.3.3. Standards zur Repräsentation konzeptueller Modelle

Die Repräsentation konzeptueller Modelle in maschinenlesbarer Form ist mit verschiedenen Standards möglich. Sie unterscheiden sich hinsichtlich ihrer formalen Fundierung und angestrebten Verwendungsbereichen. Hauptzweck des **Topic Map** Standards (Pepper und Moore, 2001) ist z.B. die Beschreibung von Topics bzw. den Konzepten eines konzeptuellen Models. Ein konzeptuelles Modell wird dabei formal weitgehend unbestimmt belassen. Die Standards des **Resource Description Framework** (Manola und Miller, 2004), insbesondere **RDF** selbst (Lassila und Swick, 1999), dienen zunächst nur der Identifikation von **Ressourcen**, wobei URI (vgl. Abschnitt 2.2.1.2) eine zentrale Rolle spielen. Mittels URI werden die Ressourcen identifiziert. **RDFS** (RDF Schema, vgl. Brickley und Guha (2004)) erweitert die Ausdrucksmöglichkeite von RDF um z.B.

die für konzeptuelle Modelle grundlegende Subsumptionsbeziehung. Auf RDFS aufbauend wird schließlich **OWL** (Web ontology language, vgl. McGuinness und v. Harmelen (2004)) definiert.

Im Vergleich zu RDF beinhalten Topic Maps weitreichende, vordefinierte Differenzierungen des Modellierungsinventars. So kann z.B. der Geltungsbereich, d.h. der Skopus von Topics festgelegt werden. RDF selbst umfasst nur **ressources**, **properties** und **statements**. Sie stellen eine unmittelbare Abbildung der Charakterisierung konzeptueller Modelle durch ein aussagenlogisches Inventar dar. Aussagen werden in Form so genannter **RDF Tripel** getroffen. In RDF Schema existieren vordefinierte Konstrukte zur Beschreibung konzeptueller Modelle: `rdfs:class` zur Beschreibung von Konzepten, `rdfs:subClassOf` zur Bildung der Konzepthierarchie, `rdf:property` zur Beschreibung von Eigenschaften, `rdfs:domain` und `rdfs:range` zur Bestimmung der Subjekt- und Objektposition von Aussagen über Konzepte. Da den Topic Maps die formale Verankerung in der Aussagenlogik fehlt, sind widersprüchliche Aussagen prinzipiell möglich. Diese Gefahr ist in RDF, RDFS und OWL nicht gegeben. Die Erweiterungen von **OWL DL** gegenüber RDFS basieren auf der **Description Logic**, vgl. Baader et al. (2003). Die Verankerung von RDF, RDF Schema und OWL DL in der Aussagenlogik schafft eine Brücke zu informationellen Ressourcen der primären Informationsstrukturierung, vgl. Abschnitt 2.2.1.2. Deshalb wird diesen Standards in der vorliegenden Arbeit der Vorzug gegenüber Topic Maps gegeben.

2.2.4. Sekundäre Informationsstrukturierung

Während es in den bisherigen Abschnitten hauptsächlich darum ging, bestehende Standards vorzustellen und ihre für die Verbindung informationeller Ressourcen relevanten Eigenschaften zu untersuchen, besitzt dieser Abschnitt einen anderen Schwerpunkt. Zwar gibt es bereits Verfahren zur sekundären Informationsstrukturierung, allerdings ist in diesem Bereich noch Vieles im Entstehen begriffen. Deshalb kann im Folgenden nicht immer von texttechnologischen Standards gesprochen werden, sondern es werden konkurrierende, teilweise in der Forschung begriffene Ansätze diskutiert.

2.2.4.1. Anwendungsszenarien

Eine Motivation für eine sekundäre Informationsstrukturierung wurde bereits in Abschnitt 2.2.1.4 dargelegt: Um vordefinierte Vokabulare als informationelle Ressourcen

miteinander kombinieren zu können, müssen ihre Beziehungen untereinander formal beschrieben werden. Das Beispiel 2.17, eine Variante von Beispiel 2.2, verdeutlicht diese Thematik.

(2.17)
```
<korpus>
 <satz
  uebersetzung="Leg bitte den langen Block nach unten.">
   <wort
    uebersetzung="lang" kategorie="Adjektiv">nagai</wort>
   <wort
    uebersetzung="Block" kategorie="Nomen">burokku</wort>
   <wort kategorie="Akkusativ-Marker">wo</wort>
   <wort
    uebersetzung="unten" kategorie="Nomen">shita</wort>
   <wort kategorie="Lokativ-Marker">ni</wort>
   <wort
    uebersetzung="legen" kategorie="Verb-Te-Form">oite</wort>
   <wort
    uebersetzung="bitte"
    kategorie="Verb-Imperativ">kudasai</wort>
 </satz>
</korpus>
```

Beispiel 2.17: Einfache Anwendung sekundärer Informationsstrukturierung

Die beiden Beispiele unterscheiden sich zum einen hinsichtlich der Benennung von Attributen und Elementen, vgl. z.B. `<s>` versus `<satz>` oder `cat` versus `kategorie`. Zum anderen sind die Attributwerte verschieden, vgl. z.B. `N` versus `Nomen`. Diese Form der sekundären Informationsstrukturierung stellt die einfachste Variante dar: Die unterschiedlichen Auszeichnungsvokabulare enthalten die gleichen Informationen, sie werden jedoch durch unterschiedliche Benennungen der Attribute und Elemente realisiert. Einen Sonderfall stellen die Attributwerte dar, die in beiden Beispielen unterschiedlich sind, aber z.B. durch unterschiedliche Datentypen in XML Schema wiedergegeben werden können. Eine Reformulierung des Datentyps aus Beispiel 2.8 für Beispiel 2.17 setzt z.B. voraus, dass nicht nur einzelne Werte, sondern eine Liste von Werten verändert werden.

Eine komplexere Variante sekundärer Informationsstrukturierung liegt vor, wenn die Verwendung von Attributen oder Elementen bei den Auszeichnungsvokabularen verschieden ist. Dies verdeutlicht das Beispiel 2.18. Der Informationsgehalt entspricht den Beispielen 2.2 und 2.17. Das Dokument in Beispiel 2.18 besitzt jedoch eine reichhaltigere Strukturierung.

(2.18)
```
<korpus>
  <satz>
    <uebersetzung>Leg bitte den langen Block nach unten.
    </uebersetzung>
    <wort>
    <uebersetzung>lang</uebersetzung>
    <kategorie>Nomen</kategorie>
    ...
  </satz>
 </korpus>
```

Beispiel 2.18: Komplexe Anwendung sekundärer Informationsstrukturierung

Die für die vorliegende Arbeit interessanteste Variante der sekundären Informationsstrukturierung ist diejenige, bei der nicht die Restrukturierung eines Auszeichnungsvokabulars zu einem anderen im Vordergrund steht, sondern die Relationierung mehrerer Auszeichnungsvokabulare zueinander. Durch die Relationierung werden genau diejenigen Informationen zugänglich, die für den jeweiligen Anwendungszweck nötig sind. Dies sei durch Beispiel 2.19 verdeutlicht.

Das Beispiel enthält den Text aus den vorangegangenen Beispielen in dreifacher Auszeichnung: in Bezug auf die Intention des Sprechers im Element `<imperative>`, die Markierung von Höflichkeit im Element `<honorific>` und syntaktische Funktion. Die drei Auszeichnungen entsprechen verschiedenen linguistischen Analyseebenen. Je nachdem, welche der drei Ebenen als sekundär angesehen wird und welche als primär, lassen sich verschiedene Prioritisierungen zwischen den Auszeichnungen beschreiben. Beispiel 2.20 beschreibt eine Prioritisierung für eine Klasse von Auszeichnungen, im Format einer XML-DTD.

Prioritisiert wird die Auszeichnung von Intentionen. Das Inhaltsmodell eines `<imperative>` Elements beinhaltet textuelle Daten oder ein `<marker>` Element. Die

```
(2.19) <corpus>
     <imperative>
      nagai burokku wo shita ni oite <marker>kudasai</marker>
     </imperative>
     <honorific>
      nagai burokku wo shita ni oite <marker>kudasai</marker>
     </honorific>
     <syntactic>
      nagai burokku wo shita ni oite <marker>kudasai</marker>
     </syntactic>
     </corpus>
```

Beispiel 2.19: Relationierung unterschiedlicher Auszeichnungsvokabulare

```
(2.20) <!ELEMENT imperative (#PCDATA|marker)*>
     <!ELEMENT marker (#PCDATA)>
     <!ATTLIST marker
       type-social   "honorific" #REQUIRED
       type-syntactic "sentence" #REQUIRED>
```

Beispiel 2.20: Aus verschiedenen Auszeichnungsvokabularien gebildete Dokumentgrammatik

Informationen aus den anderen beiden Ebenen werden als Attribute `type-social` und `type-syntactic` am `<marker>` Element realisiert. Diese Prioritisierung ist beliebig. Es wäre genauso möglich, eine der anderen Auszeichnungen zu prioritisieren und sie als Wurzelelement in der DTD zu deklarieren.

In Sasaki et al. (2002) wurde mittels dieses Verfahrens eine exemplarische, sekundäre Informationsstrukturierung für die Modellierung koreferentieller Phänomene in typologisch verschiedenen Sprachen unternommen. Für die beteiligten Sprachen, Japanisch und Kilivila[16], wurde eine Reihe von einfachen Dokumentgrammatiken entwickelt, die der Annotation verschiedener Charakteristika von Korefenz dienten, wie z.B. Mittel des

[16]Zu Kilivila vgl. z.B. Senft (1986).

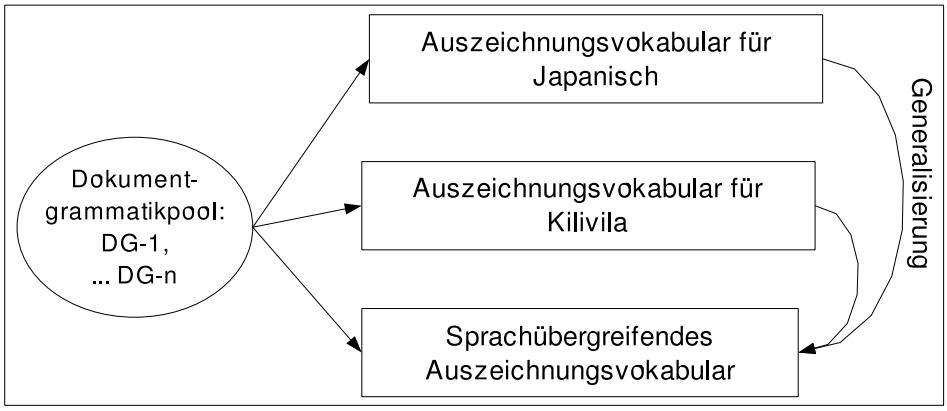

Abbildung 2.7.: Sekundäre Informationsstrukturierung zur Modellierung von Koreferenz

sprachlichen Ausdrucks von Koreferenz oder bestimmte Subtypen von Koreferenz. Die Dokumentgrammtiken waren nach der in Witt (2002) beschriebenen Methode Bestandteil eines ungeordneten Pools, in welchem keine Beziehungen zwischen den Dokumentgrammatiken bestehen. Anhand von annotierten Daten wurden die Beziehungen der Dokumentgrammatiken untersucht und anschließend komplexe Dokumentgrammatiken nach dem oben beschriebenen Muster verfasst. Diese komplexen Dokumentgrammatiken können der sprachspezifischen oder sprachübergreifenden Annotation und Analyse koreferentieller Phänomene dienen. Abbildung 2.7 visualisiert den Dokumentgrammatikpool und die verschiedenen komplexen Dokumentgrammatiken.

Dieses Beispiel stellt eine der noch seltenen Anwendungen[17] der sekundären Informationsstrukturierung dar. Der Grund für die geringe Verbreitung dieses Verfahrens liegt in der Komplexität seiner Realisierung, welche im nächsten Abschnitt thematisiert wird.

2.2.4.2. Realisierungsmöglichkeiten

Die Beispiele zur sekundären Informationsstrukturierung im vorherigen Abschnitt waren mit einer Veränderung von Dokumentgrammatiken verbunden. Wenn es jedoch nur darum geht, die Dokumentgrammatiken zu modularisiern, reicht eventuell die Verwendung von **Parameter-Entitäten** in DTDs aus. Beispiel 2.21 zeigt eine Anwendung von Parameter-Entitäten.

[17] Eine weitere Anwendung stellt z.B. Simons (1999) vor, vgl. Abschnitt 3.3.

```
(2.21) <![ %satz-generell; [
     <!ELEMENT satz (artikel | nomen | verb)*>
     ]>
     <![ %satz-speziell; [
     <!ELEMENT satz (artikel?,nomen,verb)>
     ]>
     <!ENTITY %satz-generell "INCLUDE">
     <!ENTITY %satz-speziell "IGNORE">
```

Beispiel 2.21: Anwendung von Parameter-Entitäten

Die Parameter-Entität `satz-generell` definiert ein `<satz>` Element mit einem generellen Inhaltsmodell. Es kann die Elemente `<artikel>`, `<nomen>` und `<verb>` beliebig oft und in beliebiger Reihenfolge enthalten. Die Parameter-Entität `satz-speziell` definiert ein anderes, spezielleres Inhaltsmodell für das `<satz>` Element. Beide Parameter-Entitäten stehen in so genannten **Marked Sections**. Durch die Schlüsselwörter `INCLUDE` bzw. `IGNORE` lassen sich die beiden Parameter-Entitäten ein- bzw. ausschalten. Die beiden Module der DTD können also alternativ verwendet werden.

Wenn die Modularisierung zugleich mit einer weitreichenden Veränderung der Dokumentgrammatiken verbunden ist, sind Parameter-Entitäten nicht geeignet. Für SGML gibt es als Teil des **HyTime-Standards** die Möglichkeit, sogenannte **architektonische Beziehungen** oder **Architekturen** zwischen Dokumentgrammatiken zu definieren und diese zur automatischen Transformationen von Dokumentinstanzen zu verwenden. Lobin (2000) beschreibt ausführlich die Verwendung von Architekturen[18]. Architekturen bestehen immer zwischen einer **Meta-DTD** und einer **Client-DTD**, wobei jede SGML-DTD beide dieser Rollen annehmen kann. Die Client-DTD deklariert dasjenige Auszeichnungsvokabular, welches auf das Auszeichnungsvokabular der Meta-DTD abgebildet werden soll. In der Client-DTD oder in einem zu den DTDs separaten Dokument wird festgelegt, welche Beziehungen zwischen den Auszeichnungsvokabularen bestehen sollen. Mit Architekturen kann man z.B. die Abbildung von Element- und Attributnamen realisieren,

[18] Da jede XML-Dokumentinstanz gleichzeitig eine SGML-Instanz ist, kann dieses Verfahren auch für XML angewendet werden.

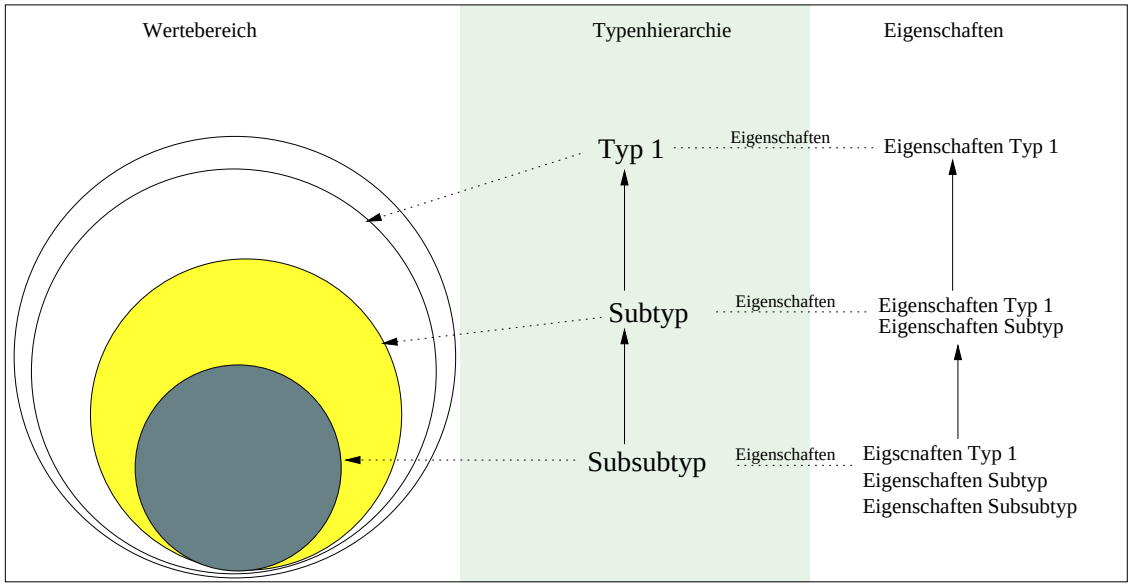

Abbildung 2.8.: Funktion von Typenhierarchien in XML Schema

vgl. Beispiel 2.17, oder die Generierung bestimmter Attributwerte in Abhängigkeit von Attributwerten in der Client-DTD, vgl. die Werte der `cat` Attribute in Beispiel 2.2 versus die Werte der `kategorie` Attribute in Beispiel 2.17, sowie die Filterung bestimmter Attribute oder Elemente. Die Dokumentstruktur kann neben der Filterung auch dahingehend geändert werden, dass der textuelle Inhalt von Elementen in Attribute überführt wird oder umgekehrt. Dadurch könnte z.B. eine Beziehung definiert werden, die das Beispiel 2.2 in das Beispiel 2.18 überführt. Der Inhalt der `cat` Attribute wird dann zum Inhalt von `<uebersetzung>` Elementen.

Wie bereits in Abschnitt 2.2.1.3 angedeutet, erlaubt XML Schema ebenfalls eine Form der sekundären Informationsstrukturierung. Drei Verfahren in XML Schema[19] sind hierfür entscheidend: Die Definition von Datentypen für Element- und Attributinhalte, die Definition von komplexen Typen für Inhaltsmodelle, und die Definition von Ersetzungsgruppen. Allen drei Verfahren liegt das Konzept der **Typenhierarchien** zu Grunde. Die Funktion dieser Hierarchien wird in Abbildung 2.8 visualisiert.

Jeder Typ besitzt Eigenschaften, anhand derer der Wertebereich des Typs beschrieben wird. Der Datentyp `integer` legt z.B. fest, dass als Werte nur ganze Zahlen infrage

[19] Ein weiteres Verfahren, das hier nicht näher behandelt wird, ist das Redefinieren von Typen.

kommen. Der subordinierte Typ `positiveInteger` besitzt die zusätzliche Eigenschaft, dass mögliche Werte nur positiv sein dürfen. Diese in XML Schema eingebauten Typen können vom Benutzer zur Definition weiterer Subtypen vewendet werden.

In Beispiel 2.8 in Abschnitt 2.2.1.3 wurde gezeigt, wie diese Funktionalität für die Definition von Datentypen, d.h. Typen für die Inhalte von Elementen oder Attributen, eingesetzt wird. Das Beispiel beinhaltete einen benutzerdefinierten Datentypen `japaneseCat`, der auf dem eingebauten Datentyp `string` beruht. Datentypen können **restringiert** oder **erweitert** werden; in dem Beispiel wird der Datentyp dahingehend restringiert, dass nicht mehr beliebige Zeichenketten als Wertebereich zugelassen werden, sondern nur noch die in der angegebenen Liste enthaltenen.

Komplexe Typen nutzen die Typenhierarchie zur Relationierung von Inhaltsmodellen. Dies demonstriert das Beispiel 2.22.

(2.22)
```
<complexType name="oneWordSentence">
 <sequence>
  <element name="w" type="xs:string"/>
 </sequence>
</complexType>
<complexType name="multipleWordSentence">
 <complexContent>
  <extension base="oneWordSentence">
   <sequence>
    <element name="w"
     type="xs:string" maxOccurs="unbounded"/>
   </sequence>
  </extension>
 </complexContent>
</complexType>
<element name="sentence" type="multipleWordSentence"/>
```

Beispiel 2.22: Nutzung der Typenhierarchie in XML Schema

Das Beispiel definiert einen komplexen Typen `oneWordSentence`, der ein Inhaltsmodell mit einem `<w>` Element bereit stellt. Daraus wird ein komplexer Typ `multipleWordSentence` abgeleitet, der ein Inhaltsmodell mit mehreren `<w>` Elementen

erlaubt. Dieser Typ dient schließlich der Deklaration eines `<sentence>` Elements.

Komplexe Typen erlauben nur eine lineare Erweiterung des Inhaltsmodells, d. h. Extension, oder eine vollständige Ersetzung, d.h. Restriction. Die Typisierungsinformationen finden Verwendung in der Abfragesprache Querysprachen **XQuery** bzw. dem zu Grunde liegenden Datenmodell XPath 2.0., vgl. Fernandez et al. (2003). In XQuery sind z.B. Suchanfragen nach Elementen möglich, deren Typ den gleichen übergeordneten Typ besitzt. Die Typinformationen sind jedoch nicht im XML Information Set einer Dokumentinstanz enthalten, sondern werden erst nach der Validierung anhand eines XML Schema Dokuments im **PSVI** (Post Schema Validation Infoset) ausgegeben. Das PSVI unterscheidet XML Schema auch von RELAX NG (vgl. Abschnitt 2.2.1.3). In der Spezifikation von RELAX NG ist ausdrücklich festgelegt, dass die Validierung einer XML-Dokumentinstanz gegenüber einem RELAX NG Schema nicht zur Veränderung des Information Set der Instanz führen darf.

Die Definition von Ersetzungsgruppen in XML Schema dient dazu, einem **Head Element** verschiedene **Substitute Elements** zuzuordnen. In einer Dokumentinstanz können dann die Elemente alternativ verwendet werden, wobei der Typ von Head Element und Substitute Element gleich sein muss. Die Substitute Elements haben also die gleiche Position in der Typenhierarchie wie das Head Element. Der Zweck dieser Ersetzung kann z.B. in der Verwendung eines sprachspezifischen Auszeichnungsvokabulars liegen, wie im Beispiel 2.17 angedeutet.

Während in XML Schema Verfahren der primären und sekundären Informationsstrukturierung in *einer* Schemasprache integriert sind, gibt es – wie bei den Architekturen – auch Ansätze, welche die sekundäre Informationsstrukturierung von der eigentlichen Dokumentgrammatik trennen. Kimber und Heintz (2001) demonstrieren, wie man mittels **UML** (Unified Modeling Language) die Konstrukte von XML-DTDs oder – bis zu einem gewissen Grad – von XML Schema Dokumenten auf einer abstrakten, konzeptuellen Ebene nachbilden kann. Für XML Schema existiert inzwischen eine Spezifikation, die sämtliche Konstrukte der Schemasprache auf Konstrukte in der abstrakten Spezifikationssprache **XMI** abbildet, vgl. IBM et al. (2001). So können Dokumentgrammatiken in Softwarearchitekturen integriert werden, die in UML oder XMI beschrieben sind.

Ein weiteres Verfahren, welches primäre und sekundäre Informationsstrukturierung voneinander trennt, stellen Lenz et al. (2002) vor. Sie beschreiben die Relationierung von Dokumentgrammatiken in Form einer Topic Map Repräsentation, die dokumentgrammatische Beziehungen auf einer konzeptuellen Ebene ausdrückt. Die Beziehungen sind

	Arch. Formen	XML Schema	UML	Topic Maps
Erweiterung von Inhaltsmodellen	x	x	x	x
Ersetzung von Inhaltsmodellen	x	x	x	x
Umbenennung	x	x	x	x
Filterung von Attributen / Elementen	x	x	x	x
Austausch von Dateninhalten Element - Attribut	x	-	-	x
Trennung primärer vs. sekundärer Informationsstrukturierung	x	-	-	x
Identifizierbarkeit sekundärer Informationsstrukturierung	-	x	x	x
Differenzierung sekundärer Informationsstrukturierungsverfahren	-	x	x	x
Unabhängigkeit von einem bestimmten Dokumentgrammatikformat	-	-	x	x
Bestimmte Implementation	x	x	-	-
Bisherige Anwendungen	x	x	-	-

Tabelle 2.1.: Vergleich von Realisierungsmöglichkeiten zur sekundären Informationsstrukturierung

unter Anwendung der Architekturen formuliert. Die konzeptuelle Realisierung bietet den Vorteil, nicht auf eine Implementation der Architekturen und bestimmte Schemasprachen wie SGML- bzw. XML-DTDs angewiesen zu sein. Es können aus der konzeptuellen Ebene auch andere Schema-Dokumente generiert werden. Mit Hilfe von Transformationssprachen, z.B. XSLT, lassen sich die entsprechenden Dokumentinstanzen ineinander überführen.

2.2.4.3. Vergleich der Realisierungsmöglichkeiten

Die wichtigsten Aspekte der verschiedenen Realisierungsmöglichkeiten zur sekundären Informationsstrukturierung sind in Tabelle 2.1 zusammengefasst. Alle vier Ansätze erlauben eine Erweiterung eines Inhaltsmodells oder seine Ersetzung, sowie die Filterung von Attributen. Der Austausch von Daten- und Elementinhalten, also die weitreichenste Form der sekundären Informationsstrukturierung, kann nur mittels Architekturen realisiert oder in Topic Maps repräsentiert werden. Architekturen erlauben es, primäre und sekundäre Informationsstrukturierung aufeinander zu beziehen, sie aber getrennt voneinander zu beschreiben. XML Schema hingegen verbindet die Aufgaben der primären Informationsstrukturierung – die Deklaration von Elementen und Attributen – mit der sekundären Informationsstrukturierung – die Definition von Datentypen, komplexen Inhaltsmodellen und der Ableitung von Typen. Bei den Architekturen gibt es allerdings – im Gegensatz zu den anderen Ansätzen – keine Möglichkeit, die Verfahren der sekundären Informationsstrukturierung zu benennen und zu differenzieren. Bei XML Schema und der XML Schema Repräsentation in UML ist dies durch die Trennung von Erweiterung und Restriktion bei Typdefinitionen gegeben. Zudem kann die primäre Informationsstrukturierung aus einem XML Schema Dokument separat als informationelle Ressource identifiziert werden, durch die Trennung der Deklarationen von Elementen und Attributen einerseits und von Typdefinitionen andererseits. Bei der Repräsentation von Architekturen durch Topic Maps gibt es zumindest potentiell die Möglichkeit, diese differenzierenden Informationen zu integrieren.

Architekturen sind für SGML- bzw. XML-DTDs entwickelt worden, sie sind also abhängig von diesem Format der primären Informationsstrukturierung. Die abstrakteren Formen der Repräsentation sekundärer Informationsstrukturierung, UML und Topic Maps, sind zwar auf jeweils eins dieser konkreten Formate hin entwickelt worden; sie können jedoch auch als Austauschformat auf einer konzeptuellen Ebene verstanden werden, aus dem verschiedene konkrete, implementationsnahe Formate generiert werden. Für Architekturen und XML Schema existieren Implementationen, für die Verarbeitung der Repräsentationen in Topic Maps oder UML bisher jedoch nicht. XML Schema scheint am besten dazu geeignet, unterschiedliche informationelle Ressourcen zu verbinden. Da ein XML Schema Dokument auch ein XML Dokument ist, können die Informationseinheiten des XML Schema Dokumentes mittels URI identifiziert und entsprechend klassifiziert werden. Dies erlaubt die separierte Nutzung von Informatio-

Informationelle Ressourcen	Standard	Abschnitt
Strukturen in Dokumentinstanzen	XML 1.0, Information Set	2.2.1.2
Regeln für Dokumentklassen	RELAX NG	2.2.1.3
Zeichenketten	Unicode, Information Set	2.2.2
Sekundäre Informationsstrukturierung	noch unklar	2.2.4
Konzeptuelle Ressourcen	RDF, RDF Schema, OWL DL	2.2.3
Verbindende Standards: URI, Namensräume		

Tabelle 2.2.: Informationelle Ressourcen und texttechnologische Standards

nen zu primärer und sekundärer Informationsstrukturierung. Dennoch erscheint es nicht sinnvoll, XML Schema für die primäre Informationsstrukturierung bzw. sekundäre Informationsstrukturierung zu verwenden. Dies liegt in der Beziehung zwischen der Validierung und dem XML Information Set einer XML-Dokumentinstanz begründet. Das PSVI vermischt informationelle Ressourcen in einer Weise, die ihre separate Operationalisierung erschwert. Wenn beispielsweise verschiedene dokumentgrammatische Regeln in die gleiche sekundäre Strukturierung einbezogen werden sollen, entstehen mit XML Schema äußerst komplexe Dokumentgrammatiken. Mit RELAX NG hingegen ist es möglich, nur Regeln in der Dokumentgrammatik zu formulieren, die separat in einer sekundären Informationsstrukturierung Einfluss finden. RELAX NG wird wie bereits erwähnt deshalb in dieser Arbeit als Dokumentgrammatikformat in der primären Informationsstrukturierungen verwendet.

2.3. Zwischenresümee: Grenzen texttechnologischer Standards aus vertikaler Sicht

Zu Beginn dieses Kapitels wurde in Abschnitt 2.1 beschrieben, dass ausgewählte texttechnologische Standards als Bestandteil einer physikalischen Modellierung eng auf be-

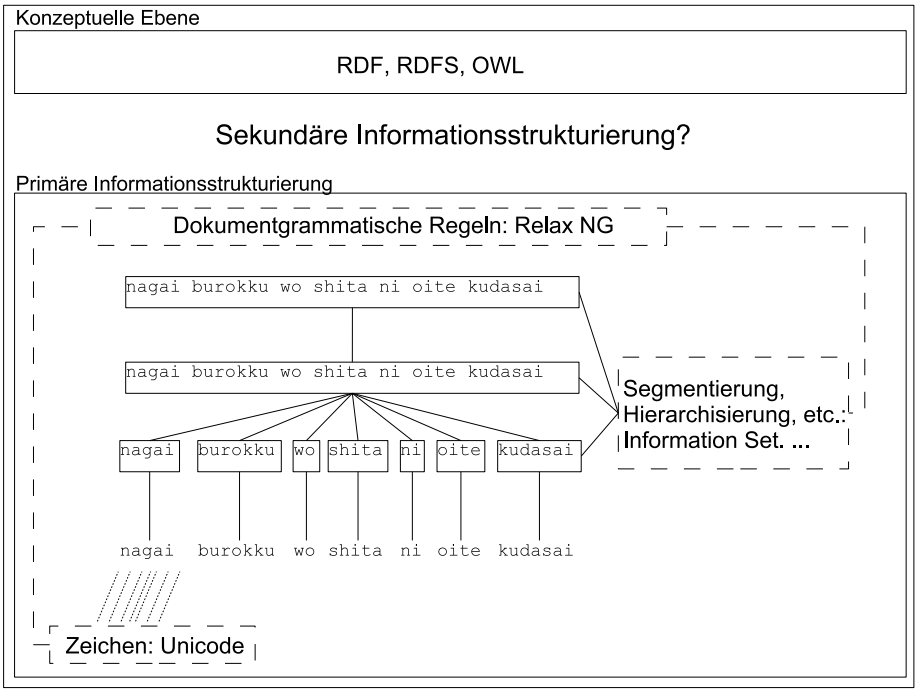

Abbildung 2.9.: Aufgaben von Standards bei der texttechnologischen Informationsmodellierung

stimmte Aufgaben der logischen Modellierung bezogen sind. Tabelle 2.2 fasst nun zusammen, welches Verhältnis zwischen informationellen Ressourcen, Modellierungsaufgaben und den texttechnologischen Standards sich aus der Diskussion in diesem Kapitel ergibt. Anhand des Beispiels „nagai burokku wo shita ni oite kudasai“ seien die Aufgaben der Standards noch einmal verdeutlicht; sie wird in Abbildung 2.9 visualisiert.

Die Zeichen, aus denen die Verschriftlichung des Satzes besteht, können mit dem Unicode-Standard repräsentiert werden. Dieser Standard hat den Vorteil, dass er in allen anderen, in dieser Arbeit relevanten Standards als Mittel zur Zeichenkodierung integriert ist. So kann der textuelle Dateninhalt als Teil der Dokumentinstanzen oder auch der dokumentgrammatischen Modellierung verwendet werden. Segmentierungen hinsichtlich z.B. Wortgrenzen und die Hierarchisierung von Satz und Wort werden im XML Dokument repräsentiert. Im Gegensatz zur Beschreibung der XML-Syntax im Standard XML 1.0 macht das Information Set die Bestandteile einer XML-Dokumentinstanz als Informationseinheiten, d.h. als informationelle Ressource, zugänglich.

Dokumentgrammatische Regeln, z.B. dass ein `<s>` Element immer ein oder mehrere `<w>` Elemente enthalten muss, werden mittels RELAX NG erfasst. Anders als XML Schema kann RELAX NG bei der Validierung nicht den Informationsgehalt der Dokumentinstanz verändern. So ist gewährleistet, dass alle Informationseinheiten eindeutig identifizierbar bleiben. XML Schema hingegen stellt Mechanismen wie Typendefinition und -ableitung zur Verfügung, die eine Rolle bei der sekundären Informationsstrukturierung spielen könnten. Da die Mechanismen jedoch integraler Bestandteil von XML Schema, also eines Formats zur primären Informationsstrukturierung, sind, werden die Modellierungsaufgaben der primären und sekundären Informationsstrukturierung vermischt. Deshalb kommt diese Schemasprache nicht zum Einsatz.

Unicode, das XML Information Set und RELAX NG sind demnach die relevanten Standards für die primäre Informationsstrukturierung. Ihre integrierte Verwendung gewährleistet übergreifenden Zugriff auf informationelle Ressourcen. XML-Dokumentinstanzen lassen sich mit URI eindeutig identifizieren. Durch die Verwendung von Fragment Identifier können auch Teilmengen von Informationseinheiten in Dokumenten adressiert werden. Dies trifft auch auf Dokumentgrammatiken bzw. darin enthaltene Deklarationen im Format von RELAX NG zu, da diese selbst XML-Dokumente sind. Namensräume identifizieren und separieren die einzelnen RELAX NG Dokumentgrammatiken. Da URI die Identifikation sowohl von Dokumentinstanzen als auch von einzelnen Informationseinheiten innerhalb von Dokumenten erlaubt, lassen sich z.B. Elements- oder Attributsdeklarationen in Dokumentgrammatiken als auch die entsprechenden Informationseinheiten in Dokumentinstanzen eindeutig bestimmen.

Die konzeptuelle Modellierung kann mit den auf dem Resource Description Framework aufbauenden Standards - RDF, RDF Schema, OWL DL - unternommen werden. Ihre aussagenlogisch fundierte Definition erleichtert die Anwendung als informationelle Ressoure. Für die sekundäre Informationsstrukturierung liegt noch kein etablierter Standard vor. Entsprechende Forschungsansätze werden in Kapitel 3 diskutiert. In Kapitel 4 folgt die Vorstellung eines eigenen Ansatzes.

2.4. Kernpunkte des Kapitels

- Textuelle Informationsmodellierung bezeichnet die Auszeichnungen von textuellen Dokumenten auf verschiedenen Ebenen. Die Auszeichnungen stehen in hierarchischen, regelhaften Beziehungen zueinander.

- Texttechnologische Informationsmodellierung nutzt standardisierte Formate. Primäre Informationsstrukturierung dient der Modellierung textueller Eigenschaften unter Anwendung von Auszeichnungssprachen wie XML. Modellierung auf einer konzeptuellen Ebene umfasst die Beschreibung abstrakter, konzeptueller Modelle. Die standardisierten Formate erlauben es, die physikalische, implementationsnahe Repräsentation informationeller Ressourcen eng an die eigentliche, formale Informationsmodellierung zu knüpfen. Dabei stellt sich die Frage, welche Standards für unterschiedliche Modellierungsaufgaben geeignet sind:

- Die Aufgabe der Auszeichnung von Dokumenten auf verschiedenen Auszeichnungsebenen. Diese Aufgabe erfüllt die Auszeichnungssprache XML. Dabei ist ein Dokument als informationelle Ressource von Bedeutung. Deshalb werden Dokumente als Mengen von Informationseinheiten aufgefasst, wie sie der Standard des XML Information Set definiert. Unter Anwendung des URI-Standards lassen sich ausgewählte Informationseinheiten eindeutig identifizieren. Der Standard der Namensräume hilft, Informationseinheiten in Teilmengen zu separieren. Problematisch bei der Anwendung von XML ist die Beschränkung auf eine hierarchische Dokumentstrukturierung, welche für viele Modellierungsaufgaben ungenügend erscheint.

- Die Aufgabe der Beschreibung von Regeln für Dokumentklassen, die in Dokumentinstanzen instanziiert werden. Die Regeln werden in Dokumentgrammatikformaten, so genannten Schemasprachen deklariert, in Formaten wie XML-DTDs, XML Schema oder RELAX NG. In einer Taxonomie formaler Sprachen besitzt RELAX NG die Ausdruckskraft Regular. XML Schema schränkt diese Ausdruckskraft ein, zum Zweck der eindeutigen Typisierbarkeit von Informationseinheiten. In dieser Arbeit kommt RELAX NG zur Anwendung, da eine Typisierung bzw. semantische Beschreibung von Informationseinheiten nicht mit der Aufgabenstellung der Regelbeschreibung vermischt werden soll.

- Die Aufgabe der Beschreibung von Bedingungen für Dokumentinstanzen. Eigenschaften von Informationseinheiten lassen sich nicht unbedingt unter Rückgriff auf Regeln beschreiben. Deshalb ist ein Format zur Beschreibung von Bedingungen nötig. Auch ein derartiges Format wird oft als Schemasprache – im weiteren Sinne – verstanden. Ein Beispiel ist Schematron. Ein Standard existiert zur Zeit nicht.

- Die Aufgabe der Repräsentation von Zeichen. Hierfür kommt der Unicode-Standard

zum Einsatz, der Zeichen aus unterschiedlichen Sprach- und Kulturräumen vereinigt. Wenn eine Modellierung von den Zeichen ausgeht, um z.B. korpusbasierte Lexika zu erstellen, kann sie nur in eingeschränkter Weise auf Regeln in Dokumentgrammatiken zurückgreifen. Dies zeigt einen Bereich der Anwendung von Bedingungen.

- Die Aufgabe der Modularisierung von Regeln. Umfangreiche Dokumentgrammatiken wie HTML oder die TEI liegen in modularisierter Form vor. Diese Dokumentgrammatiken haben einen Anspruch auf generelle Einsetzbarkeit. Für projektspezifische Anwendungen werden teilweise spezialisierte Fassungen der Dokumentgrammatiken erzeugt, die verschiedene Module kombinieren.

- Die Aufgabe der Repräsentation abstrakter, konzeptueller Ressourcen. Derartige informationelle Ressourcen werden häufig als Ontologien bezeichnet. Um die Unschärfe des Begriffs „Ontologie“ zu meiden, ist in dieser Arbeit von einer konzeptuellen Ebene die Rede, welche konzeptuelle Modelle umfasst. Die Repräsentation der konzeptuellen Ebene greift auf RDF und darauf aufbauende Standards zurück. Im Gegensatz zum Standard der Topic Maps gründet RDF auf einem aussagenlogischen Format. Die konzeptuelle Ebene lässt sich so als Menge von Aussagen repräsentieren. In einer vertikalen Verbindung informationeller Ressourcen sind deshalb drei Mengen relevant: die Menge dokumentgrammatischer Konstrukte sowie von Informationseinheiten in der primären Informationsstrukturierung, und die Menge von Aussagen in der konzeptuellen Ebene. Der zentrale Standard, der die generelle Identifikation informationeller Ressourcen in diesen drei Mengen erlaubt, lautet URI.

- Die Aufgabe der Beschreibung von Beziehungen zwischen informationellen Ressourcen der primären Informationsstrukturierung untereinander und von Beziehungen zur konzeptuellen Ebene. Hierfür dient die sekundäre Informationsstrukturierung. Bisher gibt es keine Standards, welche diese Aufgabe hinreichend umsetzen können. Die vorliegende Arbeit entwickelt nach der Diskussion bestehender Ansätze zur Verbindung informationeller Ressourcen eine entsprechende Methodologie.

3. Forschungsansätze zur Verbindung informationeller Ressourcen

3.1. Bedeutungsbeschreibung für Auszeichnungen versus semantische Auszeichnung

Die im Folgenden diskutierten Ansätze sind noch weiter von einer Standardisierung entfernt, als es bei der sekundären Informationsstrukturierung der Fall ist. Sowohl was den Gegenstand der Modellierung anbetrifft, als auch die Ziele und Methoden, herrscht derzeit (noch) große Diversität. Einige Autoren ordnen die bereits beschriebenen Methoden der sekundären Informationsstrukturierung sogar der Bedeutungsbeschreibung für Auszeichnung zu, vgl. Renear et al. (2002). Deshalb steht am Anfang dieses Kapitels eine Begriffsbestimmung, der die Beschreibung unterschiedlicher Anwendungsszenarien für die Verfahren folgt.

3.1.1. Begriffsbestimmung

In den bisherigen Beispiele für Dokumentinstanzen kamen Namen für Elemente und Attribute vor, deren Bedeutung sich für den menschlichen Leser unmittelbar erschließt. Das Attribut `cat` oder die deutsche Variante `kategorie` zeigen, dass diese Attribute Informationen über – linguistsche – Kategorien ausdrücken. Diese Interpretation ist jedoch nur für den menschlichen Leser nachvollziehbar, wie das Beispiel 3.1 zeigt. Diese Dokumentinstanz enthält sowohl in den Attributen als auch in den Elementen die gleichen textuellen Daten wie in Beispiel 2.2. Zudem sind die Elemente gleich strukturiert, und die gleichen Attribute sind gegeben. Dennoch wird hier auch einem menschlichen Benutzer die Bedeutung des Auszeichnungsvokabulars nicht sofort verständlich.

Lange Zeit lag der gebräuchliche Weg einer Bedeutungsbeschreibung für Auszeichnungsvokabulare in der natürlichsprachlichen Dokumentation. Die umfangreichen Richtlinien der TEI sind ein gutes Beispiel hierfür. Diese Dokumentationen sind jedoch nur

(3.1)
```
<a>
  <b c="Leg bitte den langen Block nach unten.">
    <e c="lang" d="ADJ">nagai</e>
    <e c="Block" d="N">burokku</e>
    <e d="ACC">wo</e>
    <e c="unten" d="N">shita</e>
    <e d="LOK">ni</e>
    <e c="legen" d="V-TE">oite</e>
    <e c="bitte" d="V-IMP">kudasai</e>
  </b>
</a>
```

Beispiel 3.1: Dokumentinstanz ohne natürlichsprachlich verständliche Semantik

dem menschlichen Leser[1], nicht aber einer maschinellen Interpretation zugänglich. Derzeit werden zwei Wege beschritten, die Bedeutung von Auszeichnungen auch formal zu spezifizieren. (Renear et al., 2002, S. 120) bezeichnen die Vorgehensweisen als die **Bedeutungsbeschreibungen für Auszeichnung** (markup semantics) versus die Entwicklung von **semantischer Auszeichnung** (semantic markup). Bedeutungsbeschreibungen für Auszeichnungen gehen von gegebenen Dokumentgrammatiken und Dokumentinstanzen aus und weisen ihnen eine Bedeutung zu. Oft sind diese Bedeutungsbeschreibungen auf Konfigurationen bzw. Positionen von Informationseinheiten bezogen. Sowohl in Beispiel 2.2 als auch in Beispiel 3.1 lassen sich z. B. die gleichen positionsbezogenen Bedeutungsbeschreibungen für alle Elemente und Attribute verfassen. Die Bedeutungsgleichheit der `c` und `cat` Attribute besteht darin, dass sie an einem ausgesuchten Typ von Element vorkommen. Dieser Typ – `<e>` oder `<w>` Elemente – hat eine eindeutige Postition im Verhältnis zu anderen Elementen, z.B. eine direkte Unterordnung zu `<b>` oder `<s>` Elementen, die wiederum andere strukturbezogene Eigenschafen aufweisen. Je nachdem, ob Konfigurationen mit Regeln oder Bedingungen – im Sinne der Diskussion aus 2.2.1.3 – beschrieben werden, ist eine Dokumentklasse oder eine Menge

[1] Wiemer (1999) definiert ein Format, das die Anforderungen natürlichsprachlicher Dokumentation und maschineller Verarbeitung verbindet. Aus diesem Format lassen sich eine kanonische Dokumentgrammatik und verschiedene Dokumentationen generieren.

von Dokumentinstanzen Gegenstand der Bedeutungsbeschreibung.

Die Bedeutungsbeschreibung für Auszeichnungen ist zumeist Bottom-Up angelegt. Informationseinheiten oder dokumentgrammatische Konstrukte erhalten durch unterschiedliche Verfahren eine Bedeutungsbeschreibung. Die Entwicklung einer semantischen Auszeichnung geht hingegen Top-Down vor. Sie zielt darauf ab, konzeptuelle Modelle zu Auszeichnungen und Auszeichnungsvokabularen zu relationieren. Ein derartiges Modell enthält z.B. ein Konzept `Satz` und ein Konzept `Wort`. Die Konzepte können auf Elementnamen wie `<s>` bzw. `<w>`, als auch auf Elementnamen wie `<b>` bzw. `<e>` bezogen werden.

Gegenstand der Bedeutungsbeschreibung für Auszeichnungen können Auszeichnungsvokabulare oder Informationseinheiten in einer Dokumentinstanz sein. Für die Entwicklung semantischer Auszeichnungen, unabhängig von bestimmten Inferenzen, ist dieser Unterschied oft bedeutungslos. Es muss nur entschieden werden, ob ein Konzept zu einem Element bzw. Attribut in einer Dokumentinstanz relationiert werden soll, oder zu dessen Deklaration in der entsprechenden Dokumentgrammatik. Die Bedeutungsbeschreibung für Auszeichnungen stößt hingegen vor besondere Probleme, wenn sie eine Dokumentklasse zum Gegenstand hat. So muss bei Attributen oder Elementen ihr Status – fakultativ, obligatorisch etc. – für die Beschreibung von Konfigurationen berücksichtigt werden. Die `<w>` bzw. `<e>` Elemente haben z.B. nicht immer ein `trans` Attribut bzw. ein `c` Attribut, weshalb diese Attribute nicht Bestandteil dokumentklassenbezogener Bedeutungsbeschreibungen sein können.

3.1.2. Anwendungsszenarien

Für eine semantische Auszeichnung liegt die Hauptmotivation in der Suche nach Informationseinheiten und ihrer Wiederverwendung. Entwicklungen entsprechender Verfahren sind deshalb auch oft mit der Vision eines **Semantic Web** verbunden, die Berners-Lee (2000) formuliert hat. Der Zugriff auf Informationen in dem unüberschaubaren Datenmeer „Internet“ soll durch eine semantische Anreicherung der Dokumente verbessert werden. Zwei Projekte aus dem linguistischen Bereich verfolgen entsprechende Anstrengungen. Das u.a. von Farrar et al. (2002) beschriebene Projekt **EMELD** (Electronic metastructure for endangered languages data) entwickelt eine Ontologie **GOLD** (General Ontology for Linguistic Description), die u.a. gebräuchliche Konzepte zur Beschreibung linguistischer Kategorien umfasst. Diese Konzepte sollen als kleinster gemeinsamer

Nenner für linguistische Daten dienen, welche auf Grund ihrer Theorie- oder Sprachspezifik nicht unmittelbar relationierbar sind. Durch die Ontologie wird ein Zugriff auf linguistische Daten angestrebt, der über einzelnen Domänen, Sprachen oder Theorien hinausreicht. XCES, die von Ide und Romary (2003) dargestellte XML-Version des Corpus Encoding Standards, unternimmt ebenfalls eine Relationierung von linguistischen, spezifischen Daten zu abstrakteren Kategorien. Im Gegensatz zu GOLD wird jedoch nicht eine nahezu unvermittelte semantische Auszeichnung vorgenommen, sondern es gibt verschiedene Zwischenrepräsentationen, die in Abschnitt 6.3.2 angesprochen werden. Für die Abbildung der einzelnen Stufen aufeinander werden Verfahren angewendet, bei denen primär strukturierte Daten schrittweise Transformationen durchlaufen. Hier überschneiden sich Methoden und Ziele sekundärer Informationsstrukturierung einerseits und der Bedeutungsbeschreibung für Auszeichnung andererseits.

Ein großer Nutzen einer Bedeutungsbeschreibung für Auszeichnungen liegt in der semantischen Validierbarkeit für Dokumente. Das Beispiel 3.2, ein Auschnitt aus dem Roman „1984“, verdeutlicht die Problematik.

(3.2)
```
<corpus>
<s><antecedent>The Ministry of Truth</antecedent> -
<reformulation>Minitrue</reformulation>, in Newspeak* - was
startlingly different from any other object in sight.</s>
<s><pronoun>It</pronoun> was an enormous pyramidal
structure ...</s>...
</corpus>
\begin{exStyle}
```

Beispiel 3.2: Nutzen semantischer Validierbarkeit von Auszeichnungen

In dem Beispiel sind semantische, koreferentielle[2] Beziehungen vorhanden, die in zwei Arten unterschieden werden können. Die Beziehung zwischen dem `<antecedent>` Element und dem `<reformulation>` Element besteht innerhalb eines `<s>` Elements. Die Beziehung zwischen dem `<antecedent>` Element und dem `<pronoun>` Element besteht über die Grenzen des `<s>` Elements hinweg. Mit einer Dokumentgrammatik ist es zwar möglich, Inhaltsmodelle mit `<pronoun>` Element versus Inhaltsmodelle mit

[2]Koreferentielle Phänomene wurden bereits in Abschnitt 2.2.4.1 angesprochen. Der Begriff **Koreferenz** wird in Abschnitt 6.3 näher erläutert.

`<antecedent>` und `<reformulation>` Element zu definieren. Die Bedingung, dass der Antezedent eines Pronomens im unmittelbar vorhergehenden Satz steht, ist jedoch mit dokumentgrammatischen Mitteln nur schwer ausdrückbar. Durch den Einsatz der `<key>` und `<keyref>` Elemente in XML Schema lässt sich eine derartige Restriktion bis zu einem gewissen Grad ausdrücken, allerdings nur als Bedingung für XML-Dokumentinstanzen. Für eine Auszeichnung koreferentieller Phänomene wäre es wünschenswert, derartige Konfigurationen zu validieren.

3.2. Verbindung Top-Down: Von konzeptuellen Modellen zu Auszeichnungen

Zwei der hier vorgestellten Ansätze zur semantischen Auszeichnung gehen von einem gegebenen, konzeptuellen Modell aus, aus dem Dokumentgrammatiken generiert werden. Diese wiederum dienen der Erstellung von Dokumentinstanzen. Für die Verbindung informationeller Ressourcen ergibt sich daraus ein Mehrwert hinsichtlich der formalen Interpretierbarkeit von Auszeichnungen: Da die Dokumentinstanzen zumindest mittelbar auf den konzeptuellen Modellen beruhen, werden deren Inferenzmechanismen in den Auszeichnungen einsetzbar. Unterschiede zwischen den Ansätzen gibt es hinsichtlich der Realisierung der Top-Down Verfahren.

Erdmann und Studer (1999, S. 8 ff.) geben Regeln für die Abbildung von Konzepten und Konzepteigenschaften auf dokumentgrammatische Konstrukte vor. Beispiel 3.3 verdeutlicht ihren Ansatz anhand eines konzeptuellen Modells und einer Dokumentgrammatik für die Modellierung von Koreferenz. Dabei wird das Beispiel 3.2 aus Abschnitt 3.1.2 wieder aufgegriffen. Das konzeptuelle Modell basiert auf dem Formalismus der **Frame Logic** im Sinne von Kifer et al. (1995). Konzepte werden auf Elementdeklarationen abgebildet. Den Konzepten Referent oder Antecedent entsprechen Elemente wie `<Referent>` oder `<Antecedent>`. Die Konzepthierarchie, konstituiert durch die Eigenschaft *is-a*, wird durch Parameter-Entitäten, vgl. Abschnitt 2.2.4.2, in der Dokumentgrammatik ausgedrückt. Ein Beispiel ist `LinguisticExpression "Referent | Antecedent"`. Eigenschaften, deren Wert andere Konzepte sind, drücken sich in Elementverschachtelungen aus, z.B. das Auftreten des `<Antecedent>` Elements im Inhaltsmodell des `<Referent>` Elements. Sind die Eigenschaften der Konzepte als textueller Inhalt definiert, so enthält das Element in der XML-DTD ein Inhaltsmodell mit `#PCDATA`. Um

```
(3.3) PhysicalObject.
      AbstractObject.
      Corpus[] :: PhysicalObject.
      DiscourseEntity[] :: AbstractObject.
      Referent[] :: DiscourseEntity.
      Antecedent[] :: DiscourseEntity.
      Corpus[discourseEntity =>> DiscourseEntity;].
      Referent[antecedent =>> Antecedent;].
      Antecedent[referent =>> Referent;].
      <!ENTITY % DiscourseEntity "Referent | Antecedent">
      <!ELEMENT corpus (#PCDATA | %DiscourseEntity;)*>
      <!ELEMENT Referent (#PCDATA | Antecedent)*>
      <!ELEMENT Antecedent (#PCDATA | Referent)*>
```

Beispiel 3.3: Generische Abbildung von Konzepten auf Elementdeklarationen

große Flexibilität bei der Erstellung von Dokumentinstanzen zu erhalten, sind sämtliche Inhaltsmodelle als Mixed Content (vgl. Abschnitt 2.2.1.3) definiert.

Eine Inferenz, die sich im konzeptuellen Modell ziehen und auf ausgezeichnete Dokumente anwenden lässt, zeigt Beispiel 3.4. Eine Instanz zum Konzept Antecedent, namentlich Antecedent1, hat eine Eigenschaft *referent*. Ihr Wert ist eine Instanz des Konzepts Referent, namentlich Referent1. Referent1 wiederum hat eine Eigenschaft *antecedent*, deren Wert Antecedent1 ist. In der XML-Dokumentinstanz lässt sich durch diese Inferenz sichern, dass für ein `<Referent>` Element mit einem verschachtelten `<Antecedent>` Element immer ein entsprechendes `<Antecedent>` Element existiert. Ist diese Eigenschaft nicht gegeben, ist die Dokumentinstanz semantisch nicht validierbar.

Der Nachteil dieses Verfahrens wird offensichtlich anhand der Struktur der Dokumentgrammatik. Um Bestandteile des konzeptuellen Modells generisch in die Dokumentgrammatik überführen zu können, wird in der zu Grunde liegenden DTD nur auf Elementdeklarationen zurückgegriffen. XML-Attribute kommen nicht zum Einsatz. Die Generizität der Transformation wird also durch eine schwache Ausnutzung der Modellierungsmöglichkeiten von XML erkauft.

(3.4)
```
-------------------------------------------------
Inferenz:
-------------------------------------------------
FORALL Referent1, Antecedent1
Referent1:Referent[antecedent ->> Antecedent1] -->
Antecedent1:Antecedent[referent ->> Referent1].
-------------------------------------------------
Dokumentinstanz:
-------------------------------------------------
<corpus>
<Antecedent>
<Referent>It</Referent>The Ministry of Truth</Antecedent>...
<Referent>
<Antecedent>The Ministry of Truth</Antecedent>It</Referent>
was an enormous pyramidal structure ....
</corpus>
```

Beispiel 3.4: Inferenz zur semantischen Validierung einer Dokumentinstanz

Auch Klein et al. (2000) gehen den Weg von einem gegebenen konzeptuellen Modell zu Dokumentgrammatiken, aus denen Instanzen generiert werden. Sie greifen dabei auf XML Schema zurück. Wie in Abschnitt 2.2.4.2 gezeigt, bietet XML Schema die Möglichkeit zur Definition von Typenhierarchien. Klein et al. (2000) bilden Konzepthierarchien aus konzeptuellen Modellen nahezu unmittelbar auf diese ab. Auch dieses Vorgehen hat jedoch Grenzen, wie die Autoren beschreiben (ebd., S. 17):

- Die Definitionsmöglichkeiten für Typen sind auf Einschränkungen oder Erweiterungen restringiert. Die Vererbungsmechanismen in der Konzepthierarchie sind so nur rudimentär nachbildbar.

- Auf der konzeptuellen Ebene sind Instanzen zu subordinierten Konzepten zugleich Instanzen zu superordinierten Konzepten. Auch diese Eigenschaft lässt sich nicht in XML-Schema wiedergeben.

Der Ansatz weist zudem die gleiche Problematik wie Erdmann und Studer (1999) auf, namentlich die Beschränkung auf XML-Elemente. XML-Attribute kommen bei der Generierung von Dokumentgrammatiken nicht zum Einsatz.

Im Rahmen der Entwicklung der Standards des Resource Description Framework, vgl. Abschnitt 2.2.3.3, hat es ebenfalls Versuche gegeben, Konzepthierarchien für konzeptuelle Modelle und Typenhierarchien für ausgezeichnete Dokumente aufeinander zu beziehen, vgl. z.B. Hayes et al. (2002). Das Ziel lag in der Bildung von Datentypen für RDF. Dabei wurde auch diskutiert, die formale Charakterisierung der Datentypen in RDF auf Inhaltsmodelle in XML Schema anzuwenden (ebd., Abschnitt 7.3). Im gegenwärtigen Stand der RDF-bezogenen Standards hat sich hingegen ein anderes Vorgehen durchgesetzt. RDF definiert eine allgemeine und abstrakte, formale Semantik für Datentypendefinitionen. Konkret genutzt wird in den Standards zu RDF eine Teilmenge der vordefinierten, einfachen Datentypen aus XML Schema. D.h. die Datentypen beziehen sich nur auf textuellen Inhalt. Zudem werden keine Mechanismen zur Validierung von Instanzen definiert. Dieser Prozess bleibt nicht weiter spezifizierter Software, z.B. einem XML Schema Prozessor, überlassen.

Datentypen wurden bereits in den Abschnitten 2.2.1.3 und 2.2.4.2 angesprochen. Hayes und McBride (2004, Abschnitt 5.1) beschreiben Datentypen aus Sicht von RDF. Sie umfassen drei Bestandteile:

- eine nicht leere Menge von Zeichen, den **lexikalischen Bereich** (lexical space) von d;
- eine nicht leere Menge, den **Wertebereich** (value space) von d;
- eine Abbildung des lexikalischen Bereichs auf den Wertebereich von d.

Diese Charakterisierung ist konform mit der Beschreibung, wie sie von Biron und Malhotra (2001, Abschnitt 2.1) für XML Schema gemacht wird. Allerdings lassen sich in RDF unter Rückgriff auf Datentypdefinitionen Aussagen treffen, die in XML Schema zu einem **Datatype Clash** führen würden, vgl. Beispiel 3.5.

(3.5)
```
japPos: rdf:type xsd:string.
japPos: rdf:type xsd:decimal.
```

Beispiel 3.5: Datatype Clash

Einer Instanz des Konzepts japPos wird durch das Prädikat *rdf:type* der Datentyp xsd:string zugewiesen, zum anderen der Datentyp xsd:decimal. In XML Schema haben diese Datentypen disjunkte Wertebereiche. Die für XML Schema spezifische Interpretation der Aussagen führt deshalb zu einen Datatype Clash. Die Inkonsistenz ist jedoch nicht durch die formale Semantik von RDF erfasst.

Für textuelle, d.h. durch Datentypen charakterisierbare Daten bietet RDF also Mechanismen, die Beziehung zu konzeptuellen Modellen zu beschreiben. Die Validierung der Beschreibungen bleibt allerdings entsprechender Software überlassen. Einheiten zur Auszeichnung von Dokumenten wie Elemente und Attribute sind aus Sicht von RDF beliebige, abstrakte Ressourcen, vgl. Abschnitt 2.2.3.3. Zur Überprüfung der Existenz der Ressourcen und ihrer Eigenschaften bietet RDF keine Mechanismen.

3.3. Verbindung Bottom-Up: Von Auszeichnungen zu semantischen Beschreibungen

Am Anfang von Abschnitt 3.1 wurde darauf hingewiesen, dass Verfahren zur sekundären Informationsstrukturierung und zur Bedeutungsbeschreibung für Auszeichnungen sich oft überschneiden. Im Folgenden werden Ansätze zur Bedeutungsbeschreibung für Auszeichnungen vorgestellt, geordnet nach der Nähe zur sekundären Informationsstrukturierung:

- **Architekturen** zur Abbildung von Dokumentgrammatiken und Dokumentinstanzen in eine objektorientierte Datenbank, vgl. Simons (1997) und Simons (1999);
- der Ansatz der **Metaschema**, vgl. Simons (2003);
- die Abbildung der primären Informationsstrukturierung auf semantische Netzwerke, entwickelt im Rahmen des Projektes **SemDoc**, vgl. Lobin (2001);
- das **XDD**-Framework (XML Declarative Description, vgl. Wuwongse et al. (2001) und Wuwongse et al. (2003)); sowie
- das **BECHAMEL-Projekt**, vgl. Sperberg-McQueen et al. (2002).

Simons (1997) demonstriert die Anwendung von SGML-Architekturen zur Beschreibung von Auszeichnungssemantik. In Simons (1990) wurde bereits früh der Nutzen einer

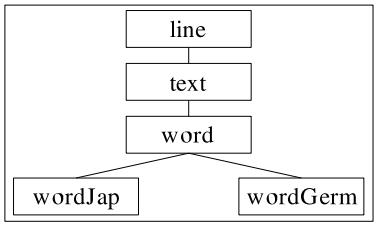

Abbildung 3.1.: Klassendefinitionen zur linguistischen Glossierung nach Simons (1990)

Validierbarkeit für die konzeptuelle Modellierung von Lexika und Glossierungen exemplifiziert, welche den gleichen Ansatz verwendet. Simons definiert eine Architektur für Glossierungen, die Element- und Attributdeklarationen in einem objektorientierten Paradigma realisiert. Die Anwendung sei unter Rückgriff auf Abbildung 3.1 erklärt.

Die Superklasse line definiert für Zeilen die Eigenschaft, dass sie textuelle Daten enthalten. Eine Subklasse text kann für alle Zeilen die Eigenschaft bestimmen, dass Text aus Zeichen des UCS bestehen soll, vgl. Abschnitt 2.2.2.1. So ist die Repräsentierbarkeit japanischer Zeichen und deutscher Umlaute gewährleistet. Eine untergeordnete Subklasse word definiert, dass Wörter im Text enthalten sind. Diese Subklasse schließt z.B. bestimmte Steuerzeichen wie Zeilenwechsel aus der Zeichenmenge aus, erlaubt sind nur japanische Schriftzeichen sowie das Alphabet mit Umlauten. Dieser Subklasse untergeordnet sind schließlich die Subklassen wordJap und wordGerm, welche die Menge der Wörter sprachbezogen differenzieren. Ein Unterschied zwischen den Subklassen liegt in der Rolle von Leerzeichen, die in japanischen Texten nicht vorkommen. Die Wortsegmentierung kann in der Subklasse wordJap also nicht anhand des textuellen Datums bestimmt werden.

Das Beispiel zeigt, wie nahe der Ansatz von Simons an der in XML Schema realisierten, sekundären Informationsstrukturierung mittels Datentypen ist. Sie wurde in Abschnitt 2.2.4 vorgestellt. Wie in Simons (1999) demonstriert, ermöglicht das Vorgehen von Simons jedoch objektorientierte Repräsentationen, die sich mit XML Schema nicht ausdrücken lassen. Dies ist z.B. der Fall, wenn ein XML-Element im objektorientierten Modell einmal ein Objekt, einmal ein Attribut und einmal beides repräsentiert.

In Simons (2003) wird der Ansatz nicht auf eine objektorientierte Datenbank, sondern auf in OWL erfasste, hauptsächlich linguistische Ontologien bezogen. Als Abbildungs-

mechanismus von XML-Elementen, Attributen und textuellen Inhalten nach OWL kommen statt der Architekturen so genannte **Metaschema** zum Einsatz. Die Ausführung der Abbildung, d.h. die Transformation von sprach-, theorie- oder auch projektspezifischen XML-Vokabularen nach OWL, wird mittels XSLT realisiert. Eine zentrale Rolle spielt dabei XPath, durch welches die transformationsrelevanten Informationseinheiten selektiert werden.

Der Ansatz von Simons ist strikt Bottom-Up ausgerichtet, von den spezifischen Auszeichnungen zu generellen Begriffsbestimmungen. Im Gegensatz dazu stellt Lobin (2001) eine auch Top-Down anwendbare Methodologie vor. Sie ist im Rahmen des Forschungsprojekts **SemDoc** (Semantik generischer Dokumentstrukturen) entwickelt worden. Im Zentrum von SemDoc stehen Ansätze zur Bedeutungsbeschreibung für Auszeichnung. Am Beispiel der Beschreibung einer Tabelle zeigt Lobin, welche semantischen Aspekte durch die hierarchische Strukturierung von XML-Dokumenten nicht ausgedrückt werden können. Eine Tabellenzelle läßt sich z.B. als Teil einer Spalte auffassen, zugleich aber auch als Teil einer Zeile, als Teil des Tabellenkopfs oder -körpers, als Zelle an einer bestimmten Position etc. Die Beziehungen zwischen diesen verschiedenen Interpretationen repräsentiert Lobin unter Anwendung des Topic Maps Standards in einem semantischen Netz, wobei die Interpretationen durch XPath-Ausdrücke adressiert werden. Das Beispiel 3.6 vollzieht dieses Vorgehen für die XML-Dokumentinstanz in Beispiel 2.2 auf Seite 18 nach.

(3.6)

Konzept	*Adressierung*
Corpus	`corpus`
Sentence	`corpus/s`
Word	`corpus/s/w`
BeginOfS	`corpus/s/w[position()=1]`
EndOfS	`corpus/s/w[position()=last()]`
MiddleOfS	`corpus/s/w[position()>1 and position()<last()]`

Beispiel 3.6: Bedeutungsbeschreibung von Auszeichnungen mittels XPath-Ausdrücken

Im semantischen Netz werden die Konzepte verknüpft, durch Relationen wie *enthält* (zwischen Corpus und Sentence) oder *ist-ein* (zwischen BeginOfS und Word). Aus dem semantischen Netz werden zum einen Konzept-Adressierungen zu Informationseinheiten

in Dokumentinstanzen vorgenommen, zum anderen kann daraus auf Elementdeklarationen in einer Dokumentgrammatik verwiesen werden. Charakteristisches Kriterium bei der Verknüpfung zwischen semantischem Netz und Dokumentgrammatik bzw. -instanz ist die Verwendung von Pfadausdrücken. Die Informationseinheiten in Dokumentinstanzen werden in konzeptbezogene Teilmengen unterteilt, wobei die Realisierbarkeit eines Pfades als Kriterium der Mengenzugehörigkeit gilt. Mengen können sich überschneiden, Teilmengen können wiederum Teilmengen enthalten, vgl. die Mengen zu den Konzepten Word und BeginOfS etc.

In einem noch stärkeren Maße als Lobin gehen Welty und Ide (1999) von einem Szenario aus, in welchem sowohl konzeptuelle informationelle Ressourcen einerseits, sowie Dokumentinstanzen und dazugehörige Dokumentgrammatiken andererseits als gegeben angesehen werden. In ihrem System **CLASSIC** entwickeln sie dementsprechend ein Verfahren, diese unterschiedlichen informationellen Ressourcen zu verknüpfen. In CLASSIC wird zunächst für eine bestehende Dokumentgrammatik ein grundlegendes konzeptuelles Modell generiert, welches Konzepte nahezu unmittelbar aus Element- und Attributsdeklarationen ableitet. Diese Konzepte werden manuell erweitert, je nachdem welche Domäne abgedeckt werden soll. Dabei werden gleichnamige Elemente subklassifiziert – vgl. den Namen eines Autors versus den Namen eines Herausgebers – und teilweise in Abhängigkeit vom Kontext in der Dokumentinstanz differenziert – vgl. Namen in Überschriften versus Namen in Paragraphen. Erweiterungen des grundlegenden konzeptuellen Modells werden automatisch zur Informationsanreicherung ausgezeichneter Dokumentinstanzen und zur Dokumentation der Dokumentgrammatik verwendet. Filter können spezifische Dokumentinstanzen generieren, die eine bestimmte Sicht auf das Dokument beinhalten – vgl. eine linguistische versus eine typographische Sicht.

Im Gegensatz zu den Ansätzen von Simons, Lobin sowie Welty und Ide baut das von Wuwongse et al. (2001) sowie Wuwongse et al. (2003) vorgestellte **XDD**-Framework (XML Declarative Description) eine semantische Beschreibung für XML-Auszeichnungen *unabhängig* von existierenden, konzeptuellen informationellen Ressourcen auf. Dies geschieht in einer inkrementellen Vorgehensweise. Für die Spezifizierung der Semantik stellt das XDD-Framework eine deklarative Beschreibungssprache zur Verfügung. Die Menge der XML-Elemente `N` bzw. ihre Deklarationen in Dokumentgrammatiken und der Zeichenketten `C` dienen als Ausgangsmengen einer Spezialisierung, die auf folgende Variablen zurückgreift:

1. Namens-Variablen `$N` für Namen `N`;
2. String-Variablen `$S` für Zeichenketten `C`;
3. AW-Variablen `$P` für Sequenzen von Attribut-Wert-Paaren;
4. Variablen für Sequenzen von XML-Expressions `$E`;
5. Variablen für Bestandteile von XML-Expressions `$I`.

```
(3.7) a   <s lang="Japanese"> a'  <s lang="Japanese">
          <w>nagai</w>            $e:japaneseWords
          <w>burokku</w>          </s>
          </s>

      a'' <s $P:Japanese>
          $e:japaneseWords
          </s>
```

Beispiel 3.7: Spezialisierung von Informationseinheiten im XDD-Framework

Die Rolle dieser Variablen verdeutlicht Beispiel 3.7. Eine variblenfreie, so genannte Ground XML-Expression a bildet die Ausgangsmenge. Daraus abgeleitet wird die Menge a', in welcher die Sequenz der <w> Elemente die XML-Expression $e:japaneseWords bindet. Die weitere Ableitung führt zur Menge a''. Hier bindet das Attribut-Wert-Paar `lang="Japanese"` die Variable $P:Japanese. Die durch XDD definierten Mengenoperationen gewähren die Reversibiliät der Spezialisierung. Es wird möglich, aus a'' mittelbar wieder die Ground XML-Expression a zu expandieren.

Diese Eigenschaft der bidirektionalen Abbildung zwischen formaler Bedeutungsbeschreibung und Auszeichnungen unterscheidet XDD von den anderen, bisher vorgestellten Ansätzen. XDD verwendet keine komplexe, modelltheoretische Semantik wie z.B. RDF, sondern greift nur auf Mengenoperationen zurück. Die Ausnutzung der hierarchischen Dokumentstruktur ist allerdings gering. Pfadausdrücke stehen bei der Teilmengenbildung nicht zur Verfügung. Nur eine Inklusionsbeziehung zwischen XML-Expressions ist durch Variablen vom Typ $I spezifizierbar. Nachahmung von Pfadbeschreibungen kann bis zu einem gewissen Grad durch rekursive Variablendefinitionen erreicht werden.

Wie das XDD-Framework thematisiert auch der Ansatz des **BECHAMEL**-Projekts, den Sperberg-McQueen et al. (2002) vorstellen, einen schrittweisen Aufbau von Bedeutungsbeschreibungen. Anders als XDD greift BECHAMEL nicht auf Variablen zur Definition von Teilmengen der Auszeichnungen zurück. Ausgangspunkt ist das komplette Information Set einer Dokumentinstanz. Es wird unmittelbar abgebildet in eine Sammlung von **Prolog**-Fakten. Diese dient der Erzeugung von sogenannten **sceleton sentences**, d.h. Prolog-Prädikaten mit Leerstellen, die durch Verweise, sogenannte **deiktische Ausdrücke** (deictic expressions, z.B. realisiert mittels XPath), auf Informationseinheiten in der Dokumentinstanz gefüllt werden. Folgende Sätze werden dabei unterschieden:

- Image Sentences repräsentieren Informationseinheiten als Prolog-Fakten, sie bilden also den Informationsgehalt der Dokumentinstanz unmittelbar in eine abstraktere Repräsentation ab.
- Property Rules verknüpfen bestimmte Eigenschaften mit bestimmten Element- oder Attributdeklarationen des Auszeichnungsvokabulars. Z. B. kann die Deklaration eines `<w>` Elements mit der Bedeutung word verknüpft werden. Property Rules spezifizieren auch die Vererbbarkeit von Eigenschaften an untergeordnete Informationseinheiten.
- Propagation Sentences verknüpfen Image Sentences innerhalb der Prolog-Faktenbasis mit Property Rules. Jede der Image Sentences für `<w>` Elemente hat dann z.B. die Eigenschaft w *interpreted-as* word.
- Mapping Rules wenden domänenspezifische Regeln auf Propagation Sentences an.
- Application Sentences resultieren aus der Anwendung einer Mapping Rule auf Propagation Sentences.
- Weitere Axiome und Weltwissen sind ebenfalls im Inferenzmechanismus vorgesehen, aber nicht weiter spezifiziert.

Die Anwendung dieser Sätze sei an Beispiel 3.8 demonstriert. Für alle Informationseinheiten werden zunächst Image Sentences gebildet. Das `lang` Attribut wird durch eine Property Rule mit der Eigenschaft *Sprachdeklaration* verknüpft. Dabei wird die Vererbung der Sprachdeklaration auf untergeordnete Informationseinheiten definiert. Ein Propagation Sentence verknüpft die Prolog-Repräsentation des `<s>` Elements, des

```
(3.8) <corpus lang="japanese" glossLang="english"
              transLang="german">
       <s cat="imperatice-sentence">
        <w trans="lang" cat="ADJ">nagai</w>
        ...
       </s>
      </corpus>
```

Beispiel 3.8: Dokumentinstanz zur Anwendung von Sceleton Sentences

`<w>` Elements sowie des textuellen Inhalts „nagai“ mit dieser Property Rule. Eine domänenspezifische Mapping Rule für die `cat` Attribute lautet, dass sie nicht die Sprachdeklaration im `lang` Attribut erben, stattdessen wird die Sprachdeklaration im `glossLang` Attribut übernommen. Das gleiche gilt für die Vererbung des `transLang` Attributs an das `trans` Attribut. Application Sentences führen dazu, dass die Mapping Rule sowohl für das `cat` Attribut am `<s>` Element als auch für das `cat` Attribut am `<w>` Element gilt.

Der Einsatz dieser Sätze kann auf zweifache Weise geschehen. Entweder werden sie sukzessive auf eine Dokumentinstanz und die dazugehörige Dokumentgrammatik angewandt, oder es werden unmittelbar Application Sentences generiert. Bei letzterem Verfahren kommen XSLT-Stylesheets zum Einsatz, die für einzelne Dokumenttypen entwickelt werden. Die zu Grunde liegenden, semantischen Beschreibungen sind jedoch dokumenttyp-unabhängig: Eine Eigenschaft wie „Die Eigenschaft a wird an alle untergeordneten Informationseinheiten vererbt“ kann so auf ein `lang` Attribut angewendet werden, aber auch auf ein `<sprache>` Element.

3.4. Diskussion der Ansätze

Die Ansätze zur semantischen Auszeichnung unterscheidet untereinander die Ausnutzung von Schemasprachen für XML-Dokumentinstanzen. Erdmann und Studer (1999) nutzen nur ein dokumentgrammatisches Konstrukt, namentlich Elemente in XML-DTDs. Diese Einschränkung erleichtert die Formulierung automatischer Abbildungsverfahren: Konzeptuelle Modelle werden auf generische Weise zur Generierung von Dokumentgram-

matiken genutzt. Der Nachteil besteht darin, dass die Modellierungsmöglichkeiten von Dokumentgrammatiken im Sinne der logischen Modellierung, vgl. Abschnitt 2.1.2, nicht ausgenutzt werden. Da jede Eigenschaft eines Konzepts durch Verschachtelung von Elementen wiedergegeben ist, entfällt z.B. die Möglichkeit zur Trennung von Segmentierungsinformationen (XML Elemente) versus Zusatzinformationen (XML Attribute).

Klein et al. (2000) nutzen ebenfalls nur Elemente in XML. Die zu Grunde liegende Schemasprache ist jedoch XML Schema. Die Autoren beschreiben ein mehrstufiges, semi-automatisch durchführbares Verfahren, dessen Kern die Abbildung von Konzepthierarchien auf Typenhierarchien in XML Schema bietet. Auf diese Weise finden Informationen aus dem konzeptuellen Modell Einzug in die Dokumentgrammatik. Offen bleibt allerdings die Frage, wie diese Informationen operationalisiert werden können. Zwar kann eine Dokumentinstanz gegenüber einem XML Schema Dokument validiert werden, wodurch die Typisierungsinformation ausgenutzt wird. Die Information bleibt jedoch zu einem wesentlichen Teil implizit: Sie lässt sich nicht nutzen, um in Dokumentinstanzen die Instanzen subordinierter Typen zu finden.

Von den vorgestellen Ansätzen zur semantischen Auszeichnung sind die auf RDF basierenden Standards am meisten auf Schemasprachen für XML-Dokumentinstanzen bezogen. Sie fokussieren dabei die Schemasprache XML Schema und deren Datentypen. Als informationelle Ressource in ausgezeichneten Dokumenten lassen sich die Zeichen selbst erfassen, nicht jedoch andere Informationseinheiten wie Elemente oder Attribute. Zwar lässt sich ein RDF Dokument verfassen, das mittels URI Bezüge zu XML-Dokumentinstanzen herstellt. Die Überprüfung dieser Bezüge ist jedoch nicht Gegenstand von RDF.

Das Verhältnis der drei diskutierten Ansätze zur semantischen Auszeichnung visualisiert Abbildung 3.2. Die Ansätze sind verankert in einem Koordinatensystem bezüglich der Ausnutzung dokumentgrammatischer Konstrukte – die horizontale Achse –, und des Grades der Formalisierung, welche die Interpretation der primären Informationsstrukturierung besitzt – die vertikale Achse. Die formale Beschreibung der Datentypen in RDF legt am detailliertesten das Verhältnis zwischen einem konzeptuellen Modell und Informationseinheiten in Dokumentinstanzen fest. Allerdings bezieht sie sich nur auf einfache Datentypen in XML Schema, andere dokumentgrammatische Konstrukte werden außer Acht gelassen. D.h. nur die informationelle Ressource „Text“ erhält eine Bedeutungsbeschreibung, nicht aber die Ressourcen Segmentierung oder Hierarchisierung. Die Ansätze von Klein et al. und Erdmann et al. nutzen zwar die entsprechenden dokument-

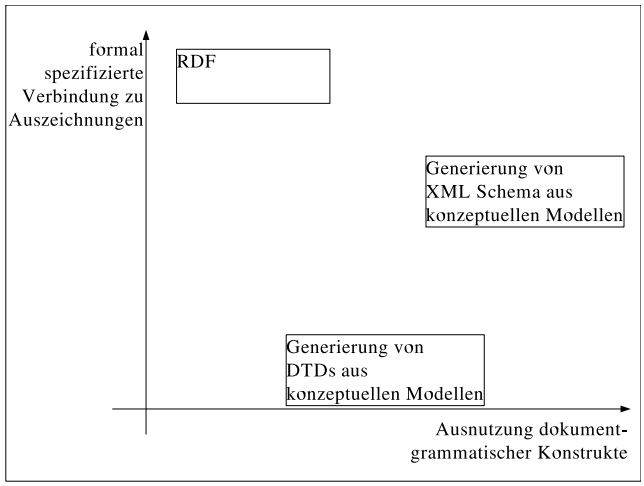

Abbildung 3.2.: Vergleich der Ansätze zur semantischen Auszeichnung

grammatischen Konstrukte, z.B. die Typenhierarchie in XML Schema, sie beschreiben aber nur eine unmittelbare Abbildung, von der konzeptuellen Ebene zu ausgezeichneten Dokumenten.

Gemeinsam haben alle drei Ansätze die Eigenschaft, dass sie keine Operationen für die vertikale Verbindung definieren. Es bleibt weitgehend unklar, wie die Repräsentation der Konzepthierarchie als Parameter-Entitäten in XML-DTDs oder komplexe Typen in XML Schema ausgenutzt wird. Mit RDF ist zunächst alles als eine Ressource interpretierbar, so auch Informationseinheiten in Dokumentinstanzen oder Deklarationen in einer Dokumentgrammatik. Es fehlen Verfahren, die diese Interpretation umsetzen, und mit denen auf informationellen Ressourcen operiert werden kann.

Die Ansätze zur Bedeutungsbeschreibung für Auszeichnungen, vgl. Abbildung 3.3, sind in andere Koordinaten einzuordnen, hinsichtlich der Ausrichtung der Bedeutungsbeschreibung auf Spezifika von Auszeichnungen – die horizontale Achse – und der Integrierbarkeit bestehender konzeptueller Modelle – die vertikale Achse.

In diesen Koordinaten treten deutliche Unterschiede hervor. Der Ansatz von Welty et al. beschreibt weitreichende Möglichkeiten der Integration bestehender konzeptueller Ressourcen. Die Abbildung der Konzepte auf Informationseinheiten und deren Deklarationen in Dokumentgrammatiken ist jedoch nahezu unvermittelt. Simons nutzt ein objektorientiertes konzeptuelles Modell. In einer neueren Fassung des Ansatzes von Si-

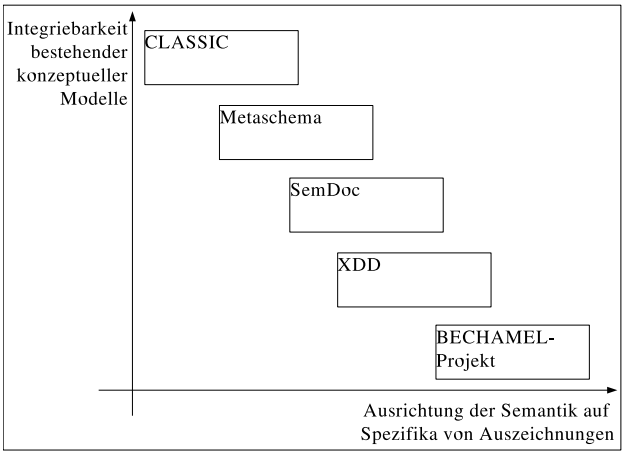

Abbildung 3.3.: Vergleich der Ansätze zur Bedeutungsbeschreibung für Auszeichnungen

mons (2003) kommt statt des objektorientierten Modells ein in OWL repräsentiertes, konzeptuelles Modell zum Einsatz. Bei der Relationierung von OWL Repräsentationen zu XML-Dokumentinstanzen kommt XPath zum Einsatz. XPath erlaubt komplexere, positionsbezogene Selektionen von Mengen von Informationseinheiten in Dokumentinstanzen bzw. deren Deklarationen in Dokumentgrammatiken. Diese Pfadsprache kommt auch bei Lobin zum Einsatz. Er nutzt jedoch im Gegensatz zu Simons ein semantisches Netz statt einer objektorientierten Repräsentation als konzeptuelles Modell. Der Ansatz im BECHAMEL-Projekt beschreibt am wenigsten den Rückgriff auf bestehende konzeptuelle Ressourcen. Grundlage der inkrementellen Bedeutungsbeschreibung ist das komplette Information Set von Dokumentinstanzen. D.h. die Bedeutung für nahezu beliebige Konfigurationen von Informationseinheiten wird - zumindest prinzipiell - erfassbar. Die Deixis, mit denen diese Konfigurationen erfasst werden, sollen ebenfalls auf XPath zurückgreifen. Das XDD-Framework steht zwischen den Ansätzen von Lobin, Simons und dem BECHAMEL-Projekt. Zwar sind keine Pfadausdrücke wie in diesen Ansätzen vorgesehen, aber durch entsprechende Kombinationen von Variablen können diese bis zu einem gewissen Grad nachgebildet werden.

3.5. Bidirektionale, vertikale Verbindung informationeller Ressourcen

Die vorgestellten Ansätze zur Beschreibung von Auszeichnungssemantik und zur semantischen Auszeichnung ermöglichen eine vertikale Verbindung informationeller Ressourcen. Viele der Ansätze sind hinsichtlich ihrer Gerichtetheit komplementär. Sie gehen von vorgebenen konzeptuellen Modellen aus, die auf Auszeichnungen bezogen werden, oder von vorgebenen Auszeichnungen, die eine semantische Interpretation erhalten. In diesem Sinne haben sie die Gemeinsamkeit der **Unidirektionalität**. Es wird nur in geringem Maße von einem Szenario ausgegangen, in welchem bestehende informationelle Ressourcen vertikal miteinander verknüpft sind. Das Ziel liegt vielmehr in der Erzeugung neuer informationeller Ressourcen: Neue Auszeichnungen (Dokumente, Dokumentgrammatiken) für die semantische Auszeichnung, oder neue semantische Beschreibungen für die Bedeutungsbeschreibung für Auszeichnung.

Dementsprechend haben sie die Gemeinsamkeit, dass sie selten Operationen für die bidirektionale, vertikale Verbindung definieren. Die Ansätze zur Bedeutungsbeschreibung von Auszeichnungen definieren Bottom-Up gerichtete Operationen, von den Auszeichnungen zur Bedeutungsbeschreibung. Komplementär verhält es sich mit den Ansätzen zur semantischen Auszeichnung. Einzige Ausnahme bildet das XDD-Framework. Der Ansatz ist bidirektional, die Bedeutungsbeschreibung für Auszeichnung wird durch sukzessive Bindungen von Variablen an Informationseinheiten erlangt. Die Variablen können wieder expandiert werden, und die Informationseinheiten sind z.B. für Suchanfragen zugänglich.

Wozu diese Bidirektionalität aber überhaupt von Nutzen ist, verdeutlicht Beispiel 3.9. Das Beispiel enthält zwei Auszeichnungen der gleichen textuellen Primärdaten. Sie werden in beiden Auszeichnungen als Inhalt einer Liste wiedergegeben. Die obere Auszeichnung greift dabei auf das Namensvokabular von HTML zurück, d.h. die `<DL>`, `<DT>` und `<DL>` Elemente. Die untere Auszeichnung zeichnet den Text in TEI-spezifischer Weise aus. Das `<list>` Element enthält ein Attribut `type`, zudem gibt es ein `<head>` und ein `<item>` Element.

Trotz der unterschiedlichen Benennung der Auszeichnungen und der Variation in Verteilung auf Elemente bzw. Attribute ist die Bedeutung der Auszeichnung offensichtlich gleich. Es handelt sich um Definitionen, die den Namen eines Terms und eine definitorische Beschreibung enthalten. Mit Top-Down ausgerichteten Ansätzen zur semantischen

(3.9)
```
<DL>
  <DT>V-TE</DT>
  <DD>Label for the annotation of japanese verbs
      in the assimilation form</DD>
</DL>
-----------------------------------------------------
<list type="gloss">
 <head>V-TE</head>
  <item>Label for the annotation of japanese verbs
        in the assimilation form</item>
</list>
```

Beispiel 3.9: Verwendung unterschiedlicher Auszeichnungensvokabulare mit gleicher Bedeutung

Auszeichnung fällt es schwer, für verschiedene Einheiten der primären Informationsstrukturierung diese Bedeutungsidentität zu beschreiben. Mit Bottom-Up ausgerichteten Ansätzen zur Bedeutungsbeschreibung für Auszeichnungen besteht die Gefahr, dass die Beschreibung immer nur für *eine* Dokumentgrammatik gilt. Bei Verwendung von XPath-Ausdrücken ist sie möglicherweise nur für eine abgeschlossene Menge von Dokumentinstanzen haltbar. Eine bidirektionale Abbildung, d.h. eine Vermittlung verschiedener informationeller Ressourcen der primären Informationsstrukturierung mit verschiedenen informationenellen Ressourcen der konzeptuellen Ebene, würde diese Mankos nicht besitzen.

Für beide Auszeichnungen erscheinen verschiedene Operationen als sinnvoll:

- Auf Grund der Bedeutungsgleichheit der Auszeichnungen sollte ihre Bedeutung nur einmal zu spezifizieren sein. Die gleiche Bedeutungsbeschreibung für Auszeichnungen sollte also für verschiedene Auszeichnungen einsetzbar sein.

- Suchanfragen für Auszeichnungen sollten auf der konzeptuellen Ebene formuliert werden können. Die Suche nach dem Konzept definitorische Liste sollte sowohl für das `<DL>` Element als auch für das `<list>` Element erfolgreich sein.

- Verschiedene Validierungsoperationen sind nötig. Zum einen müssen die Auszeichnungen gegenüber dem konzeptuellen Modell validiert werden. Gilt zum Beispiel die Beschreibung, dass eine Liste einen Listenkopf enthält, für beide Auszeichnungen? Zum anderen muss das konzeptuelle Modell hinsichtlich Widersprüchen gegenüber den Auszeichnungen überprüft werden. Ist z.B. der Listenkopf ein fakultativer Bestandteil des Modells oder ist er obligatorisch?

- Transformationen der Auszeichnungen sollen möglich sein. Dabei ist ein zentrales Kriterium die Reversibilität der Transformation.

Die Reversibilität von Transformationen ist ein außerordentliches Problem. Es ist natürlich möglich, die XML-Dokumentinstanzen durch den Einsatz von XSLT-Stylesheets ineinander zu überführen. Die Umkehrbarkeit des Prozesses ist jedoch nicht gewährleistet. Bei einer Transformation von HTML nach TEI ist z. B. dem `<head>` Element nicht anzusehen, ob es aus einem `<DT>` oder einem anderen Element generiert wurde. Die Transformationsoperationen müssen also derart spezifizierbar sein, dass diese Semantik nicht verloren geht.

Es sollte möglich sein, aus Auszeichnungen die Bedeutungsbeschreibung zu erzeugen, und aus dieser wiederum die Auszeichnungen. Das Kriterium der Reversibilität lässt sich also nicht nur anwenden auf die Transformation zwischen XML-Dokumentinstanzen, sondern auch auf die Beziehung zwischen konzeptueller Ebene und Auszeichnungen. Die Forderung nach Reversibilität erfüllen die diskutierten Ansätze zur semantischen Auszeichnung in unterschiedlichen Maße. Die Arbeiten von Erdmann und Studer (1999) und Klein et al. (2000) bilden jede informationelle Ressource auf der konzeptuellen Ebene – Konzepte, Konzepthierarchie, Eigenschaften – auf dokumentgrammatische Konstrukte ab. Wenn das Verfahren Top-Down angestoßen wird, ist die Umkehrung von Auszeichnungen zur konzeptuellen Ebene möglich. Die Standards des RDF sind so einsetzbar, dass ein konzeptuelles Modell auf beliebige Auszeichnungen referieren kann. Reversibilität ist dabei prinzipiell gegeben: Beliebige Auszeichnungen haben beliebige Relationen zu konzeptuellen Modellen. Die formale Semantik von RDF beschränkt sich jedoch auf eine intensionale Beschreibung, so dass Mechanismen zur Ausführung von Transformationen fehlen. Nahezu alle Ansätze zur Bedeutungsbeschreibung für Auszeichnungen erfüllen das Kriterium der Reversibilität nicht. Einzig das XDD-Framework erlaubt Reversibilität: Mengen von Informationseinheiten binden Variablen, die wieder expandiert werden können.

3.6. Inhaltsseitige versus ausdrucksseitige Beschreibung

Die Thematik der Bidirektionalität und Reversibilität von Transformationen wurde von Swick und Thompson (1999, Abschnitt 3, Listenpunkt 7) im Rahmen des sogenannten **Cambridge Communique** diskutiert. Gegenstand des Cambridge Communique sind die Standardisierungsverfahren für RDF und XML. RDF kann als Repräsentationsformat für konzeptuelle Modelle eingesetzt werden, XML für Auszeichnungen von Texten. Bei RDF ist zusätzlich zu unterscheiden zwischen den Konstrukten von RDF und ihrer maschinellen Kodierung, die u.a. in XML erfolgen kann. Hierbei wird XML datenzentriert genutzt, d.h. nicht mehr als Format zur Segmentierung, Hierarchisierung und Regelbeschreibung für textuelle Daten benutzt, sondern als generelles Datenaustauschformat.

Die Formate bzw. sie erweiternde Standards weisen Ähnlichkeiten auf. In RDF Schema gibt es Methoden zur Konstruktion einer Konzepthierarchie, vgl. Abschnitt 2.2.3.3, in XML Schema sind Typenhierarchien definierbar, vgl. Abschnitt 2.2.4.2. Das Cambridge Communique vertritt jedoch den Standpunkt, dass diese Methoden verschiedenen Zwecken dienen und nicht vereint werden müssen. Nur die einfachen, in XML Schema vordefinierten Datentypen sollten in beiden Standards Berücksichtigung finden.

Hinsichtlich der Abbildungen zwischen XML und RDF kamen die Verfasser zu dem Schluss, dass dafür ein anwendungsspezifischer Abbildungsmechanismus nötig sei. Die Umkehrbarkeit der Abbildung, also der Transformation zwischen XML und RDF, lässt sich nicht abschließend klären.

Das Cambridge Communique hat zwar Repräsentationsformate und nicht informationelle Ressourcen zum Gegenstand. Dennoch ist es hilfreich, um das Verhältnis von konzeptueller Ebene und primärer Informationsstrukturierung zu klären. Eine unmittelbare Abbildung informationeller Ressourcen, z.B. der Konzepthierarchie auf die Typenhierarchie oder Parameter-Entitäten, schafft eine große Abhängigkeit zwischen den Modellierungsmöglichkeiten der verschiedenen informationellen Ressourcen. Dies führt zu Dokumentgramatiken oder zu konzeptuellen Modellen, welche die Modelleriungsmöglichkeiten der jeweiligen Repräsentationsformate nur wenig ausnutzen. Konzeptuelle Ebene und primäre Informationsstrukturierung können also nicht vereint bzw. unmittelbar voneinander abgeleitet werden. Vielmehr ist ein Abbildungsmechanismus von Nöten, der zwischen beiden vermittelt. Eine besondere Problematik stellt dabei die Reversibilität der Abbildung dar.

Die Aussagen des Cambridge Communique sind zwar auf die Datenformate RDF und

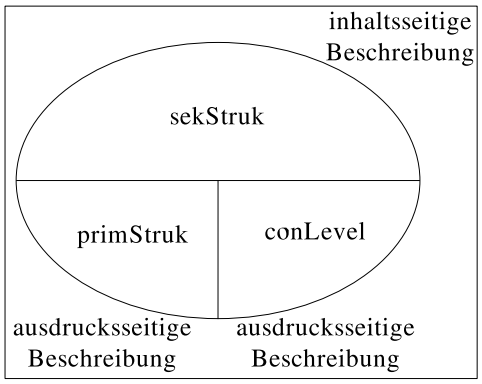

Abbildung 3.4.: Ein Abbildungsmechanismus für informationelle Ressourcen

XML bezogen. Sie lassen sich aber auch auf informationelle Ressourcen der konzeptuellen Ebene und der primären Informationsstrukturierung übertragen. Standards zur Repräsentation konzeptueller Modelle dienen der **inhaltsseitigen Beschreibung**. Ob Instanzen der beschriebenen Konzepte diese Eigenschaften tragen: Diese Frage liegt außerhalb des Skopus. Standards in der primären Informationsstrukturierung hingegen dienen der **ausdrucksseitigen Beschreibung**. Gegenstand der Beschreibung sind der Aufbau von informationellen Ressourcen, z.B. von Dokumentinstanzen, und ihr Informationsgehalt, z.B. repräsentiert als Informationseinheiten. Die Beschreibung charakterisiert jedoch nicht die Bedeutungen der Informationen im inhaltsseitigen Sinne. Ein Abbildungsmechanismus zwischen primärer Informationsstrukturierung und konzeptueller Ebene sollte zwischen inhaltsseitiger und ausdrucksseitiger Beschreibung in komplementärer Weise vermitteln. Abbildung 3.4 visualisiert, wie dieser Abbildungsmechanismus in der vorliegenden Arbeit konzipiert ist.

Inhaltsseitige Beschreibungen werden in der sekundären Informationsstrukturierung fomuliert. Sie können sich auf verschiedene ausdrucksseitige Beschreibungen, d.h. verschiedene informationelle Ressourcen beziehen, namentlich Einheiten der primären Informationsstrukturierung oder der konzeptuellen Ebene. Aus Sicht der sekundären Informationsstrukturierung ist also auch die – für sich genommen inhaltsseitige – konzeptuelle Ebene eine ausdrucksseitige Beschreibung. Die Art der Vermittlung zwischen inhaltsseitiger und ausdrucksseitiger Beschreibung darf keinen Automatismus aufzwingen, denn nicht für jede inhaltsseitige Beschreibung gibt es Einheiten der primären Informa-

tionsstrukturierung oder der konzeptuellen Ebene, die als ausdrucksseitige Beschreibung aufgefasst werden können. Mit anderen Worten, in der sekundären Informationsstrukturierung wird die **Selektion von informationellen Ressourcen als ausdrucksseitige Beschreibungen** bestimmt. Die Selektionsmechanismen hängen von den Spezifika der informationellen Ressourcen ab.

Die konzeptuelle Ebene und die primäre Informationsstrukturierung miteinander zu verbinden, bedeutet verschiedene Zwecke von und Ansprüche an Informationsmodellierung aufeinander zu beziehen. Manola und Miller (2004, Abschnitt 5.3) beschreiben den Zweck von RDF Schema in der Beschreibung von Klassen, Eigenschaften und ihren Instanzen. Auch objektorientierte Programmiersprachen wie Java besitzen diese Konstrukte, ihr Zweck und Anspruch ist jedoch ein anderer. Eigenschaften, d.h. Attribute von Klassen in Java sind immer klassenspezifisch. Der gleiche Name kann in verschiedenen Klassen Anwendung finden. In RDF Schema hingegen werden Namen global definiert. Zudem sind sie in Java präskriptiv, d.h. die Instanz einer Klasse muss die definierten Eigenschaften tragen. Bei RDF Schema hingegen ist der Grad der Präskription anwendungsspezifisch. Die Verbindung dieser unterschiedlichen Zwecke und Ansprüche kann durch die beschriebene Unterscheidung von ausdrucksseitig versus inhaltsseitig, bzw. **Extension** versus **Intension** geschehen, vgl. Abschnitt 4.4.2.1.

Die sekundäre Informationsstrukturierung kann auch ohne eine bestehende konzeptuelle Ebene eingesetzt werden. Dann beinhaltet sie eine inhaltsseitige Bescheibung, die dokumentgrammatische Konstrukte und Informationseinheiten in Dokumentinstanzen erfasst und zueinander in Beziehung setzt. Wenn allerdings bestehende konzeptuelle Ressourcen zum Einsatz kommen, erfüllt die sekundäre Informationsstrukturierung zusätzlich eine Mittlerfunktion zwischen diesen und der primären Informationsstrukturierung.

Im Idealfall kann eine Klasse von Dokumenten als Extension interpretiert und einer intensionalen Beschreibung zugeordnet werden. Allerdings ist dies nicht immer der Fall. Selbst ohne Verbindung zur konzeptuellen Ebene sind in der primären Informationsstrukturierung Regeln *und* Bedingungen zur Charakterisierung von (Klassen von) Informationseinheiten sinnvoll. Neuere Auszeichnungsvokabulare wie die in Abschnitt 2.2.1.4 angesprochene Version P5 der TEI beruhen z.B. nicht nur auf Dokumengrammatiken, sondern nutzen auch Bedingungsbeschreibungen. Auch die Verbindung zur konzeptuellen Ebene muss demnach trennen zwischen einer **dokumentgrammatikbezogenen sekundären Informationsstrukturierung** und einer **dokumentinstanzbezogenen**

sekundären Informationsstrukturierung. Welches von beiden den höheren Stellenwert einnimmt, ist abhängig von den zu verbindenden informationellen Ressourcen.

Im folgenden Kapitel wird der Ansatz der sekundären Informationsstrukturierung aus Kapitel 2.2.4 wieder aufgenommen. Er wird in der Form erweitert, dass die vertikale Verbindung informationeller Ressourcen der primären Informationsstrukturierung untereinander sowie zur konzeptuellen Ebene realisiert werden kann.

3.7. Kernpunkte des Kapitels

- Zwei vertikale Verbindungen informationeller Ressourcen sind möglich: semantische Auszeichnung, d.h. Top-Down, von der konzeptuellen Ebene zur primären Informationsstrukturierung; oder die Bedeutungsbeschreibung für Auszeichnungen, d.h. Bottom-Up, ausgehend von Dokumentgrammatiken und -instanzen.

- Das vornehmliche Anwendungsszenario für semantische Auszeichnung bildet das Semantic Web. Die Bedeutungsbeschreibung für Auszeichnungen ist zum Beispiel motiviert durch das Ziel, informationelle Ressourcen der primären Informationsstrukturierung semantisch zu validieren. Hierfür sind Regeln und Bedingungen in der primären Informationsstrukturierung nicht ausreichend.

- Ansätze zur semantischen Auszeichnung erzeugen zum einen auf generische Weise Dokumentgrammatiken und -instanzen aus bestehenden konzeptuellen Modellen. Konzepte und Konzepteigenschaften werden in Form ausgewählter dokumentgrammatischer Konstrukte realisiert. Dieses Vorgehen hat den Nachteil, dass die Möglichkeiten zur Dokumentstrukturbeschreibung in der primären Informationsstrukturierung nur in geringem Maße ausgenutzt werden. Zum anderen, im Falle von RDF, können beliebige Aussagen über informationelle Ressourcen gemacht werden. Allerdings bietet RDF keinen Mechanismus, um die Gültigkeit der Aussagen in Hinblick auf informationelle Ressourcen der primären Informationsstrukturierung zu überprüfen.

- Ansätze zur Bedeutungsbeschreibung für Auszeichnungen greifen in mehr oder weniger hohem Maße auf bestehende konzeptuelle informationelle Ressourcen zurück, um die Bedeutung von Dokumentgrammatiken oder -instanzen zu spezifizieren. Bis auf den Ansatz des XDD-Frameworks sind alle Verfahren Bottom-Up ausgerichtet

und nicht reversibel: Ein Top-Down Zugriff, d.h. aus einer Bedeutungsbeschreibung auf die informationellen Ressourcen der primären Informationsstrukturierung, ist nicht vorgesehen.

- Das gemeinsame Merkmal der meisten bestehenden Ansätze zur vertikalen Verbindung informationeller Ressourcen ist die Vermischung oder enge Verknüpfung der Ressourcen. Im Gegensatz dazu beschreibt die vorliegende Arbeit einen Ansatz, der bestehende informationelle Ressourcen unverändert belässt. Die sekundäre Informationsstrukturierung ist eine inhaltsseitige Beschreibung von Beziehungen zwischen informationellen Ressourcen. Die Ausdrucksseite bilden die informationellen Ressourcen selbst, d.h. die primäre Informationsstrukturierung und die konzeptuelle Ebene. Die sekundäre Informationsstrukturierung kann einerseits dokumentgrammatische Konstrukte und Informationseinheiten zueinander relationieren. Andererseits kann sie zwischen primärer Informationsstrukturierung und bestehender konzeptueller Ebene vermitteln.

4. Sekundäre Informationsstrukturierung

4.1. Vorbemerkung

Dieses Kapitel beschreibt die Methodologie der sekundären Informationsstrukturierung. Die Aufgabe der sekundären Informationsstrukturierung wird zum einen verstanden im Sinne von Lobin (2000): Dokumentgrammatische Konstrukte werden zueinander in Beziehung gesetzt. Dieses Vorgehen greift die vorliegende Arbeit als dokumentgrammatikbezogene sekundäre Informationsstrukturierung auf. In Ergänzung tritt die Aufgabe einer dokumentinstanzbezogenen sekundären Informationsstrukturierung. Es werden Bedingungen formuliert, die sich nur in Bezug auf Instanzen dokumentgrammatischer Konstrukte, namentlich Informationseinheiten in Dokumentinstanzen überprüfen lassen. Ein weiterer Unterschied zum Vorgehen von Lobin ist die Nutzung der sekundären Informationsstrukturierung. Sie dient zum einen – wie bei Lobin – der Beschreibung von Beziehungen zwischen informationellen Ressourcen der primären Informationsstrukturierung. Zum anderen erlaubt sie die Verbindung der primären Informationsstrukturierung und der konzeptuellen Ebene.

Das Kapitel fasst zunächst die Desiderata für eine Verbindung informationeller Ressourcen zusammen (Abschnitt 4.2), welche sich aus den Kapiteln 2 und 3 ergeben. Es folgt eine Vorstellung der Kernpunkte der Methodologie (Abschnitt 4.3), sowie ihre detaillierte Beschreibung (Abschnitt 4.4). Eine Beschreibung der in der Entwicklung befindlichen Implementation des Ansatzes (Abschnitt 4.5) schließt das Kapitel ab.

4.2. Desiderata für eine Verbindung informationeller Ressourcen

Verwendung von Standards zur Repräsentation informationeller Ressourcen Dieses Desideratum ergibt sich aus Kapitel 2. Die ausgewählten Standards sind in Tabelle 2.2 in Abschnitt 2.3 zusammengefasst. Zentrale Standards sind:

- für die primäre Informationsstrukturierung: XML 1.0, das XML Information Set, die Schemasprache RELAX NG.
- für die konzeptuelle Ebene: auf RDF aufbauende Standards, d.h. RDF Schema bzw. OWL.
- als verbindende Standards: URI und Namensräume.
- als Standard zur Zeichenkodierung: Unicode.

URI und Namensräumen kommen bei der Beschreibung der sekundären Informationsstrukturierung eine tragende Rolle zu, da sie sowohl für die konzeptuelle Ebene als auch für die primäre und sekundäre Informationsstrukturierung einsetzbar sind. Die Zeichenkodierung basiert auf Unicode.

Alle hier aufgelisteten Standards kommen zu einem festgelegten Modellierungszweck zum Einsatz, vgl. Abbildung 2.2 in Abschnitt 2.1.2. Elemente und Attribute in datenzentrierten XML-Dokumentinstanzen sind z.B. nicht Gegenstand der Modellierung. Zum einen spielen in datenzentrierten Dokumentinstanzen Konfigurationen von Informationseinheiten, d.h. Segmentierungen bzw. Hierarchisierungen, eine geringe Rolle; wichtig ist vielmehr die eindeutige Typisierbarkeit von Informationseinheiten. Da jedoch die Methode der sekundären Informationsstrukturierung den Segmentierungen und Hierarchisierungen einen hohen Stellenwert einräumt, ist sie für datenzentrierte Dokumentinstanzen nur in geringem Maße geeignet. Zum anderen sind textuelle Daten in der vorliegenden Arbeit die grundlegende Basis für die primäre Informationsstrukturierung. In datenzentrierten XML-Dokumentinstanzen ist dies nicht unbedingt der Fall.

Festlegung relevanter Eigenschaften der sekundären Informationsstrukturierung Die sekundäre Informationsstrukturierung benötigt eine formal hinreichende Beschreibung in der Art, dass sie in die vertikale Verbindung einordbar und operationalisierbar ist.

Gewährleistung von Bidirektionalität von Verbindungen zwischen informationellen Ressourcen und Reversibilität von Transformationen In Abschnitt 3.5 wurde der Nutzen einer reversiblen Transformation zwischen verschiedenen Auszeichnungsvokabulare demonstriert. Zudem wurde der Nutzen von Bidirektionalität für die Verbindung der konzeptuellen Ebene und der primären Informationsstrukturierung deutlich gemacht.

Aus der Sicht dieser informationellen Ressourcen ist die sekundäre Informationsstrukturierung „unsichtbar“. Im Gegensatz zu Ansätzen der semantischen Auszeichnung, vgl. Abschnitt 3.2, bleiben z.B. Dokumentgrammatiken unbeeinflusst von den zu verbindenden konzeptuellen Modellen auf der konzeptuellen Ebene. Anders auch als die meisten Ansätze zur Bedeutungsbeschreibung für Auszeichnung, vgl. Abschnitt 3.3, ist es nicht nötig, ein konzeptuelles Modell für die Bedeutungsbeschreibung zu erstellen. Vorhandene Modelle auf der konzeptuellen Ebene können auf informationelle Ressourcen der primären Informationsstrukturierung bezogen werden.

Beschreibung von Operationen Ein Defizit vieler in Kapitel 3 diskutierten Ansätze liegt in ihrer mangelnden Operationalisierbarkeit. Die Notwendigkeit für Operationen zwischen informationellen Ressourcen wird zwar gesehen. Ihr wird jedoch selten durch eine Explikation der Operationen Genüge getan. Die vorliegende Arbeit definiert hingegen Operationen und beschreibt sie in semi-formaler Weise.

Eine Exemplifikation des Ansatzes Die Methodologie wird exemplifiziert in der Domäne linguistischer, textueller Korpora. Diese werden im zweiten Teil der vorliegenden Arbeit behandelt.

4.3. Kernpunkte der Methodologie

Sekundäre Informationsstrukturierung beinhaltet eine Menge von **Aussagen**, die dazu dienen, informationelle Ressourcen aufeinander zu beziehen. Exemplarische Aussagen, welche die Dokumentgrammatik der TEI, entsprechende Dokumentinstanzen und – als Beispiel für die konzeptuelle Ebene – einen Auschnitt aus der lexikalischen Datenbank **WordNet** Fellbaum (1998) miteinander verknüpfen, sind in Abbildung 4.1 dargestellt. Sie zeigt eine Konkretisierung der in der Einleitung zu dieser Arbeit schematisch dargestellten Kernpunkte der Methodologie.

Gegenstand der Aussagen[1] sind **Konzepte**, d.h. unäre **Prädikate**, die allein durch ihren Namen definiert sind. Ein Beispiel ist sekStruk:Paragraph. Die Aussagen über Konzepte erfüllen die folgenden sechs Aufgaben:

(1) Selektion ausdrucksseitiger Beschreibungen aus der primären Informationsstrukturierung. Hierzu kommt das Prädikat *sekStruk2primStruk* zum Einsatz. Es wird differen-

[1] Die Aussagen sind in der Tripel-Notation wiedergegeben, welche in Abschnitt 2.2.3.2 vorgestellt wurde.

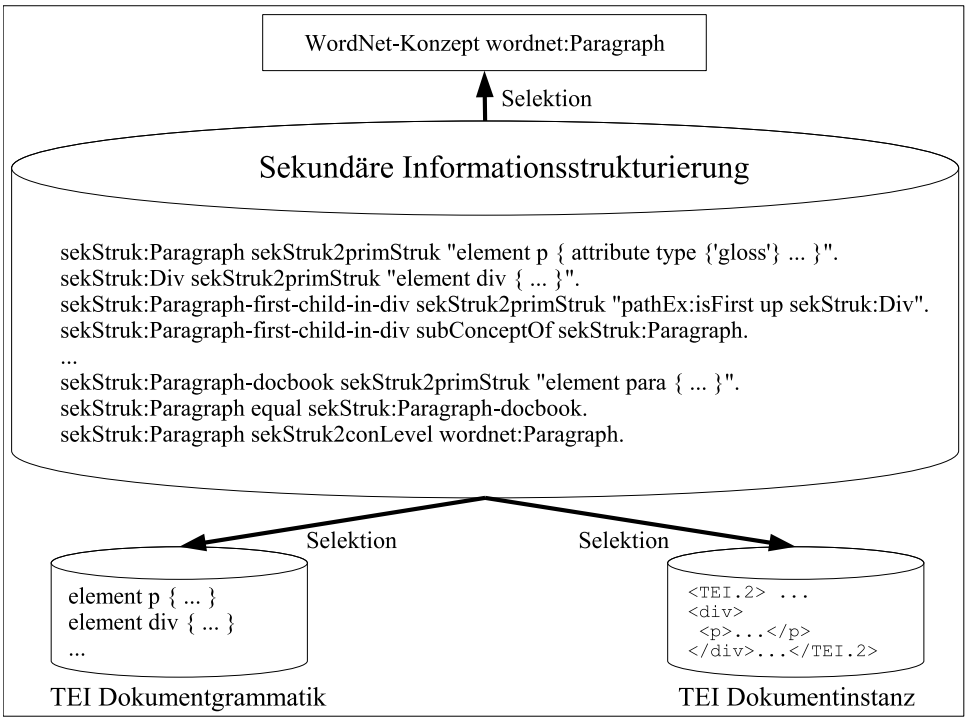

Abbildung 4.1.: Exemplifizierung der vertikalen Interrelationierung informationeller Ressourcen

ziert zwischen der Selektion von informationellen Ressourcen aus Dokumentgrammatiken versus Selektionen von informationellen Ressourcen aus Dokumentinstanzen. Im Rahmen dokumentgrammatikbezogener sekundärer Informationsstrukturierung werden Deklarationen dokumentgrammatischer Konstrukte selektiert. Die Aussage (sekStruk:Paragraph *sekStruk2primStruk* `"element p { attribute type { 'gloss' } ... }"`.) selektiert z.B. für das Konzept sekStruk:Paragraph die Deklaration eines `<p>` Elements mit einem `type` Attribut, welches den Wert `gloss` hat[2]. Die dokumentgrammatischen Konstrukte greifen auf die kompakte Syntax von RELAX NG zurück, vgl. Clark (2002). Sie wird einleitend zu Abschnitt 4.4 näher erläutert.

(2) Im Rahmen dokumentinstanzbezogener sekundärer Informationsstrukturierung eine Selektion von Informationseinheiten aus einzelnen Dokumentinstanzen. In diesem Fall erhält das Prädikat *sekStruk2primStruk* als Argument einen Pfadausdruck, der durch das Präfix `pathEx:` eingeleitet ist. Eine Beispielaussage lautet (sekStruk:Paragraph-first-child-in-div *sekStruk2primStruk* `"pathEx:isFirst up sekStruk:Div"`.). Der Pfadausdruck selektiert diejenigen `<p>` Elemente, welche das erste Tochterelement eines `<div>` Elementes sind.

(3) Ebenfalls im Rahmen dokumentinstanzbezogener sekundärer Informationsstrukturierung eine Selektion von Informationseinheiten aus mehreren Dokumentinstanzen[3]. Derartige Selektionen werden ebenfalls mit dem Prädikat *sekStruk2primStruk* durchgeführt und durch ein Präfix `layerRel:` ausgedrückt.

(4) Die Nutzung von Inferenzen in der sekundären Informationsstrukturierung. Die vorgenommenen Selektionen von Einheiten der primären Informationsstrukturierung gewinnen eine zweifache inferentielle Kraft, zum einen hinsichtlich der Konzepthierarchie, zum anderen hinsichtlich interkonzeptueller Beziehungen. In der Konzepthierarchie, konstituiert durch das Prädikat *subConceptOf*, erben untergeordnete Konzepte von übergeordneten Konzepten deren Eigenschaften. Z.B. selektiert das Konzept sekStruk:Paragraph alle Elementknoten mit dem Namen `p`. Nur diese Elementknoten finden Eingang in die Bestimmung der ausdrucksseitigen Beschreibung des subordinierten Konzepts sekStruk:Paragraph-first-child-in-div. Der Pfadausdruck

[2] Bei der Selektion dokumentgrammatischer Konstrukte ist es möglich, Spezialisierungen der Konstrukte vorzunehmen, z.B. von `attribute type { text }` nach `attribute type { 'gloss' }`. Die Regeln zur Überprüfung der Spezialisierbarkeit dokumentgrammatischer Konstrukte werden in Abschnitt 4.4.2.2 vorgestellt.

[3] Das spezifische Auszeichnungsformat, dem die Dokumentinstanzen genügen müssen, wird in Abschnitt 4.4.1.3 vorgestellt.

`isFirst up 'sekStruk:Div'` wird also nur für `<p>` Elemente realisiert. In dem Pfadausdruck zeigt sich auch die Rolle interkonzeptueller Beziehungen: Der Test von Eigenschaften eines Knotens referiert hier auf ein anderes Konzept, d.h. das Konzept sekStruk:Div. Für dieses Konzept ist die Menge aller Knoten mit dem Namen `div` selektiert. Löst man die Inferenzen auf, welche sich durch Konzepthierarchie und interkonzeptuelle Beziehungen ergeben, lautet der Pfadausdruck `'p' isFirst up 'div'`.

(5) Die Interrelationierung dokumentgrammatischer Konstrukte bzw. der Konzepte, die die Konstrukte selektieren. Die Interrelationierung wird durch Aussagen mit dem Prädikat *equal* beschrieben. Eine Beispielaussage lautet (sekStruk:Paragraph *equal* sekStruk:Paragraph-docbook.). Diese Aussage beschreibt die Identität der Elemente `<p>` und `<para>`, die in den beiden Konzepten sekStruk:Paragraph-docbook selektiert sind.

(6) Die Selektion informationeller Ressourcen aus der konzeptuellen Ebene. Hierzu kommt das Prädikat *sekStruk2conLevel* zur Anwendung. Im Beispiel ist das Konzept wordnet:Paragraph aus WordNet selektiert. Die Aussage lautet (sekStruk:Paragraph *sekStruk2conLevel* wordnet:Paragraph.).

Derartige Aussagen in der sekundären Informationsstrukturierung bilden die Grundlage für verschiedene **Operationen**: konzeptbezogene Validierung, Suche und Transformation. Informationelle Ressourcen der primären Informationsstrukturierung sind validierbar hinsichtlich der Eigenschaften, die durch Aussagen mit dem Prädikat *sekStruk2primStruk* postuliert sind. Dabei werden die Konzepthierarchie und interkonzeptuelle Beziehungen in der exemplifizierten Weise berücksichtigt. So kann für das Konzept sekStruk:Paragraph überprüft werden, ob die Deklaration des `<p>` Elements in einer Dokumentgrammatik tatsächlich ein Attribut `type` mit dem Wert `gloss` enthält. Eine Validierung ist auch in Dokumentinstanzen möglich. Es kann z.B. durch eine Aussage beschrieben werden, dass alle `<p>` Elemente als ausdrucksseitige Beschreibung des Konzeptes sekStruk:Paragraph-first-child-in-div aufzufassen sind[4].

Komplementär zur Validierungsoperation stehen Suchanfragen, deren Auswertung alle ausdrucksseitigen Beschreibungen eines Konzepts in Form von URI liefert, z.B. URI für alle Elementknoten mit dem Namen `p`, die erstes Kindelement eines `<div>` Elementes sind. Die Eingabe zu Validierungs- und Suchoperationen können auch informationelle Ressourcen aus der konzeptuellen Ebene sein. Berücksichtigt werden dabei Aussagen mit dem Prädikat *sekStruk2conLevel*, in denen auf diese informationellen Ressourcen

[4]Der Validierungsmechanismus wird detailliert in Abschnitt 4.4.2.3 vorgestellt.

referiert wird. Bei Eingabe des Konzepts wordnet:Paragraph in eine Suchanfrage werden z.B. als Ergebnis URI für die entsprechende Deklaration in der TEI Dokumentgrammatik und für eine Menge von Elementknoten in Dokumentinstanzen ausgegeben.

Schließlich können Aussagen mit dem Prädikat *equal* ausgewertet werden, um Dokumentinstanzen zu transformieren. Wenn Dokumentinstanzen der TEI Dokumentgrammatik gegeben sind, führt die Auswertung der Aussage (sekStruk:Paragraph *equal* sekStruk:Paragraph-docbook.) zur Transformation der `<p>` Elemente in `<para>` Elemente.

4.4. Charakteristika sekundärer Informationsstrukturierung

Im Folgenden wird die Anwendung der im vorhergehenden Abschnitt exemplifizierten Prädikate und die Operationalisierung der Aussagen detailliert beschrieben. Als exemplarische informationelle Ressourcen der primären Informationsstrukturierung kommen hauptsächlich die Dokumentinstanzen in Abbildung 4.1 zur Anwendung. Diese wurden teilweise bereits in Abschnitt 3.5 eingeführt, als prototypische Exemplare für eine Anforderung nach Bidirektionalität.

Alle Dokumentinstanzen nutzen fiktive Namensräume, die auf existierenden Auszeichnungsvokabularen beruhen. Die erste Dokumentinstanz verwendet ein Fragment der Dokumentgrammatik von XHTML, vgl. Pemperton et al. (2002). Die zweite Dokumentinstanz nutzt einen Ausschnitt der Dokumentgrammatik der TEI in der Version P4. Beide Dokumentinstanzen beinhalten Auszeichnungen einer definitorischen Liste. Die dritte Dokumentinstanz beinhaltet Auszeichnungen der gleichen, textuellen Primärdaten, allerdings nach anderen, d.h. typographischen Gesichtspunkten.

Die drei zu Grunde liegenden Dokumentgrammatiken sind in Beispiel 4.2 dargestellt. Die Darstellung der Dokumentgrammatiken nutzt wieder die kompakte Syntax von RELAX NG. Das Schlüsselwort `start` bestimmt das Startsymbol einer Grammatik, d.h. das Wurzelelement der Dokumentinstanz. Im Falle von XHTML ist dies im Beispiel das `<html:DL>` Element. Elementdeklarationen werden durch das Schlüsselwort `element` eingeleitet. Die Inhaltsmodelle stehen in geschweiften Klammern. Die Abfolge von Element- und Attributdeklarationen in Inhaltsmodellen ist irrelevant. Eine Sequenz von Elementen wird durch Kommata ausgedrückt. Statusbezeichnungen für Elemente bzw. Attribute stehen hinter den deklarierten Einheiten. Die Deklaration `element paragraph { element line { text }*}` erlaubt z.B. optionales bis beliebig häufiges Auftreten des `<line>` Elements. Textuellen Inhalt von Elementen oder Attributen, also den Datentyp

(4.1)
```
<html:DL xmlns:html="http://example.com/defList-html">
<html:DT>V-TE</html:DT>
<html:DD>Label for the annotation of Japanese verbs in
the assimilation form</html:DD>
</html:DL>
-----------------------------------------------------
<tei:list xmlns:tei="http://example.com/defList-tei"
       type="gloss">
<tei:head>V-TE</tei:head>
<tei:item>Label for the annotation of Japanese verbs in
the assimilation form</tei:item>
</tei:list>
-----------------------------------------------------
<layout:paragraph
xmlns:layout="http://example.com/defList-layout">
<layout:line>V-TE</layout:line>
<layout:line>Label for the annotation of Japanese verbs in
</layout:line>
<layout:line>the assimilation form</layout:line>
</layout:paragraph>
```

Beispiel 4.1: Exemplarische Dokumentinstanzen als Grundlage für die formale Beschreibung informationeller Ressourcen

`xsd:string`, signalisiert das Schlüsselwort `text`. In der Dokumentgrammatik für die zweite Dokumentinstanz kommt ein weiterer Datentyp vor, `xsd:Name`. Er wird für das Attribut `type` verwendet und entstammt der von Biron und Malhotra (2001) beschriebenen Datentypdefinition von XML Schema, welche auch in RELAX NG einsetzbar ist.

Für diese exemplarischen informationellen Ressourcen der primären Informationsstrukturierung sollen mit der Methodologie der sekundären Informationsstrukturierung folgende Ziele erreicht werden:

Anforderung 1 : Die identische Bedeutung der definitorischen Listen soll formal be-

(4.2)
```
namespace html="http://example.com/defList-html"
start =
  element html:DL {
    element html:DT { text },
    element html:DD { text }
  }
-------------------------------------------------
namespace tei="http://example.com/defList-tei"
start =
  element tei:list {
    attribute type { xsd:Name },
    element tei:head { text },
    element tei:item { text }
  }
-------------------------------------------------
namespace layout="http://example.com/defList-layout"
start =
  element layout:paragraph {
    element layout:line { text }*
  }
```

Beispiel 4.2: Exemplarische Dokumentgrammatiken für die formale Beschreibung informationeller Ressourcen

schrieben werden. Die Bedeutungsbeschreibung soll reversibel sein und sich auf dokumentgrammatische Konstrukte beziehen und nicht auf Informationseinheiten in Dokumentinstanzen.

Anforderung 2 : Eine Transformationsprozedur zwischen den Auszeichnungen nach der TEI und denen nach XHTML soll definiert werden. Die Transformation soll reversibel anwendbar sein, d.h. die Umkehrbarkeit des Prozesses soll gewährleistet sein.

Anforderung 3 : Diese Anforderung ist ein Beispiel für dokumentinstanzbezogene se-

kundäre Informationsstrukturierung, welche Beziehungen zwischen verschiedenen XML-Dokumentinstanzen als Selektionsmechanismus anwendet. Folgende Bedingung soll für diese Dokumentinstanzen erfüllt sein: Definitorische Listen müssen aus drei bis vier Linien bestehen, wobei die erste Linie für den zu definierenden Ausdruck, die weiteren Zeilen für die Definition reserviert sind. Die Bedingung soll unabhängig von den spezifischen Dokumentgrammatiken gelten, also auch für andere Dokumentgrammatiken bzw. Auszeichnungsvokabulare überprüfbar sein.

4.4.1. Aussagen über informationelle Ressourcen

4.4.1.1. Selektion dokumentgrammatischer Konstrukte

Ein Konzept, dass durch dokumentgrammatikbezogene sekundäre Informationsstrukturierung Konstrukte aus Dokumentgrammatiken selektiert, ist funktional äquivalent zu einem **Pattern in RELAX NG**. Pattern versehen Deklarationen von Elementen, Attributen oder Datentypen[5] mit einem eindeutig identifizierbaren Namen, auf den referenziert werden kann. Beispiel 4.3 demonstriert die Funktionsweise von Pattern.

(4.3)
```
namespace tei="http://example.com/defList-tei"
namespace html="http://example.com/defList-html"
start = definitionList-html | definitionList-tei
definitionList-html = element html:DL { dt-html, def-html }
definitionList-tei =
 element tei:list { attribute type { "gloss" },
  dt-tei, def-tei }
dt-html = element html:DL { text }
dt-tei = element tei:head { text }
def-html = element html:DD { text }
def-tei = element tei:item { text }
```

Beispiel 4.3: Beispiel für die Verwendung von Pattern in RELAX NG

[5]RELAX NG erlaubt es, auch andere Bestandteile von Dokumentgrammatiken wie z.B. Verbindungen von Element- und Attributdeklarationen als Pattern zu beschreiben. Von diesen Möglichkeiten wird in der vorliegenden Arbeit jedoch abgesehen. In der sekundären Informationsstrukturierung selektierte Pattern wären sonst nicht operationalisierbar für die konzeptbezogene Suche, Validierung etc., vgl. Abschnitt 4.4.2.3.

Das Startsymbol kann entweder dem Pattern `definitionList-html` entsprechen oder dem Pattern `defintionList-tei`. Die Pattern referenzieren auf die alternativen Wurzelelemente `<html:DL>` oder `<tei:list>`. Ihre Inhaltsmodelle enthalten wieder Pattern, namentlich `dt-hmtl, def-html` bzw. `dt-tei, def-tei`. Diese Pattern schließlich enthalten keine weiteren Pattern, sondern allein Deklarationen von Elementen.

Diese Dokumentgrammatik verbindet die Elementdeklarationen der ersten beiden Dokumentgrammatiken aus Beispiel 4.2 und erlaubt es, die ersten beiden Dokumentinstanzen in Beispiel 4.1 zu validieren. Durch Aussagen mit dem Prädikat *sekStruk2primStruk* in der sekundären Informationsstrukturierung werden nun die Pattern nicht unmittelbar in die Dokumentgrammatik integriert, sondern separat definiert. Der Name des Patterns ist in diesen Aussagen der Name des Konzepts, und der Inhalt das Pattern ist das zweite Argument der Aussage. Die in Beispiel 4.3 demonstrierten Pattern lassen sich in der sekundären Informationsstrukturierung wie in Beispiel 4.4 wiedergeben.

(4.4)
```
sekStruk:definitionList-html sekStruk2primStruk
     "element html:DL { sekStruk:dt-html, sekStruk:def-html }".
    sekStruk:definitionList-tei sekStruk2primStruk
     "element tei:list { attribute type { "gloss" },
      sekStruk:dt-tei, sekStruk:def-tei }".
    sekStruk:dt-html sekStruk2primStruk "element html:DL { text }".
    sekStruk:def-html sekStruk2primStruk "element html:DD { text }".
    sekStruk:dt-tei sekStruk2primStruk "element tei:head { text }".
    sekStruk:def-tei sekStruk2primStruk "element tei:item { text }".
```

Beispiel 4.4: Pattern aus Beispiel 4.3 in Aussagen in der sekundären Informationsstrukturierung

Was ist durch die von Dokumentgrammatiken separierte Definition von Pattern gewonnen? Der Vorteil der separierten Definition besteht darin, dass die informationelle Ressource „Dokumentgrammatik“ nicht verändert werden muss, um ein Konzept in der sekundären Informationsstrukturierung zu beschreiben. Nur diejenigen Konzepte, respektive Pattern werden definiert, welche für die sekundäre Informationsstrukturierung relevant sind. Zudem ist eine Spezialisierung dokumentgrammatischer Konstrukte, vgl. Abschnitt 4.4.2.2, möglich, die keine Auswirkungen auf bestehende Dokumentinstanzen hat. So ist in der ursprünglich vorliegenden Dokumentgrammatik möglicherweise der

Wertebereich des `type` Attributs nicht auf die Zeichenfolge `"gloss"` festgelegt. Würde man dies direkt in der Dokumentgrammatik tun, wären bestehende Dokumentinstanzen möglicherweise nicht mehr validierbar.

Bestehende Pattern der Dokumentgrammatiken lassen sich ebenfalls durch Aussagen mit dem Prädikat *sekStruk2primStruk* selektieren. Insbesondere bei komplexen Dokumentgrammatiken wie der TEI und DocBook geben Pattern bzw. ihr Pendant in DTDs, d.h. Parameter-Entitäten, Aufschluß über die Bedeutung der Deklarationen. So trennt die DocBook Dokumentgrammatik mittels Parameter-Entitäten Elemente zur Auszeichnung von Strukturen wie Abschnitten, Paragraphen etc. versus Elemente zur Auszeichnung von Inhalten wie Elemente für Verweise etc. Auch der Ansatz von Welty und Ide (1999), der in Abschnitt 3.3 vorgestellt wurde, nutzt bestehende Parameter-Entitäten als Ausgangspunkt für eine Bedeutungsbeschreibung für Auszeichnungen.

Im Gegensatz zu RELAX NG Pattern bzw. Parameter-Entitäten erlaubt es die sekundäre Informationsstrukturierung, die Beziehung zwischen Konzepten der dokumentgrammatikbezogenen sekundären Informationsstrukturierung zu beschreiben. Hierfür wird das Prädikat *componentOf* verwendet, vgl. Beispiel 4.5.

(4.5)
```
sekStruk:dt-tei componentOf sekStruk:definitionList-tei.
sekStruk:def-tei componentOf sekStruk:definitionList-tei.
```

Beispiel 4.5: Anwendung des Prädikats *componentOf*

Die Aussagen in dem Beispiel machen explizit, dass die Konzepte sekStruk:dt-tei bzw. sekStruk:def-tei Bestandteil der Definition des Konzepts sekStruk:definitionList-tei sind. Die Selektion des `<tei:item>` Elements für das Konzept sekStruk:def-tei ist an die entsprechende Selektion des `<tei:list>` Elements gebunden. Potentiell vorhandene, weitere Deklarationen des `<tei:item>` Elements in der gleichen Dokumentgrammatik sind also nicht selektiert.

Die Definition des *componentOf* Prädikats realisiert ein Verfahren, welches im Rahmen des XDD-Frameworks entwickelt wurde, vgl. Abschnitt 3.3. Im XDD-Framework werden Dokumentgramatiken und Dokumentinstanzen als Mengen von Ausdrücken beschrieben. Ausdrücke sind teilweise von anderen Ausdrücken abgeleitet. Die Ableitung erfolgt schrittweise, wie in Beispiel 4.6 demonstriert wird.

Im ersten Ableitungsschritt `a` ist die gesamte Dokumentgrammatik für eine definitorische Liste wiedergegeben. Im nächsten Ableitungsschritt `a'` sind die Deklarationen

```
(4.6) a   start =                        a' start =
          element html:DL {                element html:DL {
          element html:DT { text },        $dt-html, $def-html
          element html:DD { text }}
      a'' start = $definitionList-html
      b  start =                                b' start=
          element tei:list {                     element tei:list
          attribute type { xsd:Name },        { attribute type
          element tei:head { text },               { "gloss" },
          element tei:item { text }}               $dt-tei, $def-tei }
      b'' start = $definitionList-tei
```

Beispiel 4.6: Ableitungsschritte nach dem XDD-Framework

der Elemente `html:DT` und `html:DD` als Ausdrücke repräsentiert, namentlich `$dt-html` und `$def-html`. In der sekundären Informationsstrukturierung werden die Ableitungsschritte durch das Prädikat *sekStruk2primStruk* wiedergegeben, z.B. in der Aussage (sekStruk:dt-html *sekStruk2primStruk* `"element html:DL { text }"`.). Im Ableitungsschritt `a''` ist der Ausdruck `$definitionList-html` enthalten. Er beinhaltet die Ausdrücke des vorherigen Ableitungsschrittes, z.B. `$dt-html`. Diese Beziehung wird in der sekundären Informationsstrukturierung durch das beschriebene Prädikat *componentOf* explizit. *componentOf* erfüllt somit die Anforderung 1 nach Reversibilität, die einleitend zu Abschnitt 4.4 auf Seite 97 formuliert wurde. Aussagen mit dem Prädikat machen explizit, in welcher Abfolge die Selektionen aufgelöst werden müssen, um zur ursprünglichen Dokumentgrammatik zu gelangen.

4.4.1.2. Selektion von Informationseinheiten aus singulären Dokumentinstanzen

Informationseinheiten in singulären Dokumentinstanzen werden durch Pfadausdrücke selektiert. Das Selektionskriterium ist die Realisierbarkeit des von der Informationseinheit ausgehenden Pfades. Diese Art der Selektion lässt sich als Beschreibung eines Kontextkriteriums für Knoten auffassen. Ein Knoten wird selektiert, wenn er das Kontextkriterium erfüllt. Die Selektion bedeutet also eine **Kontextspezifikation anhand**

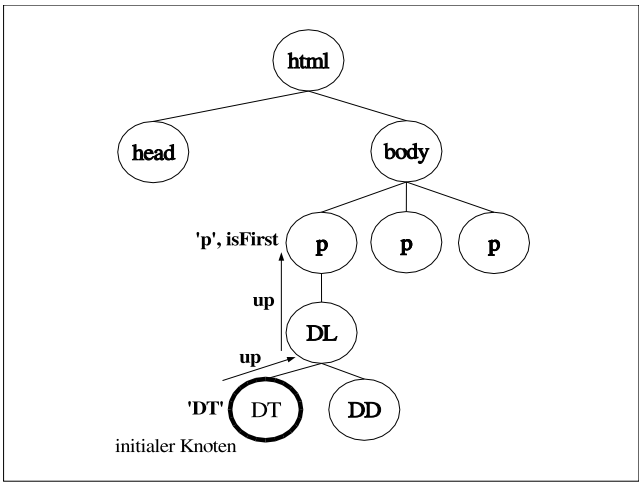

Abbildung 4.2.: Visualisierung der Kontextspezifikation mit Caterpillar-Ausdrücken

strukturbezogener Eigenschaften von Informationseinheiten.

Als Pfadsprache zur Spezifikation der Kontexte kommen in dieser Arbeit so genannte **Caterpillar-Ausdrücke** zum Einsatz, vgl. Brüggemann-Klein und Wood (2000). Sie umfassen ein Alphabet von Bewegungen und Tests über die Knoten in der Dokumentstruktur: `up`, `left`, `right`, `first`, `last`, `isFirst`, `isLast`, `isLeaf`, `isRoot`. Hinzu kommen der Kleene-Star Operator und ein Test auf Eigenschaften eines Knotens. Abbildung 4.2 visualisiert die Anwendung von Caterpillar-Ausdrücken in einer Dokumentinstanz. Der Kontext des `<DT>` Elements wird durch zwei Bewegungen nach oben spezifiziert. Es folgt ein Namenstest auf das `<p>` Element. Ein weiterer Test verifiziert die initiale Position dieses Elements. Der vollständige Caterpillar-Ausdruck lautet `‘DT’ up up isFirst ‘p’`.

Die hier vorgestelle Kontextspezifikation mit Caterpillar-Ausdrücken spielt eine Rolle bei der dokumentinstanzbezogenen sekundären Informationsstrukturierung. Ihre Funktion entspricht denen der deiktischen Ausdrücke im BECHAMEL-Projekt, vgl. Abschnitt 3.3, die mittels XPath realisiert werden. Caterpillar-Ausdrücke lassen sich jedoch, im Gegensatz zu XPath, in die Taxonomie formaler Sprachen einordnen, welche in Abschnitt 2.2.1.3 vorgestellt wurde. Caterpillar-Ausdrücke beschreiben Caterpillar-Automaten, die Bewegungen und Tests über Baumstrukturen ausführen. Die Baumstruktur kann als eine Klasse von Bäumen betrachtet werden, die von einer regulären Baumgrammatik erzeugt

wird. Jede Klasse von Bäumen, die sich durch reguläre Baumgrammatiken erzeugen lässt, kann auch durch Caterpillar-Ausdrücke erzeugt werden. Den formalen Beweis hierfür liefern Brüggemann-Klein und Wood (2000). Sie kommen außerdem zu dem Schluss, dass der umgekehrte Fall nicht immer gilt: Nicht alle durch Caterpillar-Ausdrücke erzeugten Bäume lassen sich durch reguläre Baumgrammatiken beschreiben.

Beschreibungen von Selektionen in Dokumentinstanzen sind von Selektionen in Dokumentgrammatiken entkoppelt. Insofern wäre es sinnvoll, diese beiden Formen der sekundären Informationsstrukturierung – dokumentgrammatikbezogen versus dokumentinstanzbezogen – aufeinander zu beziehen. Zwar gibt es Ansätze, Anfragesprachen und Dokumentgrammatiken formal zueinander in Beziehung zu setzen, vgl. z.B. Mönnich et al. (2001). Diese befinden sich jedoch derzeit im Stadium einer theoretischen Diskussion. Eine implementationsnahe Algorithmisierung ist noch nicht abzusehen. Die in dieser Arbeit verwendeten Caterpillar-Ausdrücke sind zumindest in die beschriebene Taxonomie integrierbar. Dies erleichtert die Formulierung von Bedingungen bzw. Selektionen, die von gegebenen Regeln in Dokumentgrammatiken lizensierbar sind.

Ein weiterer Grund für den Einsatz von Caterpillar-Ausdrücken wurde bereits in Abschnitt 4.3 angesprochen. Im Gegensatz zu komplexeren Pfadsprachen wie XPath stellen Caterpillar-Ausdrücke nur ein geringes Inventar an Tests und Bewegungen über die Dokumentstruktur bereit. Deshalb ist es auf generische Weise möglich, die Tests auf Eigenschaften von Knoten mit Inferenzen auf andere Konzepte in der sekundären Informationsstrukturierung zu füllen und zudem die inferentielle Kraft der Konzepthierarchie zu nutzen.

Abbildung 4.3 visualisiert beide Formen von Inferenzen. Das Konzept sekStruk:Paragraph selektiert die Deklaration des <p> Elements in der Dokumentgrammatik und die Elementknoten in Dokumentinstanzen mit dem Namen p. Durch das Prädikat *subConceptOf* ist das Konzept sekStruk:Paragraph-in-division dem Konzept sekStruk:Paragraph untergeordnet. Diese Anordnung in der Konzepthierarchie bewirkt, dass die Selektion von Informationseinheiten im Konzept sekStruk:Paragraph-in-division nicht von allen Informationseinheiten in der primären Informationsstrukturierung ausgeht, sondern nur von den <p> Elementen. Der nicht explizit angegebene Startknoten für den Caterpillar-Ausdruck lautet also p. Ebenfalls implizit ist im Caterpillar-Ausdruck der Test, der für den Knoten überhalb von <p> Elementen ausgeführt wird. Der Test wird nur explizit durch die Auflösung der Referenz auf das Konzept sekStruk:division. Wenn alle impliziten Informationen explizit gemacht werden, lautet der Caterpillar-Ausdruck

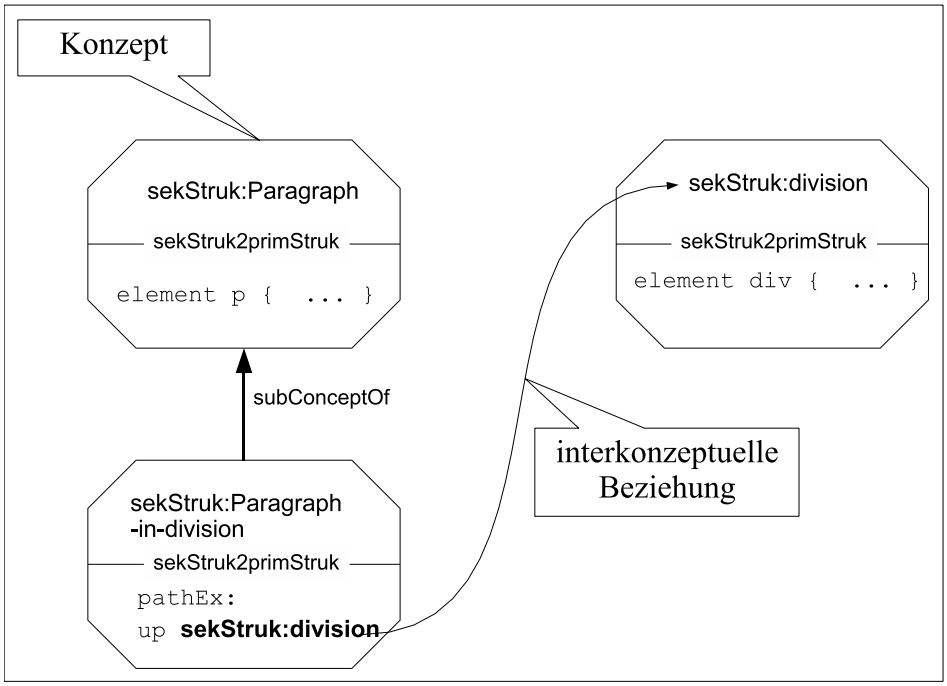

Abbildung 4.3.: Caterpillar-Ausdruck, der Inferenzen aus der Konzepthierarchie und aus interkonzeptuellen Beziehungen nutzt

demnach `'p' up 'div'`.

Die Nutzung der inferentiellen Kraft von Konzepthierarchie und interkonzeptuellen Beziehungen erlaubt es, Pfadausdrücke in unspezifischer Weise, d.h. ohne Bezug auf Elementnamen etc. aus spezifischen Dokumentgrammatiken zu formulieren. Allein in übergeordneten Konzepten muss durch Aussagen zur dokumentgrammatikbezogenen sekundären Informationsstrukturierung festgelegt werden, welche dokumentgrammatischen Konstrukte in den Pfadausdrücken bei Knotentests zur Anwendung kommen sollen. Der Caterpillar-Ausdruck `up sekStruk:division` kann z.B. auch für Dokumentinstanzen verwendet werden, die nach der DocBook Dokumentgrammatik ausgezeichnet sind. Hierzu müssen nur statt der `<p>` und `<div>` Elemente die `<para>` bzw. `<section>` Elemente in den entsprechenden Aussagen selektiert werden.

Im Ansatz des BECHAMEL-Projekts, vgl. Abschnitt 3.3, wird eine ähliche Entkopplung konkreter Namen in Dokumentinstanzen von ihrer Anwendung beschrieben. Dies wird durch die Trennung in Propagation Sentences versus Application Sentences erreicht.

Propagation Sentences verküpfen konkrete Namen mit einfachen Eigenschaftsbeschreibungen, Application Sentences enthalten komplexere Eigenschaftsbeschreibungen, die sich nur auf Propagation Sentences beziehen, nicht aber auf die konkreten Namen der Elemente etc. Der Unterschied zwischen Application Sentences und sekundärer Informationsstrukturierung besteht darin, dass sekundäre Informationsstrukturierung deklarativ vorgeht, während im BECHAMEL-Projekt die verschiedenen Sentences prozedural ineinander überführt werden.

4.4.1.3. Selektion von Informationseinheiten aus multiplen Dokumentinstanzen

Bereits in Abschnitt 2.2.1.5 wurde darauf hingewiesen, dass der Zwang zur hierarchischen Strukturierung in Dokumentinstanzen nicht immer den zu repräsentierenden Objekten gerecht wird. Die Problematik wurde einleitend zu Abschnitt 4.4, in Beispiel 4.1 exemplifiziert. Die Auszeichnungen für das Layout der definitorischen Liste lassen sich nicht in einer hierarchischen Form mit den anderen beiden Dokumentinstanzen repräsentieren.

Der in dieser Arbeit favorisierte, in Witt (2002) ausführlich vorgestellte Ansatz zur Lösung des Problems beinhaltet eine **multiple Auszeichnung primärdatenidentischer, textueller Daten**. Bei diesem Verfahren werden die gleichen **textuellen Primärdaten** mehrfach, d.h. auf separaten **Auszeichnungsebenen** verwendet. Jede Auszeichnungsebene ist eine separate XML-Dokumentinstanz. In dieser können, unter Berücksichtigung der Wohlgeformtheitsbedingungen von XML, sowohl Segmentierungen als auch Hierarchisierungen vorgenommen werden. Die Relationierung der Dokumentinstanzen wird als separater Prozess realisiert. Die eindeutige Ordnung der Zeichen in der Primärdatenbasis und in den Kopien dient dabei als Referenzpunkt für alle Auszeichnungen. Zur Demonstration dieses Verfahrens wird in Beispiel 4.7 eine Sequenz der Zeichen des Primärdatums aus Beispiel 4.1 auf Seite 96 enumeriert.

(4.7)
```
V  -  T E
1  2  3 4
```

Beispiel 4.7: Enumerierung textueller Primärdaten

Die Auszeichnung der Zeichen `V-TE` als `<html:DT>` Element ist z.B. ein Segment, das bei dem Zeichen Nummer 1 beginnt und bei dem Zeichen Nummer 4 endet. Die Stärke des Ansatzes besteht in der Analysemethodologie, welche diese Sicht auf textuelle

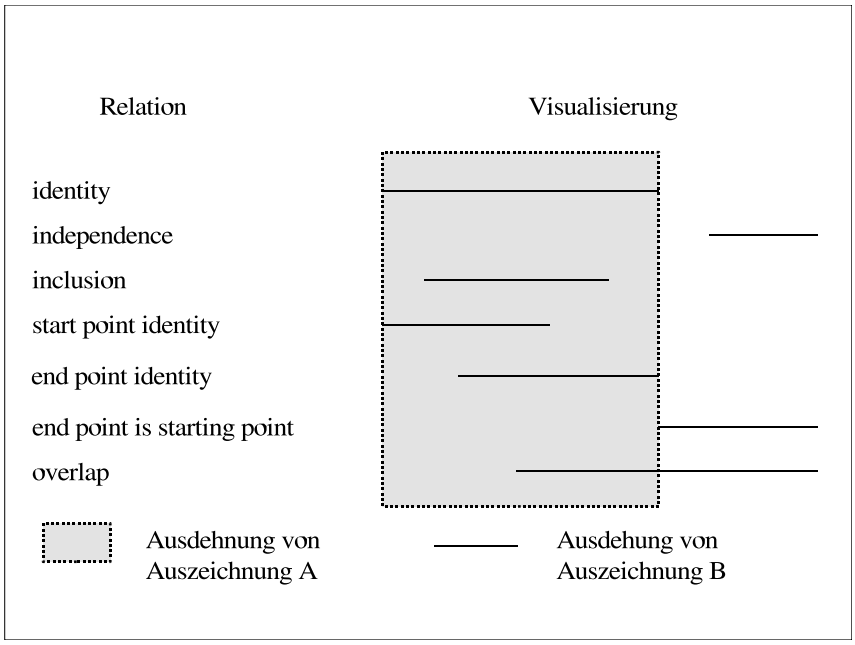

Abbildung 4.4.: Zeitlogische Beziehungen zwischen Auszeichnungseinheiten

Daten ermöglicht. Abstrakte, **zeitlogische Relationen**[6], vgl. Abbildung 4.4, dienen der Beschreibungen von Beziehungen zwischen den Auszeichnungsebenen.

In Beispiel 4.1 besteht die Beziehung *identity* z.B. zwischen den Elementen `<html:DL>`, `<tei:list>` und `<layout:paragraph>`. Die Beziehung *start_point_identity* besteht z.B. zwischen dem zweiten `<layout:line>` Element einerseits und den `<html:DD>` bzw. `<tei:item>` Elementen andererseits. In der sekundären Informationsstrukturierung lassen sich die Beziehungen durch Aussagen mit einer fixen Menge von Prädikaten beschreiben. Eine Beispielaussage ist (definitionList-html *sekStruk:identity* definitionList-tei.). Die Beschreibung von Beziehungen zwischen Auszeichnungsebenen ist aber in der sekundären Informationsstrukturierung ein Mechanismus zur Selektion von Informationseinheiten, so wie die vorgestellten Caterpillar-Ausdrücke. Durch die obige Aussage würden nur diejenigen `<html:DL>` Elemente selektiert, welche eine Identitätsbeziehung zu `<tei:list>` Elementen haben. Um auch diesen Selektionsmechanismus in gleicher Weise wie die Se-

[6]Eine Diskussion zeitlicher Relationen findet sich in Allen und Ferguson (1994). Ihre Anwendung auf die Beziehung verschiedener textueller Auszeichnungssebenen beschreiben Durusau und O'Donnell (2002).

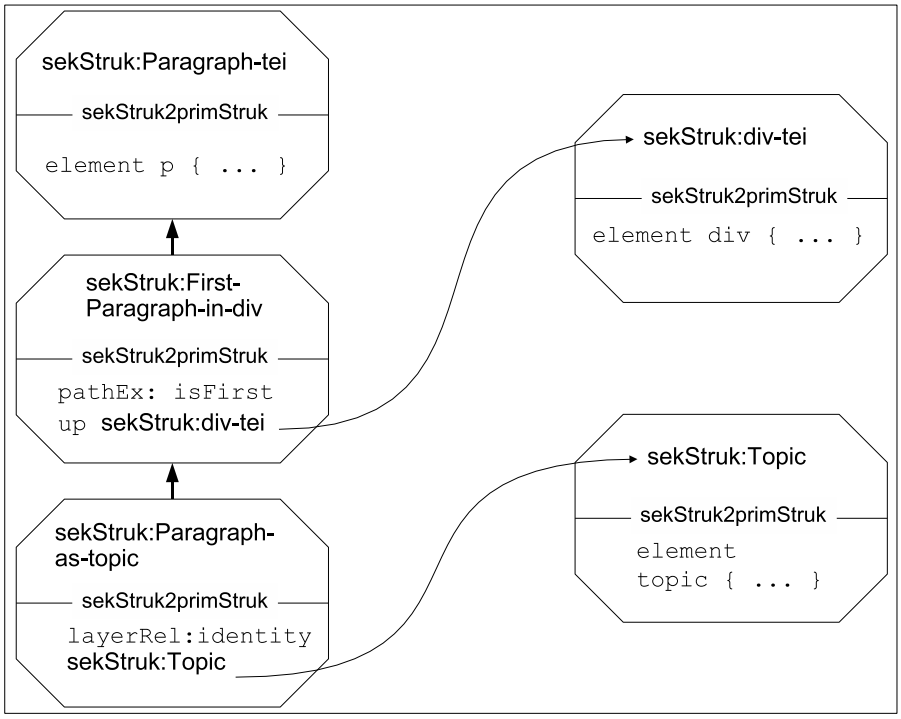

Abbildung 4.5.: Verbindung dokumentgrammatikbezogener und dokumentinstanzbezogener sekundärer Informationsstrukturierung mit mehrfach ausgezeichneten Dokumentinstanzen

lektionen mittels dokumentgrammatikbezogener Informationsstrukturierung und Selektion mittels Pfadausdrücken anwenden zu können, wird er in Aussagen mit dem Prädikat *sekStruk2primStruk* integriert. Ein Beispiel ist in Abbildung 4.5 visualisiert.

Das Beispiel zeigt, wie sekundäre Informationsstrukturierung für die deklarative Beschreibung von Beziehungen zwischen thematischen und dokumentstrukturbezogenen Auszeichnungen verwendet werden kann. Eine empirische Analyse derartiger Beziehungen, unter Anwendung der Methodologie multipler Auszeichnungen, wird z.B. in Bayerl et al. (2003) vorgenommen. In dem Beispiel werden alle drei Formen der Selektion von informationellen Ressourcen aus der primären Informationsstrukturierung kombiniert, die in der sekundären Informationsstrukturierung zur Verfügung stehen: Selektionen dokumentgrammatischer Konstrukte, von Informationseinheiten aus einzelnen Dokumentinstanzen durch Pfadausdrücke und von Informationseinheiten in mehreren Dokumentinstanzen.

Wie bei der Vorstellung der Caterpillar-Ausdrücke wird in dem Beispiel die inferentielle Kraft der Konzepthierarchie und interkonzeptueller Beziehungen genutzt. Im Konzept sekStruk:Paragraph-tei werden die Deklaration des <p> Elements in der Dokumentgrammatik und alle <p> Elementknoten in Dokumentinstanzen selektiert. In Konzept sekStruk:First-Paragraph-in-div werden diejenigen <p> Elemente selektiert, welche das erste Kindelement unmittelbar unter einem Element sind. Diese Untermenge der <p> Elemente wird schließlich durch die Selektion im Konzept sekStruk:Paragraph-as-Topic weiter eingeschränkt durch eine Aussage mit dem Prädikat *sekStruk:identity*[7] Hier werden nur diejenigen <p> Elemente selektiert, welche in einer Identitätsbeziehung zu Auszeichnungen mit dem <topic> Element stehen. Auf das <topic> Element wird nur mittelbar, d.h. über das Konzept seStruk:Topic referiert. Das Element kann in der gleichen oder einer anderen, primärdatenidentischen Dokumentinstanz vorkommen.

Das Verfahren der multiplen Auszeichnung primärdatenidentischer textueller Daten an sich ist als Ergänzung zur primären Informationsstrukturierung zu verstehen. Die Möglichkeiten der Auszeichnungssprache XML zur Segmentierung und Hierarchisierung textueller Daten werden nicht aufgegeben, sondern erweitert. Ein anderer Weg besteht in der Definition neuer Auszeichnungssprachen, die auf verschiedene Weise die Problematik sich überlappender, nicht hierarchisch repräsentierbarer Auszeichnungen angehen. Beispiele sind z.B. das von Sperberg-McQueen und Huitfeldt (1998) beschriebene Format **MECS**, das Format **TexMECS**, vgl. Huitfeldt und Sperberg-McQueen (2001), sowie die von Tennison und Piez (2002) beschriebene Auszeichnungssprache **LMNL**. Diese Auszeichnungssprachen haben jedoch zumindest gegenwärtig den Nachteil, dass ihre Implementationen zumeist[8] nicht weit fortgeschritten sind, und sie nur in geringem Maße auf andere Standards Bezug nehmen. Die für die vorliegende Arbeit zentralen Standards wie Namensräume und URI sind z.B. nicht ihm Datenmodells von LMNL integriert. Dies erschwert die vertikale Verbindung informationeller Ressourcen. Es bleibt abzuwarten, ob eine der Auszeichnungssprachen diesen Zustand überwindet.

[7] Das Präfix `layerRel` ist motiviert durch den hauptsächlichen Anwendungsbereich der Methodologie, namentlich die Beschreibung von Beziehungen linguistischer **Annotationsebenen** (annotation layers), vgl. auch Abschnitt 5.1.

[8] Czmiel (2004) stellt eine Implementation vor, die Teile des LMNL-Datenmodells in XML wiedergibt.

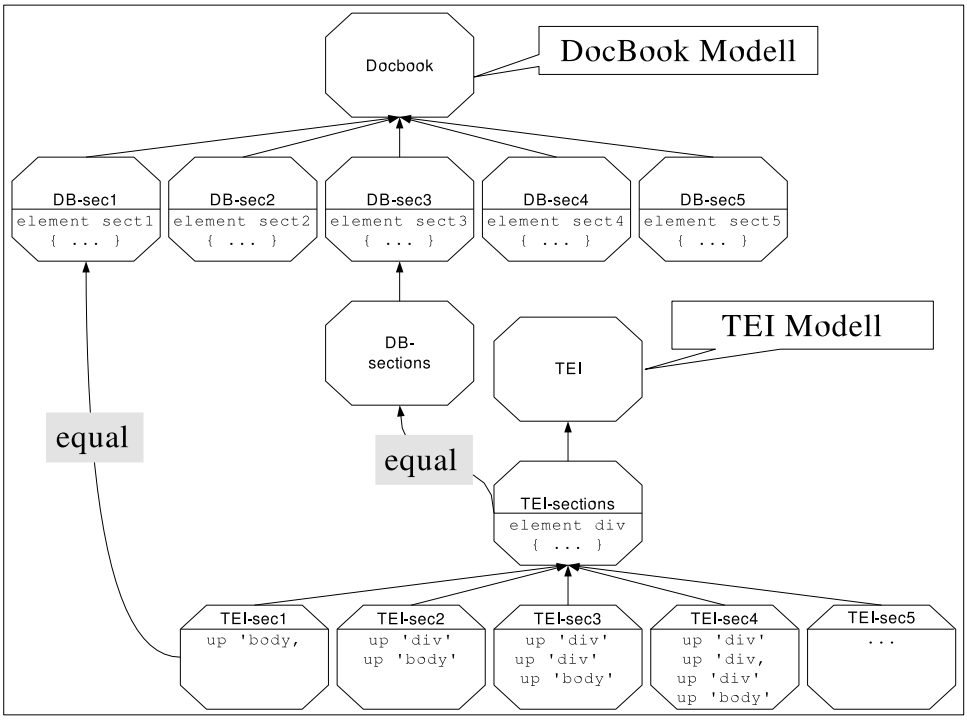

Abbildung 4.6.: Relationierung von Dokumentgrammatiken und Dokumentinstanzen mittels sekundärer Informationsstrukturierung

4.4.1.4. Interrelationierung von informationellen Ressourcen der primären Informationsstrukturierung

Aussagen zur Interrelationierung von informationellen Ressourcen der primären Informationsstrukturierung werden mit dem Prädikat *equal* gemacht. Eine exemplarische Anwendung des Prädikats, bei der Teile der TEI Dokumentgrammatik und der DocBook Dokumentgrammatik bzw. entsprechende Dokumentinstanzen miteinander in Beziehung gesetzt werden, ist in Abbildung 4.6 visualisiert.

Die Aussagen zu den Dokumentgrammatiken werden in separaten **Modellen** gemacht. Ein Modell wird durch ein Konzept definiert, welches durch das Prädikat *subConceptOf* unmittelbar dem vordefinierten Konzept sekStruk:models untergeordnet ist. Eine entsprechende Aussage lautet (sekStruk:Docbook *subConceptOf* sekStruk:models.). Auf diese Weise wird die Beziehung *partOf* zwischen Modell und Konzepten explizit: Alle weiteren Konzepte, welche durch das Prädikat *subConceptOf* dem Konzept sekStruk:Docbook

mittelbar oder unmittelbar untergeordnet sind, gehören zu dem Modell für die Docbook Dokumentgrammatik. Für die Aussagen zur TEI Dokumentgrammatik gilt entsprechendes.

Die Modelle in der Abbildung fokussieren Elemente für die Auszeichnung von Abschnitten. In der Docbook Dokumentgrammatik kann die Verschachtelungstiefe von Elementen zur Auszeichnung von Abschnitten explizit durch den Elementnamen angegeben werden, z.B. `<sect1>` oder `<sect2>`. In dem Modell sekStruk:Docbook ist für jede Verschachtelungstiefe ein entsprechendes Konzept deklariert, z. B. durch die Aussage (DB-sec1 *sekStruk2primStruk* `"element sect1 { ... }"`.). Die TEI Dokumentgrammatik erlaubt es ebenfalls, durch Elementnamen die Verschachtelungstiefe explizit zu machen. Alternativ kann der Benutzer das `<div>` Element verwenden, welches Rekursion beliebiger Tiefe erlaubt. Um `<div>` Elemente der TEI nun zu den angesprochenen Elementen von Docbook in Beziehung zu setzen, wird in Abbildung 4.6 in den TEI-spezifischen Konzepten bzw. ihren Selektionen die Verschachtelungstiefe durch Caterpillar-Ausdrücke bestimmt. Die Aussage (TEI-sec1 *sekStruk2primStruk* `"up 'body'"`.) selektiert z.B. nur diejenigen `<div>` Elemente, die unmittelbare Kindknoten eines `<body>` Elements sind.

Die Aussagen mit dem Prädikat *equal* machen die Beziehungen zwischen den Elementen der beiden Dokumentgrammatiken explizit. Ein wichtiger Aspekt bei dieser Relationierung ist die Vererbung von Informationen über die Konzepthierarchie in der sekundären Informationsstrukturierung. Das Konzept DB-sections erbt die Selektionen von Elementdeklarationen der übergeordneten Konzepte, z.B. DB-sec1 oder DB-sec2. Die Aussage (TEI-sections *equal* DB-sections.) bezieht sich also auf alle Elementdeklarationen für Abschnitte im Docbook-Modell. Derartige Aussagen können auch als Grundlage für Transformationen von Dokumentinstanzen dienen. Auf diese Weise wird die Anforderung 2, die einleitend zu Abschnitt 4.4 auf Seite 97 formuliert wurde, erfüllt. Für reversible Transformationen muss allerdings generell Bedingungen Genüge getan werden, welche in Abschnitt 4.4.2.4 erläutert werden.

4.4.1.5. Selektion von Konzepten und interkonzeptuellen Beziehungen aus der konzeptuellen Ebene

Die sekundäre Informationsstrukturierung selektiert informationelle Ressourcen aus der konzeptuellen Ebene durch Aussagen mit dem Prädikat *sekStruk2conLevel*. Nichtselektierte informationelle Ressourcen aus der konzeptuellen Ebene sind aus Sicht der

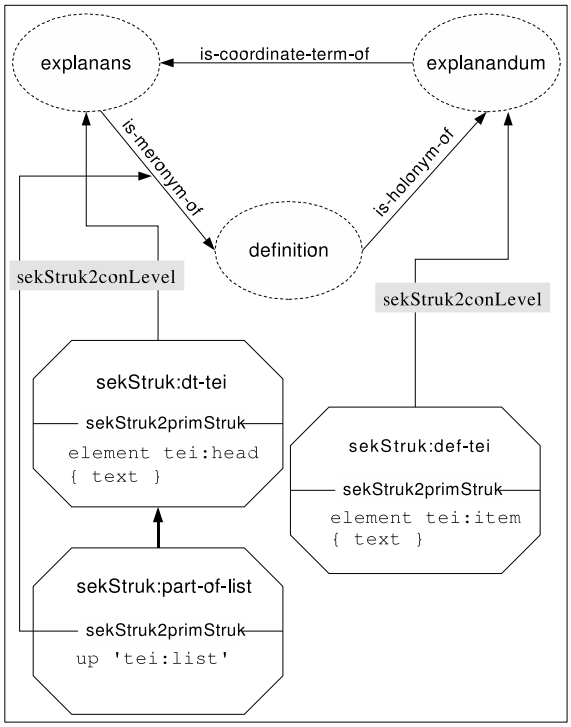

Abbildung 4.7.: Selektion von Konzepten und interkonzeptuellen Beziehungen aus WordNet mittels des Prädikats *sekStruk2conLevel*

sekundären Informationsstrukturierung nicht vorhanden. Dies entspricht dem Vorgehen bei der Selektion von informationellen Ressourcen aus der primären Informationsstrukturierung. Die sekundäre Informationsstrukturierung stellt eine vertikale Beziehung nicht zwischen allen vorhandenen informationellen Ressourcen her, sondern nur zwischen Teilmengen.

Terminologisch wird bei vielen Begriffen der sekundären Informationsstrukturierung und der konzeptuellen Ebene keine Unterscheidung vorgenommen. In beiden Bereichen spielen Konzepte, Eigenschaften bzw. interkonzeptuelle Beziehungen sowie die Konzepthierarchie eine Rolle. Dies wirft die Frage auf, was genau aus der konzeptuellen Ebene selektiert werden kann. Sie soll an einem Beispiel beantwortet werden, welches in Abbildung 4.7 visualisiert ist. Die Abbildung zeigt in der konzeptuellen Ebene einen Ausschnitt aus der bereits angesprochenen lexikalischen Datenbank WordNet. Hier sind die drei Konzepte explanans, explanadum und definition enthalten. Sie sind durch interkon-

zeptuelle Beziehungen wie *is-meronym-of* oder *is-holonym-of* miteinander verbunden. Die Aussage (sekStruk:dt-tei *sekStruk2conLevel* explanans.) selektiert das Konzept explanans für das Konzept sekStruk:dt-tei. Die Aussage (sekStruk:def-tei *sekStruk2conLevel* explanandum.) selektiert das Konzept explanandum für das Konzept sekStruk:def-tei. Neben der Selektion von Konzepten aus der konzeptuellen Ebene ist es nun möglich, Eigenschaften von Konzepten zu selektieren. Die Aussage (sekStruk:part-of-list *sekStruk2conLevel* *is-meronym-of* .) selektiert die Meronymiebeziehung, welche zwischen den Konzepten explanans und definition aus WordNet besteht. Die Verbindung eines Konzepts aus der sekundären Informationsstrukturierung mit einer Relation aus der konzeptuellen Ebene ist begründet in den Operationalisierungsmöglichkeiten, die sich daraus ergeben, vgl. Abschnitt 4.4.2.3. Eine Suchanfrage kann ausdrucksseitige Beschreibungen aus der konzeptuellen Ebene auf ausdrucksseitige Beschreibungen in der primären Informationsstrukturierung beziehen, z.B. das Konzept explanans auf das Element `<tei:head>`. D.h. eine Suchanfrage kann auch ausgewählte Konzepteigenschaften aus der konzeptuellen Ebene einbeziehen. Im vorliegenden Fall ist dies die besagte Meronymiebeziehung. In der primären Informationsstrukturierung wird sie durch die Verschachtelung des `<tei:head>` Elements in ein `<tei:list>` Element ausgedrückt. Um die Verbindung zur konzeptuellen Ebene so weit wie möglich offen zu halten, selektiert das Prädikat *sekStruk2conLevel* eine nicht leere Menge von URI. Ob sie sich auf Eigenschaften oder Konzepte in der konzeptuellen Ebene beziehen, bleibt unbestimmt.

Die Verbindung zwischen sekundärer Informationsstrukturierung und konzeptueller Ebene verspricht in zweierlei Hinsicht Gewinn. Erstens wird die sekundäre Informationsstrukturierung mit reichhaltigen Bedeutungsbeschreibungen aus der konzeptuellen Ebene verknüpft. Für die vorliegende informationelle Ressource WordNet gibt es z.B. Varianten in verschiedenen Sprachen. Definitionen von Konzepten in den sprachspezifischen Varianten sind miteinander verknüpft. Dies eröffnet den Weg zu einer semi-automatischen, multilingualen Dokumentation von Dokumentgrammatiken, die in Sasaki (2004) exemplifiziert wurde. Die Bedeutung dokumentgrammatischer Konstrukte muss einmalig in der sekundären Informationsstrukturierung spezifiziert und mit dem englischsprachigen WordNet vebunden werden. Die multilinguale Dokumentation wird erreicht durch existierende Verbindungen von WordNet zu anderen, sprachspezifischen informationellen Ressourcen, wie z.B. EuroWordNet[9].

[9]Vgl. http://www.illc.uva.nl/EuroWordNet/

Zweitens ist die Verbindung aus Sicht der konzeptuellen Ebene sinnvoll. Zusätzliche Informationen über Beziehungen zwischen Konzepten in der konzeptuellen Ebene können eingebracht werden, welche sich nur aus der sekundären Informationsstrukturierung ergeben. So kann das Konzept definitionList-tei aus der sekundären Informationsstrukturierung sowohl auf das Konzept definition als auch auf das Konzept list abgebildet werden. Die Beziehung zwischen definition und list, welche sich hieraus ergibt, ist wiederum spezifisch für die sekundäre Informationsstrukturierung.

Die Art der Verbindung zwischen sekundärer Informationsstrukturierung und konzeptueller Ebene ist in hohem Grade abhängig von den Eigenschaften, die letztere mit sich bringt. So beinhaltet WordNet z.B. eine hierarchische Anordnung von Konzepten, organisiert in so genannten **Synsets**. Das Konzept definition[10] ist Teil einer solchen Konzepthierarchie. Speziellere Terme sind z.B. contextual definition oder dictionary definition, generellere Terme sind explanation oder statement. Diese Konzepthierarchie lässt sich gewinnbringend einsetzen, wenn die vorhandenen informationellen Ressourcen der primären Informationsstrukturierung auf sie anwendbar sind. Ansonsten sind sie aus Sicht der sekundären Informationsstrukturierung nicht vorhanden. Die vorliegende Arbeit strebt deshalb keine generische, automatische Abbildung zwischen sekundärer Informationsstrukturierung und konzeptueller Ebene an.

4.4.1.6. Zusammenfassung der vordefinierten Konstrukte und eine Beispielanwendung

Die folgende Liste fasst die vordefinierten Bestandteile der sekundären Informationsstrukturierung zusammen.

- **sekStruk**: Namensraumpräfix für Konzepte in der sekundären Informationsstrukturierung; Namensraum `http://example.com/sekStruk`. Zugleich: Menge aller Einheiten der sekundären Informationsstrukturierung.
- **subConceptOf**: Prädikat für die Bildung der Konzepthierarchie in der sekundären Informationsstrukturierung und die Beziehung *partOf* von Konzepten zu einem Modell in der sekundären Informationsstrukturierung.

[10] In WordNet werden Synsets mit eindeutigen Identifikatoren versehen. Die Synsets beinhalten verschiedene Lexeme, die das Konzept ausdrücken können. Die hier vorgenommene Identifikation des Konzepts definition durch das Lexem „definition“ ist also eine verkürzte, der besseren Verständlichkeit halber gewählte Darstellung.

- **primStruk**: Menge aller informationellen Ressourcen der primären Informationsstrukturierung.
- **sekStruk2primStruk**: Prädikat zur Selektion von informationellen Ressourcen aus der primären Informationsstrukturierung.
- **componentOf**: Prädikat zur Beschreibung von Abhängigkeitsbeziehungen zwischen Konzepten in der sekundären Informationsstrukturierung, die dokumentgrammatische Konstrukte selektieren.
- **pathEx**: Präfix für Pfadausdrücke, die Informationseinheiten in einzelnen Dokumentinstanzen selektieren.
- **layerRel:** Präfix für Prädikate, die der Selektion von Informationseinheiten in mehrfach ausgezeichneten Dokumentinstanzen dienen.
- **grammarBasedSekStruk**: Dokumentgrammatikbezogene sekundäre Informationsstrukturierung; ist gegeben, wenn dokumentgrammatische Konstrukte in einer Aussage mit dem Prädikat *sekStruk2primStruk* selektiert werden. Nutzt die inferentielle Kraft der Konzepthierarchie.
- **instanceBasedSekStruk**: Dokumentinstanzbezogene sekundäre Informationsstrukturierung. Ist gegeben, wenn das Prädikat *sekStruk2primStruk* Informationseinheiten in einzelnen Dokumentinstanzen – durch Pfadausdrücke – selektiert oder – durch die Beschreibung von Beziehungen zwischen Auszeichnungsebenen – in mehrfach ausgezeichneten Dokumentinstanzen. Nutzt die inferentielle Kraft der Konzepthierarchie und für Knotentests von interkonzeptuellen Beziehungen.
- **equal**: Prädikat zur Beschreibung von Beziehungen zwischen Konzepten in der sekundären Strukturierung, die Bestandteil verschiedener Modelle sind.
- **sekStruk2conLevel**: Prädikat zur Selektion von informationellen Ressourcen aus der konzeptuellen Ebene.
- **abstract="true"**: Prädikat *abstract* mit Wert `true`. Ein Argument von Aussagen mit *abstract* ist ein Konzept der sekundären Informationsstrukturierung. Das andere Argument ist der literale Wert `true`. Das Prädikat wird benötigt für die Operation der konzeptbezogenen Validierung, vgl. Abschnitt 4.4.2.3.

In Abschnitt 4.4 wurden drei Anforderungen formuliert, welche die sekundäre Informationsstrukturierung für die exemplarischen informationellen Ressourcen der primären Informationsstrukturierung erfüllen soll. Sie sind hier noch einmal verkürzt wiedergegeben.

Anforderung 1 : Die identische Bedeutung der definitorischen Listen soll formal beschrieben werden.

Anforderung 2 : Eine Transformationsprozedur zwischen den Auszeichnungen nach der TEI und denen nach XHTML soll definiert werden.

Anforderung 3 : Definitorische Listen müssen aus drei bis vier Linien bestehen, wobei die erste Linie für den zu definierenden Ausdruck, die weiteren Zeilen für die Definition reserviert sind.

Wie die Anforderung 1 zu erfüllen ist, zeigten bereits die Beispiele 4.4 und 4.5 in Abschnitt 4.4.1.1. Anforderung 2 wurde in Abschnitt 4.4.1.4 diskutiert. Anforderung 3 wird hier nun durch ein Modell sekStruk:layout erfüllt, welches in Beispiel 4.8 wiedergegeben ist.

Das Konzept sekStruk:line-general selektiert die Deklaration des `<layout:line>` Elements. Einige der `<layout:line>` Elemente, die vom subordinierten Konzept sekStruk:line-dt selektiert werden, stehen in der Identitätsbeziehung zu `<tei:item>` Elementen. Diese wiederum werden vom Konzept sekStruk:dt-tei selektiert; die entsprechende Aussage ist hier aus Gründen der Übersichtlichkeit nicht mit aufgeführt. Die angesprochene Identitätsbeziehung zwischen den Elementen wird nun durch die Aussage (sekStruk:line-dt *sekStruk2primStruk* `"layerRel:identity sekStruk:dt-tei"`.) ausgedrückt. Für die Beziehung zwischen sekStruk:line-def und dem Konzept sekStruk:def-tei gilt entsprechendes. Das Konzept sekStruk:paragraph-definitionList dient dazu, die Anforderung 3 zu erfüllen. Durch einen Caterpillar-Ausdruck wird die Abfolge innerhalb von `<paragraph>` Elementen spezifiziert.

Diese Beschreibung der dokumentinstanzbezogenen Bedingungen bzw. Selektionen kann für verschiedene Dokumentgramamtiken wiederverwendet werden, wenn die übergeordneten Konzepte andere dokumentgrammatische Konstrukte selektieren. Die Beziehung, die zwischen der Auszeichnung von Layout und der TEI-spezifischen Auszeichnung beschrieben wurde, lässt sich z.B. ohne weiteres auf die in diesem Kapitel vorgestellen HTML-spezifischen Auszeichnungen anpassen.

(4.8)
```
sekStruk:layout subConceptOf sekStruk:models.
sekStruk:line-general subConceptOf sekStruk:layout.
sekStruk:line-dt subConceptOf sekStruk:line-general.
sekStruk:line-def subConceptOf sekStruk:line-general.
sekStruk:paragraph-general subConceptOf sekStruk:layout.
sekStruk:paragraph-definitionList subConceptOf sekStruk:paragraph.
sekStruk:line-general sekStruk2primStruk
 "element layout:line { ... }".
sekStruk:line-dt sekStruk2primStruk
 "layerRel:identity sekStruk:dt-tei".
sekStruk:line-def sekStruk2primStruk
 "sekStruk:line-general identity sekStruk:def-tei".
sekStruk:paragraph-general sekStruk2primStruk
 "element layout:paragraph"
sekStruk:paragraph-definitionList sekStruk2primStruk
 "pathEx: first sekStruk:line-dt, right sekStruk:line-def,
  right sekStruk:line-def, right sekStruk:line-def,
  (right sekStruk:line-def)*".
```

Beispiel 4.8: Sekundäre Informationsstrukturierung zur Beschreibung von Beziehungen zwischen mehreren Auszeichnungsebenen

4.4.2. Operationalisierung der Aussagen

4.4.2.1. Formale Beschreibung der sekundären Informationsstrukturierung: Eine terminologische Ontologie

Im Folgenden wird die Operationalisierung der im letzten Abschnitt vorgestelten Aussagen diskutiert. Ausgangspunkt der Operationalisierung ist die Darstellung der informationellen Ressourcen als Mengen. In Abschnitt 2.2.1.2 wurde demonstriert, wie z.B. der Informationsgehalts von Dokumentinstanzen unter Rückgriff auf das XML Information Set als eine Menge von Aussagen repräsentiert werden kann. In Abschnitt 2.2.3.2 wurde die konzeptuelle Ebene ebenfalls als eine Menge von Aussagen beschrieben. Die Rolle der Operationalisierung der sekundären Informationsstrukturierung besteht nun darin

zu bestimmen, wann diese Mengen aufeinander beziehbar sind und wann nicht.

Die Grundlage für die Beantwortung dieser Frage bildet die Arbeit von Fischer (1998). Unter Bezug auf Sowa (1996) definiert er Eigenschaften einer **terminologischen Ontologie** (terminological ontology). Auch die sekundäre Informationsstrukturierung lässt sich als terminologische Ontologie auffassen. Grundbestandteil sind Konzepte und Eigenschaften, vgl. Fischer (1998, Abschnitt 2.2):

> A noun concept is a unary predicate c and its extension ext(c) is the set of objects x for which c(x) is true. [...] ext(c,t) [is] the set of objects for which c(x) is true at time / in context t. A concept c is narrower at time / in context t than concept b if ext(c,t) is a subset of ext(b,t).

Ein Konzept wie sekStruk:Paragraph in Abbildung 4.1 ist ein unäres Prädikat, definiert durch seinen Namen. Die Extension des Konzepts `ext(c)` ist die Menge aller dokumentgrammatischen Konstrukte und Informationseinheiten, die durch das Prädikat *sekStruk2primStruk* selektiert sind, und die Menge aller URI, die durch das Prädkat *sekStruk2conLevel* als informationelle Ressourcen der konzeptuellen Ebene selektiert werden. Im Falle des Konzepts sekStruk:Paragraph, wie es in den letzten Abschnitten beschrieben wurde, sind dies `<p>` Elemente mit dem Attribut `type`, das den Wert `"gloss"` hat, sowie ein URI für das Konzept wordnet:Paragraph. Der Kontextparameter `t` wird in der sekundären Informationsstrukturierung durch die Beschreibung separater Modelle wiedergegeben, vgl. Abschnitt 4.4.1.4. Das `<p>` Element kann z.B. für ein Konzept sekStruk:Paragraph selektiert sein, welches Teil eines Modells für die Dokumentstruktur ist. Es kann aber auch für ein Konzept sekStruk:ThematicUnit selektiert sein, welches Teil eines Modells für den thematischen Aufbau von Dokumenten ist.

Für die Operationalisierung der sekundären Informationsstrukturierung ist die Trennung in **intensionale versus extensionale Interpretationen der terminologischen Ontologie** entscheidend. In der bisher verwendeten Benennung entspricht die Trennung von Intension und Extension der Unterscheidung von inhaltsseitiger und ausdrucksseitiger Beschreibung. Dabei spielt die Beziehung der **Subsumption**, d.h. der Unterordnung von Konzepten, eine wichtige Rolle. Intensional werden Konzepte anhand ihrer Eigenschaften beschrieben. Sie sind anderen Konzepten untergeordnet, wenn sie deren Eigenschaften und zumindest eine weitere umfassen. In der sekundären Informationsstrukturierung werden Eigenschaften, d.h. intensionale Beschreibungen durch Prädikate wie *sekStruk2primStruk*, *subConceptOf*, *componentOf* charakterisiert. Exten-

(1)	if <aaa,x> is in sekStruk then I(aaa) = x
(2)	if <aaa,x> is in sekStruk then ICEXT(x) is the sekStruk of x and is a subset of sekStruk
(3)	if <aaa,x> is in sekStruk then for any primStruk "ppp" ^^sss in V with I(sss) = x, if ppp is in primStruk of x then I("ppp" ^^sss) = sekStruk2primStruk(x)(sss), otherwise I("ppp" ^^sss) is not in sekStruk
(4)	if <aaa,x> is in sekStruk then I(aaa) is in ICEXT(I(sekStruk))

Tabelle 4.1.: Konditionen für die Komplementarität der extensionalen und intensionalen Interpretation der sekundären Informationsstrukturierung

sional gesehen ist ein Konzept einem anderen Konzept untergeordnet, wenn die Menge von Objekten kleiner ist, welche der Extension des Konzeptes zugehörig sind. Die Extensionen sind im vorliegenden Fall informationelle Ressourcen der primären Informationsstrukturierung oder der konzeptuellen Ebene, wobei in der vorliegenden Arbeit für letztere keine Prüfung der Subsumptionsbeziehung vorgesehen ist. Mit anderen Worten, die intensionale und extensionale Beschreibung von Subsumption sind zueinander komplementär, vgl. Fischer (1998, Abschnitt 2.3):

> [...] the "meaning" of an intensional subsumption can be found in the complementary extensional subsumption[...].

Die Frage lautet nun, wie sich die Beziehung zwischen intensionaler und extensionaler Beschreibung, zwischen Aussagen in der sekundären Informationsstrukturierung und informationellen Ressourcen in der primären Informationsstrukturierung bzw. der konzeptuellen Ebene, operationalisieren lässt. Zur Beantwortung dieser Frage dienen vier Konditionen, welche die extensionale Interpretation der sekundären Informationsstrukturierung konstituieren, vgl. Tabelle 4.1. Die vier Konditionen basieren auf den Beschreibungen von Datentypen in RDF Schema, vgl. Hayes und McBride (2004, Abschnitt 5.1). Wie in Abschnitt 3.2 bereits thematisiert wurde, hat es bei der Entwicklung von RDF Schema Überlegungen gegeben, die Charakterisierungen von Datentypen in RDF

Schema auch auf Inhaltsmodelle von Elementen in Dokumentgrammatiken anzuwenden. Dies würde eine komplexe Integration informationeller Ressourcen bedeuten, deren Auswirkungen auf die Berechenbarkeit von Inferenzen in RDF Schema unabschätzbar sind. Deshalb wurde von dem Vorhaben Abstand genommen. Die vorliegende Arbeit folgt diesem Vorgehen: Die vier Konditionen, welche die extensionale Interpretation sekundärer Informationsstrukturierung konstituieren, vermittelten zwischen informationellen Ressourcen, anstatt sie ineinander zu integrieren.

Kondition (1) sichert, dass die Interpretation `I` eines URI `aaa`, welche auf ein Konzept `x` in der sekundären Informationsstrukturierung verweist, Bestandteil der sekundären Informationsstrukturierung ist. Es mag verschiedene URI geben, die auf dieses Konzept verweisen. Bedeutsam ist nur die eindeutige Identifizierbarkeit des Konzepts.

Kondition (2) sichert, dass die Extension eines Konzepts in der sekundären Informationsstrukturierung, `ICEXT(x)`, Teil der Menge aller Extensionen von Konzepten in der sekundären Informationsstrukturierung ist.

Kondition (3) ist zentral für die Verbindung von primärer und sekundärer Informationsstrukturierung[11]. Für URI der primären Informationsstrukturierung, z.B. `ppp`, existiert eine Abbildung auf einen URI der sekundären Informationsstrukturierung, z.B. `sss`. Die Auswertung der Abbildungsfunktion `sekStruk2primStruk(x)(sss)` ergibt die Interpretation, dass der URI der primären Informationsstrukturierung `ppp` als Extension der URI der sekundären Informationsstrukturierung aufzufassen ist, also `I("ppp" ^^sss)`. Wenn für einen URI der primären Informationsstrukturierung keine derartige Abbildungsfunktion existiert, ist der URI aus der Sicht der sekundären Informationsstrukturierung nicht existent. Die Operationalisierung dieser Abbildungsfunktion entspricht der konzeptbezogenen Suchoperation, welche in Abschnitt 4.4.2.3 vorgestellt wird. Kondition (3) ist deshalb von großer Bedeutung, weil sie den Skopus der Verbindung informationeller Ressourcen beschreibt. Einheiten der primären Informationsstrukturierung, für die keine Abbildungsfunktion definiert ist, sind aus Sicht der sekundären Informationsstrukturierung nicht existent. Die Abbildungsfunktion erlaubt es, die informationellen Ressourcen klar zu separieren und variabel aufeinander zu beziehen. So können verschiedene Einheiten der primären Informationsstrukturierung als Extensionen aufgefasst werden.

Die letzte Kondition (4) schließlich gewährleistet, dass die Interpretation einer URI

[11] Für die Verbindung zur konzeptuellen Ebene lässt sich eine ähnliche Kondition beschreiben. Sie wird hier jedoch nicht aufgeführt, da sie nicht operationalisiert wird.

`aaa` als Konzept `x` in der sekundären Informationsstrukturierung in der Menge aller Konzepte in der sekundären Informationsstrukturierung enthalten ist.

Durch die Trennung in die intensionale und extensionale Interpretation der sekundären Informationsstrukturierung, bzw. inhaltsseitige versus ausdrucksseitige Beschreibung, sowie anhand der vier Konditionen, lässt sich eine ontologische Frage im Bereich der Bedeutungsbeschreibung für Auszeichnungen beantworten, welche von Sperberg-McQueen et al. (2002) gestellt wurde. Die Frage lautet, wie die **Polysemie** von Auszeichnungen in einer Bedeutungsbeschreibung erfasst werden kann. Ein `<p>` Element wird z.B. oft zur Auszeichnung von Paragraphen benutzt, es kann aber auch und eventuell gleichzeitig der Segmentierung thematischer Einheiten dienen, vgl. Abschnitt 4.4.1.3. Durch die Trennung von intensionaler und extensionaler Beschreibung wird eine derartige Polysemie greifbar. Ein Konzept sekStruk:ThematicUnit kann die Deklaration des `<p>` Elements in einem Modell für thematische Strukturen selektieren. Die Operationalisierung der Selektion in einer Suchanfrage geschieht durch die Auswertung der Abbildungsfunktion `sekStruk2primStruk(p)(sekStruk:ThematicUnit)`. Ein Konzept sekStruk:Paragraph kann das gleiche Element in einem Modell für die Dokumentstruktur selektieren. In diesem Fall entspricht die Suchanfrage der Auswertung der Abbildungsfunktion `sekStruk2primStruk(p)(sekStruk:Paragraph)`. Die Kondition (3) der extensionalen Interpretation der sekundären Informationsstrukturierung sichert, dass die Interpretation des `<p>` Elements in beiden Fällen eindeutig ist. Im Gegensatz zu diesem **wissensbasierten Vorgehen** stehen **objektorientierte Ansätze** wie XML Schema, vgl. Abschnitt 2.2.4.2. Hier muss jede Informationseinheit eindeutig typisiert sein. Typ-Ambiguität oder die Zuweisung mehrerer Typen zu einem Knoten in der Dokumentinstanz ist nicht erlaubt.

4.4.2.2. Verifikation der Spezialisierung dokumentgrammtischer Konstrukte

Bei der dokumentinstanzbezogenen sekundären Informationsstrukturierung kann die extensionale Interpretation der Subsumptionsbeziehung anhand der Mengen von Informationseinheiten überprüft werden, welche Extensionen der betreffenden Konzepte sind. Bei der dokumentgrammatikbezogenen sekundären Informationsstrukturierung hingegen muss überprüft werden, ob die Subsumptionsbeziehung für Klassen von Informationseinheiten gilt. Die in Abschnitt 4.4.1.1 beschriebene Spezialisierbarkeit dokumentgrammatischer Konstrukte, z.B. von `attribute type { text }` nach `attribute type`

{"gloss"}, muss überprüft werden. Im vorliegenden Abschnitt wird ein Verfahren vorgestellt, welches dies leistet.

Eine Spezialisierung muss immer dann überprüft werden, wenn zwei Konzepte der sekundären Informationsstrukturierung Konstrukte aus Dokumentgrammatiken selektieren und zugleich ein Konzept durch das Prädikat *subConceptOf* dem anderen subordiniert ist. Eine Spezialisierung gab es in Beispiel 4.6. Das `type` Attribut besitzt in der Dokumentgrammatik den Datentyp `xsd:name`. In der Ableitung b' wurde der Wertebereich des Attributs eingeschränkt auf die Zeichenfolge `gloss`. Eine XML-Dokumentinstanz zu dieser spezialisierten Dokumentgrammatik ist konform zur generelleren Dokumentgrammatik.

Um die Konformität von Spezialisierungen zu gewährleisten, kommen in dieser Arbeit **Spezialisierungsregeln** zum Einsatz. Eine Spezialisierung muss einer dieser Regeln entsprechen, oder sie führt zu Widersprüchen zur generellen Dokumentgrammatik. Das Prinzip, welches hinter der Regelbildung steckt, wurde motiviert durch den DTD-Generator **FRED**, vgl. Shafer (1995). FRED erzeugt aus XML- oder SGML-Dokumentinstanzen eine DTD, mit Hilfe von Generalisierungs- und Reduktionsregeln. Eine Generalisierungsregel für Sequenzen gleichnamiger Elemente führt von *body* → *p p p p* zu *body* → *p+*. Eine Reduktionsregel für Sequenzen von Elementen mit verschiedenen Namen führt von *Z* → *(A B) C* zu *Z* → *A B C*. In der vorliegenden Arbeit werden diese Regeln diametral zu ihrer Motivation in FRED verwendet. Sie dienen der Überprüfung von Spezialisierungen. Die Spezialisierung muss sich durch wenigstens eine rechte Seite der folgenden Regeln beschreiben lassen.

1. Die Statusbezeichnung `*` kann spezialisiert werden zu `?`, `+`, einmaligen Vorkommen oder Wegfallen. Beispiele: *a** → *a+*; *a** → *a;*, *a** → *a,a*. Die Einführung von UND- oder ODER-Sequenzen ist ebenfalls möglich. Beispiele: *a** → `a|a`; *a** → `(a|a)a*`. Auch `interleave` kann eingesetzt werden, z.B. *a** → *a & a*.

2. Die Statusbezeichnung `?` kann spezialisiert werden zu obligatorischem Vorkommen.

3. Die Statusbezeichnung `+` erlaubt die Einführung von UND- oder ODER-Sequenzen. Beispiel: *a+* → `a|a+`. Im Gegensatz zur Statusbezeichnung `*` muss in der Sequenz eine Statutsbezeichnung `+` oder eine obligatorische Einheit vorkommen.

4. Mixed Content, der in einer ODER-Sequenz steht, kann zu einer UND-Sequenz spezialisiert werden.

5. Einführung von Konzepten der sekundären Informationsstrukturierung für einzelne dokumentgrammatische Konstrukte ohne eine Veränderung ihres Statusbezeichners ist in jedem Fall möglich. Dabei können die gleichen dokumentgrammatischen Konstrukte mehrfach, d.h. durch verschiedene Konzepte selektiert werden, wenn die Konzepte Bestandteil separater Modelle sind, vgl. Abschnitt 4.4.2.1.

6. ODER-Sequenzen ohne Statusbezeichner können zu einzelnen Einheiten spezialisiert werden. Beispiel: `a|b` → *a*.

Natürlich kann es zu Ambiguitäten kommen, d.h. mehrere Regeln lassen sich eventuell parallel anwenden. Für eine Typisierung von Inhaltsmodellen, wie z.B. bei der Definition komplexer Typen in XML Schema, sind die Regeln deshalb ungeeignet. Ihr Zweck liegt vielmehr in der Bestimmung von mindestens einer legalen Regel. Entscheidend ist, *ob* es eine linke Seite oder mehrere gibt. Die Anwendung der Regeln sei anhand einer generellen Dokumentgrammatik (vgl. Beispiel 4.9) und einer Dokumentgrammatik mit Spezialisierungen (vgl. Beispiel 4.10) demonstriert.

(4.9)
```
start =
element a {
  element b{ mixed {
   (element c { text }
   | element d { text }
   | element e { text })*
} }* }
```

Beispiel 4.9: Generelle Dokumentgrammatik

Der Ausdruck `a` in der Dokumentgrammatik mit Spezialisierungen enthält die generelle Dokumentgrammatik. `a'` ist nach drei Ableitungsschritten zu Stande gekommen. Zunächst selektiert das Konzept sekStruk:b1 das <b> Element. Zudem wird das <b> Element durch ein anderes Konzept sekStruk:b2 selektiert. Dies entspricht der Anwendung der Regel 5. Anschließend wird das Inhaltsmodell des <a> Elements spezialisiert, nach (sekStruk:b1 — sekStruk:b2)*. Auf Grund von Regel 1 ist diese Spezialisierung möglich. In der Ableitung `a''` wird dieses Inhaltsmodell weiter spezialisiert, was ebenfalls zur Regel 1 konform ist. In `a'''` wird die zu Regel 1 konforme Spezialisierung weitergeführt und auf das Inhaltsmodell der <b> Elemente angewendet.

```
(4.10) a    start = element a { ... }    a' start = element a {
    REGEL1                  (sekStruk:b1 |
                             sekStruk:b2)*}
    REGEL5    sekStruk:b1 = element b { ...}
    REGEL5    sekStruk:b2 = element b { ...}
    a'' start =  element a {
    REGEL1       ((sekStruk:b1) | (sekStruk:b1, sekStruk:b2))*}
    a''' start = element a {
        ((element b{ mixed {
          (element e { text } | element d { text }),
    REGEL1     (element d { text } | element c { text })? }}) |
    REGEL1     (element b{ mixed {
               (element e { text } | element d { text }),
               (element d { text } | element c { text })?}},
               element b{ mixed { element d { text }
               | element c { text }}}))*}
```

Beispiel 4.10: Regelkonforme Spezialisierung der generellen Dokumentgrammatik

Die vielen Schritte in der Ableitung von `a` zu `a'''` sind gerechtfertigt, weil so die Konformität der Regeln zur ursprünglichen Dokumentgrammatik überprüft werden kann. Deutlich wird auch die Funktion der Abfolge: Die Spezialisierung muss in der vorgestellten Abfolge geschehen, sonst lässt sich die Konformität zur generellen Dokumentgrammatik nicht durch die vorgestellten Regeln überprüfen.

Einen Sonderfall stellt die Spezialisierung von Datentypen dar. Für sie existieren Regeln, die sich aus der Hierarchie von Datentypen in der jeweils eingesetzten Bibliothek von Datentypen ergeben. Das Vorgehen ist in Abbildung 4.8 visualisiert. Von denjenigen Datentypen, denen selbst kein weiterer Datentyp untergeordnet ist, lassen sich von der Wurzel der Datentypenhierarchie ausgehende Pfade beschreiben. Eine Spezialisierung eines Datentyps ist anwendbar, wenn der spezialisierte Datentyp in einem der Pfade rechts vom ursprünglichen Datentyp steht. Dies trifft z.B. für eine Spezialisierung von `d` nach `g` zu, nicht aber von `e` nach `h`.

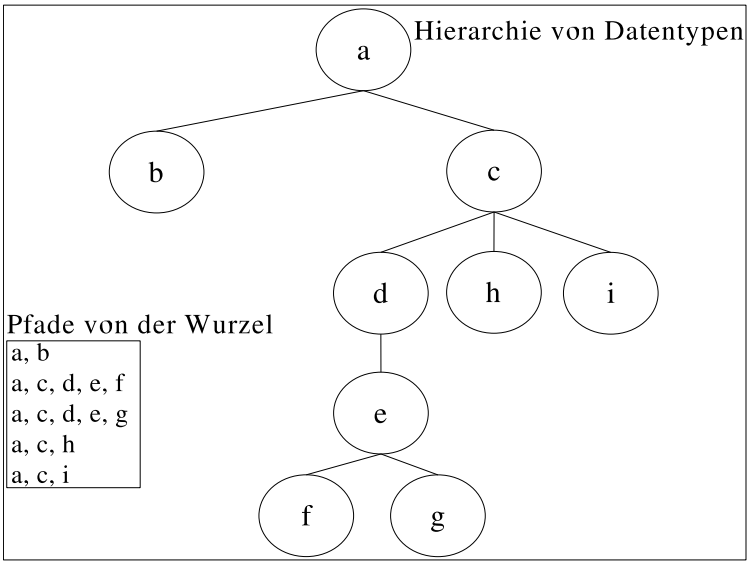

Abbildung 4.8.: Datentypenhierarchie und mögliche Pfade zur Wurzel

4.4.2.3. Konzeptbezogene Suche und Validierung von Informationen der primären Informationsstrukturierung

Die konzeptbezogene Suche hat als Eingabe den Namen eines Konzepts aus der sekundären Informationsstrukturierung, oder einen URI, welcher sich auf eine informationelle Ressource aus der konzeptuellen Ebene bezieht[12]. Der URI für die Extensionen des Konzepts sind die Ausgabe der Suchprozedur. Sie bedeutet die Auswertung der Aussagen mit dem Prädikat *sekStruk2primStruk*, d.h. eine Auswertung der Abbildungsfunktion `sekStruk2primStruk(x)(sss)`, vgl. Abschnitt 4.4.2.1. Das Konzept sekStruk:Paragraph aus Abbildung 4.1 selektiert z.B. die Deklaration des `<p>` Elements in der Dokumentgrammatik und die entsprechenden Elementknoten in Dokumentinstanzen. Ausgabe der konzeptbezogenen Suche sind URI-Referenzen auf diese informationellen Ressourcen.

Die konzeptbezogene Validierung gestaltet sich komplexer als die Suchprozedur. Sie beruht auf der Trennung in **abstrakte versus nicht abstrakte Konzepte**, welche in Abbildung 4.9 visualisiert sind.

[12]Der URI der konzeptuellen Ebene wird ausgewertet hinsichtlich des oder derjenigen Konzepte bzw. Eigenschaften der konzeptuellen Ebene, die der URI durch eine Aussage mit dem Prädikat *sekStruk2conLevel* selektiert.

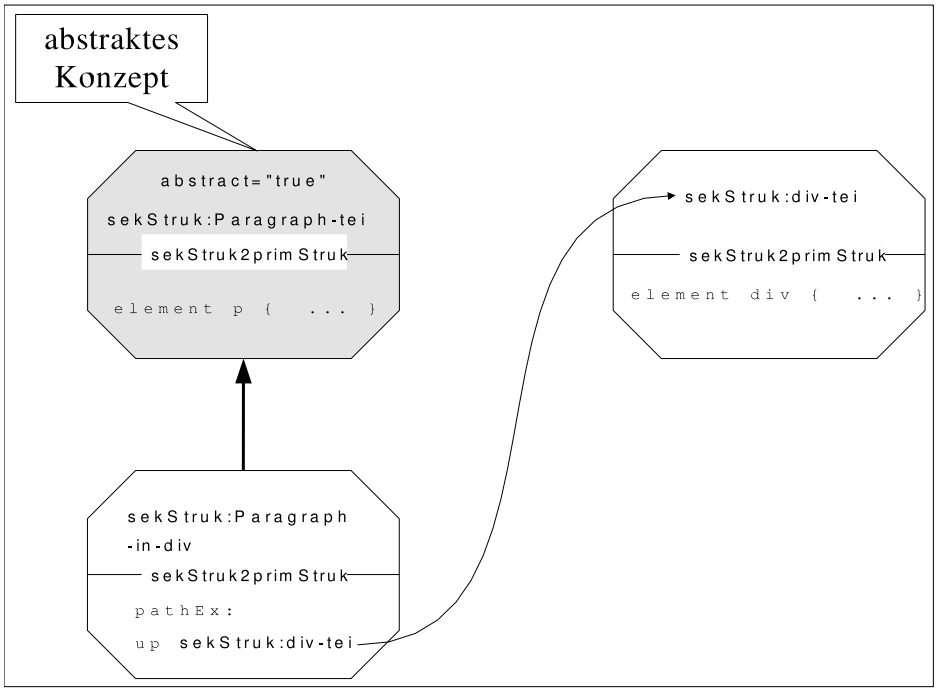

Abbildung 4.9.: Konzeptuelle Validierung von Dokumentgrammatiken und Dokumenteninstanzen durch die Beschreibung abstrakter Konzepte

Die Abbildung stellt eine abgeänderte Version von Abbildung 4.3 dar. Das Konzept sekStruk:Paragraph-tei ist als abstrakt deklariert. Dies wird durch die Aussage (sekStruk:Paragraph-tei *abstract* `true`.) wiedergegeben. Diese Aussage bewirkt, dass es auch für subordinierte Konzepte eine Extension geben muss, ansonsten schlägt die konzeptbezogene Validierung fehl. Die Dokumentgrammatik der TEI erlaubt es, z.B. `<p>` Elemente als Kindelement verschiedener Elemente in Dokumentinstanzen zu verwenden. Durch die Aussage wären nur diejenigen `<p>` Elemente valide, welche innerhalb von `<div>` Elementen vorkommen. Die Validität bezieht sich natürlich nicht auf die Dokumentgrammatik, die unverändert bleibt, sondern nur auf das Konzept sekStruk:Paragraph-tei.

Das Ergebnis der konzeptbezogenen Validierungsprozedur bildet eine Menge von URI, die auf nicht valide dokumentgrammatische Konstrukte bzw. Informationseinheiten verweisen. Diese stellen eine ausdruckseitige Beschreibung bzw. Extension eines abtrakten Konzeptes dar. Die Ausführung der Validierungsprozedur kann im Sinne von Sperberg-McQueen (2003) als **parsing as instructions for a proof** (Parsing als Anweisung

für eine Beweisführung) verstanden werden. Er beschreibt Parsing auf diese Weise im Rahmen einer Logik-basierten, in Prolog wiedergegebenen Repräsentation von Dokumentgrammatiken im Format von XML Schema. Im Gegensatz zu diesem Vorgehen bezieht sich die sekundäre Informationssstrukturierung nur potentiell auf eine gesamte Dokumentgrammatik. Ob die Dokumentgrammatik als Ganzes oder nur ein Teil davon Gegenstand der konzeptbezogenen Suche bzw. Validierung sind, ist abhängig von der Auswertung der Aussagen in der sekundären Informationsstrukturierung.

4.4.2.4. Relationierung von Dokumentgrammatiken und Transformation von Dokumentinstanzen

Aussagen zur Relationierung von Dokumentgrammatiken wurden in Abschnitt 4.4.1.4 vorgestellt. Die Transformation von Dokumentinstanzen soll möglichst reversibel spezifiziert werden. D.h. in der sekundären Informationsstrukturierung muss die Bedeutung von Einheiten der primären Informationsstrukturierung hinreichend spezifiziert sein.

Wenn in einer Dokumentgrammatik jedoch verschiedene Konstrukte die gleiche Bedeutung besitzen, stößt die Reversibilität an Grenzen. So ist es denkbar, dass in einer TEI-basierten Dokumentinstanz definitorische Listen zum einen als `<tei:list>` Element mit einem `type="gloss"` Attribut ausgezeichnet werden. Zum anderen könnten sie aber mit anderen Attribut-Wert-Paaren, mit anderen Elementen etc. ausgezeichnet werden. Bei einer Transformation von TEI-basierten Auszeichnungen zu HTML-basierten Auszeichnungen tritt so das Reversibilitätsproblem auf. Es kann nicht grundsätzlich gewährleistet werden, dass von der HTML-Dokumentinstanz wieder die intendierte TEI-Dokumentinstanz generiert wird.

Eine Lösung des Problems bestünde in der Integration von Zusatzinformationen in Dokumentinstanzen. So bietet HTML ein universell einsetzbares `class` Attribut, welches z.B. für die TEI geeignete Parameter zur Transformation beinhalten könnte. Dieses Vorgehen würde jedoch dem Anspruch widersprechen, informationelle Ressourcen nicht zu vermischen. Die `class` Attribute müssten Informationen zu jeweils einem Modell in der sekundären Informationsstrukturierung enthalten, so dass keine anderen Modelle als die TEI Zielpunkt einer Transformation sein könnten.

Die vorliegende Arbeit geht das Problem durch die Beschreibung separater Modelle an. Für eine Dokumentgrammatik bzw. verschiedene Teilmengen dieser können dabei verschiedene Modelle entwickelt werden. So lässt sich in einem Modell für HTML eine

andere sekundäre Informationsstrukturierung für das `<DL>` Element beschreiben als in einem anderen Modell. Je nachdem, welche sekundäre Informationsstrukturierung in der TEI-basierten Dokumentgrammatik intendiert ist, greift der Benutzer auf die jeweiligen Modelle zurück.

Die separate Beschreibung von Modellen stößt jedoch auf Probleme, wenn innerhalb eines Modells in der sekundären Informationsstrukturierung mehrere dokumentgrammatische Konstrukte anwendbar sind. Dies ist z.B. der Fall, wenn die TEI-basierte Dokumentgrammatik in *einem* Modell zugleich das `<html:list>` Element mit `type="gloss"` Attribut-Wert-Paar sowie ein anderes dokumentgrammatisches Konstrukt zulässt. Dieses Problem lässt sich nicht prinzipiell, aber pragmatisch durch verschiedene Transformationsphasen angehen:

1. In der ersten Phase werden Dokumentinstanzen unter Berücksichtigung der sekundären Informationsstrukturierung im Ausgangsmodell transformiert, z.B. von sekStruk-modell-html nach sekStruk-modell-tei. Wenn Ambiguitäten wie die beschriebene auftreten, werden sie in der Ziel-Dokumentinstanz durch Attribute vermerkt, z.B. `sekStruk:matches="dt-tei1 dt-tei2"`.

2. In der zweiten Phase werden die Ambiguitäten unter Zuhilfename der dokumentinstanzbezogenen sekundären Informationsstrukturierung aufgelöst. Varianten definitorischer Listen können anhand ihrer Position (in Kapiteln, im Anhang etc.) unterschieden werden. Die dokumentinstanzbezogenen Kontextbeschreibungen der potentiellen Zielstrukturen helfen, die geeignete Variante auszuwählen.

Eine prinzipielle Lösung des Problems ist nicht möglich. Angenommen, alle Varianten von Listen in HTML – definitorische Listen, ungeordnete Listen etc. – werden in `<list>` Elemente transformiert, dann wäre die Reversibilität nicht realisierbar. Sind die Bedeutungen der dokumentgrammatischen Konstrukte jedoch hinreichend in der sekundären Informationsstrukturierung spezifiziert, bietet die Methodologie der vorliegenden Arbeit einen Ansatz zur reversiblen Transformation von Auszeichnungen.

4.4.2.5. Heuristikbeschreibung für die Unifikation primärdatenidentischer Dokumentinstanzen

Zweck dieser Operation, welche in Witt et al. (2004) geschildert wird, ist die Eingabe von Heuristiken in den Prozess der Unifikation primärdatenidentischer, mehrfach

ausgezeichneter Dokumentinstanzen[13]. Die Operation nutzt als Repräsentationsformat für informationelle Ressourcen von Dokumentinstanzen Prolog-Fakten, welche aus primärdatenidentischen XML-Dokumentinstanzen generiert werden[14]. Die Prädikate, zur Beschreibung von Beziehungen zwischen Auszeichnungsebenen, welche in Abschnitt 4.4.1.3 vorgestellt wurden, können zur Analyse der Prolog-Fakten verwendet werden. Beispiel 4.11 enthält einen Ausschnitt aus einer Faktenbasis.

(4.11)
```
node('levelA', 0, 18, [1, 1], element('x')).
    attr('levelA', 0, 18, [1, 1], 'type', 's1').
    node('levelB', 0, 18, [1, 1], element('y')).
```

Beispiel 4.11: Prolog-Fakten, generiert aus primärdatenidentischen XML-Dokumentinstanzen

Die Faktenbasis enthält zwei Arten von Prädikaten, `node` und `attr`. Das Prädikat `node` enthält Informationen über die Auszeichnungsebene (`levelA`), den Start- bzw. Endpunkt hinsichtlich der enumerierten Zeichen (`0, 18`), die Knotenposition in der Dokumentstruktur (`[1, 1]` sowie den Namen des Elements (`element('X')`. `attr` enthält statt des Elementnamens den Attributnamen (`'type'`) und den Attributwert (`'s1'`).

Anhand der Zeichenenumerierung sind die Beziehungen zwischen den Auszeichnungen erkennbar. Das dritte Prädikat in Beispiel 4.11 entstammt z.B. einer anderen Auszeichnungsebene (`'levelB'`), besitzt aber die gleiche Enumerierung wie die ersten beiden Prädikate. Auf diese Weise lassen sich die Identitätsbeziehungen erkennen.

Aus der Faktenbasis soll ein XML-Dokument generiert werden. Bei den Fakten in Beispiel 4.11 kommt es zu Konflikten zwischen den Auszeichnungen: Ohne zusätzliches Wissen kann nicht bestimmt werden, wie das Verhältnis zwischen `<x>` Element und `<y>` Element aus den beiden Ebenen im Zieldokument sein soll. Für derartige Fälle werden Regel- und Bedingungsbeschreibungen wie in Beispiel 4.12 eingesetzt.

Drei Lösungsstrategien für Konflikte sind vorgesehen: `hierarchy` erzeugt eine Hierarchie zwischen den konkurrierenden Elementen, `delete` löscht eines der Elemente, `attributes` löscht eines der Elemente und fügt seine Attribute an das zweite Element

[13] Die Heuristikbeschreibung ist nur eine optionale Eingabe in die Unifikation, die u.a. auf Analysen der Dokumentinstanzen zurückgreift.

[14] Der Ablauf des Generierungsprozesses und der Zusammenführung von XML-Dokumentinstanzen ist im Rahmen der Beschreibung der Tools dargestellt, vgl. http://coli.lili.uni-bielefeld.de/Texttechnologie/Forschergruppe/sekimo/semt/semt-doku.html

```
(4.12) ruleIdentity(
       (x,levelA,START1,END1,NODE1),
       (y,levelB,START2,END2,NODE2),
        attributes(x)):-
        attr(levelA,START1,END1,NODE1,type,s1),
        one_relation
        (included_A_in_B,y,levelB,NODE2,
         z,levelC,NODE3,START2,END2,START3,END3).
```

Beispiel 4.12: Bedingungen zur Zusammenführung primärdatenidentischer XML-Dokumentinstanzen

an. In Beispiel 4.12 wird die dritte Strategie verfolgt. Das `<x>` Element wird gelöscht, und sein `type` Attribut wird an das `<y>` Element angefügt. Die Ausführung dieser Strategie ist zum einen an eine Regel geknüpft: Das `<y>` Element muss ein Attribut-Wert-Paar `type="s2"` besitzen. Derartige Regeln können aus der dokumentgrammatikbezogenen sekundären Informationsstrukturierung übernommen werden. Wenn die dokumentgrammatikbezogene sekundäre Informationsstrukturierung angewendet wird, sind Lösungsstrategien generierbar, welche sich auf Dokumentklassen beziehen. Voraussetzung ist allerdings, dass die Dokumentgrammatiken in der sekundären Informationsstrukturierung entsprechend spezialisiert wurden und die Spezialisierungen konform sind gegenüber den in Abschnitt 4.4.2.2 vorgestellten Regeln. Beispiele für Spezialisierungen sind z.B. von `element y { attribute type { text } }` zu `element y { attribute type { "s1"} }`. Zum anderen kann die dokumentinstanzbezogene sekundäre Informationsstrukturierung die Formulierung von Bedingungen für eine geschlossene Menge von Dokumentinstanzen speisen. In Beispiel 4.12 lautet die Bedingung, dass das `<y>` Element aus der Ebene `levelB` in der Inklusionsbeziehung mit einem `<z>` Element aus der Ebene `levelB` steht. Zur Formulierung dieser Bedingung kommt das Prädikat `one_relation` zum Einsatz.

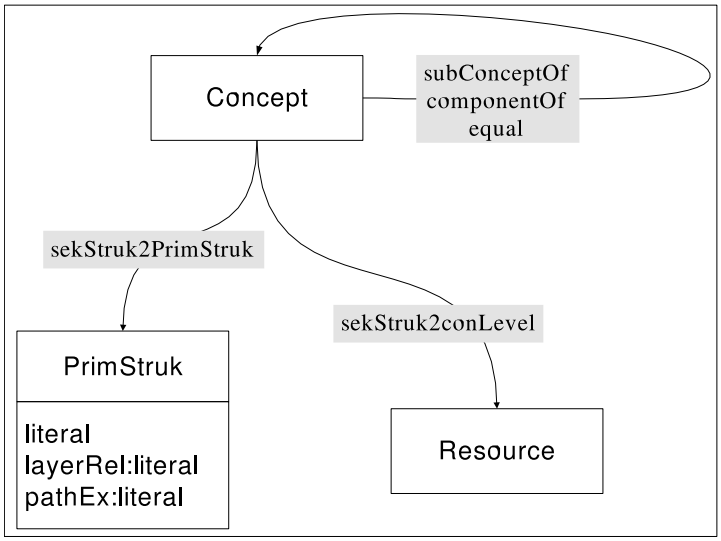

Abbildung 4.10.: Visualisierung des RDF Schema für sekundäre Informationsstrukturierung

4.5. Implementation

Im Folgenden wird die in der Entwicklung befindliche Implementation der Operationen, die im letzten Abschnitt diskutiert wurden, vorgestellt. Ihr wird hier nur wenig Platz eingeräumt, da der Schwerpunkt in dieser Arbeit auf der Entwicklung einer Methodologie liegt.

4.5.1. Syntax zur Repräsentation sekundärer Informationsstrukturierung

RDF-basierte Syntax Zwei Syntaxen für die Repräsentation der sekundären Informationsstrukturierung sind entwickelt worden. Eine Syntax ist als RDF Schema wiedergeben, vgl. Abbildung 4.10. Das RDF Schema ist vollständig in Anhang D wiedergeben. Es dient der Explikation der Aussagen in der sekundären Informationsstrukturierung. Die eigentliche Erzeugung der Aussagen greift auf die im Folgenden beschriebene XML-basierte Syntax zurück. RDF-Dokumente lassen sich daraus generieren.

Das RDF Schema besitzt zwei Klassen, die als rdfs:Class wiedergegeben werden. Concept dient der Beschreibung aller Konzepte, welche in der sekundären Informationsstrukturierung enthalten sind. PrimStruk dient der Beschreibung von informationellen

Ressourcen der primären Informationsstrukturierung. Das Prädikat *sekStruk2primStruk* ist durch eine **Eigenschaft** (property) in RDF Schema wiedergegeben, welche für Concept verwendet werden kann. Wenn es sich um dokumentgrammatikbezogene sekundäre Informationsstrukturierung handelt, ist der Wert dieser Eigenschaft ein Literal. Für die Selektion von Informationseinheiten in einzelnen oder multiplen Dokumentinstanzen ist der Wert ein Literal, welcher ein entsprechendes Präfix beinhaltet. Eine exemplarische Anwendung des Schemas findet sich in Abschnitt E. Die Selektion von informationellen Ressourcen der konzeptuellen Ebene wird durch einen URI wiedergeben, der auf eine beliebige Menge von Resourcen verweisen kann.

XML-basierte Syntax: Annotation von RELAX NG mit Informationen zur sekundären Informationsstrukturierung Eine XML-basierte Syntax ist ebenfalls entwickelt worden. Hierfür gibt es zwei Gründe. Erstens wird so die Implementation der Operationen erleichtert, da das Datenmodell von RDF bzw. RDF Schema keine XML-Serialisierung besitzt, welche durch eine Dokumentgrammatik validierbar ist. Und zweitens ist die XML-Serialisierung eine definierte Menge von Konstrukten zur Annotation von Dokumentgrammatiken im Format von RELAX NG. Mit anderen Worten, die Dokumentgrammatik der XML-Serialisierung lizensiert jede RELAX NG Dokumentgrammatik. Auf diese Weise können in RELAX NG vordefinierte Pattern unverändert in Aussagen der sekundären Informationsstrukturierung übernommen werden.

Beispiel 4.13 zeigt die XML-basierte Syntax für die Konzepte sekStruk:list-tei-general und definitionList-tei. Für jedes Konzept gibt es ein `<define>` Element, welches in RELAX NG zur Definition von Pattern verwendet wird. Dass es sich bei den Pattern zugleich um Beschreibungen von Konzepten handelt, zeigt das Attribut `sekStruk:sekStruk2primStruk`. Die Subordinierung von Konzepten zeigt das Attribut `sekStruk:subConceptOf`, die Abbildung auf die konzeptuelle Ebene das Attribut `sekStruk:sekStruk2conLevel`. Im Beispiel wird ein URI aus dem RDF Schema für WordNet verwendet, welches Melnik Decker entwickelt hat, vgl. `http://www.semanticweb.org/library`

4.5.2. Implementation der Operationen

Gegenwärtig ist eine Implementation des Ansatzes in Entwicklung, die auf die Anfragesprache XQuery zurückgreift. Informationen aus Dokumentgrammatiken, aus einzelnen

```
(4.13) <sekStruk:models
        xmlns:tei="http://example.com/defList-tei"
        xmlns="http://relaxng.org/ns/structure/1.0"
        xmlns:sekStruk="http://example.com/sekStruk">
       <sekStruk:model name="sekStruk-tei">
       <define name="list-tei-general"
        abstract="true"
        sekStruk:sekStruk2primStruk="mapping2">
       <element name="tei:list">...</element>
       </define>
       <define name="definitionList-tei"
        sekStruk:sekStruk2primStruk="mapping4"
        sekStruk:subConceptOf="list-tei-general"
        sekStruk:sekStruk2conLevel=
       "http://www.cogsci.princeton.edu/~wn/concept#103701336">
       <attribute name="type"><value>gloss</value></attribute>
       </define>
       </sekStruk:model>
       </sekStruk:models>
```

Beispiel 4.13: XML-basierte Syntax für sekundäre Informationsstrukturierung

Dokumentinstanzen und aus mehrfach ausgezeichneten, primärdatenidentischen Dokumentinstanzen werden durch die beschriebenen Verfahren selektiert. Im gegenwärtigen Stand der Implementation wird die Information über Start- und Endpunkte von Elementen, die unverzichtbar für die Analyse von Beziehungen zwischen mehreren Dokumentinstanzen ist, nach dem Muster in Beispiel 4.14 in die Dokumentinstanzen eingefügt.

Die Attribute `sekStruk:start` und `sekStruk:end` enthalten numerische Werte, welche die Position des Elements hinsichtlich der Ordnung der primären, textuellen Daten wiedergeben. Die Generierung der Attribute erfolgt automatisch. Dabei kommt das Tool **NEXUS** zum Einsatz, welches von Maas (2003) entwickelt wurde. Dieses Vorgehen hat zwei Nachteile. Zum einen werden informationelle Ressourcen, d.h. die Doku-

```
(4.14) <corpus xmlns:sekStruk="http://example.com/sekStruk" ...>
     <word sekStruk:start="1" sekStruk:end="4">V-TE</word>
     ...<corpus>
```

Beispiel 4.14: Informationen über Start- und Endpunkte von Elementen in Dokumentinstanzen

mentinstanzen verändert, zum anderen ist die Abfrage von Beziehungen zwischen mehreren Dokumentinstanzen sehr rechenaufwändig. Die Repräsentation der Dokumentinstanzen als Prolog-Fakten, vgl. Abschnitt 4.4.1.3 und Abschnitt 4.4.2.5, erlaubt eine wesentlich effizientere Analyse der Beziehungen. Deshalb ist in einer zukünftigen Implementation vorgesehen, diese Repräsentation in die Implementation der Operationen mit einzubeziehen.

Für die Beschreibung der Operationen wurde eine Dokumentgrammatik im Format von RELAX NG entwickelt. Sie ist in Anhang F vollständig wiedergegeben. Exemplarische Operationen, die u.a. auf die sekundäre Informationsstrukturierung in Beispiel 4.13 Bezug nehmen, zeigt Beispiel 4.15.

Die Operationen nutzen in XQuery verwendbare, so genannte **Collections** (Sammlungen von XML Dokumenten). Für Dokumentgrammatiken und Dokumentinstanzen muss eine entsprechende Collection bestehen, nur die Collection von informationellen Ressourcen der konzeptuellen Ebene ist optional. Es gibt vier Operationen: `conceptualQuery`, `conceptualValidation`, `conceptualTransformation` und `generateSolutionStrategy`. Sie werden durch verschiedene XML Elemente spezifiziert. Eingabe der konzeptbezogenen Suchanfrage `conceptualQuery` ist zum einen ein Modell, z.B. sekStruk-tei, und ein Konzept, z.B. sekStruk:definitionList-tei. Das Konzept stammt aus der sekundären Informationsstrukturierung. Zum anderen kann – statt des Konzepts aus der sekundären Informationsstrukturierung – eine Menge von URI aus der konzeptuellen Ebene als Eingabe dienen. In diesem Fall werden die URI in einem `<uriOfConceptualLevel>` Element, im Attribut `href` spezifiziert.

Die konzeptbezogene Validierung beruht auf den gleichen Eingabeinformationen wie die konzeptbezogene Suchanfrage. In Beispiel 4.15 wird das Konzept definitionList-tei validiert. Da in Beispiel 4.13 das übergeordnete Konzept list-tei-general als abstrakt beschrieben ist (`abstract="true"`), müssen alle `<tei:list>` Elemente in Dokumentin-

```
(4.15) <operations>
        <informationResources
       collectionSekStruk="http://www.example.com/SekStruk"
       collectionDocuGram="http://www.example.com/DocuGram"
       collectionInstanceDocs="http://www.example.com/instanceDocs"
       collectionConceptualLevel=
        "http://www.example.com/conceptualLevel"/>
      <conceptualQuery model="sekStruk-tei"
       conceptSekStruk="definitionList-tei"/>
      <conceptualQuery model="sekStruk-tei"
      <uriOfConceptualLevel href=
        "http://www.cogsci.princeton.edu/~wn/concept#103701336"/>
      <conceptualValidation model="sekStruk"
       conceptSekStruk="definitionList-tei"/>
      <conceptualTransformation
       sourceModel="definitionList-tei"
       targetModel="definitionList-html"/>
      </operations>
```

Beispiel 4.15: Exemplarische Operationen

stanzen ein Attribut `type` mit dem Wert `gloss` haben, sonst schlägt die Validierung fehl. Die konzeptbezogene Transformation nimmt als Eingabe ein Ausgangsmodell, z.B. definitionList-tei, und ein Zielmodell, z. B. definitionList-html. Bei der Transformation werden alle Konzepte – bzw. die selektierten Informationseinheiten in Dokumentinstanzen – im Ausgangsmodell, die durch eine Aussage mit dem Prädikat *equal* zu Konzepten aus dem Zielmodell relationiert sind, berücksichtigt.

Eine weitere Operation dient der Erzeugung von Regeln, die Konflikte bei der Unifikation primärdatenidentischer Dokumente auflösen, vgl. Abschnitt 4.4.2.5. Es wird dabei nur der Identitätskonflikt behandelt, vgl. Beispiel 4.16. Das `strategy` Attribut im `<solutionStrategy>` Element bestimmt die Strategie. Die beschriebene Operation führt zur Attributierung vom Element des `levelB` an das Element des `levelA`, und zur

Löschung des Elements des `levelB`. Die Bezüge von `levelA` und `levelB` werden durch entsprechende Verweise auf Modelle und Konzepte aufgelöst.

(4.16)
```
<solutionStrategy strategy="attributes-levelB">
 <levelA model="sekStruk-tei"
  conceptSekStruk="definitionList-tei"/>
 <levelB model="sekStruk-html"
  conceptSekStruk="definitionList-html"/>
</solutionStrategy>
```

Beispiel 4.16: Anwendung einer Lösungsstrategie für konfligierende Elemente

4.6. Kernpunkte des Kapitels

- Das Kapitel fasst zunächt die Desiderata zusammen, die sich aus der vorangegangenen Diskussion ergeben: die Verwendung von Standards zur Repräsentation informationeller Ressourcen, die bidirektional operationalisierare Verbindung der informationellen Ressourcen, und eine Explikation des Ansatzes.

- Ausgangspunkt der Methodologie sekundärer Informationsstrukturierung bildet eine Trennung in inhaltsseitige bzw. intensionale Beschreibung versus eine ausdrucksseitige bzw. extensionale Beschreibung von Konzepten. Die sekundäre Informationsstrukturierung beinhaltet eine intensionale Beschreibung von Konzepten, einer Konzepthierarchie und interkonzeptueller Beziehungen, in Aussagen mit einer Reihe vordefinierter Prädikate. Sie dienen der Selektion und Interrelation verschiedener ausdrucksseitiger Beschreibungen aus der primären Informationsstrukturierung und der konzeptuellen Ebene. Die sekundäre Informationsstrukturierung nutzt Selektionsmechanismen, welche für die verschiedenen informationellen Ressourcen spezifisch sind: für Dokumentgrammatiken sowie einzelne oder mehrere, primärdatenidentische Dokumentinstanzen.

- Dokumentgrammatische Konstrukte werden in Form von Pattern in RELAX NG selektiert. Die Selektion erfolgt allerdings nur innerhalb der Aussagen als Bestandteil der sekundären Informationsstrukturierung, d.h. die ursprünglichen Dokumentgrammatiken müssen nicht verändert werden. Zudem wird die inferentielle Kraft

der Konzepthierarchie in der sekundären Informationsstrukturierung genutzt: Ein dokumentgrammatisches Konstrukt spezialisiert z.B. ein dokumentgrammatisches Konstrukt, welches in übergeordneten Konzepten selektiert ist. Das Prädikat *componentOf* macht die Beziehungen zwischen den Selektionen der dokumentgrammatikbezogenen sekundären Informationsstrukturierung explizit und gewährleistet so die Reversibilität der Selektionen.

- In Dokumentinstanzen werden Mengen von Informationseinheiten selektiert. Das Selektionskriterium in einzelnen Dokumentinstanzen lässt sich als Kontextspezifikation für Knoten auffassen. Zu diesem Zweck kommt die Pfadsprache der Caterpillar-Ausdrücke zum Einsatz. Sie besitzt gegenüber Pfadsprachen wie XPath den Vorteil, dass ihre Ausdruckskraft sich in die Taxonomie der Schemasprachen einordnen lässt. Ein weiterer Vorteil dieser Pfadsprache ist die Beschränkungen des Sprachumfangs. Der darin enthaltene Testoperator für Knoteneigenschaften kann in der sekundären Informationsstrukturierung interkonzeptuelle Beziehungen nutzen. Die zu testenden Eigenschaften wie Elementnamen, Attribut-Werte-Paare etc. werden anhand interkonzeptueller Beziehungen aus anderen Konzepten bzw. deren Selektionen dokumentgrammatischer Konstrukten oder von Informationseinheiten inferiert. Dies hat den Vorteil, dass in der sekundären Informationsstrukturierung ein unmittelbarer Bezug zur Ausdrucksseite „primäre Informationsstrukturierung“ nur für übergeordnete Konzepte erfolgen muss. Untergeordnete Konzepte werden rein inhaltsbezogen definiert und nutzen die inferentielle Kraft der Konzepthierarchie und der interkonzeptuellen Beziehungen.

- Diese Inferenzen können auch genutzt werden, wenn in der primären Informationsstrukturierung mehrfach ausgezeichnete, primärdatenidentische Dokumentinstanzen vorliegen. Sie erlauben die Repräsentation multipler Hierarchien – in den einzelnen Dokumentinstanzen – und die Relationierung der Auszeichnungen zwischen Dokumentinstanzen. Vordefinierte Prädikate beschreiben die Beziehungen der Auszeichnungen und stellen neben der beschriebenen Pfadsprache einen weiteren Selektionsmechanismus der dokumentinstanzbezogenen sekundären Informationsstrukturierung dar.

- Die mit den vorgestellten Mechanismen selektierten informationellen Ressourcen der primären Informationsstrukturierung lassen sich nun durch Aussagen in der

sekundären Informationsstrukturierung zueinander in Beziehung setzen. Voraussetzung dafür ist, dass dokumentgrammatikspezifische Konzepte bzw. Selektionen in separaten Modellen erfasst sind. Ein Anwendungsszenario für die Relationierung von Modellen ist die Beschreibung von Beziehungen zwischen umfangreichen, standardisierten Dokumentgrammatiken.

- Aus der konzeptuellen Ebene kann eine Menge von URI selektiert werden. Ob die URI auf ein Konzept, eine Eigenschaft oder beides verweisen, ist unerheblich. Eine ausgewählte informationelle Ressource der konzeptuellen Ebene, die lexikalische Datenbank WordNet, kann für eine multilinguale Dokumentation von Dokumentgrammatiken eingesetzt werden.

- Die Operationen, welche auf den geschilderten Aussagen zur Selektion und Verbindung informationeller Ressourcen beruhen, gründen auf der Beschreibung der sekundären Informationsstrukturierung als terminologische Ontologie. Zentrales Charakteristikum der terminologischen Ontologie ist die Komplementarität von Intension und Extension, also von inhaltsseitiger und ausdrucksseitiger Beschreibung. Konzepte teilen die Eigenschaften von in der Konzepthierarchie übergeordneten Konzepten. Ihre Extension, d.h. die Menge der Objekte, welche sie beschreiben, ist eine Teilmenge der Extension übergeordneter Konzepte. Hieraus ergeben sich eine Reihe von Konditionen, die für die Bestimmung der Extension eines Konzepts, d.h. von informationellen Ressourcen der primären Informationsstrukturierung bzw. der konzeptuellen Ebene, erfüllt sein müssen.

- Operationen umfassen die Prüfung der Spezialisierbarkeit dokumentgrammatischer Konstrukte; eine konzeptbezogene Suche, Validierung und Transformation; sowie die Beschreibung von Heuristiken in Form von Regeln bzw. Bedingungen für die Zusammenführung primärdatenidentischer Dokumente. Die konzeptbezogene Suche entspricht der Auswertung der Abbildungsfunktion, die – als Teil der beschriebenen Konditionen – informationelle Ressourcen der primären Informationsstrukturierung zu ihrer intensionalen, inhaltsbezogenen Beschreibung, d.h. der sekundären Informationsstrukturierung relationiert. Konzeptbezogene Validierung wird möglich durch die Definition von abstrakten Konzepten, die keine unmittelbar zugeordnete Extensionen haben dürfen. Die konzeptbezogene Transformation ist auf Grund des Informationsverlustes beim Transformationsprozess prinzipiell

nicht immer umkehrbar.

- Die Repräsentation der Aussagen in der sekundären Informationsstrukturierung greift auf eine Syntax in RDF Schema und eine Syntax in XML zurück, letztere wurde als Annotation zu RELAX NG konzipiert. Das RDF Schema macht die Aussagen in der sekundären Informationsstrukturierung durch die Notation als Tripel explizit. Die Annotation zu Relax NG erlaubt es, Pattern in bestehenden Dokumentgrammatiken zu selektieren. Zur Repräsentation der Eingabewerte für Operationen ist eine weitere Syntax in XML definiert worden. Eine gegenwärtig in der Entwicklung befindliche Implementation der Operationen nutzt die Abfragesprache XQuery.

Teil II.

Anwendungen

5. Die Domäne: Linguistische Korpora

5.1. Problemstellung

Unter linguistischen Korpora werden in dieser Arbeit sprachliche Daten verstanden, die zwei Merkmale besitzen, vgl. Sasaki und Witt (2004b): (1) Im Zentrum stehen textuelle Daten, d.h. Verschriftlichungen von Gesprächen oder Texten. (2) Die Daten sind mit Informationen angereichert, wobei von Auszeichnungssprachen, z.B. XML, und Auszeichnungsvokabularen, z.B. dem bereits erwähnten XCES, Gebrauch gemacht wird. Diese Eigenschaften machen linguistische Korpora zu einem idealen Anwendungsfeld für die Methodologie der sekundären Informationsstrukturierung, die ebenfalls textuelle Daten ins Zentrum der primären Informationsstrukturierung stellt.

Die Auszeichung linguistischer Daten wird als **Annotation** bezeichnet. Linguistische Korpora eignen sich besonders für eine primäre Informationsstrukturierung durch Segmentierung, Hierarchisierung und Beschreibung dokumentgrammatischer Regeln im Sinne von Abschnitt 2.2.1.3. Insbesondere wenn verschiedene – linguistische – konzeptuelle Modelle als informationelle Ressourcen zur Verfügung stehen, reichen diese Verfahrensweisen jedoch nicht aus. Die Problematik veranschaulicht Abbildung 5.1, die das zentrale Beispiel in diesem Kapitel visualisiert.

Der auch in den Kapiteln 2 und 3 verwendete Beispielsatz ist auf dreizehn **Annotationsebenen** annotiert. Keine Hierarchisierungen wurden vorgenommen, sondern nur Segmentierungen auf den verschiedenen Ebenen. Ebene 0 beinhaltet das textuelle Ausgangsdatum. Unter den Segmentierungen stehen ihre Bezeichnungen. Die Ebenen sind in 4 Bereiche unterteilt: Syntax (Ebene 1 bis 4), Satztyp (Ebene 5 bis 6), Illokution (Ebene 7 bis 9), Argumentstruktur (Ebene 10 und 11) und interpersonale Beziehungen (Ebene 12 und 13). Für sich gesehen lassen sich die Ebenen problemlos unter Anwendung von XML repräsentieren. Innerhalb der einzelnen Bereiche gibt es auch keine Probleme bei der Hierarchisierung und der Beschreibung dokumentgrammatischer Regeln. So sind z.B. die Annotationseinheiten der Syntax als Nichtterminale einer Phrasenstrukturgramma-

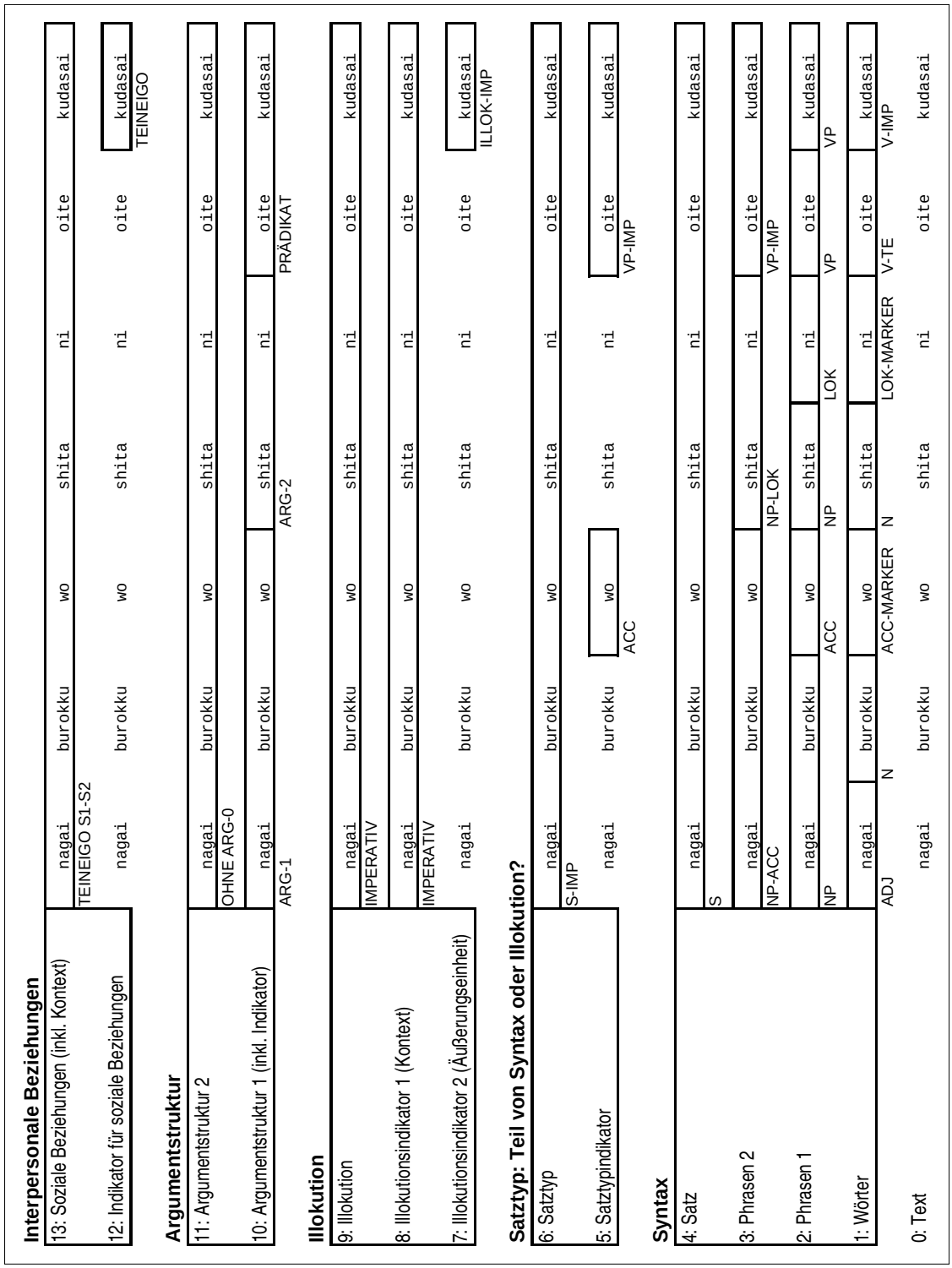

Abbildung 5.1.: Annotation nach verschiedenen konzeptuellen Modellen

tik beschreibbar. Problematisch ist jedoch die Kombination verschiedener konzeptueller Modelle und Annotationen: Wie verhalten sich die verschiedenen Ebenen zueinander, wie die – dokumentgrammatischen – Regeln? Die Zeichenfolge „kudasai“ hat z.B. auf vier Ebenen (Ebene 1, 2, 7 und 12) unterschiedliche Annotationen: als `V-IMP`, `VP`, `ILLOK-IMP` bzw. `TEINEIGO`. Eine für die Syntax relevante Regel würde `V-IMP` und `VP` als Nichtterminale in eine Phrasenstrukturgrammatik integrieren. Für die Annotation `ILLOK-IMP` wäre jedoch eine Bedingungsbeschreibung adäquater. Mit anderen Worten, je nachdem, welche Beschreibungen in welcher Kombination, d.h. Prioritisierung zur Verfügung stehen, lassen sich andere Hierarchisierungen der Annotationen sowie andere Regeln und Bedingungen rechtfertigen.

Annotationen linguistischer Korpora fokussieren oft morphosyntaktische Merkmale wie Wortarten und Kategorien zur syntaktischen Strukturbeschreibung. Ein bekanntes, morphosyntaktisch annotiertes Korpus ist das **BNC** (British National Corpus), welches von Aston und Burnard (1998) beschrieben wird. Es enthält mehr als 100 Millionen Wortformen geschriebener und gesprochener Sprache. Um beispielsweise den Gebrauch verschiedener morphosyntaktischen Konstruktionen analysieren zu können, werden in den letzten Jahren zunehmend **Metadaten** standardisiert. Sie beinhalten zum Beispiel Informationen über Herkunft, Geschlecht und Alter der Sprecher. Die Vergleichbarkeit der Metadaten über Varietäts- und Sprachgrenzen hinweg sollen Metadatenstandards sichern. Beispiele solcher Standards sind **IMDI** (EAGLES/ISLE Meta Data Initiative), vgl. Wittenburg et al. (2000), **OLAC** (Open Language Archives Community), vgl. Simons und Bird (2003), oder **Dublin Core**[1].

Metadaten sind den eigentlichen textuellen Daten und Annotationen übergeordnet. In den letzten Jahren gewinnen Annotationen zunehmend an Bedeutung, die tiefergehende Informationen in Korpora integrieren. Während **Annotationen von Oberflächenstrukturen** wie Wortarten und bestimmte syntaktische Strukturen sich unmittelbar ausgewählten Segmenten eines Textes zuordnen lassen, ist dies z.B. bei **tiefergehenden Strukturen** wie der Prädikat-Argumentstruktur häufig nicht der Fall, vgl. Beispiel 5.1.

Das Beispiel beinhaltet eine Annotation, welche die Ebenen 10 und 11 aus Abbildung 5.1 wiedergibt. Wie beim Imperativsatztyp im Deutschen bleibt die Subjektposition implizit, vgl. ‚Leg den langen Block nach unten‘. Derartige linguistische Phänomene, die

[1] Siehe `http://dublincore.org/`.

```
(5.1) <s empty-argument="ARG-0">
       <argument type="ARG-1">nagai burokku wo</argument>
       <argument type="ARG-2">shita ni</argument>
       <predicate>oite kudasai</predicate>
      </s>
```

Beispiel 5.1: Tiefergehende Strukturen in Annotationen

sich der unmittelbaren Zuordung von Zeichenketten und Annotationsssegmenten entziehen, bilden ein großes Problem für eine linguistische Annotation. Im Beispiel ist die Subjektposition durch ein Attribut am `<s>` Element repräsentiert. Genauso könnte sie jedoch als leeres Element innerhalb des Satzes stehen oder als Attribut am `<predicate>` Element. Welche dieser Alternativen gewählt wird, hat weitreichende Folgen für Analysemöglichkeiten, die ein Korpus bietet. Eine Repräsentation als leeres Element bedeutet die Integration tiefergehender Strukturen in das Inhaltsmodell eines Elements, mit entsprechenden positionalen Restriktionen. Eine Repräsentation als Attribut hingegen beinhaltet weniger derartige Restriktionen.

Die Wahl der Alternativen hat auch Folgen für die Analysemöglichkeiten, welche ein Korpus bietet. Die Abfrage eines Korpus soll Belegexemplare linguistischer Phänomene hervorbringen. Durch welche Konfigurationen von Informationseinheiten die Phänomene repräsentiert sind, ist für den Linguisten oft unerheblich. Der hohe Freiheitsgrad bei der Annotation komplexer linguistischer Phänomene erschwert jedoch die Formulierung generell anwendbarer Suchprozeduren.

Ein Entscheidungskriterium bei der Wahl adäquater Annotationsverfahren bildet der Anwendungszweck für das Korpus. Sollen z.B. **Treebanks** (Baumbanken), d.h. Sammlungen syntaktisch ausgezeichneter Sätze oder Äußerungen, zum Training von syntaktischen Parsern aufgebaut werden, bietet sich die Annotation der Subjektkategorie als Attribut am `<s>` Element an. Die Prädikat-Argument-Struktur ist so unmittelbar abgebildet auf die syntaktische Struktur und kann zur Disambiguierung unterschiedlicher syntaktischer Lesartern verwendet werden. Anders verhält es sich, wenn das Korpus für Sprachlerner entwickelt wird. Für diese sind kommunikativ-funktionale Phänomene, wie soziale Beziehungen, von größerer Bedeutungen als formale, syntaktische Strukturen, vgl.

die Annotationsebenen 12 und 13 in Abbildung 5.1. Diese Anwendungsszenarien[2] stellen jeweils einen Bereich linguistischer Beschreibung in den Vordergrund. Die Korpusdaten müssen dementsprechend angepasst werden: Im Anwendungsszenario „Spracherwerb“ muss die Syntaxannotation gefiltert oder den kommunikativ-funktionalen Annotationen untergeordnet werden, im Anwendungsszenario „Trainingskorpus für syntaktische Parser“ verhält es sich umgekehrt.

Aus diesen Diskussionen ergeben sich folgende Modellierungsanforderungen[3], die mit der in Kapitel 4 entwickelten sekundären Informationsstrukturierung erfüllt werden sollen:

1. Die **Multidimensionalität der Sprache**, d.h. unterschiedliche Sichten und Annotationen der gleichen Daten, muss realisierbar sein.

2. Die Annotation **tiefergehender Strukturen** muss möglich sein.

3. Je nach Art der Repräsentation tiefergehender Strukturen, z.B. als Attribut oder als leeres Element, müssen andere **Suchräume** anwendbar sein, um dokumentgrammatische Regeln und dokumentinstanzbezogene Bedingungen zu formulieren.

4. Die anwendungsspezifische Kombination linguistischer Beschreibungen muss übertragbar sein auf die Anforderungen 1-3. D.h. verschiedene Prioritisierungen erfordern verschiedene Annotationen und Annotationsrelationierungen.

Im Folgenden werden diese Anforderungen im Detail diskutiert und anschließend mit der im ersten Teil dieser Arbeit entwickelten Methodologie angegangen. Dabei kommen nur exemplarische linguistische Daten und Phänomene zur Anwendung. Konkrete Phänomenbereiche werden in Kapitel 6 thematisiert. Besonders zu beachten ist in beiden Kapiteln der terminologische Unterschied zum ersten Teil dieser Arbeit. So ist im Folgenden zum Beispiel von Semantik *im linguistischen Sinne* die Rede, während der Begriff „Semantik“ im ersten Teil, insbesondere in Kapitel 3 der vorliegenden Arbeit, auf informationelle Ressourcen bezogen ist.

[2] Einen ausführlichen Überblick über den Einsatz linguistischer Korpora bietet Garside et al. (1997).
[3] Die Anforderungen 1 und 4 wurden erstmals von Simons (1998) formuliert.

5.2. Anforderungen an eine linguistische Informationsmodellierung

5.2.1. Repräsentation von Multidimensionalität

5.2.1.1. Lösungsansätze zur Repräsentation von Multidimensionalität in der Linguistik

Die Multidimensionalität der Sprache führt zur Falsifizierung der **OHCO-Hypothese**, die schon in Abschnitt 2.2.1.5 vorgestellt wurde. Die Beispiele in Abbildung 5.1 haben bereits gezeigt, dass sich bei linguistischen Daten dieses Problem auf vielfältige Weise äußert. Für linguistische Annotationen wurde deshalb eine Vielzahl von Formaten [4] entwickelt, die zur Lösung dieses Problems beitragen. Die folgende Diskussion der Formate beschreibt ihre Eigenschaften als logische Modelle im Sinne der Definition in Abschnitt 2.1.2. Von ihrer physikalischen Repräsentation, z.B. in XML, wird dabei abgesehen. Anders als die texttechnologische Modellierung in dieser Arbeit fallen bei den meisten dieser Modelle die Ebene des physikalischen Modells und des logischen Modells *nicht* zusammen.

Bei der Verwendung von so genanntem **Standoff Markup**, wie z.B. im Projekt MATE, vgl. Mengel (1999), werden die eigentlichen Daten und die Annotationen voneinander getrennt. Mittels Zeigermechanismen wie XLink, vgl. deRose et al. (1999), kann von einer Annotation aus auf die Daten verwiesen werden. Standoff Markup bietet den Vorteil, nicht auf eine Hierarchie der Annotationen angewiesen zu sein. Jede Annotation kann beliebige Segmentierungen vornehmen. Verweise müssen zudem nicht immer auf das Ausgangsdatum zeigen, sondern können eine bereits annotierte Ebene zum Verweisziel haben. So kann die Phrasenstruktur in Abbildung 5.1 (Ebene 2 und 3) auf die Wortebene (Ebene 1) referenzieren. Die Beschreibung von formalen, z.B. dokumentgrammatischen Regeln, wie z.B. im syntaktischen Modell (Ebene 1 bis 4), ist jedoch mit Standoff Markup schwierig zu realisieren. Bisher existieren keine Implementationen, die eine Ebenen übergreifende Analyse und Validierung der Regeln und Bedingungen erlauben.

Audio- und Videosignalannotationen lassen sich in noch geringerem Maße als textuelle Daten hierarchisieren. Als Beispiel mag die Beziehung von Silben, Satzintonation und Gesten dienen. In den letzten Jahren ist für derartige Annotationen von Bird und Liberman (2001) das Format der **Annotationgraphen** entwickelt worden. Unter den hier

[4] Eine ausführliche Abhandlung des Themas bietet Witt (2002).

diskutierten Annotationsformaten stellen Annotationsgraphen das generellste dar: Eine Annotation beinhaltet einen Startpunkt, einen Endpunkt und eine benannte Kante. Die Punkte referenzieren auf eine **Zeitachse**, welche durch das Audio- oder Videosignal vorgegeben wird. Die Annotation des Wortes „nagai“ beinhaltet z.B. einen Startpunkt 0,03 Sekunden, einen Endpunkt 1,29 Sekunden und die Kantenbenennung `ADJ`. Mehrere Annotationen führen zu einer Graphenstruktur, in der jedes beliebige Segment annotiert werden kann, unabhängig von hierarchischen oder anderen strukturellen Beziehungen zwischen Annotationen.

Dieser Freiraum lässt Annotationsgraphen als ideales Annotationsformat erscheinen. Er stellt aber zugleich die Schwäche des Ansatzes dar. In noch größerem Maße als bei Standoff Markup gestaltet sich die Beschreibung von Ebenen übergreifenden Beziehungen als äußerst komplex. Die von Laprun et al. (2002) vorgestellte, gegenwärtige Implementation des Ansatzes beinhaltet eine Basisfunktionalität zum Ausdruck von Inklusionsbeziehungen zwischen Ebenen, vgl. z.B. Wörter und Phrasen. Die Validierungs- und Analysemöglichkeiten bleiben also weit hinter denen von Auszeichnungssprachen zurück.

Lezius (2002) stellt einen Ansatz vor, der speziell für die Annotation von Treebanks entwickelt wurde. Das Format ist ein gerichteter azyklischer Graph mit einem Wurzelknoten. Die Kanten des Graphen werden explizit durch ineinander verschachtelte XML-Elemente repräsentiert. Baumstrukturelle Beziehungen sind deshalb durch XML-Parser validierbar. Das Konzept der sekundären Kanten erlaubt zudem die Annotation von Beziehungen, die über die Baumstrukturierung hinausgehen, etwa bei diskontinuierlichen Konstitutenen. Für die gewählte Domäne hat dieses Format eine adäquate Ausdruckskraft, wenn die Beziehungen der Kanten zueinander geklärt ist. Dies ist jedoch bei einer Annotation wie in Abbildung 5.1 nicht der Fall. Wie bereits erläutert wurde, können z.B. Satztypen als Teil einer syntaktischen Beschreibung verstanden werden oder als Teil einer illokutionsbezogenen Beschreibung. In dem von Lezius entwickelten Format ist jedoch die einmal getroffene Entscheidung, welche Annotationseinheit den Status einer normalen bzw. sekundären Kante hat, nicht revidierbar.

Das **NOM** (NITE Object Model) wurde von Evert et al. (2003) beschrieben und baut auf einem generellen Graphenmodell auf, welches die folgenden strukturellen Einschränkungen definiert:

1. Bildung von multiplen Hierarchien;

2. Dominanz- und Präzedenzrelationen innerhalb einer Hierarchie, mit definierter ver-

tikaler bzw. sequentieller Distanz;

3. horizontale Distanzen innerhalb einer Hierarchie;
4. Pointer Graphen, die unabhängig von den Hierarchien sind;
5. zeitliche Ordnung von Annotationen.

Jede Hierarchie ist in einer XML-Dokumentinstanz repräsentiert, so dass die Präzedenz- und Dominanzbeziehungen mit variablen, vertikalen bzw. sequentiellen Distanzen validierbar sind. Horizontale Distanzen bestehen zwischen zwei Annotationseinheiten in einer Hierarchie, wobei Elementtypen bei der Distanzermittlung berücksichtigt werden. Es wird zudem eine horizontale Achse erzeugt, die z.B. zwischen zwei Phrasenannotationen nur die Wortannotationen zählt, nicht aber andere Phrasenannotationen. Pointer Graphen dienen der Beschreibung von koreferentiellen oder anaphorischen Beziehungen. Die zeitliche Ordnung von Annotationen wird berücksichtigt, wenn eine Verankerung im entsprechenden Audio- oder Videosignal gegeben ist. Jede Hierarchie kann auf Zeitmarken in einem Signal aufbauen oder auf anderen Annotationsebenen, z.B. Phrasen auf Wörtern.

Welche Annotationen als Hierarchie, durch zeitliche Beziehungen oder Pointer repräsentiert werden, wird für das NOM in einer separaten Definition festgelegt. Aus dieser lassen sich auch Dokumentgrammatiken generieren, mit denen einzelne Hierarchien validierbar sind. Es fällt jedoch schwer, Annotationen aus verschiedenen Hierarchien in einer Hierarchie zu vereinigen. Vor der Korpuserstellung muss die Defintion feststehen. Es sind keine Verfahren beschrieben, welche eine Anpassung der Daten an veränderte Definitionen erlauben.

5.2.1.2. Repräsentation von Multidimensionalität mittels sekundärer Informationsstrukturierung

Es soll nun die Frage geklärt werden, welche Vorteile die Methodologie der sekundären Informationsstrukturierung gegenüber den im letzten Abschnitt diskutierten Ansätzen bietet. Die Stärke der sekundären Informationsstrukturierung liegt in der Möglichkeit, Regeln und Bedingungen auf flexible Weise auf textuelle Annotationen verschiedener Dimensionen anzuwenden. Standoff Markup erlaubt zwar die Repräsentation beliebiger Beziehungen zwischen Annotationen durch Hyperlinks. Die Validierung der Annotationen ist jedoch äußerst komplex. Die Generalität des Ansatzes der Annotationsgraphen

erschwert diese Aufgabe zusätzlich. Sinnvolle Einschränkungen der allgemeinen Graphenstruktur, die den Kern der Annotationsgraphen ausmacht, sind jedoch im Entstehen begriffen. So spezifiziert der Ansatz der **Transkriptionsgraphen**, den (Schmidt, i.V.) entwickelt hat, verschiedene Formen von Graphen, welche für die Visualisierung von theoriespezifischen Diskurstranskriptionen relevant sind.

Die Datenmodelle von Lezius (2002) und des NOM beinhalten für ausgesuchte Domänen spezifizierte, formale Regeln: Treebanks, vgl. auch Abschnitt 6.2, versus multimodale Annotationen. Beide Ansätze stellen leistungsfähige Tools zur Korpusbearbeitung zur Verfügung. Voraussetzung ist allerdings immer, dass die Korpusmodelle *a priori* bestehen. Es fällt schwer, bestehende Daten an eine geänderte Definition anzupassen. Die Methodologie der sekundären Informationsstrukturierung bietet hingegen mehr Flexibilität. Annotationsebenen können beliebig entfernt und hinzugefügt werden. Voraussetzung ist allerdings die Primärdatenidentität der XML-Dokumentinstanzen, vgl. Abschnitt 4.4.1.3.

Zwei Anwendungsbeispiele der sekundären Informationsstrukturierung zur Repräsentation von Multidimensionalität sollen hier vorgestellt werden.

(1) Primärdatenidentische XML-Dokumentinstanzen werden zusammengeführt. Dabei wird auf die Beispiele zurückgegriffen, welche in den dreizehn annotierten Ebenen in Abbildung 5.1 enthalten sind. Beispiel 5.2 beinhaltet eine Dokumentinstanz, die der Argumentstruktur das Primat über andere Annotationen gibt. Dies verdeutlicht das `<argStruc>` Element, das dem `<corpus>` Wurzelelement unmittelbar untergeordnet ist. Wenn die Dokumente in Form einer Prolog-Faktenbasis vorliegen, kann diese Struktur durch den Einsatz der Lösungsstrategie `hierarchy` für die Zusammenführung primärdatenidentischer Dokumente erzielt werden, vgl. Beispiel 4.12 in Abschnitt 4.4.2.5. Bei dieser XML-Dokumentinstanz gehen bestimmte Prämissen einer phrasenstrukturellen Annotation verloren. So gibt es z.B. keine unmittelbare Dominanz aller `<phr>` über die `<wort>` Elemente. Die Beziehungen innerhalb der syntaktischen Phrasenstruktur sind nicht mehr repräsentiert. Im Gegensatz zur syntaktischen, phrasenstrukturellen Annotation lässt sich die Annotation für soziale Beziehungen in diese Dokumentinstanz integrieren, ohne dass die Adäquatheit linguistischer Beschreibung beeinträchtigt wird. Das `social-Rel-default` Attribut am Wurzelelement und das `socRel-type` Attribut am letzten `<w>` Element beinhalten die notwendigen Informationen. Diese Attributierung der Informationen kann durch den Einsatz der Lösungstrategie `attributes` erreicht werden.

```
(5.2) <corpus socialRel-default="TEINEIGO">
      <argStruc socRel-type="TEINEIGO" speechAct-type="IMPERATIVE"
      speechAct-contextInformation="IMPERATIVE-ILLOCUTION"
      sentence-type="IMPERATIVE">
       <implicitArgument type="NO-0"/>
       <argument type="NO-0" syntax-phr-type="NP-ACC">
        <phr type="NP">
         <w cat="N">nagai</w>
         <w cat="N">burokku</w>
        </phr>
        <phr type="ACC">
         <w cat="ACC-MARKER">wo</w>
        </phr>
       </argument>...
        <w cat="V-IMP" socRel-type="TEINEIGO"
         speechAct-type="IMPERATIVE"
         sentence-type-marker="IMPERATIVE">kudasai</w>...
      </argStruc>
      </corpus>
```

Beispiel 5.2: Zusammengeführte Dokumentinstanz I

Beispiel 5.3 gibt der syntaktischen Annotation den Vorang. In dieser Dokumentinstanz sind die phrasenstrukturellen Regeln durch Elemente und ihre Anordnung repräsentiert. Das `<s>` Element dominiert die `<phr>` Elemente, die wiederum die `<w>` Elemente dominieren. Argumentstrukturelle Annotationen lassen sich in dieser Dokumentinstanz realisieren, wenn die entsprechenden Informationen als Attributionen der syntaktischen Einheiten integriert werden. Die Lösungsstratgie `attributes` kann dementsprechend eingesetzt werden.

(2) Primärdatenidentische Annotationen werden zur Relationierung annotierter Daten und maschinenlesbarer Lexika eingesetzt, vgl. Trippel et al. (2003). Die textuellen Annotationen werden dabei durch Signalannotationen ergänzt. D.h. nicht nur verschie-

```
(5.3) <corpus socialRel-default="TEINEIGO">
     <s socRel-type="TEINEIGO" implicitArgument-type="N0-0"
      speechAct-type="IMPERATIVE"
      speechAct-contextInformation="IMPERATIVE-ILLOCUTION"
      sentence-type="IMPERATIVE">
      <phr type="NP-ACC" argument-type="N0-1">
       <phr type="NP">
        <w cat="N">nagai</w>
        <w cat="N">burokku</w>
       </phr>
       <phr type="ACC">
        <w cat="ACC-MARKER">wo</w>
       </phr>
      </phr>...
     </s>
     </corpus>
```

Beispiel 5.3: Zusammengeführte Dokumentinstanz II

dene Dimensionen der Analyse textueller Daten werden kombiniert, sondern zugleich verschiedene Modalitäten, vgl. Abbildung 5.2.

Die Abbildung visualisiert eine Annotation, welche unter Nutzung von **TASX** (Time Aligned Signal data eXchange, vgl. Milde und Gut (2002)) erstellt wurde. TASX umfasst ein Annotationsformat und eine Reihe von Tools, die speziell für die Erzeugung und Verarbeitung multimodaler Daten ausgelegt sind. Die Annotation zeigt – von oben beschrieben – signalbezogene Annotation der Äußerung 'Nimm diese Schraube' hinsichtlich Ortographie, Wortarten, Lemmatisierung, **absoluter Zeit** und so genannter **kategorialer Zeit**. Die Unterscheidung von absoluter Zeit und kategorialer Zeit geht auf Carson-Berndsen (1998) zurück. Die Enumerierung von Zeichen in textuellen Daten lässt sich als Beschreibung einer kategorialen Zeitordnung auffassen. Die textbezogenen Annotationen im unteren Teil von Abbildung 5.2 beinhalten eine solche Annotation.

Lexikalische Einträge werden durch eine Kombination dokumentgrammatikbezoge-

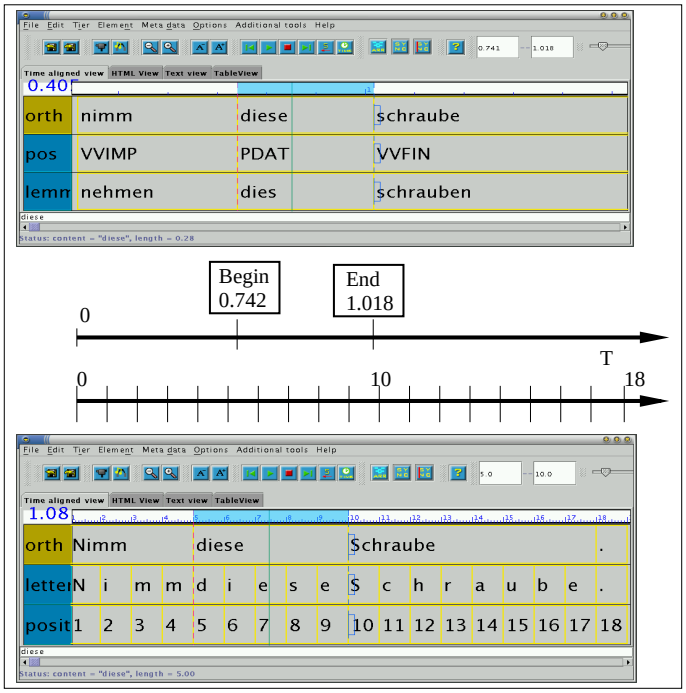

Abbildung 5.2.: Beziehung zwischen absoluter und kategorialer Zeit

ner sekundärer Informationsstrukturierung mit dokumentinstanzbezogener sekundärer Informationsstrukturierung beschrieben. Für letztere kommen korpusbasierte Analysen von Beziehungen zwischen den Annotationsebenen zum Einsatz. Generelle Beschreibungen einer lexikalischen Einheit werden in der sekundären Informationsstrukturierung durch Konzepte definiert, deren Adäquatheit unabhängig von annotierten Daten verifizierbar ist. In der Konzepthierarchie subordinierte Konzepte, d.h. spezifischere lexikalische Beschreibungen, sind nur auf Teilmengen der Instanzen anwendbar. Die Teilmengenbildung erfolgt anhand annotierter Korpora. Ein exemplarischer, lexikalischer Eintrag[5] ist in Beispiel 5.4 wiedergegeben.

Für das Lexikon wird ein Modell sekStruk:lexicon erzeugt. Es beinhaltet eine Konzepthierarchie, die u.a. das Konzept sekStruk:pronouns umfasst. sekStruk:pronouns wird durch die Aussage (sekStruk:pronouns *sekStruk2primStruk*

[5]Trippel et al. (2003) verwenden eine RDF-basierte Syntax, welche zur Repräsentation lexikalischer Einträge spezialisiert ist. Sie ist funktional äquivalent zur sekundären Informationsstrukturierung.

(5.4)
```
sekStruk:sentenceType subConceptOf sekStruk:models.
sekStruk:interrogative subConceptOf sekStruk:sentenceType.
sekStruk:lexicon subConceptOf sekStruk:models.
sekStruk:nominals subConceptOf sekStruk:lexicon.
sekStruk:pronouns subConceptOf sekStruk:lexicon.
sekStruk:pronounSore subConceptOf sekStruk:pronouns.
sekStruk:pronounSore-1 subConceptOf sekStruk:pronounSore.
sekStruk:interrogative sekStruk2primStruk
 "element sentence { attribute type {'interrogative'}, ... }".
sekStruk:pronouns sekStruk2primStruk
 "element word { attribute type { 'pronoun' },  ... }".
sekStruk:pronounSore sekStruk2primStruk
 "..., 'sore'".
sekStrukpronounSore-1 sekStruk2primStruk
 "layerRel-starting_point_A sekStruk:interrogative ".
```

Beispiel 5.4: Lexikonbeschreibung mittels sekundärer Informationsstrukturierung

`"element word { attribute type { 'pronoun' }, ... }".`) auf ein `<word>` Element mit entsprechendem Attribut-Wert-Paar abgebildet. Das Konzept sekStruk:pronounSore beinhaltet eine Spezialisierung des textuellen Inhalts des `<word>` Elements, d.h. zu der Zeichenfolge `sore`. Hier findet sich die Integration der Zeichenebene in die sekundäre Informationsstrukturierung, wie sie in Abschnitt 2.2.2.2 angesprochen wurde. Zugleich ist dies ein Beispiel für eine Spezialisierung, die sich durch die Analyse der Datentypenhierarchie verifizieren lässt, vgl. Abschnitt 4.4.2.2. Auf diese dokumentgrammatikbezogene sekundäre Informationsstrukturierung folgt für das Konzept sekStruk:pronounSore-1 eine dokumentinstanzbezogene sekundäre Informationsstrukturierung. Dabei kommt die Beziehung *starting_point_A* zwischen sekStruk:pronounSore und sekStruk:interrogative zum Einsatz. sekStruk:interrogative selektiert ein `<sentence>` Element mit entsprechendem Attribut-Wert-Paar.

5.2.2. Annotation tiefergehender Strukturen

5.2.2.1. Varianten tiefergehender Strukturen

Neben der Repräsentation von Multidimensionalität ist die Annotation tiefergehender Strukturen der zweite Bereich linguistischer Informationsmodellierung, zu dem die Methodologie der sekundären Informationsstrukturierung einen Beitrag leisten kann. Tiefergehende Strukturen sind in vielen Bereichen Gegenstand der Annotation. Dabei muss keine linguistische Beschreibung vorliegen, die zwischen Oberflächen- und Tiefenrepräsentation unterscheidet. Der Grund für die Annotation tiefergehender Strukturen rührt vielmehr in zweifacher Weise aus dem Verhältnis von Daten und linguistischen Phänomenen her. Erstens differenziert das Symbolinventar der Primärdatenrepräsentation eventuell nicht genug, und zweitens lassen sich die zu annotierenden Phänomene möglicherweise nicht eindeutig bestimmten Sequenzen in den Primärdaten zuordnen.

Der erste Fall tritt zum Beispiel bei der Annotation morphologischer Strukturen im Japanischen auf. Da die Standardverschriftlichung des Japanischen auf ein syllabisches Symbolsystem zurückgreift, sind morphologische, zu Silben inkongruente Annotation nicht möglich. Einen Ausweg bietet der Rückgriff auf ein anderes Symbolsystem, z.B. in einer Lateinverschriftlichung. Dies bedeutet jedoch nur eine Verschiebung des Problems, da z.B. prosodische Annotationen eine noch größere Anzahl an Annotationsmerkmalen benötigen. Eine mögliche Lösung des Problems liegt in der regelhaften Zuordnung von Segmentierungen eines Symbolsystems zu Segmentierungen eines anderen. In Sasaki (2002) wird für bestimmte Wortformen die silbenbasierte Segmentierung japanischer Originalverschriftlichungen auf eine morphembasierte Segmentierung in einer automatisch generierten Lateinumschrift abgebildet. Das Verfahren setzt aber eine regelhafte Inkongruenz der Ebenen voraus, wie sie z.B. bestimmte japanische Wortarten aufweisen.

Der zweite Fall tritt auf, wenn bestimmte Phänomene unabhängig von der Granularität des Symbolsystems nicht an einzelnen Segmenten festzumachen sind. Ein Beispiel, vgl. Abbildung 5.1, ist die Verbform „kudasai“, die durch Stammmodifikation – „kudasar“ wird zu „kusasa“ – und eine Suffigierung – „i“ suffigiert an „kudasa“ – gebildet wird. Die Stammmodifikation lässt sich nicht in den Daten lokalisieren. Die Annotation `ACC` drückt hingegen eine morphologische Markierung des Akkusativ aus, welche sich an dem Segment `wo` festmachen lässt. Allerdings ist für die Beschreibung von `ACC` als

Akkusativmarker insbesondere die Eigenschaft bedeutsam, dass sie auf eine Annotation NP, d.h. die markierte Nominalphrase, folgt. Eine weitere Variation des zweiten Falls tritt auf, wenn ein Segment mehrere Phänomene vereinigt. So vereinigt die Wortform „kudasai“ die satztyp- und illokutionsbezogene Kategorie Imperativ und die für soziale Beziehungen relevante Kategorie TEINEIGO.

Die Unlokalisierbarkeit von Wortformvarianten und die Überlagerung mehrerer Funktionen auf einem Segment ließen sich durch einen Verweis auf ein wortformenbezogenes Paradigma annotieren. Es wird also deutlich, dass der Ausdruck tiefergehender Strukturen in Annotionsvokabularen zum einen **theorieabhängig** ist. Der Verweis auf ein wortformenbezogenes Paradigma realisiert das Modell **Word and Paradigm**, die vorangehende Analyse der Annotation ACC das Modell **Item and Arrangement**. Wenn nicht die Position – ACC steht hinter NP –, sondern der Prozess – ACC suffigiert NP – hervorgehoben wird, entspricht dies dem Modell **Item and Process**[6]. Zum anderen ist der Ausdruck tiefergehender Strukturen **sprachabhängig**. Ausgewählte Sprachen sind durch die jeweiligen Modelle adäquater abgedeckt als andere. Die **Word and Paradigm** Analyse der Stammmodifikation ist nur für einen kleinen Bereich der japanischen Verbmorphologie von Bedeutung. Sie zeichnet sich durch eine Vielfalt von Serialisierungsvarianten der Suffixe aus. Die meisten Verbformen beinhalten keine Stammmodifikation, weshalb die anderen beiden Modelle eine aufschlußreichere Analyse erlauben.

Eine besondere Problematik tritt im Japanischen auf, weil die Standardverschriftlichung keine Wortsegmentierungen beinhaltet. In Abbildung 5.1 sind z.B. in den Ebenen 1 bis 3 Annotationen vorgenommen, die sich auf folgende Art analysieren lassen:

1. Die Phrasen der Ebene 3 enthalten Phrasen der Ebene 2, die wiederum Wörter der Ebene 1 enthält.

2. Die Phrasen der Ebene 3 entsprechen einem bestimmten Paradigma, das durch Wortformen der Ebene 1 realisiert wird. Ebene 2 ist für diese Analyse irrelevant.

3. Die Phrasen der Ebene 3 entsprechen einem bestimmten Paradigma, das durch Wortformen der Ebene 2 realisiert wird. Genauso verhält es sich mit der Beziehung zwischen den Ebenen 2 und 1.

Die erste Analyse ist z.B. für ein automatisches Tagging morphologischer und phrasenstruktureller Informationen von Nutzen. Die stochastische Gewichtung der Wahr-

[6] Zu einer genaueren Beschreibung der Modelle vgl. Hockett (1954).

scheinlichkeit von Übergängen zwischen Segmenten fungiert dabei als Indikator für die adäquate Kategorienzuweisung. Allerdings ist diese Analyse – wie sich gezeigt hat – nicht immer realisierbar, so dass auf die zweite bzw. dritte Analyse zurückgegriffen werden muss. Asahara et al. (2002) nutzen bei ihrer Entwicklung eines integrierten Repräsentationsformates für japanische Lexika und Korpora deshalb alle drei Analysen und beschreiben Überführungskonventionen – soweit dies möglich ist.

Die bisherigen Beispiele für die Annotation tiefergehender Strukturen beinhalteten Phänomene, die sprachlich realisiert sind. Annotationsprobleme entstehen bei der Zuordnung von Segmenten in Daten und theorie- oder sprachspezifischen Phänomenanalysen. Eine andere Problematik tritt auf, wenn die Phänomene sprachlich implizit bleiben, d.h. nur als sogenannte **leere Kategorien** analysiert werden können. Dies ist z.B. in Ebene 11 in Abbildung 5.1 der Fall. Ein Argument des Prädikats, d.h. die Subjektposition, bleibt implizit. Die in diesem Abschnitt vorgenommene Differenzierung tiefergehender Strukturen hat den Vorteil, dass auch derartige Phänomene annotierbar sind. Sie unterscheiden sich von den anderen Varianten des zweiten Falls tiefergehender Strukturen durch den **Suchraum**, der für ihre Interpretation notwendig ist. Bei einer Wortformannotation besteht der Suchraum aus einem Verweis auf das Paradigma, dessen Annotation *unmittelbar* am annotierten Segment erfolgt. In XML kann die Annotation etwa in Form eines Attributs `cat="V-IMP"` ausgedrückt werden. Bei der Annotation eines impliziten Arguments `OHNE ARG-0` in Ebene 11 hingegen muss eventuell die Konfiguration anderer Annotationseinheiten ermittelt werden. Umfasst die Annotation des Prädikats auf Ebene 10 eine Information über erforderliche Argumente, kann daraus die Implizitheit eines Arguments inferiert werden. Diese Information dient wiederum der Validierung der Annotation `OHNE ARG-0` auf Ebene 11.

Die folgende Liste fasst die verschiedenen Motivationen und Annotationsvarianten tiefergehender Strukturen zusammen.

1. Das Symbolsystem ist zu undifferenziert. Ein Beispiel ist die Morphemannotation in syllabischen Schriftsystemen. Ein Verfahren ist die Annotation nach dem Modell **Word and Paradigm**, ein anderes die regelhafte Zuordnung der Segmentierungen in den jeweiligen Symbolsystemen.

2. Zu annotierende Phänomene können nicht auf ein Segment bezogen werden. Für diesen Fall gibt es drei Varianten:

a) Phänomene können auf Segmentsequenzen auf einer Ebene bezogen werden. Beispiele sind Morphemanalysen nach den Modellen **Item and Arrangement** oder **Item and Process**. Ein Verfahren ist die Annotation nach diesen Modellen.

b) Phänomene können auf Segmentkombinationen auf einer Ebene bezogen werden, diese sind zu anderen Ebenen relationierbar. Ein Beispiel ist die Suffigierung ohne Stammmodifikation für bestimmte Wortarten. Die Annotation beinhaltet die entsprechenden Regeln, z.B. in Form von Inhaltsmodellen wie „`<w>` besteht aus `<prefix>`, `<stem>` und `<suffix>`“.

c) Phänomene können auf Verweise auf Informationen bzw. Konfigurationen von Informationen bezogen werden, die auf anderen Ebenen vorliegen. Es gibt vier Subvarianten, die sich hinsichtlich des Suchraums zur Informationsermittlung unterscheiden:

   i. Die Information kann unmittelbar an dem annotierten Segment festgemacht werden. Ein Beispiel ist die Überlagerung wortformenbezogener Kategorien. Ein Annotationsverfahren besteht in der Integration unmittelbarer Informationen, z.B. durch Attribute.

   ii. Die Information kann mittelbar zum annotierten Segment an einem anderen Segment festgemacht werden. Die Segmente stehen in einer hierarchischen Beziehung zueinander. Ein Beispiel ist das Verhältnis von Phrasen- und Wörtern im Japanischen. Ein Annotationsverfahren besteht in der Integration mittelbarer Informationen, unter Verwendung eines auf die Hierarchie bezogenen Suchraums.

   iii. Die Information kann mittelbar zum annotierten Segment festgemacht werden. Sie ist nicht innerhalb einer Hierarchie erreichbar, aber innerhalb eines eingeschränkten Suchraums im Korpus. Ein Beispiel sind Illokutionsindikatoren, die unter Zuhilfenahme des Kontexts interpretiert werden, z.B. anhand der Illokution vorhergehender Äußerungen, vgl. Ebene 8 in Abbildung 5.1. Bei der Annotation kann ein entsprechender Suchraum definiert werden.

   iv. Die Information kann mittelbar zum annotierten Segment festgemacht werden. Sie ist nur unter Rückgriff auf den gesamten Korpus erreichbar. Ein Beispiel ist die Analyse sozialer Beziehungen zwischen den Ge-

sprächsteilnehmern. Sie beinhaltet Kontextinformationen zum Diskurs, zugleich bezieht sie Indikatoren in einzelnen Äußerungen mit ein.

Die Liste verdeutlicht zwei zentrale Eigenschaften tiefergehender Strukturen. Erstens, die letzte Annotationsvariante entspricht der Anreicherung von Korpora mit Metadaten, die in Abschnitt 5.1 angesprochen wurden. Üblicherweise sind diese, wie gesagt, Gegenstand einer separaten Standardisierung und nicht Bestandteil sprach- oder theoriespezifischer Annotationsvokabulare. Die vorgeschlagene Differenzierung von Suchräumen ermöglicht es, Metadaten im Prozess der Validierung und Analyse mit anderen Annotationen im Korpus zu verknüpfen. Die Integration von Metadaten zeigt als zweite Eigenschaft tiefergehender Strukturen die zentrale Rolle von Suchräumen zur Differenzierung tiefergehender Strukturen.

5.2.2.2. Bestehende Annotationsverfahren

Die Annotation tiefergehender Strukturen ist Bestandteil verschiedener Annotationsvokabulare. Im Folgenden werden vier davon diskutiert:

- Die MATE-Richtlinien für morphosyntaktische Annotation, vgl. Mengel et al. (2000);
- die Annotationsrichtlinien für die English Dependency Treebank, vgl. Rambow et al. (2002);
- Die Annotationsrichtlinien für das FrameNet-Projekt, vgl. Fillmore et al. (2001);
- Die Annotationsrichtlinien für die Proposition Bank, vgl. Kingsbury et al. (2002).

Die Ansätze lassen sich unterscheiden hinsichtlich der Rolle, die sie Suchräumen für die Interpretation tiefergehender Strukturen zuweisen. Die beiden Extreme sind zum einen der Verzicht auf eine Analyse dokumentstruktureller Beziehungen, d.h. tiefergehende Strukturen werden allein durch unterschiedliche Benennungen von Annotationseinheiten differenziert. Dies kann man als benennungsbezogene tiefergehende Strukturierung bezeichnen. Das andere, von keinem Ansatz vollkommen realisierte Extrem ist die suchraumbezogene tiefergehende Strukturierung.

Die Annotationsrichtlinien des MATE-Projekts beinhalten ein Schema zur syntaktisch-funktionalen Annotation. Mit diesem lassen sich nicht realisierte Kategorien wie in Beispiel 5.5 annotieren.

```
(5.5) <mw id="mw_001">oite</mw>
      <mw id="mw_002">kudasai</mw>
      ...
      <funct id="funct_001">
        <head id="h_001" href="mword.xml#id(mw_001)..id(mw_002)"/>
        <dep id="d_001" type="subj"/>
      </funct>
```

Beispiel 5.5: Annotationskategorien für Morphosyntax

Das Prädikat wird im `<head>` Element durch einen Verweis zu `<mw>` Elementen auf der Wortebene annotiert. Das nicht realisierte Subjekt wird durch ein `<dep>` Element ohne Verweis annotiert. Dieses Verfahren ist rein benennungsbezogen. Die Annotation des nicht realisierten Subjekts beinhaltet den `subj` Wert des `type` Attributs. Es ist jedoch keine Überprüfung von Beziehungen realisierter Einheiten möglich, welche die Annotation nicht realisierter Einheiten validierbar macht. Dazu könnte in Beispiel 5.5 die Wortform „kudasai“ zur Annotation eines Imperativ-Satztyps relationiert werden. Die Gruppierung des `<dep>` Elements mit einem bestimmten `<head>` Element innerhalb eines `<funct>` Elements ist ebenfalls nicht validierbar. Argument und Prädikat sind nur durch `ID` Attribute identifiziert. Es kann so nicht mit den Mitteln der primären Informationsstrukturierung verhindert werden, dass ein `<head>` Element mit einem `ID` Attribut `id="_001"` neben einem `<dep>` Element mit einem `ID` Attribut `id="d_058"` steht.

Die English Dependency Treebank macht – im Gegensatz zum MATE-Projekt – explizit von einem theoriespezifischen, dependenziellen Annotationsverfahren Gebrauch. Leere Kategorien werden in Abhängigkeit von ihrer Kategorienzugehörigkeit mit zusätzlichen Informationen versehen. Ein nicht realisiertes Subjekt in einem Aufforderungssatz wird durch einen Verweis auf das entsprechende Prädikat annotiert. Durch Kommata separierte Aufzählungen werden als Konjunktion interpretiert, sie besitzen also den gleichen Status wie sprachlich realisierte Konjunktionen. Dieses Vorgehen ermöglicht innerhalb des spezifischen, theoretischen Rahmens eine Validierung der Annotationen.

Im Rahmen des FrameNet-Projekts werden tieferliegende Strukturen als **Frames** annotiert. Frames ordnen lexikalischen Einheiten Konfigurationen sogenannter **Frame Entities** zu, die in etwa den Argumenten auf der Ebene 10 in Beispiel 5.1 entsprechen. Nicht realisierte, sogenannte **Implicit Frame Entities** werden in folgende Gruppen unterteilt:

- Konstruktionale Implicit Frame Entities. Die Implizitheit beruht auf bestimmten syntaktischen Konstruktionstypen, z.B. dem Aufforderungssatztyp. Die Implizitheit des Subjekts `OHNE-ARG-0` in Abbildung 5.1 lässt sich auf diese Weise erklären.
- Existentielle Implicit Frame Entities. Das Prädikat erfährt eine generische Interpretation, vgl. ‚Ich arbeite jeden Tag'. Die Realisierung einer Frame Entity, in diesem Fall der Arbeitsort, wird unnötig.
- Anaphorische Implicit Frame Entities, die sich nur durch den Diskurskontext bzw. Eigenschaften der Domäne erklären. So drücken Bedienungsanweisungen – sowohl im Deutschen als auch im Japanischen – Imperative teilweise durch Infinitivkonstruktionen aus, vgl. ‚Den Block nach unten *legen.*' versus „nagai burokku wo shita ni *oku.*".

Diese Liste macht deutlich, dass bestimmte Suchräume mit bestimmmten linguistischen Phänomenen in Beziehung stehen. Wie Annotationen validiert werden, d.h. welche Suchräume zum Einsatz kommen, hängt also von den Phänomenen bzw. der theorie-, sprach- oder domänenspezifischen Sicht auf die Phänomene ab. Die Implizitheit auf Grund von Konstruktionstypen kann auf entsprechende (Satz)typmarkierungen verweisen, welche wiederum eine Verbindung zu Argument- bzw. Frame-Struktur besitzen. Derartige Validierungen werden im FrameNet-Projekt jedoch nicht vorgenommen.

Von den vorgestellten Ansätzen sind die Annotationskonventionen der Proposition Bank am stärksten an eine tiefergehende Strukturierung gebunden. Die Motivation für eine tiefergehende Strukturierung ist ähnlich wie bei FrameNet und der English Dependency Treebank. Konzeptuelle Muster, d.h. Frames bzw. Prädikat-Argument Strukturen, sollen möglichst eindeutig zu syntaktischen Realisierungen in Beziehung gesetzt werden. Die Prädikat-Argument Strukturen werden als Standoff Markup mit den Basisannotationen verknüpft. Die Basisannotationen sind die von Bies et al. (1995) vorgestellten Phrasenstrukturannotationen der Penn Treebank, vgl. Beispiel 5.6.

In der Phrasenstruktur hat das nicht realisierte Subjekt eine bestimmte Position und wird durch ein `*` Symbol wiedergegeben. Je nach Art der nicht realisierten Einheiten

```
(5.6) (S  (NP-SBJ *)
      (VP (NP nagaiburokkuwo)
          (NP shitani)
              oitekudasai))
```

Beispiel 5.6: Phrasenstrukturannotationen der Penn Treebank

werden diese Symbole an anderen Positionen in der Annotation eingefügt. Anders als bei FrameNet werden jedoch keine Suchräume spezifiziert, die der Annotationsvalidierung oder -analyse dienen könnten.

5.2.2.3. Annotation tiefergehender Strukturen und Beschreibung von Suchräumen mittels primärer und sekundärer Informationsstrukturierung

Die Diskussion von Annotationsvokabularen für tiefergehende Strukturen hat gezeigt, dass die Validierung oder Analyse tiefergehender Strukturen bisher nur in geringem Maße möglich ist. Das Annotationsvokabular des MATE-Projekts erlaubt nur die Konstruktion von Verweisen zwischen Einheiten im ganzen Korpus. Eine Differenzierung von Suchräumen, wie sie in Abschnitt 5.2.2.1 vorgeschlagen wurde, ist mit dieser Methode verwehrt. Die Annotationsvokabulare der Dependency-Treebank und der Proposition Bank zeigen auf verschiedene Weise, wie eine bestimmte Theorie sich in den Analysemöglichkeiten von Korpusannotationen niederschlägt. Das differenzierte Paradigma nicht realisierter Kategorien in der Dependency-Treebank beinhaltet immer eine Relation zu realisierten Kategorien. In einem Aufforderungssatztyp wird z.B. das implizite Subjekt zum Verb in Beziehung gesetzt. Die Proposition Bank hingegen setzt Marken für implizite Einheiten an bestimmten Positionen in der Phrasenstrukturannotation. Dabei wird keine unmittelbare Beziehung zu realisierten Einheiten ausgedrückt. Sie ergibt sich höchstens indirekt, unter Rückgriff auf die syntaktischen Regeln und Bedingungen. Das FrameNet-Projekt schließlich sieht eine Differenzierung von Suchräumen und bestimmten Benennungen von Annotationseinheiten vor. Während die Beziehung zwischen Aufforderungssatztyp und der Implizitheit des Subjekts in der Dependency Treebank nicht weiter analysiert wird, ordnet das FrameNet-Projekt sie in eine Topologie impliziter Einheiten ein.

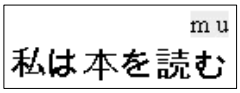

Abbildung 5.3.: Visualisierung von Ruby-Annotationen

Wie verhält es sich nun mit der sekundären Informationsstrukturierung? Sie erlaubt es, Annotationen in variabler Weise zu verknüpfen und Strukturbeschreibungen in Form inhaltsseitiger Beschreibungen auf die gleichen Annotationen zu beziehen. Im Gegensatz zu Ansätzen wie dem NOM sind die Strukturbeschreibungen jedoch nicht unabdingbar. Die annotierten Korpora können auch ohne inhaltsseitige Beschreibung, d.h. ohne Definition einer Strukturbeschreibung in der sekundären Informationsstrukturierung genutzt werden. Dann bietet das Format der primärdatenidentischen, multiplen Annotation maximale Flexibilität beim Zugriff auf die Korpora.

Wie bereits angemerkt, ist die sekundäre Informationsstrukturierung nicht nötig, um alle Varianten tiefergehender Strukturen zu beschreiben. Die Variante 1 – mangelnde Differenzierung des Symbolsystems, vgl. Seite 156 – lässt sich auch innerhalb einer Dokumentinstanz mittels primärer Informationsstrukturierung wiedergeben, vgl. Beispiel 5.7 und Abbildung 5.3.

(5.7)
```
<ruby>
<rb>Basistext</rb>
<rt>Zusatztext</rt>
</ruby>
```

Beispiel 5.7: Das `<ruby>` Element

Die Darstellung ist mittels des XHTML-Modules **Ruby** erstellt, welches von Sawicki et al. (2001) definiert wurde. Ruby ist zur Auszeichnung von Zusatzinformationen, z.B. hinsichtlich Lautungen, von hauptsächlich asiatischen, piktographischen Zeichen definiert worden. Der Originaltext steht in einem `<rb>` Element. Der ergänzende Text steht in einem `<rt>` Element. Innerhalb des `<rt>` Elements lassen sich auch vom Originaltext abweichende Segmentierungen definieren. Im Beispiel wird das japanische Zeichen durch eine Lateintranskription ergänzt, welche eine Beschreibung der Morphemgrenze `m u` erlaubt.

Der Einsatz der sekundären Informationsstrukturierung erlaubt eine Spezialisierung der Regeln, ohne dass die generelle Dokumentgrammatik für Ruby Veränderungen erfahren müsste. Eine denkbare Spezialisierung zeigt Beispiel 5.8.

(5.8)
```
sekStruk:rubyModel subConcept of sekStruk:models.
sekStruk:ruby-general subConceptOf sekStruk:rubyModel.
sekStruk:ruby-japanese-verbs
 subConceptOf sekStruk:rubyModel.
sekStruk:ruby-japanese-consonantVerb
 subConceptOf sekStruk:ruby-japanese-verb.
sekStruk:ruby-general
 sekStruk2primStruk "element rt { text }".
sekStruk:ruby-japanese-verbs sekStruk2primStruk
 "element rt { text, 'u' }".
sekStruk:ruby-japanese-consonantVerbs sekStruk2primStruk
 "element rt { list { ('b'|'m'|'r'), 'u' } }".
```

Beispiel 5.8: Relationierung von Symbolsystemen durch sekundäre Informationsstrukturierung

Für Ruby-Annotationen wird ein Model sekStruk:rubyModel beschrieben. Es enthält drei hierarchisch angeordnete Konzepte sekStruk:ruby-general, sekStruk:ruby-japanese-verb und sekStruk:ruby-consonantVerbs. sekStruk:ruby-general ist auf das `<rt>` Element abgebildet. sekStruk:ruby-japanese-verb selektiert eine Einschränkung des textuellen Inhalts: Das letzte Zeichen muss ein `u` sein. sekStruk:ruby-japanese-consonantVerbs schließlich beschreibt ein Muster für japanische Verben einer bestimmten Flexionsklasse. Diese Spezialisierung lässt sich unter Rückgriff auf die Datentypenhierarchie verifizieren, vgl. Abschnitt 4.4.2.2. Sie hätte auch innerhalb der Dokumentgrammatik vorgenommen werden können. Dies würde jedoch bedeuten, dass das XHTML-Modul für Ruby geändert werden müsste. Diese Anforderung lässt sich jedoch kaum erfüllen.

Auch die Variante tiefergehender Strukturen 2a und 2b auf Seite 157 – linguistische Phänomeme, beschrieben als Segmentsequenzen auf einer Ebene und eventuell in hierarchischer Beziehung zu anderen Annotationen stehend –, lässt sich durch ähnliche Formen sekundärer Informationsstrukturierung beschreiben. Das generelle Prinzip liegt in einer dokumentgrammatischen sekundären Informationsstrukturierung. Hierfür sind Re-

geln zur Verifikation der Spezialisierungen von Inhaltsmodellen, vgl. Abschnitt 4.4.2.2, relevant. Ein generelles Inhaltsmodell für das Verhältnis der Annotationen `NP` und `ACC` lautet z. B. `(NP | ACC)*`. Die Spezialisierungsregel 1 lizensiert ein linguistisch adäquates Inhaltsmodell wie `NP, ACC`.

Die Varianten 2(c)i und 2(c)ii auf Seite 157 – Phänomene sind durch Verweise auf ebenenübergreifende Informationen beschreibbar – lassen sich durch dokumentgrammatikbezogene sekundäre Informationsstrukturierung erfassen. Beispiel 5.9 beschreibt das Verhältnis von Phrasen und Wörtern als eine sekundäre Informationsstrukturierung für das `<N>` bzw. das `<ADJ>` Element auf der Annotationsebene eins in Abbildung 5.1.

(5.9)
```
sekStruk:immediate-dominance subClass of sekStruk:models.
sekStruk:noun subClass of sekStruk:models.
sekStruk:adjective subClass of sekStruk:models.
sekStruk:nounPhrase subClass of sekStruk:models.
sekStruk:noun sekStruk2primStruk "element N { text }".
sekStruk:adjective sekStruk2primStruk "element ADJ { text }".
sekStruk:nounPhrase sekStruk2primStruk
 "element NP { (sekStruk:noun | sekStruk:adjective)* }".
sekStruk:noun componentOf sekStruk:nounPhrase.
sekStruk:adjective componentOf sekStruk:nounPhrase.
```

Beispiel 5.9: Beschreibung unmittelbarer Dominanz duch sekundäre Informationsstrukturierung

Für die Beziehung der **unmittelbaren Dominanz** zwischen syntaktischen Einheiten wird ein Modell sekStruk:immediate-dominance beschrieben. Es enthält die Konzepte sekStruk:adjective, sekStruk:noun und sekStruk:nounPhrase. Sie selektieren durch Aussagen mit dem Prädikat *sekStruk2primStruk* entsprechende Elementdeklarationen. Die Konzepte sekStruk:noun und sekStruk:adjective sind Teil des Inhaltsmodells vom `<NP>` Element. Die Prädikate *componentOf* sichern, dass die Selektion in der adäquaten Abfolge geschieht.

Die Prädikate *componentOf* zeigen zugleich den Gewinn sekundärer Informationsstrukturierung für die Beschreibung der unmittelbaren Dominanz. Die primäre Informationsstrukturierung erlaubt es nicht, die Beziehung der unmittelbaren Dominanz als Eigenschaft der `<N>` bzw. `<ADJ>` Elemente zu beschreiben. Sie ist nur aus der Perspek-

tive der <NP> Elemente fassbar, deren Inhaltsmodell <N> bzw. <ADJ> Elemente enthält. Durch die Prädikate *componentOf* wird diese Perspektive umgedreht. Auf diese Weise lässt sich z.B. ein generelles Konzept für Wörter definieren, welches keine unmittelbare Dominanz durch Nominalphrasen mit einschließt. Diese Eigenschaft ist dann subordinierten Konzepten wie sekStruk:noun vorbehalten.

Die Varianten 2(c)iii und 2(c)iv auf Seite 157 schließlich – Informationen stehen in keinem hierarchischen Verhältnis, wobei eventuell der ganze Korpus den Suchraum darstellt – greifen auf eine Verbindung von dokumentgrammatischer sekundärer Informationsstrukturierung und dokumentinstanzbezogener sekundärer Informationsstrukturierung zurück. In Beispiel 5.10 werden die in Abschnitt 5.2.2.2 angesprochenen konstruktionalen Implicit Frame Entities durch eine derartige sekundäre Informationsstrukturierung identifiziert.

Für verschiedene Formen der Implizitheit werden entsprechende Konzepte deklariert: sekStruk:constructionalImplicity, sekStruk:existentialImplicity und sekStruk:anaphoricImplicity. Konstruktionale Implicit Frame Entities zeichnen sich durch die Implizitheit eines Arguments und das Vorhandensein eines syntaktischen Indikators aus. Dem Konzept sekStruk:imperativeMarker, welches als Indikator fungiert, entspricht in der primären Informationsstrukturierung das <V-IMP> Element. Dies beschreibt die Aussage (sekStruk:imperativeMarker *sekStruk2PrimStruk* `"element V-IMP { text }"`.). Das Konzept sekStruk:missingArgument drückt die Implizitheit eines Arguments aus. Es wird mit dem <argumentStructure> Element verknüpf, wobei ein entsprechendes Attribut-Wert-Paar angenommen wird. Die Aussage (sekStruk:missingArgument *sekStruk2PrimStruk* `"element argumentStructure { attribute missingArgument { 'zero' }, ...}"`..) beschreibt diese Verknüpfung. Die Konzepte sekStruk:missingArgument und sekStruk:imperativeMarker werden durch das Prädikat *layerRel-included_B_in_A* miteinander in Beziehung gesetzt. Dies geschieht innerhalb einer Aussage zum Konzept sekStruk:missingArgumentConstructional. Die Aussage ist die letzte von den diskutierten Aussagen, welche sich auf die Modelle sekStruk:argumentStructure und sekStruk:words beziehen. Das Konzept für konstruktionale Implicit Frame Entities wird durch das Prädikat *equal* auf diese beiden Modelle bezogen: sekStruk:constructionalImplicity *equal* sekStruk:missingArgumentConstructional.

Das Beispiel zeigt, dass sich die Varianten 2(c)iii und 2(c)iv auf Seite 157 im Prinzip nicht unterscheiden. Beide benötigen „artifizielle“ Inhaltsmodelle bzw. übergreifende An-

notationen, um die Beziehungen zwischen nicht hierarchisch relationierbaren Einheiten zu beschreiben. Das Verfahren ähnelt einem Ansatz zur variablen Analyse von Metadaten Trippel et al. (2004). In diesem Ansatz werden XML-Dokumentinstanzen auf konzeptuelle Modelle abgebildet, in einer Umkehrung des Ansatzes von Erdmann und Studer (1999), vgl. Abschnitt 3.2. Im Anschluss können die verschiedenen Metadaten – Metadaten für den ganzen Korpus, für einzelne Ausschnitte oder Äußerungen etc. – variabel miteinander kombiniert und abgefragt werden. Diese Bottom-Up gerichtet Abbildung von XML-Dokummentinstanzen auf konzeptuelle Modelle generalisiert jedoch die Konfigurationen zwischen Informationseinheiten: Die Verschachtelung von Elementen wird als *partOf* Beziehung interpretiert, weitergehende Konfigurationsbeschreibungen für dokumentgrammatische Konstrukte bzw. Informationseinheiten spielen jedoch keine Rolle.

5.2.3. Theorie-, Sprach- und Domänenspezifik konzeptueller Modelle

Hinter dem Vorgehen, welches in den letzten Abschnitten ausgeführt wurde, steht die Annahme, dass die eigentliche Voraussetzung für eine erfolgreiche Verbindung informationeller Ressourcen in ihrer differenzierten Beschreibung liegt. Selbst für die informationellen Ressourcen der primären Informationsstrukturierung ist diese Beschreibung nicht leicht zu realisieren. Die Gründe hierfür liegen in der Spezifik linguistischer informationeller Ressourcen. Linguistische konzeptuelle Modelle können theorie-, sprach oder domänenspezifisch sein. In diesem Kapitel haben hauptsächlich theoriespezifische Modelle eine Rolle gespielt, wie die morphologischen Ansätze von Hockett, die dependenziellen bzw. phrasenstrukturellen syntaktischen Analysen der Dependency Treebank bzw. der Proposition Bank etc. Oft zeigen sich Überlagerungen von Theorie-, Sprach- und Domänenspezifik, z.B. bei Auszeichnungsvokabularen für japanische, aufgabenorientierte Dialoge im Rahmen des Grammatik-Formalismus der **HPSG** (Head Driven Phrase Structure Grammar, vgl. Pollard und Sag (1994)).

Vielfach wurde und wird versucht, diese Unterschiede durch übergreifende Auszeichnungsvokabulare auszugleichen. Für spezifische Vokabulare müssen entsprechende Abbildungen auf die übergreifenden Vokabulare formuliert werden. Die bereits erwähnten Auszeichnungsvokabulare des XCES-Standards und von GOLD sind Beispiele für derartige generalisierende, von spezifischen Vokabularen abstrahierende Standardisierungsbemühungen. Für das Japanische wird derzeit z.B. von Kawata (2001) ein Referenz-Tagset entwickelt, das eine ähnliche Rolle spielen soll.

```
(5.10) sekStruk:implicitReference subConcept sekStruk:models.
       sekStruk:constructionalImplicity subConceptOf
        sekStruk:implicitReference.
       sekStruk:existentialImplicity
        subConceptOf sekStruk:implicitReference.
       sekStruk:anaphoricImplicity subConceptOf sekStruk:implicitReference.
       sekStruk:argumentStructure subClassof sekStruk:models.
       sekStruk:missingArgument subConceptOf sekStruk:argumentStructure.
       sekStruk:missingArgumentConstructional subConceptOf
        sekStruk:missingArgument.
       sekStruk:words subConceptOf sekStruk:models.
       sekStruk:imperativeMarker subConceptOf sekStruk:words.
       sekStruk:imperativeMarker sekStruk2PrimStruk
        "element V-IMP { text }".
       sekStruk:missingArgument sekStruk2PrimStruk "element
        argumentStructure { attribute missingArgument { 'zero' }, ...}".
       sekStruk:missingArgumentConstructional sekStruk2PrimStruk
        "layerRel-included_B_in_A sekStruk:imperativeMarker".
       sekStruk:constructionalImplicity equal
        sekStruk:missingArgumentConstructional.
```

Beispiel 5.10: Beschreibung von konstruktionalen Implicit Frame Entities durch sekundäre Informationsstrukturierung

Die Problematik dieser Ansätze ist ihr übergreifender Anspruch. Generalisierung bedeutet zugleich den Verlust von Spezifik, d.h. im vorliegenden Fall von informationellen Ressourcen. Die vorliegende Arbeit möchte hingegen linguistische informationelle Ressourcen verknüpfen, *ohne* sie zu verändern; dort, wo eine unmittelbare Verknüpfung nicht möglich ist, sollen die Unterschiede und notwendige Überführungsschritte beschrieben werden.

Das Verfahren der sekundären Informationsstrukturierung ist mit zwei Arbeiten aus dem linguistischen Bereich vergleichbar, die mit unterschiedlichen Zielsetzungen die Rela-

tionierung konzeptueller Modelle zum Gegenstand haben. Shieber (1987) beschreibt den Mehrwert einer formalen, vergleichenden Analyse linguistischer Theorien: Welche Formalismen liegen den Theorien zu Grunde, welche Überführungsmöglichkeiten zwischen den Formalismen gibt es? Shieber beschreibt ein Verfahren, den Unterschied zwischen linguistischen Theorien durch ihre explizite Analyse zu erfassen. Die vorliegende Arbeit stellt eine Methodologie bereit, welche die Anwendung dieses Verfahrens mit empirischen Methoden ermöglicht. Jeder Modellierungsschritt fließt bei der Verbindung informationeller Ressourcen selbst wieder als informationelle Ressource ein. So werden sämtliche angewandten Methoden transparent und verknüpfbar. Die andere Arbeit ist in Hajicova und Kucerova (2002) geschildert. Die Autoren vergleichen verschiedene Annotationsvokabulare für Treebanks, vgl. auch Abschnitt 6.2.2. Die Vokabulare unterscheiden sich hinsichtlich theoretischer Rahmenwerke – phrasenstrukturell vs. dependenziell – und zu annotierender Sprachen – Englisch vs. Tschechisch. Das Ergebnis des Vergleichs liegt nicht in einem übergreifenden Vokabular für Treebanks. Die Autoren kommen vielmehr zu dem Schluss, dass die Erstellung übergreifender Annotationsvokabulare mit einem zu hohen Informationsverlust verbunden ist. Auch können die verschiedenen Vokabulare nicht direkt ineinander überführt werden. Notwendig ist ein mehrstufiges und deklarativ ausformuliertes Verfahren, welches die überführungsrelevanten Information explizit und operationalisierbar macht. Diese Vorgehensweise wird auch in der vorliegenden Arbeit angewandt.

Es ist davon auszugehen, dass es für viele konzeptuelle Modelle einen kleinsten gemeinsamen Nenner gibt. Z.B. wird die Beziehung *partOf* zwischen den Konzepten Morphem und Wort von Modellen für Baumdatenbanken oder linguistischen Theorien wohl kaum hinterfragt werden. Das in dieser Arbeit verfolgte Vorgehen nutzt deshalb linguistisch akzeptierte, theoriearme Konzeptualisierungen als Ausgangspunkt für die Erzeugung verschiedener primärer Informationsstrukturierungen. Der Aufbau sekundärer Informationsstrukturierungen kann zum einen als ein Wechselspiel von Hypothesenbildung, Verifikation und Hypothesenerweiterung bzw. -revidierung geschehen. Dies ist das in dieser Arbeit eingesetzte Vorgehen. Zum anderen können induktive Verfahren eingesetzt werden, um aus den annotierten Daten Beschreibungen für Regeln und Bedingungen zu generieren Doest (1999). Ein derartiges Verfahren existiert für primärdatenidentische, textuelle Daten noch nicht. Seine Erstellung liegt außerhalb des Skopus dieser Arbeit. Es bleibt jedoch die Option, es in den Prozess der linguistischen Hypothesenbildung und -verifikation mit einzubeziehen.

5.3. Kernpunkte des Kapitels

- Eine Anwendungsdomäne sekundärer Informationsstrukturierung sind linguistische Korpora mit textuellen Daten. Die Auszeichnung linguistischer Daten wird als Annotation bezeichnet. Drei Themenbereiche sind relevant für die texttechnologische Informationsmodellierung in diesem Bereich: die Repräsentation von Multidimensionalität, die Annotation tiefergehender Strukturen, sowie der Zugriff auf Annotationen in genereller versus sprach-, theorie- und domänenspezifischer Weise.

- Die Multidimensionalität der Sprache führt dazu, dass unterschiedliche Annotationen hinsichtlich Morphologie, Syntax, Argumentstruktur etc. nicht in einer Hierarchie repräsentierbar sind. Lösungsansätze dieser Problematik sind u.a. der Ansatz der Annotationsgraphen oder das Nite Object Model. Die Lösungsansätze beschreiben entweder eine allgemeine Graphenstruktur, die nur schwer für Suchanfragen etc. operationalisierbar ist. Oder sie erlauben die Definition multipler hierarchischer Strukturen. Dann sind sie allerdings an die definierten Strukturen gebunden; bereits annotierte Daten lassen sich nicht auf veränderte Strukturbeschreibungen beziehen. Die sekundäre Informationsstrukturierung erlaubt es hingegen, Annotationen in variabler Weise zu relationieren und verschiedene Strukturbeschreibungen in Form inhaltsseitiger Beschreibungen auf die gleichen Annotationen zu beziehen. Exemplarische Anwendungen sind die Unifikation der hinsichtlich verschiedener linguistischer Dimensionen annotierten, primärdatenidentischen Dokumentinstanzen, sowie die Erstellung korpusbasierter, multimodaler Lexika.

- Tiefergehende Strukturen sind für die linguistische Beschreibung relevante Strukturen, die sich nicht an einzelnen Annotationssegmenten festmachen lassen. Dies kann an einem zu undifferenzierten Symbolsystem im textuellen Datum liegen, oder die fokussierten Phänomene sind nur unter Bezug auf mehrere Segmente erfassbar. In letzterem Fall muss ein Suchraum für die Selektion der Segmente spezifiziert werden.

- Bestehende Verfahren zur Annotation tiefergehender Strukturen nutzen Suchräume nur in geringem Maße. Die sekundäre Informationsstrukturierung hingegen kann durch die in Kapitel 4 beschriebenen Selektionsmechanismen verschiedene Annotationssegmente in einzelnen oder mehreren, primärdatenidentischen

Dokumentinstanzen erfassen und in einer inhaltsbezogenen Beschreibung als ausgewählte tiefergehende Strukturen spezifizieren.

- Linguistische Annotationen nutzen allgemeine Annotationskategorien bzw. konzeptuelle Modelle oder sprach-, theorie- oder domänenspezifischen Varianten. Es wäre wünschenswert, flexibel auf diese verschiedenen Granularitätsebenen zugreifen zu können. Im letzten Kapitel der Arbeit wird demonstriert, wie dieser Zugriff durch sekundäre Informationsstrukturierung realisiert werden kann.

6. Beispielanwendungen

6.1. Motivation für die Auswahl der Phänomene

In diesem Kapitel werden linguistische Anwendungen der Methodologie der sekundären Informationsstrukturierung vorgestellt: die Interelationierung theoriespezifischer, im linguistischen Sinne syntaktischer Annotationen, sogenannter Treebanks einerseits und die Beschreibung koreferentieller Phänomene andererseits, sowie die Beziehung zwischen diesen Anwendungen. Die beiden Anwendungen machen Stärken und auch Schwächen der sekundären Informationsstrukturierung deutlich. Koreferenz ist ein im linguistischen Sinne linguistisch semantisches Phänomen. Aussagen mit dem Prädikat *sekStruk2primStruk* in der sekundären Informationsstrukturierung selektieren informationelle Ressourcen jedoch hauptsächlich anhand struktureller Eigenschaften. Linguistisch semantische Phänomene zeichnen sich nun dadurch aus, dass sie zwar nicht unabhängig, aber zumindest nur mittelbar auf strukturbezogene Beschreibungen abbildbar sind. Es stellt sich deshalb die Frage, inwiefern die sekundäre Informationsstrukturierung für die Modellierung linguistisch semantischer Phänomene überhaupt geeignet ist.

Treebanks hingegen scheinen ein ideales Anwendungsfeld für die sekundäre Informationsstrukturierung zu sein. Sie beinhalten Realisierungen von linguistisch syntaktischen Phänomenen. Die Abfolge von Konstituenten in Sätzen, die Anordnung von über- und untergeordneten Konstituenten in hierarchisch orientierten Syntaxmodellen lassen sich unmittelbar in Dokumentinstanzen realisieren, durch Linearisierung und Hierarchisierung der Elemente. Mit anderen Worten, bei Treebanks sind das logische und das physikalische Modell im Sinne von Abschnitt 2.1.2 besonders eng aufeinander bezogen. Es stellt sich allerdings die Frage, wie sich Anwendungen der sekundären Informationsstrukturierung auf linguistisch semantische Phänomene zu Anwendungen auf linguistisch syntaktische Phänomene verhalten. Konkret lautet die Frage, welches Verhältnis Beschreibungen linguistischer Strukturen, realisiert in Baumbanken, zu Beschreibungen koreferentieller, d.h. semantischer Beziehungen sprachlicher Einheiten haben. Nach der

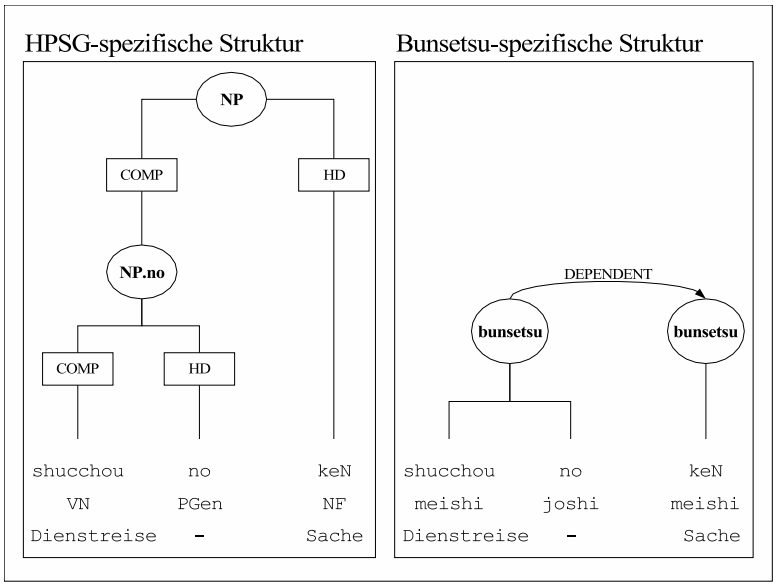

Abbildung 6.1.: Theoriespezifische, syntaktische Annotationen japanischer Daten

Vorstellung der beiden Anwendungen in Abschnitt 6.2 und Abschnitt 6.3 wird deshalb in Abschnitt 6.4 exemplifiziert, wie dieses Verhältnis durch die Anwendung eines übergreifenden, konzeptuellen Modells formal beschrieben werden kann.

6.2. Relationierung theoriespezifischer Annotationen von Treebanks

6.2.1. Motivation

Verschiedene theoriespezifische Annotationsformate für Treebanks wurden bereits in Abschnitt 5.2.2.2 diskutiert. Dabei wurden nicht nur Treebanks im engeren Sinne behandelt, d.h. Sammlungen syntaktisch annotierter Satzstrukturen. Auch semantische Annotationen von Sätzen wie im FrameNet-Projekt und in der Proposition Bank lassen sich unter dieser Form von Korpus subsumieren. Hier soll es nun um eine Thematik gehen, welche jede Art von Treebank betrifft: die Relationierung und Transformation theoriespezifischer Annotationen.

Welche Motivation gibt es, eine Relationierung theoriespezifischer Treebanks vor-

zunehmen? Diese Frage wird anhand von Abbildung 6.1 beantwortet. Die Abbildung enthält eine japanische Phrase „shucchou no ken“, ‚die Dienstreisenangelegenheit‘, welche auf zweierlei Weise syntaktisch gekennzeichnet ist. Die Visualisierung auf der linken Seite basiert auf einem Annotationsschema für HPSG, vgl. Kawata und Bartels (2000). Es wurde im **Verbmobil-Projekt** entwickelt, vgl. Wahlster (2000). Die Visualisierung auf der rechten Seite zeigt eine Segmentierung der Phrase in sogenannte **Bunsetsu**, vgl. Kurohashi und Nagao (2003). Bunsetsu sind nicht verschachtelte Phrasen, vergleichbar zu **Chunks** im Sinne von Abney (1991), die in einer dependenziellen Beziehung zueinander stehen.

Annotationen von Bunsetsu können auf Grund der einfachen syntaktischen Struktur mit verhältnismäßig geringem Aufwand und mit hoher Qualität automatisch erstellt werden. Entsprechende Verfahren beschreiben z.B. Kudo und Matsumoto (2002). Die automatische Erzeugung HPSG-spezifischer Annotationen hingegen ist zwar sehr aufwändig. Sie stellen jedoch komplexere syntaktische Strukturen bereit, welche für Anwendungsszenarien wie die automatische Übersetzung im erwähnten Verbmobil-Projekt von hohem Nutzen sind. Eine Relationierung der theoriespezifischen Treebanks stellt eine semiautomatische Transformation der Bunsetsu-Annotationen zu HPSG-spezifischen Annotationen in Aussicht. Auf diese Weise können die Vorteile beider theoriespezifischen Sichten kombiniert werden.

Die sekundäre Informationsstrukturierung und ihre Anwendung auf linguistische Korpora, die in Kapitel 5 vorgestellt wurde, ermöglichen diese Kombination. So sind die Kategorien `HD` und `COMP` differenzierbar anhand ihrer Position im Verhältnis zu `NP.no` bzw. `NP` Kategorien. Diese Unterschiede lassen sich als Varianten tiefergehender Strukturen, vgl. Abschnitt 5.2.2.1, auffassen, die durch verschiedene Suchräume, vgl. Abschnitt 5.2.2.3, operationalisierbar werden. So entspricht die Annotation der `HD` Kategorie, welche der Annotation der `NP` Kategorie unmittelbar untergeordnet ist, einer Variante der `bunsetsu` Kategorie. Im Folgenden werden bestehende Ansätze zur Transformation und Relationierung von Treebanks diskutiert und anschließend der Beitrag der sekundären Informationsstrukturierung zu dieser Thematik detaillierter vorgestellt.

6.2.2. Ansätze zur Transformation und Relationierung theoriespezifischer Treebanks

Bestehende Ansätze zur Transformation theoriespezifischer Treebanks lassen sich in zwei Gruppen aufteilen. Die eine Verfahrensweise, beschrieben z.B. von Xia und Palmer (2001), nutzt **Abbildungsalgorithmen**. Die Annotationen werden durch die sukzessive Anwendung von Regeln transformiert. Dieser Ansatz zeichnet sich durch sein prozedurales Vorgehen aus. In inkrementeller Weise werden Regeln auf die Ausgangsdaten angewandt, bis eine eindeutige Zielstruktur erkennbar ist. Die andere Gruppe von Ansätzen zielt auf die Relationierung theoriespezifischer Annotationen zu einem generellen, übergreifenden Modell ab. Ide und Romary (2003) greifen zu diesem Zweck auf die von Leech et al. (1996) beschriebenen EAGLES Empfehlungen zur Annotation zurück, Oepen et al. (2002) hingegen nutzen HPSG.

Beide Vorgehensweisen werden von Hajicova und Kucerova (2002) kritisch diskutiert. Ihrer Ansicht nach ist eine Abbildungsprozedur, welche Kategorien unmittelbar zueinander in Beziehung setzt, ungenügend. Kategorien sollten in mehreren, nachvollziehbaren Schritten relationiert werden. Die Schritte sollten Informationen aus verschiedenen linguistischen Beschreibungsebenen nutzen, z.B. hinsichtlich syntaktischer Strukturen und Wortarten. Das prozedurale Vorgehen von Xia und Palmer (2001) erschwert jedoch die Bewertung der einzelnen Schritte. Es bleibt unklar, welche Teilprozedur für ein besseres Ergebnis verändert werden muss.

Die Beschreibung eines generellen, übergreifenden Modells bedeutet hingegen einen Informationsverlust durch Generalisierung. Oepen et al. (2002) argumentieren zwar, dass das HPSG-Modell hinreichend detailierte Kategorien bereitstellen würde. HPSG-spezifische Annotationen seien die Grundlage für Transformationen zu verschiedenartigen, theoriespezifischen Annotationen. Allerdings hat der Detailiertheitsgrad von HPSG den Nachteil, den Aufwand für den Transformationsprozess bzw. die Relationierung möglicherweise unnötig zu erhöhen. Für die Transformation verschiedener Varianten dependenzieller Beschreibungen untereinander wären einfachere Strukturen ausreichend und leichter automatisch generierbar. Obwohl ein übergreifendes Modell also sinnvoll erscheint, müssen – zusätzlich – Kategorien zueinander relationiert werden. Dabei steht nicht die Generalisierung, sondern die Explikation der Unterschiede zwischen den Kategorien im Vordergrund.

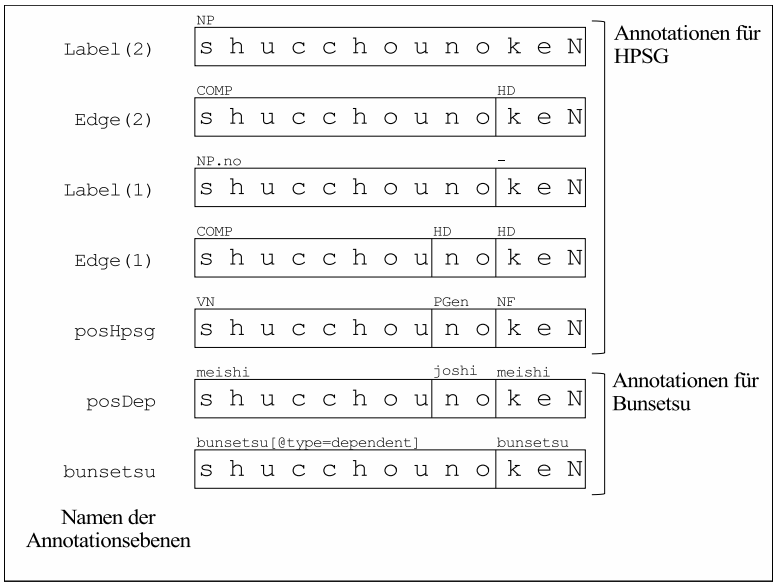

Abbildung 6.2.: Multiple Annotationen der theoriespezifischen Treebanks

6.2.3. Relationierung theoriespezifischer Treebanks mittels sekundärer Informationsstrukturierung

Die in dieser Arbeit vorgestellte Methodologie kann zur Erfüllung der von Hajicova und Kucerova (2002) formulierten Desiderata – nachvollziehbare Beschreibungen der Relationierungsschritte und eine Explikation von Unterschieden zwischen Kategorien – beitragen. Zwei Modellierungsverfahren, die in Kapitel 4 vorgestellt wurden, sind dabei von Bedeutung: die multiple Auszeichnung bzw. Annotation primärdatenidentischer, textueller Daten und die Beschreibung der Beziehungen zwischen den Annotationseinheiten mittels sekundärer Informationsstrukturierung. Exemplarische Annotationen sind in Abbildung 6.2 visualisiert.

Die multiple Auszeichnung der Daten erlaubt eine Beschreibung der Beziehungen zwischen theoriespezifischen Kategorien und deren empirischer Überprüfung, ohne auf die Beschränkungen des baumorientierten Formats von Baumbanken Rücksicht nehmen zu müssen. Trotzdem kann das baumorientierte Format genutzt werden, um in einzelnen Dokumentinstanzen mittels der Caterpillar-Ausdrücke hierarchiebezogene Selektionskriterien zu formulieren. Die diskutierten Verfahren zur Transformation von Baumbanken

hingegen nutzen nur die baumstrukturelle Sicht.

In Abbildung 6.2 wird deutlich, wie theoriespezifische, nominale Varianten von Annotationen aufeinander bezogen werden. Die Annotationen der Wortarten hinsichtlich HPSG - die Ebene `posHpsg` - stellen nominale Varianten der Wortartenannotationen für Bunsetsu dar, vgl. die Ebene `posDep`. Die Segmentierungen der textuellen Primärdaten sind identisch. Deshalb lassen sich in der sekundären Informationsstrukturierung Aussagen mit dem Prädikat *equal* machen, welche die Grundlage einer reversiblen Transformation der Annotationen bilden, vgl. Beispiel 6.1.

(6.1)
```
sekStruk:Hpsg-pos-PGen sekStruk2primStruk "element PGen { text }".
sekStruk:Bunsetsu-pos-joshi-no sekStruk2primStruk
 "element joshi { 'no' }".
sekStruk:Bunsetsu-pos-joshi-no equal sekStruk:Hpsg-pos-PGen.
```

Beispiel 6.1: Relationierung nominaler Varianten theoriespezifischer Treebankkategorien

Das `<PGen>` Element, Bestandteil der Wortartenkategorien für HPSG, wird für das Konzept sekStruk:Hpsg-pos-PGen selektiert. Das `<joshi>` Element, welches zu den Wortartenkategorien der Bunsetsu gehört, wird für das Konzept sekStruk:Bunsetsu-pos-joshi-no selektiert. Die Aussagen (sekStruk:Bunsetsu-pos-joshi-no *equal* sekStruk:Hpsg-pos-PGen.) schließlich macht explizit, dass es sich bei den selektierten Elementen um nominale Varianten handelt.

Mit der Methodologie der sekundären Informationsstrukturierung ist es auch möglich, „echte“ Varianten zueinander zu relationieren. Abbildung 6.3 visualisiert das Vorgehen, welches in Beispiel 6.2 in Tripel-Notation wiedergegeben ist.

Es werden zwei Modelle in der sekundären Informationsstrukturierung beschrieben, namentlich sekStruk:HPSG und sekStruk:Bunsetsu. Die Annotationskategorien der theoriespezifischen Treebanks werden auf modellspezifische Konzepte abgebildet. Im Beispiel sind dies die Annotationskategorien, die durch das `<hd>` Element für sekStruk:HPSG bzw. `<bunsetsu>` Element für sekStruk:Bunsetsu ausgedrückt werden. Die Varianten der theoriespezifischen Annotationskategorien werden nun als subordinierte Konzepte in der sekundären Informationsstrukturierung beschrieben. Für diese Varianten lassen sich dann mit dem Prädikat *equal* wieder Relationierungen der Treebanks vornehmen, etwa durch die Aussage (sekStruk:HeadSub *equal* sekStruk:BunsetsuGeneral.).

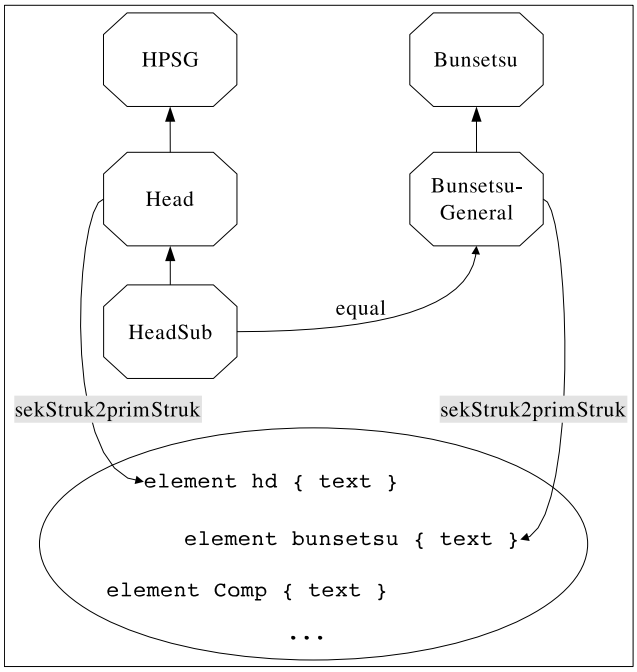

Abbildung 6.3.: Relationierung nicht nominaler Varianten theoriespezifischer Treebankkategorien

6.3. Modellierung von Koreferenz mit den Mitteln der sekundären Informationsstrukturierung

6.3.1. Abstrakte versus konkrete Merkmale bei der Modellierung von Koreferenz

Koreferentielle Phänomene wurden bereits in den Abschnitten 2.2.4.1 und 3.1.2 angerissen. Laut Poesio (2000) dient die Annotation von Koreferenz bzw. anaphorischen Beziehungen[1] der Markierung von textuellen Einheiten, die das gleiche Objekt denotieren. Beispiel 6.3 veranschaulicht koreferentielle Phänomene.

Das Beispiel beinhaltet die ersten Sätze des Romans „1984“ von George Orwell. Es wurde bereits in Abschnitt 3.1.2 eingeführt. Die koreferierenden Einheiten sind mit Indices versehen. Die Ausdrücke `Ministry of Truth`, `Minitrue` und das Pronomen `it` im

[1] Poesio trifft diese Unterscheidung zwischen korefentiellen und anaphorischen Beziehungen, welche jedoch für die Diskussion in diesem Abschnitt keine Relevanz besitzt.

```
(6.2) sekStruk:HPSG subConceptOf sekStruk:models.
      sekStruk:Bunsetsu subConceptOf sekStruk:models.
      sekStruk:Head subConceptOf sekStruk:HPSG.
      sekStruk:HeadSub subConceptOf sekStruk:Head.
      sekStruk:BunsetsuGeneral subConceptOf sekStruk:Bunsetsu.
      sekStruk:Head sekStruk2primStruk "element hd { text }".
      sekStruk:Bunsetsu sekStruk2primStruk "element bunsetsu { text }".
      sekStruk:HeadSub equal sekStruk:BunsetsuGeneral.
```

Beispiel 6.2: Tripel-Notation zu Abbildung 6.3

```
(6.3) [Ministry of Truth(i1)], - [Minitrue(i1)], in Newspeak - was
      startingly different from any other object in sight. [It(i1)]
      was an enourmous pyramidial structure of glittering white
      concrete, soaring up, terrace after terrace, 300 metres into
      the air. From where Winston stood [it(i2)] was just possible to
      read, picked out on its white face in elegant lettering,
      [the three slogans of the Party(i2)]:
      [War is peace
      Freedom is slavery
      Ignorance is strength.(i2)]
```

Beispiel 6.3: Beispiel für koreferentielle Phänomene

zweiten Satz beziehen sich auf das gleiche Objekt mit der Indexziffer `(i1)`. `Ministry of Truth` wird als **Antezedent** bezeichnet, auf welches sich ein **koreferrierender Ausdruck**, z.B. das Pronomen `it` mit der Indexziffer `(i1)` bezieht. Das Pronomen `it` mit der Indexziffer `(i2)` im dritten Satz bezieht sich auf den Antezedenten `the three slogans of the Party` und die Verbalisierung der Parteislogans: `War is peace Freedom is slavery Ignorance is strength`.

Koreferenz ist ein abstraktes, funktionsbezogenes Phänomen, d.h. es lassen sich keine konkreten, formbezogenen Merkmale sprachlicher Ausdrücke festmachen, die sie als koreferierend kennzeichen. Deshalb verwenden linguistische Modelle für Koreferenz **ab-**

strakte Merkmale, die auf **konkrete Merkmale** bezogen werden. Ein Beispiel für ein konkretes, morphosyntaktisches Merkmal im Englischen oder Deutschen ist die **Definitheit** von Nominalphrasen, ausgedrückt z.B. durch den Artikel `the` in `the three slogans of the Party`. Die Definitheit der Phrase macht deutlich, dass sich das zuvor erwähnte Pronomen `it` auf die Phrase bezieht.

Das Verhältnis abstrakter und formbezogener Merkmale ist jedoch nicht immer so eindeutig. Definitheit als eine morphosyntaktische Kategorie existiert z.B. in Sprachen wie dem Japanischen nicht. Es stellt sich deshalb für korpusbasierte Modelle die Frage, wie das Verhältnis abstrakter und konkreter Merkmale repräsentiert werden soll.

6.3.2. Gegenwärtige Ansätze zur korpusbasierten Modellierung abstrakter und konkreter Merkmale von Koreferenz

Im Hinblick auf korpusbasierte Modelle lässt sich die Modellierung von Koreferenz als Annotation tiefergehender Strukturen auffassen. In Abschnitt 5.2.2 wurden bereits Beispiele existierender Annotationsverfahren für tiefergehende Strukturen gegeben. Im Folgenden wird der Ansatz von Salmon-Alt und Romary (2004) vorgestellt, welcher speziell für die Annotation von Koreferenz entwickelt wurde.

Die Autoren greifen auf die von Ide und Romary (2003) entwickelte Methodologie zurück, deren Kern die Unterscheidung von **CAML** (Concrete Markup Languages) und **VAML** (Virtual Markup Languages) bildet. CAML beinhaltet jene Kategorien, die konkret z.B. für die morphosyntaktische Annotation einer Sprache geeignet sind. VAML beinhalten Kategorien, die zum Zwecke der Relationierung über CAML generalisieren oder eine Abstraktion darstellen. Die Beschreibung von Kategorien für die Annotation von Koreferenz von Salmon-Alt und Romary (2004) stellt eine derartige Abstraktion dar.

VAML und CAML bringen eine feste Verbindung abstrakter und konkreter Kategorien mit sich. In einem Top-Down ausgerichteten Verfahren werden aus Kategoriebeschreibungen in RDF Schema zunächst VAML in einer „abstrakten“ XML-Serialisierung[2] generiert, die über weitere Transformationsschritte der Erzeugung spezifischer CAML dienen. Z. B. wird die Kategorie Definitheit - als Teil von VAML - auf das morphosyntaktische Merkmal Definitheit - als Teil von CAML für Deutsch oder Englisch - abgebildet. Für

[2] Die XML-Serialisierung gibt die Baumstruktur der Dokumente, Attribut-Wert-Paare und Relationen zu anderen Knoten im Dokument durch die generischen Elemente `<struct>`, `<feat>` und `<rel>` wieder. Auf die textuellen Daten wird im Standoff Verfahren mittels XPointer-Ausdrücken verwiesen.

Sprachen wie das Japanische stellt sich nun die Frage, ob ein Top-Down Ansatz angemessen ist, da hier die Verbindung abstrakter und konkreter Beschreibungen von Definitheit strittig ist. Definitheit existiert nicht als morphosyntaktische Kategorie. Für das Japanische scheint es deshalb notwendig, die Verbindung abstrakter und konkreter Kategorien in annotierten Korpora empirisch zu überprüfen. Mit der Methododologie von VAML und CAML ist dieser empirische, Bottom-Up gerichtete Zugriff auf die Verbindung abstrakter und konkreter Merkmale jedoch nicht möglich.

In Sasaki und Witt (2004a) wurde beschrieben, wie mittels der multiplen Annotation primärdatenidentischer XML-Dokumentinstanzen die Modellierung von Definitheit im Japanischen empirisch, also Bottom-Up betrieben werden kann. Die sekundäre Informationsstrukturierung erlaubt es nun, derartige Beziehungen zwischen abstrakten und konkreten Merkmalen deklarativ in Form von Aussagen zu beschreiben. Im Folgenden soll ein Beispiel hierfür gegeben werden.

6.3.3. Relationierung abstrakter und konkreter Merkmale von Koreferenz mittels sekundärer Informationsstrukturierung

Das Beispiel für die Modellierung von Korefrenz mittels sekundärer Informationsstrukturierung greift auf das Korpus **Multext-East** zurück, vgl. Erjavec (2004). Das Korpus umfasst die englische Originalausgabe von „1984“ und Übersetzungen in verschiedene, hauptsächlich osteuropäische Sprachen. Die Annotation macht Gebrauch von der Dokumentgrammatik des bereits ofters behandelten Corpus Encoding Standard. Der im vorhergehenden Abschnitt vorgestellte Auschnitt aus „1984“ ist in Beispiel 6.4 wiedergegeben, eine Erweiterung von Beispiel 3.2.

Anhand der „konkreten“ Annotationen lassen sich verschiedene Formen der „abstrakten“ Koreferenz unterscheiden. So folgt das `<antecedent>` Element unmittelbar auf das `<reformulation>` Element innerhalb des gleichen `<s>` Elements. Die Abfolge des `<pronoun>` Elements und des `<antecendent>` Element im dritten `<s>` Element ist genau umgekehrt. Zudem bezieht sich dieses `<pronoun>` Element auf ein weiteres `<antecedent>` Element, welches nicht nur innerhalb eines `<s>` Elements steht, sondern selbst mehrere `<s>` Elemente umfasst.

Diese variierenden Beziehungen der `<antecedent>`, `<reformulation>`, `<prounoun>` und `<s>` Elemente drücken unterschiedliche Formen von Koreferenz aus. Die Beziehung des `<reformulation>` Elements zum `<antecedent>` Element im ersten `<s>` Element ist

(6.4)
```
<corpus>
<s><antecedent>The Ministry of Truth</antecedent> -
<reformulation>Minitrue</reformulation>, in Newspeak* - was
startlingly different from any other object in sight.</s>
<s><pronoun>It</pronoun> was an enormous pyramidal structure of
glittering white concrete, soaring up, terrace after terrace,
300 metres into the air.</s><s> From where Winston stood
<pronoun>it</pronoun> was just possible to read, picked out on its
white face in elegant lettering,
<antecedent>the three slogans of the Party:</antecedent>
<antecedent>
<s>War is peace</s>
<s>Freedom is slavery</s>
<s>Ignorance is strength</s></antecedent>.</s>
</corpus>
```

Beispiel 6.4: Koreferenz zwischen Elementen in Dokumentinstanzen

rückverweisend: Der Antezedent ist bereits zuvor genannt. Die Beziehung des `<pronoun>` Elements zu den folgenden `<antecedent>` Elementen ist vorverweisend, da der Antezedent erst noch eingeführt wird.

Mittels sekundärer Informationsstrukturierung werden diese unterschiedlichen Formen von Korefenz als Annotationen unterschiedlicher tiefergehender Strukturen und verschiedener Suchraumbeschränkungen beschrieben. Ausgangspunkt bildet eine generelle Dokumentgrammatik für Koreferenz, die in Beispiel 6.5 wiedergegeben ist.

Ein `<corpus>` Element besteht aus einer Folge von `<s>` Elementen. Diese enthalten ein `<antecedent>`, `<reformulation>` oder ein `<pronoun>` Element. Zugleich können sie Textknoten in beliebiger Abfolge enthalten, ausgedrückt durch das Schlüsselwort `mixed`. Die Tochterelemente von `<s>` sind zudem optional und können beliebig oft auftreten. Dies zeigt der Kleene-Star Operator `*` an. Das `<antecedent>` Element schließlich kann selbst `<s>` Elemente mit textuellem Inhalt enthalten, oder nur Text. In Abbildung 6.4 ist eine konzeptuelle Hierarchie einer sekundären Informationsstrukturierung visualisiert,

```
(6.5) start =
       element corpus {
         element s {
          mixed {
           element antecedent { text }
           | element reformulation { text }
           | element pronoun { text }
           | element antecedent { element s { text }+ }
         }*
       }+
     }
```

Beispiel 6.5: Generelle Dokumentgrammatik zur Beschreibung von Koreferenz

die diese allgemein gültige Deklaration des `<s>` Elements im Inhaltsmodell des `<corpus>` Elements für verschiedene Formen von Koreferenz selektiert und spezialisiert. Das `<s>` Element im Inhaltsmodell eines `<antecedent>` Elements hingegen wird nicht selektiert, es ist also - im Sinne der in Abschnitt 4.4.2.1 beschriebenen Konditionen - aus Sicht der sekundären Informationsstrukturierung nicht existent.

Die konzeptuelle Hierarchie unterscheidet Koreferenz hinsichtlich der Richtung, in welcher der Antezedent vom koreferierenden Ausdrück aus zu suchen ist, und hinsichtlich der Eigenschaften des Antezedenten. Die generellen Charakterisierungen von Koreferenz sind als abstrakt beschrieben, so dass bei einer konzeptbezogenen Validierung, vgl. Abschnitt 4.4.2.3, für ein untergeordnetes Konzept dokumentgrammatische Konstrukte bzw. Informationseinheiten in Dokumentinstanzen selektierbar sein müssen. Die Aussagen zur Selektion der dokumentgrammatischen Konstrukte sind in den Beispielen 6.6 und 6.7 wiedergegeben.

Für das Konzept Coref-general wird das `<s>` Element mit jenem Inhaltsmodell selektiert, welches in der gegebenen Dokumentgrammatik enthalten ist. Das subordinierte Konzept Coref-forward spezialisiert dieses Inhaltsmodell derart, dass auf ein `<pronoun>` Element mindestens ein `<antecedent>` Element folgen muss. Es kann sich dabei um einen nominalen Antezedenten handeln, z.B. `the three slogans of the Party`, oder

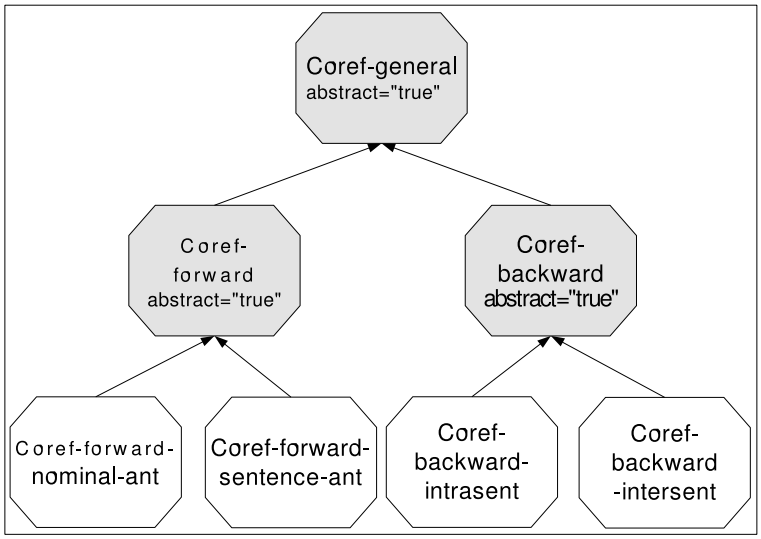

Abbildung 6.4.: Konzeptuelle Hierarchie koreferentieller Phänomene

um einen Antezedenten, der aus mehreren Sätzen besteht, also mehrere <s> Elemente enthält. Die subordinierten Konzepte Coref-forward-nominal-ant bzw. Coref-forward-sentence-ant machen diesen Unterschied explizit. Das dritte <s> Element in Beispiel 6.4 wird durch beide Konzepte selektiert. Dies ist ein Fall von Polysemie, vgl. Abschnitt 4.4.2.1: Verschiedene intensionale bzw. inhalsseitige Beschreibungen, Beschreibungen, also Aussagen in der sekundären Informationsstrukturierung, erfahren die gleiche extensionale bzw. ausdrucksseitige Interpretation, d.h. sie beziehen sich auf das gleiche dokumentgrammatische Konstrukt.

Die Spezialisierung der Inhaltsmodelle der <s> Elemente ist konform zu den Spezialisierungsregeln Regel 1 und Regel 6, vgl. Abschnitt 4.4.2.2. Dies trifft auch auf die Spezialisierungen in den Konzepten Coref-backward und Coref-backward-intersent zu, vgl. Beispiel 6.7. Coref-backward selektiert ein Inhaltsmodell für <s> Elemente, welches ein <reformulation> oder ein <pronoun> Element enthält. Coref-backward-intrasent spezialisiert dieses Inhaltsmodell konform zu Regel 6. Es darf nur ein <pronoun> Element enthalten sein. Das Konzept Coref-backward-intersent schließlich greift auf dokumentinstanzbezogene Selektion von Informationseinheiten mittels Pfadausdrücken zurück. Es werden jene <s> Elemente selektiert, in denen ein <pronoun> Element steht. Zudem muss im vorhergehenden <s> Element ein <antecedent> Element enthalten sein.

```
(6.6) Coref-general sekStruk2primStruk "element s { ... }".
      Coref-forward sekStruk2primStruk
       "element s { mixed { element pronoun { text },
        (element antecedent { text }
        | element antecedent { element s { text }+ })+ } }".
      Coref-forward-nominal-ant sekStruk2primStruk
       "element s { mixed { element pronoun { text },
        element antecedent { text },
        element antecedent { element s { text }+ }? } }".
      Coref-forward-sentence-ant sekStruk2primStruk
       "element s { mixed { element pronoun { text },
        element antecedent { text }?,
        element antecedent { element s { text }+ } } }".
```

Beispiel 6.6: Spezialisierung des <s> Elements für vorwärts gerichtete Koreferenz

6.4. Verbindung der Phänomenbeschreibungen durch ein übergreifendes konzeptuelles Modell

6.4.1. Bewertung der Modellierungen von Treebanks und Koreferenz

Obwohl die Modellierungen in diesem Kapitel nur exemplarischen Charakter haben, zeigen sie deutlich Stärken und auch Schwächen der sekundären Informationsstrukturierung. Die Relationierung theoriespezifischer Treebanks nutzt sicherlich die Stärken der Methodologie. Theoriespezifische Kategorien können durch alle Formen dokumentgrammatikbezogener und dokumentinstanzbezogener sekundärer Informationsstrukturierung zueinander in Beziehung gesetzt werden. Problematisch für diese Anwendung kann die Anzahl der notwendigen Konzepte in der sekundären Informationsstrukturierung sein, die notwendig ist, um die Treebanks aufeinander zu beziehen. Diesen Schwachpunkt teilt das Vorgehen allerdings mit den anderen diskutierten Ansätzen.

Die Modellierung von koreferentiellen Phänomenen mittels sekundärer Informationsstrukturierung ist sinnvoll, wenn die Dokumentinstanzen geeignete Annotationen enthalten. Die vorgestellten Beispiele segmentierten beispielsweise Sätze durch <s> Elemen-

```
(6.7) Coref-backward sekStruk2primStruk
       "element s { mixed { (element antecedent { text },
        element reformulation { text }) | element pronoun { text } } }".
      Coref-backward-intrasent sekStruk2primStruk
       "element s { mixed { element antecedent { text },
        element reformulation { text } } }".
      Coref-backward-intersent sekStruk2primStruk
       "pathEx: first 'pronoun' up left 's' first 'antecedent' ".
```

Beispiel 6.7: Spezialisierung des <s> Elements für rückwärts gerichtete Koreferenz, und Selektion in einzelnen Dokumentinstanzen mittels Pfadausdrücken

te. Auf diese Elemente konnte bei der Unterscheidung von Koreferenz innnerhalb von Sätzen versus Koreferenz zwischen Einheiten in verschiedenen Sätzen zurückgegriffen werden. Ein Nachteil der vorgestellten sekundären Informationsstrukturierung ist jedoch die Fokussierung auf Inhaltsmodelle. Nicht die Sätze bzw. <s> Elemente koreferieren, sondern die textuellen Einheiten, welche mittels <antecedent>, <pronoun> oder <reformulation> Elementen annotiert sind. Bis zu einem gewissen Grad lässt sich dieses Manko durch das Prädikat *componentOf* ausgleichen, vgl. Abschnitt 4.4.1.1. Und man kann die Elemente durch Pfadausdrücke selektieren, welche ihre koreferentiellen Eigenschaften beschreiben. Allerdings birgt dies die Gefahr einer sekundären Informationsstrukturierung, welche nur für eine abgeschlossene Menge von Dokumentinstanzen verifizierbar ist. Es bleibt dem Anwender der Methodologie überlassen, ob eine knotenzentrierte Beschreibung in der dokumentinstanzbezogenen Informationsstrukturierung für das zu modellierende Phänomen wirklich unerlässlich ist.

Die Beschreibung linguistisch semantischer Phänomene mittels sekundärer Informationsstrukturierung hat allerdings einen großen Vorteil. Die sekundäre Informationsstrukturierung selektiert Einheiten der primären Informationsstrukturierung anhand strukturbezogener Eigenschaften. So wird der Bezug linguistisch semantischer Phänomene zu strukturellen Eigenschaften deutlich. Als Tertium Comparationis zwischen Semantik und Struktur kann ein übergreifendes, konzeptuelles Modell dienen, auf welches sich sowohl die linguistisch semantischen Phänomene als auch die Strukturbeschreibungen beziehen

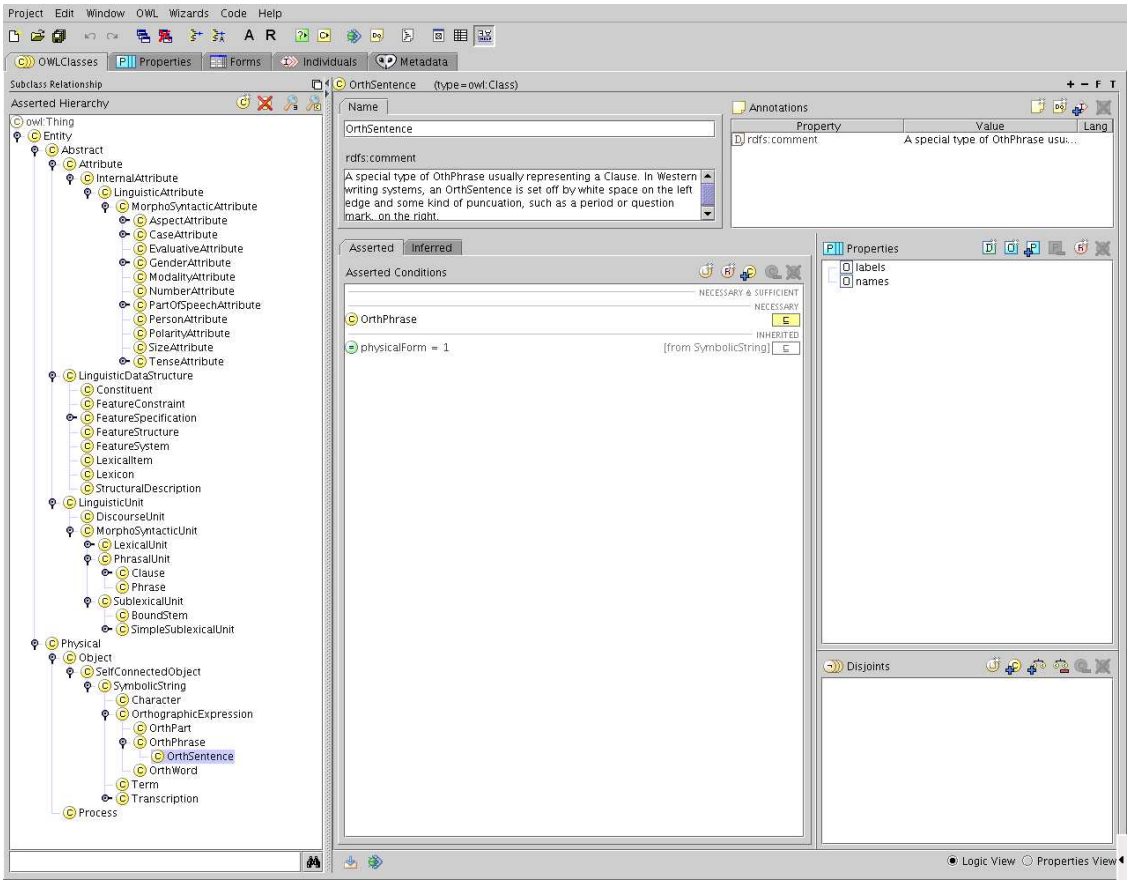

Abbildung 6.5.: Die Konzepthierarchie in GOLD, dargestellt im Ontologie-Editor Protege

lassen. Im Folgenden werden die entwickelten Beispiele für die Relationierung von Treebanks und die Beschreibung von Koreferenz zu solch einem übergreifenden Modell, der Ontologie GOLD, relationiert, abstrahiert und zu einander in Beziehung gesetzt.

6.4.2. Ausgangspunkt: Die Ontologie GOLD

Die Ontologie GOLD, vgl. Farrar et al. (2002), wurde bereits in Abschnitt 3.1.2 eingeführt. Einen Überblick über die Konzepthierarchie von GOLD, dargestellt im Ontologie-Editor Protege, gibt Abbildung 6.5. Gold wird in einer OWL-Repräsentation benutzt.

Wie in Abschnitt 3.1.2 bereits angesprochen, bietet GOLD ein Inventar von linguis-

tischen Kategorien, die als kleinster gemeinsamer Nenner linguistischer Analysen konzipiert sind. Der Schwerpunkt liegt bei GOLD in Kategorien, die für die linguistische Feldforschung wenig bekannter, oft bedrohter Sprachen geeignet sind. Da es sich dabei um typologisch sehr verschiedene Sprachen handelt, beinhaltet GOLD nicht nur eine Sammlung linguistisch weitgehend unumstrittener Kategorien, sondern generalisiert auch über diese Kategorien.

Zur Generalisierung dient dabei die Ontologie SUMO, vgl. Abschnitt 2.2.3.1. Die Konzepthierarchie von SUMO trennt Konzepte bzw. die Objekte, auf welche sie sich beziehen, in Abstract versus Physical. GOLD behält diese Trennung und basale Konzepte wie Attribute, Object oder Process bei. Spezifische Konzepte für die linguistische Beschreibung, die Abstract untergeordnet sind, betreffen z.B. linguistische Merkmale wie MorphoSyntacticAttribute oder linguistische Datenstrukturen, namentlich LinguisticDataStructure. Linguistische Konzepte, die Physical untergeordnet sind, fokussieren Objekte der Verschriftlichung von Sprache, wie das Konzept OrthographicExpression, dem das Konzept OrthSentence untergeordnet ist. Hier zeigt sich, wie GOLD die gleichen Objekte mit verschiedenen Konzepten beschreibt. So gibt es neben OrthSentence auch das Konzept PhrasalUnit, welches sich ebenfalls auf Sätze bezieht. Allerdings ist PhrasalUnit subordiniert zum Konzept LinguisticUnit, welches wiederum dem Konzept Abstract untergeordnet ist. Diese verschiedenen Beschreibungen von Sätzen – als Teil von Physical versus als Teil von Abstract – zeigen, wie GOLD zur Relationierung von abstrakten und konkreten Analyseeinheiten beitragen kann. Eine derartige Relationierung wurde in Abschnitt 6.3.3 für koreferentielle Phänomene beschrieben. Im Folgenden wird beschrieben, wie bei dieser Relationierung zugleich für Baumbanken relevante Konzepte integriert werden.

6.4.3. Verbindung von Baumbanken und korpusbasierten Modellen für Koreferenz durch sekundäre Informationsstrukturierung

Abbildung 6.6 stellt einen Ausschnitt aus der GOLD Ontologie, ein Konzept sekStruk:Sentence-antecedent aus der sekundären Informationsstrukturierung sowie die Verbindung beider mittels des in Abschnitt 4.4.1.5 beschriebenen Prädikats *sekStruk2conLevel* dar. Das Konzept sekStruk:Sentence-antecedent selektiert `<s>` Elemente, deren Inhaltsmodell mindestens ein `<antecedent>` Element und optional[3] ein `<reformulation>` bzw. ein `<pronoun>` Element enthält. Über eine Aussage mit

[3] Die Abfolge der Elemente ist beliebig, da der Interleave-Mechanismus & verwendet wird.

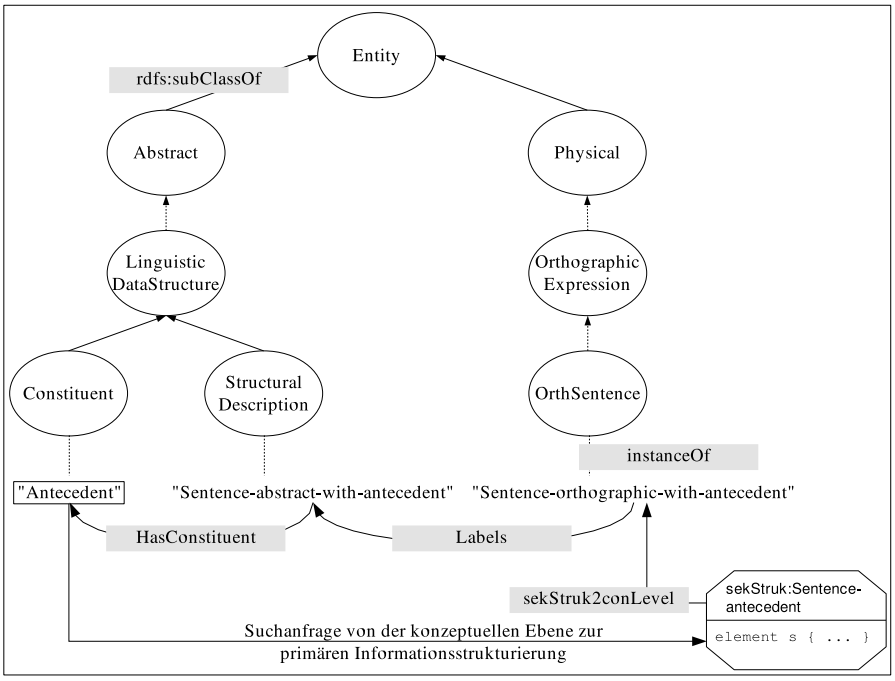

Abbildung 6.6.: Selektion von Einheiten der GOLD Ontologie aus der sekundären Informationsstrukturierung

dem Prädikat *sekStruk2conLevel* selektiert dieses Konzept die Instanz[4] Sentence-ortographic-with-antecedent aus GOLD, vgl. Beispiel 6.8. Der URI der Instanz `http://example.com/gold/gold.owl# Sentence-ortographic-with-antecedent` ist fiktiv.

Sentence-ortographic-with-antecedent ist eine Instanz zu dem Konzept OrthSentence in GOLD. Durch die Eigenschaft *Labels* ist Sentence-ortographic-with-antecedent mit der Instanz Sentence-abstract-with-antecedent verbunden. Sentence-abstract-with-antecedent ist wiederum eine Instanz zu dem Konzept StructuralDescription in GOLD. Sentence-abstract-with-antecedent hat die Eigenschaft *HasConstituent*, welche eine Verbindung zu der Instanz Antecedent schafft. Diese ist eine Instanz zu dem Konzept Constituent in GOLD.

[4]Die Trennlinie in GOLD zwischen Konzept und Instanz nutzt das Kriterium sprachübergreifend versus sprach- und domänenspezifisch. Sie wird hier beibehalten. Die Ausdrücke „Konzept“ und „Instanz“ haben allerdings keinen Bezug zu der Unterscheidung einer intensionalen Beschreibung versus einer extensionalen Beschreibung.

```
(6.8) sekStruk:Sentence-antecedent sekStruk2primStruk
       "element s { mixed { element antecedent
       { text } & element reformulation { text }? &
       element pronoun { text }? } }".
      Sentence-antecedent sekStruk2conLevel
       "http://example.com/gold/gold.owl#
       Sentence-ortographic-with-antecedent".
```

Beispiel 6.8: Verbindung von primärer Informationsstrukturierung und der konzeptuellen Ebene durch das Prädikat *sekStruk2conLevel*

Das Konzept OrthSentence ist mittelbar dem Konzept Physical untergeordnet, es bezieht sich also auf konkrete, physikalisch existente Objekte. Die Unterordnung wird durch Aussagen mit dem Prädikat *rdfs:subClassOf* beschrieben, welches in RDF Schema bzw. OWL vordefiniert ist. Die Konzepte StructuralDescription und Constituent sind mittelbar dem Konzept Abstract untergeordnet, sie beschreiben also abstrakte, nicht physikalisch erfassbare Objekte. Die Verbindung dieser konkreten und abstrakten Konzepte spiegelt die Verbindung der linguistischen Analyseebenen Syntax und Semantik wieder, welche hier exemplifiziert ist. Ein Satz als orthographische Einheit wird bezogen auf einen Satz als abstrakte Einheit. Die Eigenschaft *hasConstituent* schafft eine Verbindung dieser abstrakten Einheit zu einer linguistsch syntaktischen Analyse, welche für Baumbanken von Relevanz ist. Die Verbindung zur Koreferenz geschieht innerhalb der annotierten Daten bwz. der zu Grunde liegenden Dokumentgrammatik: Das Inhaltsmodell des selektierten `<s>` Elements muss ein `<antecedent>` Element enthalten. Um weitere Aspekte von Koreferenz zu modellieren, können zusätzlich Konzepte in der sekundären Informationsstrukturierung wie Sentence-ortographic-with-pronoun und Sentence-orthographic-with-reformulation erzeugt werden. Sie werden dann ebenfalls mittels *sekStruk2conLevel* auf den Satz als orthographische Einheit bezogen, der dann innerhalb von GOLD auf die beschriebene Weise eine Verbindung zu linguistisch syntaktischen Kategorien und Eigenschaften herstellt.

Als Anwendungsszenario der beschriebenen, exemplarischen Relationierung von sekundärer Informationsstrukturierung für Baumbanken, koreferentielle Phänomene einer-

seits und GOLD andererseits kann eine Suchanfrage gelten, die von den beschriebenen Beziehungen zwischen Konzepten in GOLD ausgeht und zur primären Informationsstrukturierung hinführt. Wie demonstriert wurde, sind Konzepte für abstrakte Objekte Abstract mit Konzepten für konkrete, physikalische Objekte Physical verknüpft. Eine Suchanfrage kann also als Eingabe die abstrakten Konzepte in GOLD nehmen, z.B. die Instanz Antecedent zum Konzept Constituent in GOLD. Als Ergebnis liefert die Anfrage Instanzen zu Konzepten in GOLD, die Physical untergeordnet sind, u.a. Sentence-ortographic-with-antecedent. Da diese Instanz in GOLD in der sekundären Informationsstrukturierung für das Konzept sekStruk:Sentence-antecedent selektiert ist, kann sie selbst nun als Eingabe einer Suchanfrage dienen. Die Suchprozedur ist in Abschnitt 4.4.2.3 beschrieben. Als Ergebnis werden nun die Deklaration des <s> Elements bzw. die entsprechenden Elementknoten in Dokumentinstanzen geliefert. Die Suchanfragen erlauben also eine empirische Überprüfung der in GOLD vorgegebenen Konzepte und ihrer Eigenschaften.

6.5. Kernpunkte des Kapitels

- Die in dieser Arbeit entwickelte Methodologie wird an Baumbanken, d.h. Korpora mit – im linguistischen Sinne – syntaktischen Phänomenen einerseits, und zur Modellierung von Koreferenz andererseits exemplifiziert. Die Unterschiedlichkeit der Phänomene zeigt Stärken und Schwächen der Methodologie.

- Die Stärken der Methodologie offenbaren sich bei der Relationierung und Transformation theoriespezifischer Baumbanken. Diese Aufgabenstellung ist motiviert durch den Wunsch, die Vorteile der spezifischen Baumbanken miteinander kombinieren zu können.

- Bisherige Ansätze zur Relationierung und Transformation von Baumbanken zielen zum einen auf eine Generalisierung der theoriespezifischen Kategorien ab. Dies führt zum Verlust von Informationen. Zum anderen beschreiben sie Transformationsprozeduren, die keinen deklarativen Zugang zum Verhältnis der Kategorien bieten. Die sekundäre Informationsstrukturierung erlaubt sowohl die Relationierung zu einem generellen, theorieunspezifischen Modell, als auch die deklarative Beschreibung von Beziehungen zwischen Baumbanken.

- Koreferenz ist ein – im linguistischen Sinne – semantisches, abstraktes Phänomen. Da es keinen unmittelbaren Bezug zu konkreten, syntaktischen Annotationen hat, greifen die Selektionsmechanismen der sekundären Informationsstrukturierung hier nicht. Dennoch erscheint die Modellierung von Koreferenz mit dieser Methodologie sinnvoll, da so die Beziehung abstrakter und konkreter linguistischer Merkmale einer empirischen Überprüfung zugänglich wird.

- Die Verbindung der beiden Phänomenbeschreibungen erfolgt über ein Modell auf der konzeptuellen Ebene. Hierzu dient die Ontologie GOLD, welche theorie- und sprachspezifische Kategorien in das übergreifende Modell SUMO integriert. Suchanfragen können von der konzeptuellen Ebene, d.h. von GOLD bzw. SUMO ausgehen, und als Ergebnis informationelle Ressourcen aus der primären Informationsstrukturierung liefern. Die in der konzeptuellen Ebene postulierten Beziehungen zwischen Baumbanken und koreferentiellen Modellierungen werden auf diese Weise einer empirischen Prüfung in annotierten Korpora zugänglich.

7. Resümee und Ausblick

Die vorliegende Arbeit hat eine Methodologie zur Verbindung informationeller Ressourcen vorgestellt. Das Verfahren der texttechnologischen Informationsmodellierung, auf dem diese Methodologie beruht, geht von einer engen Beziehung zwischen den Aufgaben einer logisch-formalen, algorithmisierbaren Modellierung auf der einen und Formaten zur physikalischen Repräsentation informationeller Ressourcen auf der anderen Seite aus. Die zentrale Aufgabe der Modellierung von Dokumentgrammatiken bzw. ausgezeichneten Dokumenten besteht in der Beschreibung von Regeln und Bedingungen, welche die Regelanwendung weiter einschränken. Die zentrale Modellierungsaufgabe im Rahmen der konzeptuellen Ebene besteht demgegenüber in der Beschreibung von Konzepten einer Konzepthierarchie und ihrer interkonzeptuellen Beziehungen. Der Kern der in der vorliegenden Arbeit entwickelten Methodologie der sekundären Informationsstrukturierung ist ein Format, welches diese Aufgaben sowohl im physikalisch-implementationsnahen Sinne, als auch im logisch-formalen, algorithmisierbaren Sinne explizit macht. Mit diesem Format wird es möglich, die ausdrucksseitigen Beschreibungen, d.h. die standardisierten Formate bzw. die in ihnen repräsentierten informationellen Ressourcen, aufeinander zu beziehen.

Ein Beispiel, das ein Defizit der vorgestellten Methodologie aufweist, ist mit ihrer Anwendung auf die Modellierung koreferentieller Phänomene gegeben. Da es sich um – im linguistischen Sinne – semantische Phänomene handelt, haben sie einen geringen Bezug zu sprachlichen, d.h. annotierbaren Strukturen. Die sekundäre Informationsstrukturierung nutzt jedoch in ihrer gegenwärtigen Form nahezu ausschließlich strukturbezogene Selektionsmechanismen, die auf dokumentgrammatische Konstrukte und Informationseinheiten zurückgreifen.

Eine Möglichkeit zur Überwindung dieses Defizits besteht darin, derartige abstrakte Phänomene in Bezug auf konkrete, im linguistischen Sinne syntaktische Auszeichnungen zu relationieren. Voraussetzung dafür ist jedoch, dass Auszeichnungen mit reichhaltigen Strukturen überhaupt zugänglich sind. Ansonsten muss eine aufwändige sekundäre In-

formationsstrukturierung vorgenommen werden, die eine Vielzahl komplexer Regeln und Bedingungen aus der primären Informationsstrukturierung zu selektieren hat.

Es gibt verschiedene Auswege aus diesem Dilemma, die in weiterführenden, an die vorliegende Arbeit anknüpfenden Forschungen beschritten werden sollen. Sie gehen zum einen von Eigenschaften der konzeptuellen Ebene und zum anderen von Verarbeitungsprozessen in der primären Informationsstrukturierung aus. Auf der konzeptuellen Ebene gibt es neben der Konzepthierarchie und den interkonzeptuellen Beziehungen weitere Modellierungsinventarien, die in die sekundäre Informationsstrukturierung Eingang finden sollen. Ein Beispiel hierfür bildet die Möglichkeit, die Disjunktion von Konzepten bzw. ihrer Instanzen festzulegen. Sie lässt sich beinahe unmittelbar als ein Mechanismus zur Selektion informationeller Ressourcen der primären Informationsstrukturierung einsetzen: Zwei als disjunkt beschriebene Konzepte der Ebene der sekundären Informationsstrukturierung dürfen nicht die gleiche Informationseinheit bzw. das gleiche Konstrukt in der jeweils betrachteten Dokumentgrammatik selektieren. Formate wie die OWL sind in diesem Zusammenhang dahingehend zu untersuchen, welche Bestandteile ihres Modellierungsinventars für eine inhaltsseitige Beschreibung als Bestandteil der sekundären Informationsstrukturierung in Frage kommen.

Im Rahmen dieser Arbeit bestand die primäre Informationsstrukturierung aus zwei Mengen: aus Strukturen, d.h. Informationseinheiten in Dokumentinstanzen, sowie aus Strukturbeschreibungen, d.h. Regeln in einer Dokumentgrammatik und Bedingungen. In Ergänzung zu dieser strukturbezogenen Sicht soll in weiterführenden Forschungen ein **prozessorientierter Zugang zur primären Informationsstrukturierung** treten. Ausdrucksseitige Beschreibungen werden dann nicht mehr (nur) anhand ihrer explizit repräsentierten Strukturen bzw. Strukturbeschreibungen selektiert, sondern unter Bezug auf einen zu realisierenden Verarbeitungsprozess.

Insbesondere für die Modellierung linguistischer Phänomene erscheinen prozedurale, prozessorientierte Selektionsmechanismen für die primäre Informationsstrukturierung sinnvoll. Hierfür existieren mehrere Beispiele aus unterschiedlichen Phänomenbereichen. Dies betrifft etwa die Koordination syntaktischer Einheiten, die als ein prozedurales Phänomen beschrieben werden kann, vgl. Lobin (1993). Auch die in dieser Arbeit strukturell analysierten, koreferentiellen Beziehungen lassen sich anhand von Suchprozeduren beschreiben. Die Suchprozedur wird von einem koreferierenden Ausdruck ausgelöst und führt zum Antezedenten. Schließlich kann die Rezeption eines Texts prozesshaft dargestellt werden, wobei textuelle Einheiten Suchprozesse nach außertextuellen Infor-

mationen, z.B. in Form von Komponenten des Vorwissens des jeweiligen Rezipienten hervorrufen, vgl. Schnotz (1994).

Eine sekundäre Informationsstrukturierung, die nicht nur Strukturen, sondern auch Prozesse als Selektionsmechanismen für die primäre Informationsstrukturierung nutzt, erlaubt dann die Integration von Verarbeitungskomponenten, welche die Prozesse realisieren. Der Mehrwert einer derartigen sekundären Informationsstrukturierung liegt in der Verbindung strukturbezogener Beschreibungen mit prozessorientierten Beschreibungen in einem inhaltsseitigen, deklarativen Format. Der Entwicklung eines solchen Formats zur Integration prozessorientierter Aspekte der sekundären Informationsstrukturierung sind zukünftige Arbeiten gewidmet.

Teil III.

Anhang

A. Typographische Kennzeichnungen

In der vorliegenden Arbeit werden informationelle Ressourcen durch typographische Kennzeichnungen differenziert. Für informationelle Ressourcen der primären Informationsstrukturierung kommt eine Typewriter-Schrift zum Einsatz. Elemente in XML werden zusätzlich durch spizte Klammern markiert. Ein Beispiel:

> Der Name des Attributes lautet `type`. Es wird für das `<link>` Element verwendet.

Für informationelle Ressourcen der konzeptuellen Ebene kommt eine serifenlose Schrift zum Einsatz. Prädikate bzw. Beschreibungen von Eigenschaften der Konzepte werden zudem kursiv gesetzt. Aussagen werden in runden Klammern geschrieben, vor der schließenden Klammer steht ein Punkt. Ein Beispiel sieht folgendermaßen aus:

> Die Aussage lautet (Physical *rdfs:subClassOf* Entity)..

Die sekundäre Informationsstrukturierung umfasst Aussagen, die zum einen aus Konzepte und Eigenschaftsbeschreibungen bzw. Prädikate bestehen. Die typographischen Konventionen ihrer Darstellung folgen denen der konzeptuellen Ebene. Zum anderen enthalten die Aussagen informationelle Ressourcen der primären Informationsstrukturierung. Ihre typographischen Merkmale entsprechen denen der primären Informationsstrukturierung. Ein Beispiel:

> Die Verbindung von konzeptueller Ebene und primärer Informationsstrukturierung wird durch die Aussage (sekStruk:Sentence *sekStruk2primStruk* "`element s { text }` ").realisiert.

In abgesetzten Passagen wird auf Grund der besseren Lesbarkeit durchgehend eine Typewriter-Schrift benutzt, z.B.

> Die Aussagen zur Verknüpfung von HPSG und Bunsetsu lauten:

```
sekStruk:Hpsg-pos-PGen sekStruk2primStruk "element PGen { text }".
sekStruk:Bunsetsu-pos-joshi-no sekStruk2primStruk
 "element joshi { 'no' }".
sekStruk:Bunsetsu-pos-joshi-no equal sekStruk:Hpsg-pos-PGen.
```

Grundlegende Begriffe und feststehende Terminologie werden durch Fettdruck wiedergegeben. Erklärungen zu Akronymen oder ergänzende Angaben stehen in Klammern. Ein Beispiel:

> Informationsanreicherung von Dokumenten wird unter Rückgriff auf **Auszeichnungssprachen** wie **XML** (eXtensible Markup Language) realisiert.

B. Glossar der wichtigsten Begriffe

Die folgende Liste stellt die wichtigsten Begriffe vor, die in der vorliegenden Arbeit zur Anwendung kommen. Dabei wird jeweils auf den Abschnitt verwiesen, der eine nähere Erläuterung des Begriffs beinhaltet.

Annotation (5.1) Auszeichnung linguistischer, – teilweise textueller – Daten mit zusätzlichen Informationen.

Annotationsebene (5.1) Eine Ebene der Annotation linguistischer Informationen, z.B. hinsichtlich Morphologie, Syntax etc. Viele Annotationsebenen lassen sich nicht in einer hierarchischen Struktur wiedergeben, d.h. nicht in einer XML-Dokumentinstanz repräsentieren.

Argument Vgl. **Aussage**.

Attribut (2.2.1.2) Eine Informationseinheit in XML. Attribute sind einem **Element** als ungeordnete Menge zugeordnet. Ein Attribut ist gekennzeichnet durch einen Namen, z.B. `type`, und einen Wert, z.B. `gloss`.

ausdrucksseitig Vgl. **Extension**.

Aussage (2.2.3.2) Eine logische Einheit, die mindestens ein Prädikat und n Argumente umfasst. Vgl. **Prädikat**.

Auszeichnungsebene (4.4.1.3) Eine XML-Dokumentinstanz, vgl. **Annotationsebene**.

Auszeichnungssprachen (2.2.1.1) Standardisierte Formate zur Auszeichnung von – zumeist textuellen – Daten. Die Auszeichnung kann dokumentzentriert erfolgen, d.h. auf verschiedenen, in hierarchischer Ordnung zueinander stehenden Ebenen, oder datenzentriert, d.h. die Abfolge und Hierarchisierung der Daten ist unerheblich.

Auszeichnungsvokabular (2.2.1.4) Eine abgeschlossene Menge von Elementnamen und Attributnamen in XML. Auszeichnungsvokabulare werden in Dokumentgrammatiken definiert.

Bedeutungsbeschreibung für Auszeichnung (3.3) Bezeichnet Ansätze, welche die Bedeutung von informationellen Ressourcen der primären Informationsstrukturierung beschreiben. Die Verfahren gehen zumeist Bottom-Up vor, von den Auszeichnungen zur Bedeutungsbeschreibung. Vgl. **semantische Auszeichnung**.

Bedingung (2.2.1.3) Wird zur Beschreibung von Eigenschaften verwendet, die für Dokumentinstanzen gelten. Vgl. **Regel**. Bedingungen in einzelnen Dokumentinstanzen können mit Hilfe von Pfadbeschreibungssprachen überprüft werden. Eine Bedingung ist für eine Informationseinheit erfüllt, wenn der beschriebene Pfad von dieser Informationseinheit aus realisierbar ist. Bedingungen in mehrfach ausgezeichneten, primärdatenidentischen Dokumentinstanzen können unter Rückgriff auf Prädikate zur Beschreibung von Beziehungen zwischen den Dokumentinstanzen überprüft werden. Vgl. **Selektion**.

datenzentiert (2.1.2) vgl. **Auszeichnungssprachen**.

Dokumentgrammatik (2.2.1.3) Beinhaltet Regeln, die eine Dokumentklasse beschreiben. Die Regeln dienen der Deklaration von Elementen und Attributen, d.h. ihrer Namen, ihres Status und – bei Elementen – ihrer Abfolge bzw. Hierarchisierung. Der Aufbau eines Elements, also die erlaubten Attribute und enthaltenen Elemente, wird Inhaltsmodell genannt. Vgl. **Schemasprachen**.

dokumentgrammatikbezogene sekundäre Informationsstrukturierung (4.4.1.1) sekundäre Informationsstrukturierung, bei der dokumentgrammatische Konstrukte selektiert werden.

Dokumentinstanz (2.2.1.2) Ein Dokument im Sinne des XML Standards. Optional ist einer Dokumentinstanz eine Dokumentgrammatik zugeordnet. Vgl. **Informationseinheit**.

dokumentinstanzbezogene sekundäre Informationsstrukturierung (4.4.1.2, 4.4.1.3) sekundäre Informationsstrukturierung, bei der Informationseinheiten in Dokumentinstanzen selektiert werden.

Dokumentklasse (2.2.1.1) Dokumente „an sich“. Der Aufbau einer Dokumentklasse wird beschrieben durch eine Dokumentgrammatik. Die Instanziierung der Dokumentklasse, d.h. der Regeln der Grammatik, heißt Dokumentinstanz.

dokumentzentiert (2.1.2) vgl. **Auszeichnungssprachen**.

Informationseinheit (2.2.1.2) Informationeller Bestandteil von Dokumentinstanzen in XML. Beispiele für Informationseinheiten sind Elemente, Attribute, Namensräume.

Extension (4.4.2.1) ausdrucksseitige Beschreibung einer Menge von Objekten, d.h. anhand ihrer Zugehörigkeit zu der Menge. Vgl. **Intension**.

Element (2.2.1.2) Eine Informationseinheit in XML. Elemente konstituieren die Hierarchie in einer Dokumentinstanz in XML.

informationelle Ressource (2.1.2) Beispiele für informationelle Ressourcen sind Texte und konzeptuelle Modelle, vgl. **textuelle Informationsmodellierung**, **konzeptuelle Ebene**. Die Eigenschaften informationeller Ressourcen sind charakterisiert durch das Verfahren der **texttechnologischen Informationsmodellierung**.

Inhaltsmodell (2.2.1.3) Vgl. Dokumentgrammatik.

inhaltsseitig Vgl. **Intension**.

Intension (4.4.2.1) inhaltsseitige Beschreibung einer Menge von Objekten, d.h. anhand ihrer Eigenschaften. Vgl. **Extension**.

Konzept (2.2.3.2) Unäres Prädikat, dessen Argument der Name des Konzepts ist.

Konzepthierarchie (2.2.3.2) Hierarchische Relationierung von Konzepten in eine Subsumptions- bzw. Vererbungsrelation. Untergeordnete Konzepte subsumieren übergeordnete Konzepte, d.h. sie besitzen die gleichen und mindestens eine weitere Eigenschaft, und sie haben eine geringere oder die gleiche Extension. Vgl. **Intension** und **Extension**.

konzeptionelles Modell (2.1.2) Adäquate, aber nicht unmittelbar algorithmisierbare Beschreibung eines Untersuchungsgegenstandes und seiner Eigenschaften. Vgl. **physikalisches Modell**, **logisches Modell**.

Konzeptuelle Ebene (2.2.3.2) Ebene der konzeptuellen Modellierung. Modelliert werden konzeptuelle Modelle, d.h. Beschreibungen von Konzepten anhand ihrer Eigenschaften für ausgewähle Domänen. Die Beschreibungen sind abstrakt in dem Sinne, dass sie nicht auf „konkrete“ informationelle Ressourcen wie z.B. ausgezeichnte Dokumente bezogen werden. Vgl. **primäre Informationsstrukturierung**, **sekundäre Informationsstrukturierung**.

logisches Modell (2.1.2) formale, algorithmisierbare Beschreibung eines Gegenstandes und seiner Eigenschaften. Vgl. **physikalisches Modell**, **konzeptionelles Modell**.

Namensraum (2.2.1.2) Eine Informationseinheit in XML. Dient der Separierung von Auszeichnungsvokabularen.

Pfadbeschreibungssprache (4.4.1.2) Dient der Beschreibung von Navigationen – Pfaden – innerhalb von Dokumentinstanzen. Vgl. **Selektion**.

physikalisches Modell (2.1.2) Implementationsnahe Beschreibung eines Gegenstandes und seiner Eigenschaften. Vgl. **konzeptionelles Modell**, **logisches Modell**.

Prädikat (2.2.3.2) Fundamentaler Bestandteil einer Aussage. Stellt Eigenschaften von und Beziehungen zwischen Objekten her, diese werden durch Argumente wiedergegeben. Die Stelligkeit eines Prädikats beschreibt die Zahl der Argumente. Für die Prädikate der sekundären Informationsstrukturierung ist ihre Stelligkeit in der Liste in Abschnitt 4.4.1.6 auf Seite 113 zusammengefasst.

Primärdatenidentische Dokumentinstanzen (4.4.1.3) Dokumentinstanzen, denen das gleiche textuelle Datum zu Grunde liegt. Selektionskriterien für Informationseinheiten in verschiedenen Dokumentinstanzen werden durch Prädikate beschrieben, welche Beziehungen zwischen den Informationseinheiten ausdrücken.

Primäre Informationsstrukturierung (2.2.1) Ebene der ausgezeichneten, textuellen Dokumente. Vgl. **konzeptuelle Ebene**, **sekundäre Informationsstrukturierung**.

Regel (2.2.1.3) Wird in Dokumentgrammatiken bzw. Schemasprachen zur Beschreibung von Eigenschaften verwendet, die für eine Dokumentklasse gelten. In der Taxono-

mie von Murata et al. (2001) gibt es 4 verschiedene Regeltypen: local, restrained-competition, single-type, regular. Vgl. **Bedingung**, **Selektion**.

Schemasprache (2.2.1.3) Beschreibt zumeist Eigenschaften einer Dokumentklasse mittels Regeln. In diesem engeren Sinne ist der Begriff „Schemasprache“ synonym zum Begriff „Dokumentgrammatik“. Schemasprachen im weiteren Sinne nutzen in Ergänzung oder alternativ zu Regeln Bedingungen, um Dokumentinstanzen zu charakterisieren. Beispiele für Schemasprachen sind XML-DTDs, RELAX NG und XML Schema.

Sekundäre Informationsstrukturierung Im engeren Sinne (2.2.4) die Beschreibung von Beziehungen zwischen dokumentgramamtischen Konstrukten. Im weiteren, dieser Arbeit zu Grunde liegenden Sinne (4) die Ebene der Verknüpfung von informationellen Ressourcen. Vgl. **Selektion**.

Selektion (4.4.1.1, 4.4.1.2, 4.4.1.3, 4.4.1.5) Mechanismus zur Verküpfung informationeller Ressourcen in der sekundären Informationsstrukturierung. Selektiert werden Konzepte oder Eigenschaften der konzeptuellen Ebene, oder informationelle Ressourcen der primären Informationsstrukturierung. In Dokumentgrammatiken werden Bestandteile von Regeln, in Dokumentinstanzen Mengen von Informationseinheiten selektiert. Das Selektionskriterium in Dokumentinstanzen ist die Erfüllbarkeit von Bedingungen, welche z.B. durch eine Pfadbeschreibungssprache wiedergegeben sind. Vgl. **Regeln**, **Bedingungen**, **Pfadbeschreibungssprache**, **primärdatenidentische Dokumentinstanzen**.

Semantische Auszeichnung (3.2) Bezeichnet Ansätze, die informationelle Ressourcen der konzeptuellen Ebene auf informationelle Ressourcen der primären Informationsstrukturierung beziehen. Die Verfahren gehen zumeist Top-Down vor, von der konzeptuellen Ebene zur primären Informationsstrukturierung. Vgl. **Bedeutungsbeschreibung für Auszeichnung**.

Spezialisierungsregel (4.4.2.2) Regeln, die das Verhältnis von Spezialisierungen dokumentgrammatischer Konstrukte bei der dokumentgrammatikbezogenen sekundären Informationsstrukturierung beschreiben.

Suchraum (5.2.2.1) Differenzierungskriterium für verschiedene Formen tiefergehender Strukturierung. Suchräume können dokumentgrammatische Konstrukte, einzelne

Dokumentinstanzen oder mehrere primärdatenidentische Dokumentinstanzen umfassen. Sie werden festgelegt durch Selektionen in der sekundären Informationsstrukturierung.

textuelle Informationsmodellierung (2.1.2) Die Modellierung der informationellen Ressource „Text“ mittels Auszeichnungssprachen.

texttechnologische Informationsmodellierung (2.1.2) Die Modellierung informationeller, vorallem textueller und konzeptueller Ressourcen unter Rückgriff auf standardisierte Formate. Die Ebenen der logischen und physikalischen Modellierung fallen dabei oft zusammen. Vgl. **textuelle Informationsmodellierung** und **konzeptuelle Ebene**.

tiefergehende Strukturen (5.2.2.1) (Linguistische) Annotationsstrukturen, die sich nicht unmittelbar an Einheiten der textuellen Oberfläche offenbaren. Vgl. **Suchraum**.

tiefergehende Strukturierung (5.2.2.1) Die Festlegung einer tieferliegenden Struktur anhand ausgewählter Suchräume.

URI (2.2.1.2) Standard zur eindeutigen Identifikation informationeller Ressourcen. URI sind anwendbar in Standards für die Repräsentation der primären und sekundären Informationsstrukturierung sowie der konzeptuellen Ebene.

Tripel (2.2.3.2) Konvention für die Notation von Aussagen.

vertikale Verbindung informationeller Ressourcen (2.1.2) Verbindung von informationellen Ressourcen der primären Informationsstrukturierung untereinander oder zu informationellen Ressourcen der konzeptuellen Ebene. Verbindungsglied ist die sekundäre Informationsstrukturierung.

XML (2.2.1.1) (eXtensible Markup Language), vgl. Bray et al. (2000). Standardisierte Auszeichnungssprache.

XML Information Set (2.2.1.2) Standard zur Beschreibung des Informationsgehalts von XML-Dokumentinstanzen. Vgl. **Informationseinheiten**.

C. RELAX NG Schema für die Repräsentation der sekundären Informationsstrukturierung

```
# sekStruc.rnc
# schema for the creation of secondary information structuring documents
# version: 0.2
# june 6th 04
# felix sasaki

# The folllowing is taken from RELAX NG, and slightly modified.
# difference to RELAX NG schema:
# optionality of attribute name at <element> and <attribute> element
# <element> and <attribute> element are not allowed to have an open-name-class
# <element> and <attribute> element are not allowed to have 'other' pattern

namespace local =""
namespace sekStruk = "http://example.com/sekStruk"
default namespace rng = "http://relaxng.org/ns/structure/1.0"
datatypes xsd="http://www.w3.org/2001/XMLSchema-datatypes"

start = element sekStruk:models
         { element sekStruk:model { attribute name { xsd:QName }, element
         sekStruk:annotation { text }?, define-element+ }+ }

pattern =
  element element {
```

```
    attribute name { xsd:QName }?,
#     | open-name-class,
    common-atts,
    open-patterns?
  }
  | element attribute {
      attribute name { xsd:QName }?,
#       | open-name-class,
      common-atts,
      pattern?
    }
  | element group { common-atts, open-patterns }
  | element interleave { common-atts, open-patterns }
  | element choice { common-atts, open-patterns }
  | element optional { common-atts, open-patterns }
  | element zeroOrMore { common-atts, open-patterns }
  | element oneOrMore { common-atts, open-patterns }
  | element list { common-atts, open-patterns }
  | element mixed { common-atts, open-patterns }
  | element ref {
      attribute name { xsd:NCName },
      common-atts
    }
#  | element parentRef {
#      attribute name { xsd:NCName },
#      common-atts
#    }
  | element empty { common-atts, other }
  | element text { common-atts, other }
  | element value {
      attribute type { xsd:NCName }?,
      common-atts,
      text
    }
```

```
  | element data {
      attribute type { xsd:NCName },
      common-atts,
      (other
       & (element param {
            attribute name { xsd:NCName },
            text
          }*,
          element except { common-atts, open-patterns }?))
    }
#  | element notAllowed { common-atts, other }
#  | element externalRef {
#      attribute href { xsd:anyURI },
#      common-atts,
#      other
#    }
#  | element grammar { common-atts, grammar-content }
#grammar-content =
#  other
#  & (start-element
#     | define-element
#     | element div { common-atts, grammar-content }
#     | element include {
#         attribute href { xsd:anyURI },
#         common-atts,
#         include-content
#       })*
#include-content =
#  other
#  & (start-element
#     | define-element
#     | element div { common-atts, include-content })*
#start-element = element start { combine-att, common-atts, open-pattern }
```

```
# DIFFERENCE TO RELAX NG:
# a <define> element is allowed to have RELAX NG open-patterns or
# a caterpillar expression or a layerRel (description of relations between
# multiple annotation layers.)
define-element =
  element define {
    attribute name { xsd:QName },
    sekStruk-atts,
    combine-att,
    common-atts,
    element sekStruk:annotation { text }?,
    (open-patterns | sekStruk-caterpillar | sekStruk-layerRel),
     define-element*
  }
combine-att = attribute combine { "choice" | "interleave" }?

# pattern+ has been relaxed to pattern*
open-patterns = other & pattern*
open-pattern = other & pattern
name-class =
  element name { common-atts, xsd:QName }
  | element anyName { common-atts, except-name-class }
  | element nsName { common-atts, except-name-class }
  | element choice { common-atts, open-name-classes }
except-name-class =
  other
  & element except { open-name-classes }?
open-name-classes = other & name-class+
open-name-class = other & name-class
common-atts =
  attribute ns { text }?,
  attribute datatypeLibrary { xsd:anyURI }?,
  attribute * - (rng:* | local:*) { text }*
other =
```

```
  element * - rng:* {
    (attribute * { text }
     | text)*
  }*
#any =
#  element * {
#    (attribute * { text }
#     | text
#     | any)*
#  }

# this is taken from CSD ("context specification document"):
sekStruk-caterpillar =
  element sekStruk:caterpillar { (movesAndTests | zeroOrMore)+}

movesAndTests = element sekStruk:up { empty } | element sekStruk:left
 { empty } | element sekStruk:right { empty } | element sekStruk:first
 { empty } | element sekStruk:last { empty } | element
 sekStruk:isFirst { empty } | element sekStruk:isLast { empty } |
 element sekStruk:isLeaf { empty } | element sekStruk:isRoot { empty }
 | element sekStruk:nodeTest { attribute concept-ref {xsd:QName},
 empty } zeroOrMore = element sekStruk:zeroOrMore { movesAndTests }

# this is taken from witt et al. 04. it has been slightly modified to be used
# within sekStruk.
sekStruk-layerRel = element sekStruk:layerRel { (
    element sekStruk:identity { layerRelAttsAndContent } |
    element sekStruk:included_A_in_B { layerRelAttsAndContent } |
    element sekStruk:included_B_in_A { layerRelAttsAndContent } |
    element sekStruk:starting_point_A { layerRelAttsAndContent } |
    element sekStruk:starting_point_B { layerRelAttsAndContent } |
    element sekStruk:end_point_A { layerRelAttsAndContent } |
    element sekStruk:end_point_B { layerRelAttsAndContent } |
    element sekStruk:overlap_A { layerRelAttsAndContent } |
```

```
    element sekStruk:overlap_B { layerRelAttsAndContent } |
    element sekStruk:endA_is_starting_pointB { layerRelAttsAndContent } |
    element sekStruk:endB_is_starting_pointA { layerRelAttsAndContent } |
    element sekStruk:before_A_B { layerRelAttsAndContent } |
    element sekStruk:before_B_A { layerRelAttsAndContent }
)+  }

layerRelAttsAndContent = attribute concept-ref {xsd:IDREF}, empty

# the following attributes can be used to encode various statements in
# secondary information strukturing. In the RDF Schema, the statements are
# encoded as tripels, here they are encoded as attribute-value pairs; the
# subject of the tripel is the name of the respective <define> element.
sekStruk-atts = attribute sekStruk:sekStruk2primStruk { xsd:ID }?,
    attribute sekStruk:subConceptOf { xsd:IDREFS }?,
    attribute sekStruk:abstract { "true" }?,
    attribute sekStruk:sekStruk2conLevel { list { xsd:anyURI+ } }?,
    attribute sekStruk:componentOf { xsd:IDREFS }?
```

D. RDF Schema für die Repräsentation der sekundären Informationsstrukturierung

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

<!-- This is an RDF Schema for Secondary Information Structuring
  written by Felix Sasaki, felix.sasaki@uni-bielefeld.de last changed:
  June 25 2004 -->

  <rdfs:Class
    rdf:about="http://www.example.com/schemas/sekStruk#Concept"
    rdfs:comment="A concept in Secondary Information Structuring."/>

  <rdfs:Class
    rdf:about="http://www.example.com/schemas/sekStruk#PrimStruk"
    rdfs:comment="Set of all things in Primary Information
    Structuring."/>

  <rdf:Property
    rdf:about="http://www.example.com/schemas/sekStruk#subConceptOf"
    rdfs:comment="predictate for the creation of the concept hierarchy
    and the relation partOf between concepts and a model, for example
    'sekStruk:dt-tei subConceptOf sekStruk:tei.'. All concepts which
    denote a model are directly subordinated to the predefined concept
    sekStruk:models, e.g. 'sekStruk:tei subConceptOf
    sekStruk:models.'"> <rdfs:range
```

```
    rdf:resource="http://www.example.com/schemas/sekStruk#Concept"/>
    <rdfs:domain
    rdf:resource="http://www.example.com/schemas/sekStruk#Concept"/>
  </rdf:Property>

  <rdf:Property
    rdf:about="http://www.example.com/schemas/sekStruk#sekStruk2primStruk"
    rdfs:comment="predicate for the mapping between Secondary and
    Primary Information Structuring, for example 'sekStruk:def-tei
    sekStruk2primStruk element dltei:item'."> <rdfs:domain
    rdf:resource="http://www.example.com/schemas/sekStruk#Concept"/>
    <rdfs:range
    rdf:resource="http://www.example.com/schemas/sekStruk#PrimStruk"/>
  </rdf:Property>

  <rdf:Property
    rdf:about="http://www.example.com/schemas/sekStruk#sekStruk2conLevel"
    rdfs:comment="predicate for the mapping between Secondary
    Information Structuring and the conceptual level."> <rdfs:domain
    rdf:resource="http://www.example.com/schemas/sekStruk#Concept"/>
    <rdfs:range
    rdf:resource="http://www.w3.org/2000/01/rdf-schema#Resource"/>
  </rdf:Property>

  <rdf:Property
    rdf:about="http://www.example.com/schemas/sekStruk#componentOf"
    rdfs:comment="predicate which is used in document grammar based
    Secondary Information Structuring. It describes the property of a
    document grammar construct / the concept which selects the
    construct, of being contained in another construct / another
    concept. An example is 'sekStruk:dt-html componentOf
    sekStruk:definitionList-html'"> <rdfs:domain
    rdf:resource="http://www.example.com/schemas/sekStruk#Concept"/>
    <rdfs:range
```

```
  rdf:resource="http://www.example.com/schemas/sekStruk#Concept"/>
</rdf:Property>

<rdf:Property
  rdf:about="http://www.example.com/schemas/sekStruk#equal"
  rdfs:comment="predicate for the description of identity relations
  between concepts in Secondary Information Structuring. This
  predicate is used only for concepts which belong to separate
  models. For example, the identity relation between the concepts
  sekStruk:dt-tei and sekStruk:dt-html can be described with the
  logical statement 'sekStruk:dt-tei equal sekStruk:dt-html.'.">
  <rdfs:domain
  rdf:resource="http://www.example.com/schemas/sekStruk#Concept"/>
  <rdfs:range
  rdf:resource="http://www.example.com/schemas/sekStruk#Concept"/>
</rdf:Property>

<rdf:Property
  rdf:about="http://www.example.com/schemas/sekStruk#grammarPattern"
  rdfs:comment="predicate to describe document grammar based
  Secondary Information Structuring"> <rdfs:domain
  rdf:resource="http://www.example.com/schemas/sekStruk#PrimStruk"/>
  <rdfs:range
  rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
</rdf:Property>

<rdf:Property
  rdf:about="http://www.example.com/schemas/sekStruk#pathEx"
  rdfs:comment="predicate to describe instance based Secondary
  Information Structuring with path expressions in a single instance
  document"> <rdfs:range
  rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  <rdfs:domain
  rdf:resource="http://www.example.com/schemas/sekStruk#PrimStruk"/>
```

```
  </rdf:Property>

  <rdf:Property
    rdf:about="http://www.example.com/schemas/sekStruk#layerRel"
    rdfs:comment="predicate to describe instance based Secondary
    Information Structuring in multiple instance documents,
    i.e. several annotation layers."> <rdfs:domain
    rdf:resource="http://www.example.com/schemas/sekStruk#PrimStruk"/>
    <rdfs:range
    rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  </rdf:Property>

</rdf:RDF>
```

E. Exemplarisches Dokument in RDF

```
<!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:sekStruk="http://www.example.com/schemas/sekStruk#"
            xml:base="http://example.org/things">

  <sekStruk:Concept rdf:ID="Model"/>

  <sekStruk:Concept rdf:ID="tei">
    <sekStruk:subConceptOf rdf:resource="#Model"/>
  </sekStruk:Concept>

  <sekStruk:Concept rdf:ID="def-tei">
    <sekStruk:subConceptOf rdf:resource="#tei"/>
    <sekStruk:componentOf rdf:resource="#definitionList-tei"/>
    <sekStruk:sekStruk2primStruk rdf:resource="#mapping1"/>
  </sekStruk:Concept>

  <sekStruk:Concept rdf:ID="list-tei-general">
    <sekStruk:subConceptOf rdf:resource="#tei"/>
    <sekStruk:sekStruk2primStruk rdf:resource="#mapping2"/>
  </sekStruk:Concept>

  <sekStruk:Concept rdf:ID="dt-tei">
    <sekStruk:subConceptOf rdf:resource="#tei"/>
    <sekStruk:componentOf rdf:resource="#definitionList-tei"/>
    <sekStruk:sekStruk2primStruk rdf:resource="#mapping3"/>
```

```
  </sekStruk:Concept>

  <sekStruk:Concept rdf:ID="definitionList-tei">
    <sekStruk:subConceptOf rdf:resource="#list-tei-general"/>
    <sekStruk:sekStruk2primStruk rdf:resource="#mapping4"/>
    <sekStruk:sekStruk2conLevel
    rdf:resource="http://www.cogsci.princeton.edu/~wn/concept#103701336"/>
  </sekStruk:Concept>

  <sekStruk:PrimStruk rdf:ID="mapping1">
   <sekStruk:grammarPattern>element
    dltei:item</sekStruk:grammarPattern>
    <sekStruk:pathEx>up 'dltei:tei.2</sekStruk:pathEx>
    <sekStruk:layerRel>identity 'dltei:line'</sekStruk:layerRel>
  </sekStruk:PrimStruk>

  <sekStruk:PrimStruk rdf:ID="mapping2">
   <sekStruk:grammarPattern>element
    dltei:list</sekStruk:grammarPattern>
  </sekStruk:PrimStruk>

  <sekStruk:PrimStruk rdf:ID="mapping3">
   <sekStruk:grammarPattern>element
    dltei:head</sekStruk:grammarPattern>
  </sekStruk:PrimStruk>

  <sekStruk:PrimStruk rdf:ID="mapping4">
   <sekStruk:grammarPattern>attribute type
    {'gloss'}</sekStruk:grammarPattern>
  </sekStruk:PrimStruk>
</rdf:RDF>
```

F. Aufbau der Ausdrücke zur Operationalisierung der sekundären Informationsstrukturierung

Die Operationen beruhen auf einer Dokumentgrammatik im Format RELAX NG, die im Folgenden wiedergegeben wird.

```
# sekStruk-query.rnc
# schema for the creation of operations which analyse secondary information
# structuring (s.i.s)
# version: 0.3
# july 8th 04
# felix sasaki
# information resources have to be specified, and at least one operation to be
# executed
start = element operations { informationResources, operation+}
# information resources encompass a collection of statements, i.e. secondary
# information structuring in XML-syntax, a collection of document grammars in
# the XML-format of RELAX NG, a collection of XML instance documents, and a
# (optional) a collection of information resources from the conceptual level,
# in any XML serialization
informationResources =
 element informationResources
 { attribute collectionSekStruk { xsd:anyURI },
   attribute collectionDocuGram { xsd:anyURI },
   attribute collectionInstanceDocs { xsd:anyURI },
   attribute collectionConceptualLevel { xsd:anyURI }?,
   empty }
```

```
# operations encompass conceptual queries, validations and transformations
operation = conceptualQuery | conceptualValidation | conceptualTransformation
# a conceptual query takes as an input a model from secondary information
# structuring and alternatively a concept from s.i.s or the conceptual level
conceptualQuery =
element conceptualQuery { model, (conceptSekStruk | conceptualLevel+) }
# a conceptual validation takes the same input as a query
conceptualValidation =
element conceptualValidation { model, (conceptSekStruk | conceptualLevel+) }
# a conceptual transformation takes as an input a source model
# and a target model
conceptualTransformation =
element conceptualTransformation { sourceModel, targetModel }
# the following is just the content of the RELAX NG patterns
conceptSekStruk =
attribute conceptSekStruk { xsd:anyURI }
conceptualLevel =
element uriOfConceptualLevel { attribute href  { xsd:anyURI }, empty }
model = attribute model { xsd:anyURI }
sourceModel = attribute sourceModel { xsd:anyURI }
targetModel = attribute targetModel { xsd:anyURI }
```

Literaturverzeichnis

Es liegt in der Thematik der vorliegenden Arbeit begründet, dass ein gewisser Anteil der zitierten Literatur nicht als Buch- oder Zeitschriftenpublikation in gedruckter Form vorliegt. In diesen Fällen ist es notwendig, auf die jeweiligen Netzadressen bzw. URI zu verweisen. Größtenteils handelt es sich bei den Quellen um Beschreibungen standardisierter Formate. Organisationen wie das World Wide Web Consortium tragen deshalb Sorge für die Nachhaltigkeit der URI. Sämtliche im Literaturverzeichnis verwendeten URI wurden am 26. Juli 2004 überprüft.

S. Abney. Parsing by Chunks. In: S. Abney, R. Berwick und C. Tenny (Hrsg.) *Principle-Based Parsing*. Kluwer, Dordrecht, 1991.

J. F. Allen und G. Ferguson. Actions and Events in Interval Temporal Logic. *Logic and Computation*, 4(5):531–579, 1994.

M. Altheim, F. Boumphrey, S. Dooley, S. McCarron, S. Schnitzenbaumer und T. Wugofski. Modularization of XHTML. W3C Recommendation, 2001. http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410/

M. Asahara, R. Yoneda, A. Yamashita, Y. Den und Y. Matsumoto. Use of XML and relational database for consistent development and maintenance of lexicon and annotated corpora. In: *Proceedings of LREC 2002*, Las Palmas, Spanien, 2002.

G. Aston und L. Burnard. *The BNC handbook: Exploring the British National Corpus with SARA*. Edinburgh University Press, Edinburgh, 1998.

P. S. Bayerl, D. Goecke, H. Lüngen und A. Witt. Methods for the semantic analysis of document markup. In: C. Roisin, E. Munson und C. Vanoirbeek (Hrsg.) *Proceedings of the 3rd ACM Symposium on Document Engineering (DocEng)*, S. 161–170, Grenoble, 2003.

T. Berners-Lee. *Weaving the Web - The original design and ultimate design of the World Wide Web by its inventor*. HarperBusiness, San Francisco, 2000.

T. Berners-Lee, R. Fielding und L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396. The Internet Society, 1998. http://www.isi.edu/in-notes/rfc2396.txt

A. Bies, M. Ferguson, K. Katz, R. MacIntyre, V. Tredinnick, G. Kim, M. A. Marcinikiewicz und B. Schasberger. *Bracketing Guidelines for Treebank II Style - Penn Treebank Project*. University of Pennsylvania, 1995.

S. Bird und M. Liberman. A Formal Framework for Linguistic Annotation. *Speech Communication*, 33(1-2):33–60, 2001.

P. V. Biron und A. V. Malhotra. XML Schema Part 2: Datatypes. W3C Recommendation, 2001. http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

T. Bray, D. Hollander und A. Layman. Namespaces in XML. W3C Recommendation, 1999. http://www.w3.org/TR/1999/REC-xml-names-19990114/

T. Bray, J. Paoli, C. M. Sperberg-McQueen und E. Maler. Extensible Markup Language 1.0 (Second Edition). W3C Recommendation, 2000. http://www.w3.org/TR/2000/REC-xml-20001006

A. Brüggemann-Klein und D. Wood. Caterpillars: A Context Specification Technique. *Markup Languages: Theory and Practice*, 2(1):81–106, 2000.

D. Brickley und R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

J. Carson-Berndsen. *Time Map Phonology*. Text, Speech and Language Technology. Kluwer Academic Publishers, Dordrecht, 1998.

P. Caton. Markup's Current Imbalance. *Markup Languages: Theory and Practice*, 3(1): 1–13, 2002.

J. Clark. XSL Transformations Version 1.0. W3C Recommendation, 1999. http://www.w3.org/TR/1999/REC-xslt-19991116

J. Clark. RELAX NG Compact Syntax. OASIS, 2002. http://www.relaxng.org/compact-20021121.html

J. Clark und S. deRose. XML Path Language (XPath) Version 1.0. W3C Recommendation, 1999. http://www.w3.org/TR/1999/REC-xpath-19991116

J. Clark und M. Murata. RELAX NG Specification. OASIS, 2001. http://www.oasis-open.org/committees/relax-ng/spec-20011203.html

J. Cowan und R. Tobin. XML Information Set (Second Edition). W3C Recommendation, 2004. http://www.w3.org/TR/2004/REC-xml-infoset-20040204/

A. Czmiel. XML for Overlapping Structures (XfOS) using a non XML Data Model. In: *Conference abstracts of ALLC / ACH 2004*, Göteborg, 2004.

S. deRose, E. Maler und D. Orchard. XML Linking Language (XLink) Version 1.0. W3C Recommendation, 1999. http://www.w3.org/TR/2001/REC-xlink-20010627/

H. Ter Doest. *Towards Probabilistic Unification-based Parsing*. Dissertation, Universiteit Twente Enschede, Enschede, 1999.

M. Dürst, F. Yergeau, R. Ishida, M. Wolf, A. Freytag und T. Texin. Character Model for the World Wide Web 1.0. W3C Working Draft, 2002. http://www.w3.org/TR/2002/WD-charmod-20020220/

P. Durusau und M. B. O'Donnell. Concurrent Markup for XML Documents. In: *Proceedings of XML Europe 2002*, Barcelona, 2002.

M. Erdmann. *Ontologien zur konzeptuellen Modellierung der Semantik von XML*. Book on Demand BoD GmbH, Norderstedt, 2001.

M. Erdmann und R. Studer. Ontologies as Conceptual Models for XML Documents. In: *KAW'99 - Twelfth Workshop on Knowledge Acquisition, Modeling and Managment*, Alberta, Kanada, 1999.

T. Erjavec. MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons & Corpora. In: *Proceedings of LREC 2004*, Lisabon, Portugal, 2004.

S. Evert, J. Carletta, T. O'Donnell, J. Kilgour, A. Vögele und H. Voormann. The NITE Object Model. NITE - Natural Interactivity Tools Engineering, 2003.

F. Baader, D. Calvanese, D. McGuinness, D. Nardi und P. Patel-Schneider (Hrsg.) *Description Logic Handbook.* Cambridge University Press, 2003.

S. Farrar, W. Lewis und T. Langendoen. A Common Ontology for Linguistic Concepts. In: *Proceedings of the Knowledge Technologies Conference*, Seattle, Washington, 2002.

C. Fellbaum (Hrsg.) *WordNet. An Electronic Lexical Database.* MIT Press, Cambridge, Mass., 1998.

M. Fernandez, A. Malhotra, J. Marsh, M. Nagy und N. Walsh. XQuery 1.0 and XPath 2.0 Data Model. W3C Working Draft, 2003. http://www.w3.org/TR/2003/WD-xpath-datamodel-20031112/

C. J. Fillmore, C. Wooters und C. F. Baker. Building a Large Lexical Databank which provides Deep Semantics. In: *Proceedings of the Pacific Asian Conference on Language, Information and Computation*, Hong Kong, 2001.

D. Fischer. From Thesauri towards Ontologies? In: W. M. el Hadi, J. Maniez und St. A. Pollit (Hrsg.) *Structures and Relations in Knowledge Organization. Proceedings of the 5th ISKO-Conference, Lille*, Würzburg, 1998. Ergon Verlag.

R. Garside, G. Leech und T. McEnery. *Corpus Annotation: Linguistic Information from Computer Text Corpora.* Longman, London, 1997.

J. Gippert. Language-specific Encoding in Multilingual Corpora: Requirements and Solutions. In: *Multilinguale Corpora - Codierung, Strukturierung, Analyse. Proceedings der GLDV-Frühjahrstagung 1999*, Enigma, Prag, 1999.

T. Graham. *Unicode: a Primer.* M and T Books, California, 2000.

E. Hajicova und I. Kucerova. Argument / Valency Structure in PropBank, LCS Database and Prague Dependency Treebank: A comparative pilot study. In: *Proceedings of LREC 2002*, Las Palmas, Spanien, 2002.

P. Hayes und B. McBride. RDF Semantics. W3C Recommendation, 2004. http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

P. Hayes, S. Melnik und P. Stickler. RDF Datatyping, 2002. http://www.coginst.uwf.edu/ phayes/RDF_Datatyping060102_draft.html

C. F. Hockett. Two Models of Grammatical Description. *Word*, 10:210–231, 1954.

C. Huitfeldt und C. M. Sperberg-McQueen. TexMECS: an experimental markup meta-language for complex documents. 2001. http://www.hit.uib.no/claus/mlcd/papers/texmecs.html

IBM, Unisys und SofTeam. XML Metadata Interchange (XMI) - XMI Production of XML Schema. OMG (Object Modeling Group), 2001.

N. Ide. Corpus Encoding Standard: SGML Guidelines for Encoding Linguistic Corpora. In: *Proceedings of LREC 1998*, Granada, Spanien, 1998.

N. Ide und R. Romary. Encoding Syntactic Annotation. In: A. Abeillé (Hrsg.) *Building and Using Parsed Corpora*. Kluwer, Dordrecht, 2003.

ISO/IEC10744. Information Technology - Hypermedia/Time-based Structuring Language (HyTime). International Organization for Standardization, Genf, 1997.

ISO/IEC8859. Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML). International Organization for Standardization, Genf, 1986.

ISO/IEC8859-1. Information technology - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1. International Organization for Standardization, Genf, 1998.

ISO/IEC9541-1. Information technology – Font information interchange – Part 1: Architecture. International Organization for Standardization, Genf, 1991.

R. Jelliffe. The Schematron Assertion Language 1.5. Academia Sinica Computing Centre, 2000. http://xml.ascc.net/resource/schematron/Schematron2000.html

Y. Kawata. Towards a Reference Tagset for Japanese. In: *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium Post-Conference Workshop on Language Resources in Asia*, Tokyo, 2001.

Y. Kawata und J. Bartels. *Stylebook for the Japanese treebank in VERBMOBIL*, 2000. Verbmobil-Report 240.

M. Kifer, G. Lausen und J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42, 1995.

E. Kimber und J. Heintz. Using UML to define XML Document Types. *Markup Languages: Theory and Practice*, S. 295–320, 2001.

P. Kingsbury, M. Palmer und M. Marcus. Adding Semantic Annotation to the Penn TreeBank. In: *Proceedings of the Human Language Technology Conference*, San Diego, 2002.

M. Klein, N. O. Bernsen, S. Davies, L. Dybkjær, J. Garrido, H. Kasch, A. Mengel, V. Pirrelli, M. Poesio, S. Quazza und C. Soria. *MATE Deliverable D 1.1 – Supported Coding Schemes*, 1998. http://mate.nis.sdu.dk/about/D1.1/

M. Klein, D. Fensel, F. v. Harmelen und I. Horrocks. The Relation Between Ontologies and XML Schemata. In: *Proceedings of the ECAI 2000 Workshop on Applications of Ontologies and Problem-solving Methods*, Berlin, 2000.

T. Kudo und Y. Matsumoto. Japanese Dependency Analyisis using Cascaded Chunking. In: *Proceedings of Sixth Conference on Natural Language Learning (CoNLL-2002)*, S. 29–35, Taipei, Taiwan, 2002.

S. Kurohashi und M. Nagao. Building a Japanese parsed corpus while improving the parsing system. In: A. Abeillé (Hrsg.) *Building and Using Parsed Corpora*. Kluwer, Dordrecht, 2003.

C. Laprun, J. G. Fiscus, S. Pajot und J. Garofolo. A Practical Introduction to ATLAS 2.0. In: *Proceedings of Human Language Technology Conference 2002*, San Diego, 2002.

O. Lassila und R. R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 1999. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

D. Lee und W. W. Chu. Comparative Analysis of Six XML Schema Languages, 2000. http://www.cobase.cs.ucla.edu/tech-docs/dongwon/ucla-200008.html

G. Leech, R. Barnett und P. Kahrel. EAGLES Recommendations for the Syntactic Annotation of Corpora. EAG-TCWG-SASG//1.8. European Communities Language Engineering Strategy Committees, 1996. http://www.ilc.cnr.it/EAGLES96/segsasg1/

E. Lenz, A. Storrer und A. Witt. Towards Declarative Descriptions of Transformations: An Approach Based on Topic Maps. In: *Conference abstracts of ALLC / ACH 2002*, Tübingen, 2002.

W. Lezius. *Ein Suchwerkzeug für syntaktisch annotierte Textkorpora.* Dissertation, Universität Stuttgart, 2002.

H. Lobin. *Koordinationssyntax als prozedurales Phänomen.* Studien zur deutschen Grammatik. Narr, Tübingen, 1993.

H. Lobin. *Informationsmodellierung in XML und SGML.* Springer, Berlin, 2000.

H. Lobin. Netzwerkbasierte Modellierung der Semantik von XML-Strukturen. In: H. Lobin (Hrsg.) *Sprach- und Texttechnologie im digitalen Medium. Proceedings der GLDV-Frühjahrstagung 2001*, Gießen, 2001.

H. Lobin. Erweiterte Dokumentgrammatiken als Grundlage innovativer XML-Tools. *Information Technology*, 45(3):143–150, 2003.

H. Lobin. Textauszeichnung und Dokumentgrammatiken. In: H. Lobin und L. Lemnitzer (Hrsg.) *Texttechnologie. Perspektiven und Anwendungen.* Stauffenburg, Tübingen, 2004a.

H. Lobin. Text(e) technologisch. In: H. Lobin und L. Lemnitzer (Hrsg.) *Texttechnologie. Perspektiven und Anwendungen.* Stauffenburg, Tübingen, 2004b.

J. F. Maas. NEXUS: Vollautomatische Konvertierung mehrfach XML-annotierter Texte in das NITE-XML Austauschformat. Magisterarbeit, Universität Bielefeld, 2003.

F. Manola und E. Miller. RDF Primer. W3C Recommendation, 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

D. L. McGuinness und F. v. Harmelen. OWL Web Ontology Language Overview. W3C Recommendation, 2004. http://www.w3.org/TR/2004/REC-owl-features-20040210/

A. Mehler und H. Lobin. Aspekte der texttechnologischen Modellierung. In: A. Mehler und H. Lobin (Hrsg.) *Systeme und Methoden zur Annotation und Analyse natürlichsprachlicher Texte.* Westdeutscher Verlag, Wiesbaden, 2004.

S. Melnik und S. Decker. A Layered Approach to Information Modeling and Interoperability on the Web. In: *ECDL Workshop 2000 on the Semantic Web*, Lisabon, 2000.

A. Mengel. Die integrierte Repräsentation linguistischer Daten. In: J. Gippert (Hrsg.) *Multilinguale Corpora - Codierung, Strukturierung, Analyse. Proceedings der GLDV-Frühjahrstagung 1999*, Enigma, Prag, 1999.

A. Mengel, L. Dybkjær, J.M. Garrido, U. Heid, M. Klein, V. Pirrelli, M. Poesio, S. Quazza, A. Schiffrin und C. Soria. *MATE Deliverable D 2.1 - Dialogue Annotation Guidelines*, 2000. http://www.ims.uni-stuttgart.de/projekte/mate/mdag/

J. T. Milde und U. Gut. The TASX environment: an XML-based toolset for time aligned speech corpora. In: *Proceedings of LREC 2002*, Las Palmas, Spanien, 2002.

U. Mönnich, F. Morawietz und S. Kepser. A Regular Query for Context-Sensitive Relations. In: S. Bird, P. Buneman und M. Liberman (Hrsg.) *IRCS Workshop on Linguistic Databases 2001*, S. 187–195, University of Pennsylvania, Philadelphia, 2001.

M. Murata, D. Lee und M. Mani. Taxonomy of XML Schema Languages using Formal Language Theory. In: *Proceedings of Extreme Markup Languages 2001*, Montreal, Canada, 2001.

I. Niles und A. Pease. Towards a Standard Upper Ontology. In: *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Ogunquit, Maine, 2001.

N. F. Noy und D. L. McGuinness. Ontology Development 101: A Guide to Creating your First Ontology. Stanford Medical Informatics, 2001.

S. Oepen, D. Flickinger, K. Toutanova und C. D. Manning. LinGO Redwoods. In: *First Workshop on treebanks and linguistic theories*, Sozopol, Bulgaria, 2002.

S. Pemperton, D. Austin, J. Axelsson et al. XHTML 1.0 The Extensible HyperText Markup Language (Second Edition). W3C Recommendation, 2002. http://www.w3.org/TR/2002/REC-xhtml1-20020801/

S. Pepper und G. Moore. XML Topic Maps (XTM) 1.0. TopicMaps.Org, 2001. http://www.topicmaps.org/xtm/1.0/xtm1-20010806.html

M. Poesio. *Coreference*, 2000. http://www.ims.uni-stuttgart.de/projekte/mate/mdag/cr/cr_1.html

C. Pollard und I. Sag. *Head-driven Phrase Structure Grammar*. University of Chicago Press, Chicago, 1994.

J. C. Ramalho, J. G. Rocha, J. J. Almeida und P. Henriques. SGML Documents: Where does Quality go? *Markup Languages: Theory and Practice*, 1(1):75–90, 1999.

O. Rambow, C. Creswell, R. Szekely, H. Taber und M. Walker. A Dependency Treebank for English. In: *Proceedings of LREC 2002*, Las Palmas, Spanien, 2002.

A. Renear, D. Dubin, C. M. Sperberg-McQueen und C. Huitfeldt. Towards a Semantics for XML Markup. In: R. Furuta, J. I. Maletic und E. Munson (Hrsg.) *Proceedings of the 2002 ACM Symposium on Document Engineering*, Virginia, 2002.

A. Renear, E. Mylonas und D. Durand. Refining our Notion of what Text really is: The Problem of Overlapping Hierarchies. In: N. Ide und S. Hockey (Hrsg.) *Research in Humanities Computing*. Oxford University Press, Oxford, 1996.

S. Salmon-Alt und L. Romary. Towards a Reference Annotation Framework. In: *Proceedings of LREC 2004*, Lisabon, Portugal, 2004.

F. Sasaki. Annotation und Repräsentation morphologischer Strukturen in syllabischen Symbolsystemen. Projektbericht. Universität Bielefeld, 2002.

F. Sasaki. Combining Semantic Markup and Markup Semantics: A Secret Marriage. In: *Conference abstracts of ALLC / ACH 2004*, Göteborg, 2004.

F. Sasaki, C. Wegener, A. Witt, D. Metzing und J. Pönninghaus. Co-reference Annotation and Resources: A Multilingual Corpus of Typologically Diverse Languages. In: *Proceedings of LREC 2002*, Las Palmas, Spanien, 2002.

F. Sasaki und A. Witt. Co-reference in Japanese Task-oriented Dialogues: A Contribution to the Development of Language-specific and Language-general Annotation Schemes and Resources. In: *Proceedings of LREC 2004*, Lisabon, Portugal, 2004a.

F. Sasaki und A. Witt. Linguistische Korpora. In: H. Lobin und L. Lemnitzer (Hrsg.) *Texttechnologie. Perspektiven und Anwendungen*. Stauffenburg, Tübingen, 2004b.

F. Sasaki, A. Witt und D. Metzing. Declarations of Relations, Differences and Transformations between Theory-specific Treebanks: A New Methodology. In: J. Nivre (Hrsg.) *The Second Workshop on Treebanks and Linguistic Theories (TLT 2003)*, Växjö University, Sweden, 2003.

M. Sawicki, M. Suignard, M. Ishikawa, M. Dürst und T. Textin. Ruby Annotation. W3C Recommendation, 2001. http://www.w3.org/TR/2001/REC-ruby-20010531/

A. Schiller und S. Teufel. Guidelines für das Tagging deutscher Textkorpora. Institut für maschinelle Sprachverarbeitung Stuttgart und Universität Tübingen, Stuttgart, Tübingen, 1995.

Thomas Schmidt. *Computergestützte Transkription als Modellierung und Visualisierung gesprochener Sprache mit texttechnologischen Mitteln.* Dissertation, Universität Dortmund, i.V.

W. Schnotz. *Aufbau von Wissensstrukturen. Untersuchungen zur Kohärenzbildung bei Wissenserwerb in Texten.* Belz, Psychologie-Verl.-Union, Weinheim, 1994.

G. Senft. *Kilivila: The Language of the Trobiand Islanders.* de Gruyter, Berlin, 1986.

K. Shafer. Creating DTDs via the GB-Engine and Fred. In: *SGML '95 Conference Proceedings*, 1995. http://www.oclc.org/fred/docs/sgml95.html

St. M. Shieber. Separating Linguistic Analysis from Linguistic Theories. In: *Linguistic Theory and Computer Applications.* Academic Press, London, 1987.

G. F. Simons. A Conceptual Modeling Language for the Analysis and Interpretation of Text. Text Encoding Initiative Committee on Text Analysis and Interpretation. Document Number: TEI AIW12, 1990. http://www.tei-c.org/Vault/AI/aiw12.txt

G. F. Simons. Conceptual Modeling versus Visual Modeling: A Technological Key to Building Consensus. *Computing in the Humanities*, 30(4):303–319, 1997.

G. F. Simons. The Nature of Linguistic Data and the Requirements of a Computing Environment for Linguistic Research. In: J. Lawler und H. Aristar Dry (Hrsg.) *Using Computers in Linguistics. A Practical Guide.* Routledge, 1998.

G. F. Simons. Using architectural Forms to map TEI Data into an Object-oriented Database. *Computing in the Humanities*, 33(1):85–101, 1999.

G. F. Simons. Developing a metaschema language to support interoperation among XML resources with different markup schemas. In: *Conference abstracts of ALLC / ACH 2003*, Athens, Georgia, USA, 2003.

G. F. Simons und S. Bird. *OLAC Metadata*. Open Language Archives Community, 2003. http://www.language-archives.org/OLAC/metadata.html

J. F. Sowa. Ontologies for Knowledge Sharing. Manuscript of the invited talk at TKE 96, 1996.

C. M. Sperberg-McQueen. Logic grammars and XML Schema. In: *Proceedings of Extreme Markup Languages 2003*, Montreal, Canada, 2003.

C. M. Sperberg-McQueen und L. Burnard (Hrsg.) *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*. The Association for Computers and the Humanities (ACH) - Oxford University Computing Services, Oxford, 1994.

C. M. Sperberg-McQueen, D. Dubin, C. Huitfeldt und A. Renear. Drawing Inferences on the Basis of Markup. In: *Proceedings of Extreme Markup Languages 2002*, Montreal, Canada, 2002.

C. M. Sperberg-McQueen und C. Huitfeldt. Concurrent Document Hierarchies in MECS and SGML. In: *Conference abstracts of ALLC / ACH 1998*, Debrecen, Ungarn, 1998.

C. M. Sperberg-McQueen, C. Huitfeldt und A. Renear. Meaning and interpretation of markup. *Markup Languages: Theory and Practice*, 2(3):215–234, 2000.

R. R. Swick und H. S. Thompson. The Cambridge Communique. W3C Note, 1999. http://www.w3.org/TR/1999/NOTE-schema-arch-19991007

J. Tennison und W. Piez. The Layered Markup and Annotation Language (LMNL). In: *Proceedings of Extreme Markup Languages 2002*, Montreal, Canada, 2002.

H. S. Thompson, D. Beech, M. Maloney und N. Mendelsohn. XML Schema Part 1: Structures. W3C Recommendation, 2001. http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

T. Trippel, F. Sasaki und D. Gibbon. Consistent Storage of Metadata in Inference Lexica: the MetaLex Approach. In: *Proceedings of LREC 2004*, Lisabon, 2004.

T. Trippel, F. Sasaki, B. Hell und D. Gibbon. Acquiring Lexical Information from Multilevel Temporal Annotations. In: *Proceedings of Eurospeech 2003*, Genf, 2003.

T. Ule. DEREKO Linguistic Markup. Seminar für Sprachwissenschaft, Universität Tübingen, 2002.

P. Vossen. Ontologies. In: R. Mitkov (Hrsg.) *The Oxford Handbook of Computational Linguistics*. Oxford University Press, Oxford, 2003.

W. Wahlster (Hrsg.) *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer, Berlin, 2000.

C. Welty und N. Ide. Using the Right Tools: Enhancing Retrieval from Marked-up Documents. *Computing in the Humanities*, 33(1-2), 1999.

S. Wiemer. Strukturierte, konzeptorientierte Dokumentation von XML DTDs. Magisterarbeit, Universität Bielefeld, 1999.

A. Witt. SGML und Linguistik. In: *Text im digitalen Medium. Linguistische Aspekte von Textdesign, Texttechnologie und Hypertext Engineering*. Westdeutscher Verlag, Wiesbaden, 1999.

A. Witt. *Multiple Informationsstrukturierung mit Auszeichnungssprachen. XML-basierte Methoden und deren Nutzen für die Sprachtechnologie*. Dissertation, Universität Bielefeld, 2002.

A. Witt. Linguistische Informationsmodellierung. In: A. Mehler und H. Lobin (Hrsg.) *Automatische Textanalyse – Systeme und Methoden zur Annotation und Analyse natürlichsprachlicher Texte*. Verlag für Sozialwissenschaften, Wiesbaden, 2004a.

A. Witt. Multiple hierarchies: New aspects of an old solution. In: *Proceedings of Extreme Markup Languages 2004*, Montreal, Canada, 2004b.

A. Witt, H. Lüngen und D. Gibbon. Enhancing speech corpus resources with multiple lexical tag layers. In: *Proceedings of LREC 2000*, Athen, 2000.

A. Witt, H. Lüngen, D. Goecke und F. Sasaki. Unification of XML Documents with Concurrent Markup. In: *Conference abstracts of ALLC / ACH 2004*, Göteborg, 2004.

P. Wittenburg, D. Broeder und B. Sloman. EAGLES/ISLE: A Proposal for a Meta Description Standard for Language Resources. In: *LREC 2000 Workshop*, Athens, 2000.

V. Wuwongse, K. Akama, C. Anutariya und E. Nantajeewarawat. A Data Model for XML Databases. *Journal of Intelligent Information Systems*, 20(1):63–80, 2003.

V. Wuwongse, C. Anutariya, K. Akama und E. Nantajeewarawat. XML Declarative Description (XDD): A Language for the Semantic Web. *IEEE Intelligent Systems*, 16 (3):54–65, 2001.

F. Xia und M. Palmer. Converting Dependency Structures to Phrase Structures. In: *Proceedings of Human Language Technology Conference 2001*, San Francisco, 2001.