*Felix Sasaki*

# Textauszeichnung im Original und in der Übersetzung: Schemasprachen und mehr

## 1 Einleitung

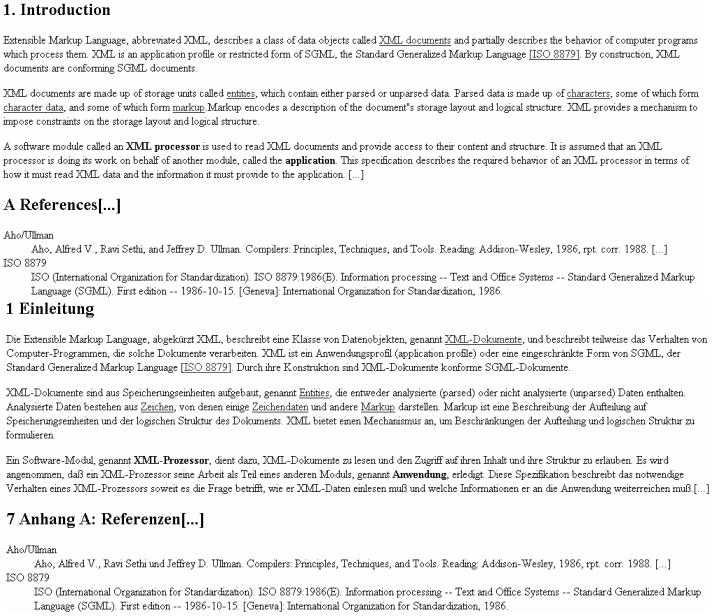
Der vorliegende Artikel behandelt Forschungsarbeiten, die in der Forschergruppe TEXTTECHNOLOGISCHE INFORMATIONSMODELLIERUNG, im Rahmen des Projekts SEKIMO (Sekundäre Informationsstrukturierung und vergleichende Diskursanalyse) entstehen. Im Projekt Sekimo[1] werden u. a. Schemasprachen[2] für einen korpusbasierten Vergleich sprachlicher Funktionen (z. B. Koreferenz) und ihren Realisierungsformen in typologisch verschiedenen Sprachen eingesetzt. In diesem Artikel werden Möglichkeiten von Schemasprachen und komplementären Ansätze bzw. Methoden, die sich innerhalb der Projektarbeit für den Sprachvergleich als hilfreich erwiesen haben, hinsichtlich ihrer Eignung als Mittel zur Wahrung und Analyse von Konsistenz textuell ausgezeichneter Dokumente untersucht. Die Dokumente stehen in einer Übersetzungsrelation zueinander. D. h. sie liegen in einer Ausgangssprache (AS) und in einer oder mehreren Zielsprache(n) (ZS) vor.

Abbildung 1 zeigt einen mittels HTML ausgezeichneten Ausschnitt der Beschreibung des XML-Standards, der XML-Empfehlung (vgl. Bray et al. 2000), in der ausgangssprachlichen, englischen Version, und einer zielsprachlichen, deutschen Version. Die Darstellung entspricht einer Browserausgabe. Die Textauszeichnungen - oft auch als Annotation bezeichnet - in AS und ZS stehen in bestimmten, in dreifacher Weise differenzierbaren Beziehungen zueinander. (1) Es wird eventuell ein gemeinsames Auszeichnungsvokabular verwendet. Im Beispiel ist dies HTML, da der Text in beiden Sprachen im gleichen Ausgabemedium präsentiert werden soll. (2) Es kann von Gemeinsamkeiten bei der Strukturierung der Dokumente in AS und ZS ausgegangen werden, was z. B. die Trennung und Anordnung der Überschriften, Absätze, Fußnoten etc. anbetrifft.

---

[1] Siehe http://www.text-technology.de [letzter Zugriff: 26.1.2003] sowie Goecke et al., in diesem Band.

[2] Eine genauere Bestimmung des Begriffs „Schemasprache“ findet in Abschnitt 2 statt.



**1. Introduction**

Extensible Markup Language, abbreviated XML, describes a class of data objects called XML documents and partially describes the behavior of computer programs which process them. XML is an application profile or restricted form of SGML, the Standard Generalized Markup Language [ISO 8879]. By construction, XML documents are conforming SGML documents.

XML documents are made up of storage units called entities, which contain either parsed or unparsed data. Parsed data is made up of characters, some of which form character data, and some of which form markup. Markup encodes a description of the document"s storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure.

A software module called an **XML processor** is used to read XML documents and provide access to their content and structure. It is assumed that an XML processor is doing its work on behalf of another module, called the **application**. This specification describes the required behavior of an XML processor in terms of how it must read XML data and the information it must provide to the application. [...]

**A References[...]**

Aho/Ullman
Aho, Alfred V., Ravi Sethi, and Jeffrey D. Ullman. Compilers: Principles, Techniques, and Tools. Reading: Addison-Wesley, 1986, rpt. corr. 1988. [...]
ISO 8879
ISO (International Organization for Standardization). ISO 8879:1986(E). Information processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML). First edition -- 1986-10-15. [Geneva]: International Organization for Standardization, 1986.

**1 Einleitung**

Die Extensible Markup Language, abgekürzt XML, beschreibt eine Klasse von Datenobjekten, genannt XML-Dokumente, und beschreibt teilweise das Verhalten von Computer-Programmen, die solche Dokumente verarbeiten. XML ist ein Anwendungsprofil (application profile) oder eine eingeschränkte Form von SGML, der Standard Generalized Markup Language [ISO 8879]. Durch ihre Konstruktion sind XML-Dokumente konforme SGML-Dokumente.

XML-Dokumente sind aus Speicherungseinheiten aufgebaut, genannt Entities, die entweder analysierte (parsed) oder nicht analysierte (unparsed) Daten enthalten. Analysierte Daten bestehen aus Zeichen, von denen einige Zeichendaten und andere Markup darstellen. Markup ist eine Beschreibung der Aufteilung auf Speicherungseinheiten und der logischen Struktur des Dokuments. XML bietet einen Mechanismus an, um Beschränkungen der Aufteilung und logischen Struktur zu formulieren.

Ein Software-Modul, genannt **XML-Prozessor**, dient dazu, XML-Dokumente zu lesen und den Zugriff auf ihren Inhalt und ihre Struktur zu erlauben. Es wird angenommen, daß ein XML-Prozessor seine Arbeit als Teil eines anderen Moduls, genannt **Anwendung**, erledigt. Diese Spezifikation beschreibt das notwendige Verhalten eines XML-Prozessors soweit es die Frage betrifft, wie er XML-Daten einlesen muß und welche Informationen er an die Anwendung weiterreichen muß.[...]

**7 Anhang A: Referenzen[...]**

Aho/Ullman
Aho, Alfred V., Ravi Sethi und Jeffrey D. Ullman. Compilers: Principles, Techniques, and Tools. Reading: Addison-Wesley, 1986, rpt. corr. 1988. [...]
ISO 8879
ISO (International Organization for Standardization). ISO 8879:1986(E). Information processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML). First edition -- 1986-10-15. [Geneva]: International Organization for Standardization, 1986.

Abb. 1: Beispieldokumente (Ausgangssprache: Englisch; Zielsprache: Deutsch).

Abbildung 2 visualisiert die - etwas vereinfachte - Dokumentstruktur[3] der Dokumente aus Abbildung 1. Sie ist für AS und ZS gleich, was durch die gegebene Domäne der formalen Spezifikationen zu erklären ist. (3) Bedeutsam sind auch Beziehungen auf der linguistisch-funktionalen Ebene. Z. B. wird die unmarkierte Fortführung des Themas bei den obigen Textausschnitten gleichermaßen in AS und ZS vor allem als syntaktisches Subjekt realisiert. Im Deutschen gibt es einige Alternativen, die für eine unmarkierte Themenfortführung zur Verfügung stehen, wie die Verwendung einer Präpositionalphrase. Derartige Beziehungen sind aus einer übersetzungstheoretischen wie auch aus einer Evaluierungsperspektive (z. B. eine Evaluierung der Qualität maschineller Übersetzungen) heraus von hoher Bedeutung (vgl. Hansen und Teich 1999). Ein möglicher Ausgangspunkt für die Ermittlung der Beziehungen sind Annotationen linguistischer Funktionen und ihrer Realisierungsformen (vgl. Abschnitt 4.3).

[3] Die Dokumentknoten sind durch Ellipsen realisiert, welche die verschiedenen Elementnamen enthalten, z. B. <html>.

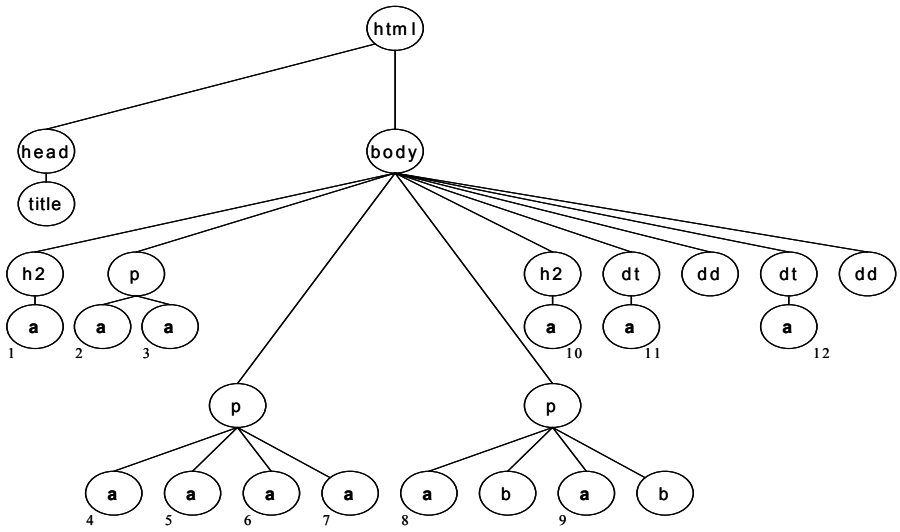



Abb. 2: Struktur der Dokumente aus Abb. 1.

Konsistenz von Textauszeichnungen in AS und ZS sollte hinsichtlich jeder dieser drei Beziehungen gewahrt werden. Was die Beziehungen der ersten und zweiten Art betrifft, so lässt sich Konsistenz bis zu einem gewissen Grad durch die Verwendung eines standardisierten Textextraktionsformats wie XLIFF (Savourel und Reid 2002) aufrechterhalten. Die zu übersetzenden Textsegmente werden dabei aus dem annotierten Dokument extrahiert, manuell oder automatisch übersetzt und mit dem strukturellen Gerüst der Ausgangsdatei zu einem zielsprachlichen Dokument zusammengefügt. Beziehungen der dritten Art sind je nach Grad der Unterschiedlichkeit der Sprachen weitaus schwerer konsistent zu halten, unabhängig davon, ob sie implizit bleiben oder textuell ausgezeichnet sind.

Ein wichtiges Mittel zur Wahrung von Konsistenzen textuell ausgezeichneter Dokumente sind Schemasprachen. Im weiteren Verlauf dieses Artikels werden bestimmte Einschränkungen diskutiert, denen man bei der Konsistenzwahrung durch Schemasprachen unterliegt (Abschnitt 2). Es folgt die Vorstellung eines komplementären Ansatzes (Abschnitt 3) und verschiedener Anwendungsszenarien (Abschnitt 4), die sich in den drei Bereichen der Konsistenzwahrung und -analyse von AS und ZS ergeben. Eine Zusammenfassung und ein Ausblick auf zukünftige Forschungsperspektiven bilden den Abschluss (Abschnitt 5).

## 2 Schemasprachen und Konsistenzwahrung von AS und ZS

Dokumentauszeichnungen werden mittels sogenannter **Auszeichnungssprachen** wie XML vorgenommen. Restriktionen über die Dokumentstruktur werden in **Schemasprachen** formuliert, z. B. XML-DTDs (Bray et al. 2000), XML SCHEMA (Thompson et al. 2001) oder RELAX NG (Clark und Murata 2001). Mittels Schemasprachen werden **Dokumentgrammatiken** verfasst, die Restriktionen für eine Klasse von **Dokumentinstanzen** enthalten.

Zumindest die gegenwärtig verfügbaren Schemasprachen weisen einige Unzulänglichkeiten auf, wenn man sie für die Konsistenzwahrung und -analyse auf die beschriebene, dreifache Art verwenden will. (1) An erster Stelle zu nennen ist die Eingeschränktheit der Restriktionen, die zur Beschreibung der Dokumentstruktur verwendet werden können. Mit Schemasprachen, die die Ausdruckskraft einer kontextfreien Grammatik besitzen, ist es z. B. schwierig, linguistische, syntagmatische Relationen wie Rektions- und Kongruenzbeziehungen zu modellieren, da diese lineare Restriktionen erfordern. (2) Ein anderes Problem ergibt sich bei der Klassifikation von Annotationseigenschaften, sei es auf linguistischer oder dokumentstruktureller Ebene. So haben z. B. alle <a> Elemente in Abbildung 2 die Eigenschaft, mittelbar dem <body> Element untergeordnet zu sein. Einige <a> Elemente stehen in <p> Elementen, von diesen stehen einige am Anfang oder in der Mitte, etc. Die Beschreibung derartiger übergreifender Eigenschaften einerseits und spezifischer Eigenschaften andererseits kann zu einer Klassifikationshierarchie für Annotationseinheiten führen. Klassifikationshierarchien sind in der Übersetzungsforschung ein wichtiges Instrument (vgl. Hansen und Teich 1999 sowie Teich 2001) zur Differenzierung verschiedener, linguistischer Form-Funktion Beziehungen. Somit wären sie auch für eine linguistische Konsistenzüberprüfung - bei entsprechenden textuellen Auszeichnungen - von Interesse. Mit Schemasprachen ist jedoch eine Klassifikation z. B. von sprachübergreifenden und sprachspezifischen Eigenschaften nur in geringem Maße möglich. (3) Schließlich zeigt sich bei verschiedenen Sprachen, dass ähnliche Phänomene schwer miteinander in Beziehung zu setzten sind, weil bei der Textauszeichnung auf verschiedene Auszeichnungsvokabulare zurückgegriffen wird. Dies gründet teils in unterschiedlichen linguistischen Schulen, teils in Phänomenen, welche verschiedene Sprachbeschreibungsebenen betreffen. Ein Mechanismus erscheint deshalb sinnvoll, Vokabulare verschiedener Dokumentgrammatiken aufeinander abzubilden.

Um die genannten Erfordernisse bei der Konsistenzanalyse und -wahrung umsetzen zu können, haben wir mit der Entwicklung eines Beschreibungsformats[4] begonnen, welches die folgenden Eigenschaften besitzt:

- Es dient der Formulierung von strukturellen Bedingungen (Kontextspezifikationen), deren Erfüllbarkeit für bestimmte Informationseinheiten[5] in einer Dokumentinstanz getestet werden.
- Die strukturellen Bedingungen können auch nicht-hierarchische Beziehungen zwischen Informationseinheiten umfassen.
- Die strukturellen Bedingungen können Restriktionen einer existierenden Dokumentgrammatik ergänzen, ohne dass diese abgeändert werden muss.
- Die Informationseinheiten in einer Dokumentinstanz lassen sich hinsichtlich der Erfüllbarkeit unterschiedlicher Bedingungen klassifizieren.
- Die strukturellen Bedingungen und die mit ihnen verbundene Klassifikation von Informationseinheiten sind übertragbar auf Vokabulare unterschiedlicher Dokumentgrammatiken.

Dokumente, die derartige Beschreibungen enthalten, nennen wir CSD-Dokumente (**Context Specification Document**). Im Folgenden wird die Funktionsweise von CSD detailliert beschrieben.

## 3 Funktionsweise von CSD[6]

### 3.1 Formulierung struktureller Bedingungen

Abbildung 3 visualisiert den Aufbau eines CSD-Dokuments. Strukturelle Bedingungen werden nur über die Dokumentknotenstruktur formuliert, Attributnamen und -werte oder Namensräume spielen keine Rolle. Die Identifikation der Knoten findet anhand von Elementnamen statt. Die Bedingungen sind zudem - im Gegensatz zu Schemasprachen - knoten- bzw. pfadzentristisch: In der Terminologie von CSD ist eine strukturelle Bedingung die Beschreibung eines Pfades, dessen Realisierbarkeit für eine bestimmte Menge von Knoten überprüft wird. Abbildung 3 enthält strukturelle Bedin-

[4] Ein bereits bestehender Ansatz, der dem hier vorgestellten ähnelt, ist Schematron (vgl. Jeliffe 2002). Allerdings lässt sich mit Schematron nur schwer die Funktionalität der Klassifikation und der Abbildungen auf unterschiedliche Vokabulare realisieren.

[5] Der Begriff INFORMATIONSEINHEIT ist der W3C-Empfehlung zum XML-Infoset entnommen (vgl. Cowan und Tobin 2001). Beispiele für Informationseinheiten sind das Dokument, Elementnamen, Attributnamen, Namensräume etc.

[6] Die genaue Syntax von CSD-Dokumenten ist in Sasaki und Pönninghaus (2003) und Sasaki (im Druck) beschrieben.

gungen für die Menge der <a> Elemente, wiedergegeben durch den Ausdruck SCOPE="A".

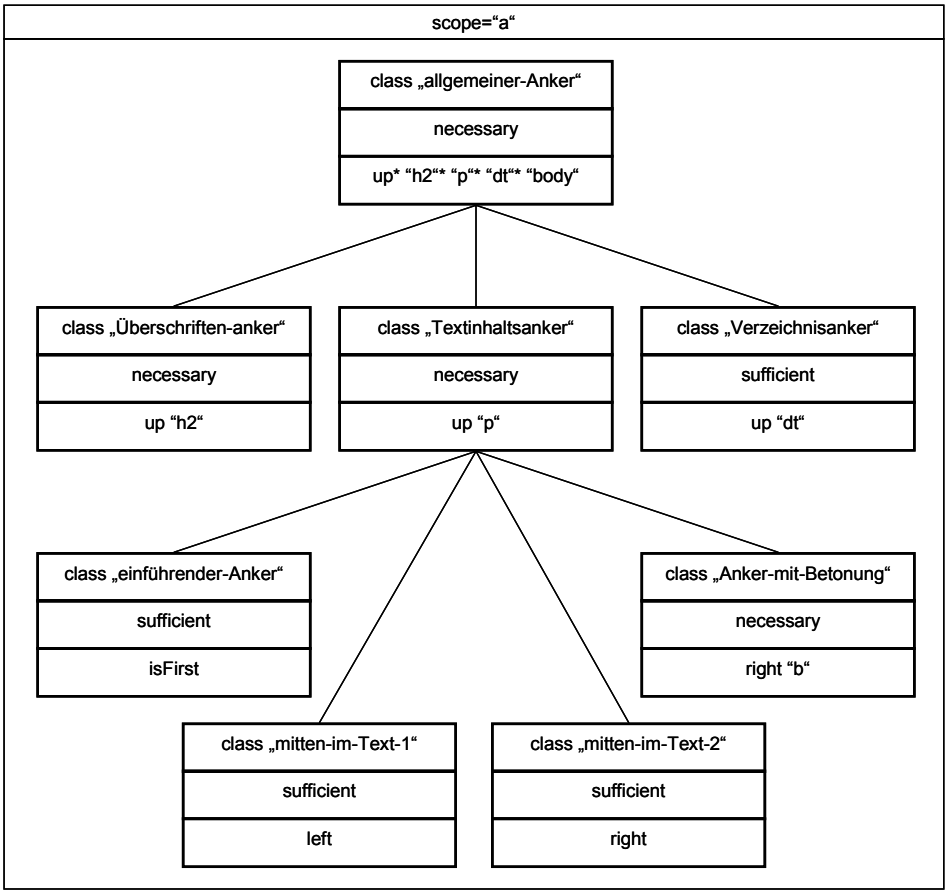



Abb. 3: Visualisierung eines CSD-Dokuments zur Klassifikation von <a> Elementen.

Die Pfade werden mittels sogenannter Caterpillar-Ausdrücke beschrieben, wie sie Brüggemann-Klein und Wood (2000) darstellen. Die ausgewählte Menge der Informationseinheiten gilt zugleich als Initialpunkt der Caterpillar. Ein Caterpillar-Ausdruck beinhaltet mindestens eine der folgenden Bewegungen und Tests über die Dokumentknotenstruktur: LEFT, RIGHT, UP, FIRST, LAST, ISFIRST, ISLAST, ISLEAF, IS-

ROOT. Zudem gibt es den Sternoperator * und einen Namenstest für den Knoten, auf dem sich der Caterpillar gerade befindet. Eine strukturelle Bedingung gilt für den jeweiligen initialen Knoten als erfüllt, wenn alle Bewegungen des entsprechenden Caterpillar-Ausdrucks ausführbar sind und die Tests zu einem positiven Ergebnis führen. Die strukturelle Bedingung UP* "H2"* "P"* "DT"* "BODY"[7] ist für alle <a> Elemente der Abbildung 2 erfüllt, die Bedingung UP "H2" nur für die <a> Elemente mit den Indize 1 und 10.

Die Caterpillar-Ausdrücke in der hier gezeigten Verwendungsweise sind eine relativ ausdrucksschwache Pfadsprache, wenn man sie z. B. mit XPath (Clark und deRose 1999) vergleicht. Es gibt keine Tests auf Attributnamen, -werte, Namensräume oder etwa den Optionalitätsoperator. Diese Beschränkung liegt in dem innerhalb des Projektes Sekimo verfolgten Forschungsziel begründet, Beziehungen zwischen knoten- und pfadzentristischen Bedingungen einerseits und den hierarchischen Strukturbeschreibungen einer Dokumentgrammatik andererseits zu erforschen. Pfadzentristische Bedingungen lassen sich nur hinsichtlich der verarbeiteten Dokumentinstanzen überprüfen. Wenn die pfadzentristischen Bedingungen jedoch in dokumentgrammatische Strukturbeschreibungen überführt werden können, sind diese für eine Klasse von Dokumenten einsetzbar. Dies wäre ein großer Gewinn für die generische Beschreibung von z. B. Übersetzungseigenschaften. Es ist anzunehmen, dass eine minimalisierte Ausdruckskraft der Pfadsprache die Überführung erleichtert.

Ein mögliches Vorgehen bei der Erforschung der Beziehungen von Caterpillar-Ausdrücken und dokumentgrammatischen Strukturbeschreibungen wäre die Formulierung von Restriktionen verschiedener Ausdruckskraft (z. B. in der Reihenfolge reguläre Grammatiken, kontextfreie Grammatiken, kontext-sensitive Grammatiken, Caterpillar-Ausdrücke) und ihr empirischer Vergleich an annotierten Korpora. Möglicherweise differiert das Maß an Einsetzbarkeit für die jeweiligen Restriktionen, in Abhängigkeit von der (linguistischen) Domäne in unterschiedlichen Sprachen. Diese Fragestellung wird hier jedoch nicht weiter behandelt.

### 3.2 Klassifikation von Informationseinheiten

Die strukturellen Bedingungen werden im CSD-Dokument zur Bildung einer Klassifikationshierarchie genutzt, die auf die betreffende Elementmenge angewandt wird. Das Differenzierungs- bzw. Klassenzugehörigkeitskriterium ist die Erfüllbarkeit der strukturellen Bedingungen. Im CSD-Dokument wird jede Klasse CLASS mit einem eindeutigen Namen versehen. In Abbildung 3 steht die Klasse mit der Bezeichnung **„**allgemeiner Anker**“** am höchsten in der Hierarchie. Die <a> Elemente werden in 8 Klassen unterteilt. „Textinhaltsanker“ sind in der Dokumentinstanz aus Abbildung 2 z. B. die

[7] Die Tests der Elementnamen sind in Anführungsstriche gesetzt.

<a> Elemente mit den Indize 2-9, „Verzeichnisanker“ die <a> Elemente mit den Indize 11 und 12. Bei der Klassifikation kommen so genannte hinreichende und notwendige strukturelle Bedingungen zum Einsatz, in Abbildung 2 als SUFFICIENT bzw. NECESSARY gekennzeichnet. So ist es möglich, bestimmte Teilmengen bei der Klassifikation unter Beibehaltung der Hierarchie auszublenden. Die Klasse **„**Textinhaltsanker**“** wird beispielsweise im CSD-Dokument in Abbildung 2 ausgeblendet. Die entsprechende Teilmenge der <a> Elemente mit den Indize 2-9 ist also nur ein Ausgangspunkt für weitere Klassifikationen.

### 3.3 Restriktion und Analyse von Strukturen in Dokumentinstanzen

Abbildung 4 visualisiert die Verarbeitung eines CSD-Dokuments und einer Dokumentinstanz.

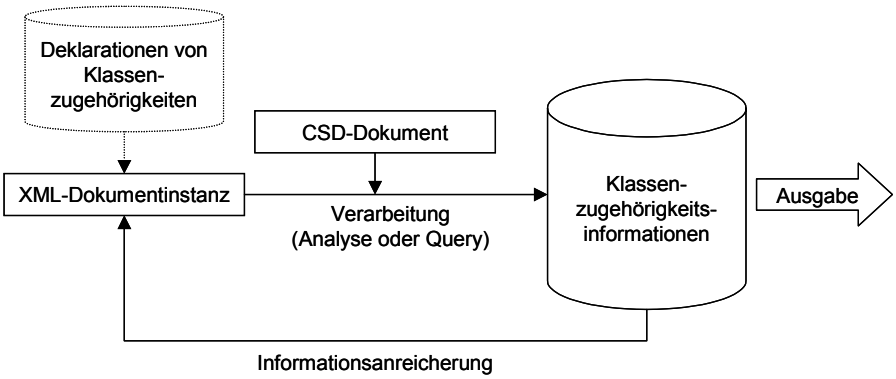



Abb. 4: Verarbeitungsprozess von CSD-Dokument und XML-Dokumentinstanz.

Die Klassifikation der strukturellen Bedingungen in einem CSD-Dokument wird auf die betreffende Elementmenge, z. B. alle <a> Elemente, angewandt. Die Informationen über Klassenzugehörigkeiten der verschiedenen Elementinstanzen kann anschließend in einem separaten Dokument ausgegeben werden, oder sie werden in die verarbeitete Dokumentinstanz integriert.

Die Verarbeitung kann als Analyse oder Validierung vollzogen werden. Bei der Analyse werden diejenigen Knoten gesucht und ausgegeben, für die eine hinreichende Bedingung im CSD-Dokument erfüllt ist. Bei einer Validierung werden die komplementären, also „nicht validierbaren“ Knoten gesucht. Im Falle der Dokumentinstanz aus Abbildung 1 und dem CSD-Dokument aus Abbildung 3 werden die <a> Elemente

mit den Indize 1 und 10 als Ergebnis einer Validierung ausgegeben, da sie keiner hinreichenden Klasse angehören.

Es ist außerdem möglich, Klassenzugehörigkeiten für Knoten in der Dokumentinstanz zu deklarieren. Dies geschieht mittels des CSD:CATERPILLAR-Attributs. Möchte man z. B., dass ein bestimmtes <a> Element immer den Klassen „mitten-im-Text-1“ und „mitten-im-Text-2“ angehören soll, so kann dies durch das Attribut CSD:CATERPILLAR="MITTEN-IM-TEXT-1 MITTEN-IM-TEXT-2" deklariert werden. Wird bei der Verarbeitung eine andere Klassenzugehörigkeit festgestellt, gilt der Knoten als nicht validierbar.

### 3.4 Adaption eines CSD-Dokuments auf verschiedene Namensvokabulare

Es erscheint sinnvoll, die strukturellen Bedingungen bzw. die Pfadausdrücke für verschiedene Elementnamensinventare verwenden zu können (vgl. Abschnitt 2). Hier zeigt sich ein weiterer Vorteil der Caterpillar-Ausdrücke gegenüber XPath: Da die Caterpillar-Ausdrücke nur Tests von Positionen und Elementnamen umfassen, können sie durch eine einfache Abbildung der Namenstests auf ein anderes Vokabular[8] für Dokumentinstanzen einer anderen Dokumentgrammatik eingesetzt werden. Die Abbildung erfolgt auf deklarative Weise in einem separaten Dokument, ohne dass das CSD-Dokument oder die Dokument-Instanz geändert werden müssen. In XPath wäre dies auf Grund der potentiell erheblichen Vielfalt von Tests hinsichtlich Elementnamen, Attributsnamen, Namensräumen, Werten etc. nicht ohne weiteres möglich.

Im Folgenden werden exemplarische Anwendungen von CSD auf die beschriebenen drei Bereiche der Konsistenzanalyse und -wahrung von Übersetzungen vorgestellt.

[8] Es erscheint auch sinnvoll, Abbildungen für bestimmte Verbindungen von Elementnamen und Attributnamen oder -werten vorzunehmen, z. B. alle <p> Elemente mit dem Attribut STYLE="BOLD" auf ein <emph> Element abzubilden. Das daraus resultierende Ausmaß kombinatorischer Möglichkeiten gefährdet jedoch die Untersuchbarkeit von Beziehungen zwischen Dokumentgrammatiken und Caterpillar-Ausdrücken.

# 4 Konsistenz von Textauszeichnungen in Original und Übersetzung

## 4.1 Konsistenz der inneren Struktur von einzelnen, ausgezeichneten Einheiten

Begonnen wird mit der eindeutigsten Beziehung zwischen AS und ZS: Eine annotierte Einheit in der AS entspricht einer Annotationseinheit in der ZS. Dabei kann es vorkommen, dass die Einheiten zwar gleich benannt sind, aber - bis zu einem gewissen Grad - eine unterschiedliche interne Strukturierung aufweisen. Je nach den Konventionen im jeweiligen Sprach- und Kulturraum sind z. B. in bibliografischen Einträgen andere Linearisierungen von Autorennamen (Vor- und Nachnamen), Jahresangaben etc. erforderlich. Zudem kann für unterschiedliche Sprachen ein unterschiedliches Vokabular nötig sein, vgl. „author“ versus „Autor“. Ein CSD-Dokument ermöglicht die Klassifikation der verschiedenen Linearisierungen. Des Weiteren können durch eine Abbildung der Elementnamenstests sprachspezifische Elementvokabulare verwendet werden. In einer Dokumentinstanz wird durch das beschriebene CSD:CATERPILLAR-Attribut die sprach- und kulturspezifische Klassenzugehörigkeit an den betreffenden Knoten deklariert.

## 4.2 Konsistenz von Einheiten hinsichtlich ihrer Position in der Dokumentstruktur

Ein Beispiel für die Konsistenzprüfung von Einheiten hinsichtlich ihrer Position in der Dokumentstruktur haben wir bereits in Abbildung 3 kennen gelernt. Die Klassifikation der <a> Elemente ist sowohl auf die AS-Version als auch auf die ZS-Version des Dokuments anwendbar. CSD kann hier eingesetzt werden, um

- im Text der AS strukturelle Bedingungen für bestimmte Elemente zu spezifizieren,
- die Bedingungen in einer Klassifikationshierarchie anzuordnen und
- mit Hilfe der Klassifikation Texte in einer ZS zu erstellen, sowie diese zu validieren.

Die Klassifikation kann zur Unterstützung multilingualer Texterstellung und -validierung verwendet werden. Dies wird besonders deutlich, wenn die zu Grunde liegende Dokumentgrammatik nicht auszeichnungsorientierte, sondern inhaltsorientierte Elemente verwendet. Ein Beispiel für eine solche Dokumentgrammatik ist die XML Specification („XMLspec“) DTD[9], die auch für die Erstellung der Beschreibung des XML-Standards verwendet wurde. Abbildung 5 visualisiert ein CSD-Dokument zur Klassifikation eines der Elemente dieser DTD, das so genannte <term-

[9] Siehe http://www.w3.org/XML/1998/06/xmlspec-report.htm [letzter Zugriff: 26.1.2003] für weitere Informationen zu dieser DTD.

def> Element, mit dem Definitionen von Termini ausgezeichnet werden. Die Klassifikation differenziert z. B. Definitionen, die Programmcode enthalten (class „3“), die innerhalb von Listen vorkommen (class „12“ und „13“), oder in denen auf andere Definitionen referenziert wird (class „10“).

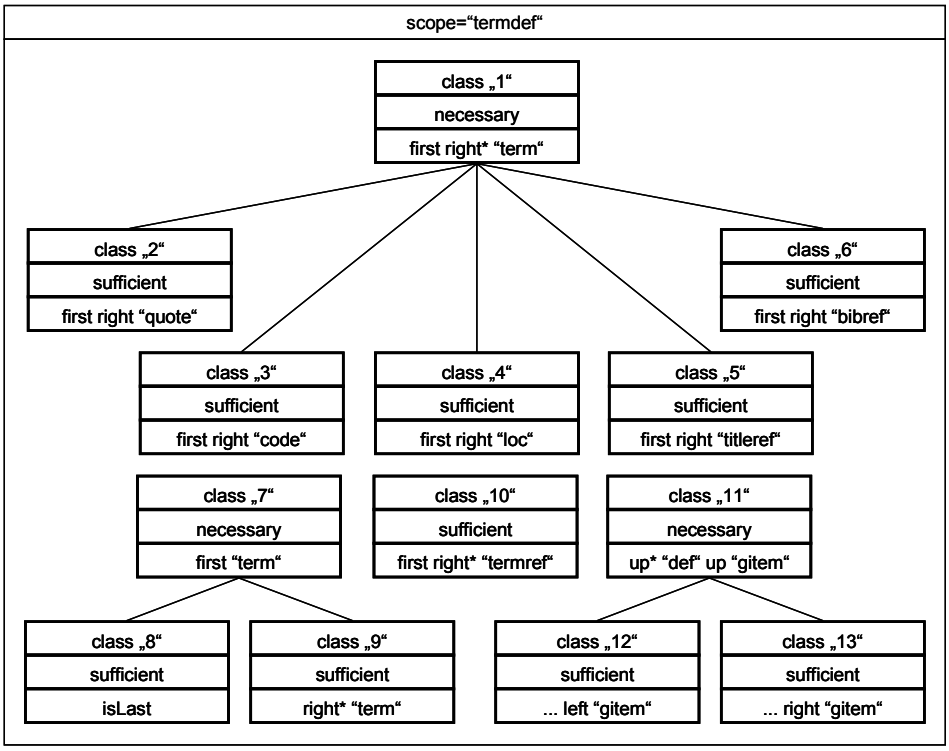



Abb. 5: Klassifikation von <termdef> Elementen, ein Bestandteil der XML Specification („XMLspec“) DTD.

Eine weitere potentielle Verwendung derartiger Klassifikationen ist die Differenzierung von Hypertextsorten. Diese gewinnen z. B. für Anwendungen im Bereich der Informationsextraktion zunehmend an Bedeutung. (vgl. Rehm im Druck) Hypertextsorten ließen sich dabei erfassen durch Klassifikationshierarchien, die sich für bestimmte Annotationseinheiten ergeben.

### 4.3 Konsistenz auf linguistischer Ebene

Die Analyse von Konsistenz auf linguistischer Ebene wird exemplarisch und versuchsweise für den Bereich unmarkierter Themenfortführung durchgeführt. In der Einleitung wurde bereits erwähnt, dass das Deutsche und das Englische einige Unterschiede aufweisen hinsichtlich der Realisierungsformen unmarkierter Themenfortführung. Das Englische nutzt hauptsächlich die Subjektposition, dass Deutsche zusätzlich z. B. auch die Objektposition oder Präpositionalphrasen. Wie sollten solche Unterschiede in linguistisch ausgezeichneten Texten gefunden werden, wie die Konsistenz der Auszeichnung überprüft werden? Abbildung 6 visualisiert ein CSD-Dokument, mit dem dies möglich wäre.

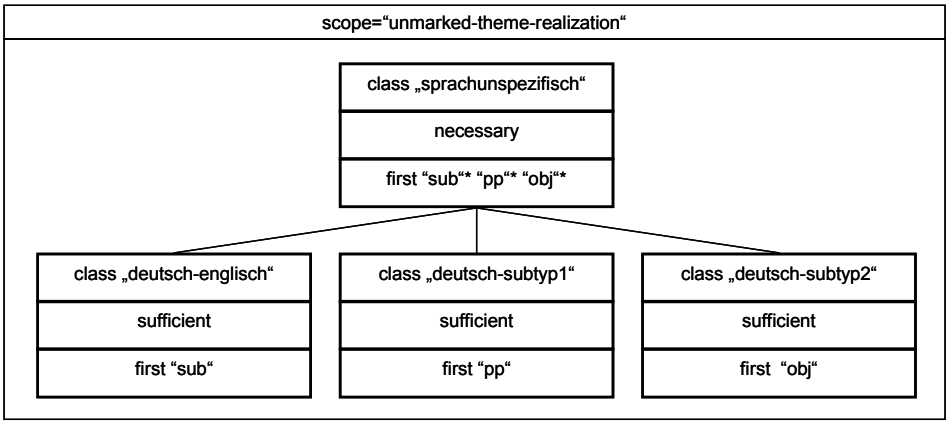



Abb. 6: Sprachübergreifende und sprachspezifische Modellierung unmarkierter Themenfortführungen.

Das CSD-Dokument beinhaltet eine Klasse „sprachunspezifisch", die innerhalb von als <unmarked-theme-realization> ausgezeichneten Einheiten unmarkierter, thematischer Fortführung alle beschriebenen Mittel erlaubt. Dieser untergeordnet sind Klassen, die u. a. spezifische Muster für das Deutsche definieren: „deutsch-subtyp1" und „deutsch-subtyp2". In der Abbildung sind die strukturellen Bedingungen für das Deutsche und das Englische als hinreichend eingeordnet. Das CSD-Dokument ist also für Texte der ZS und der AS geeignet. Wenn nur Texte der AS analysiert bzw. validiert werden sollen, kann dies durch eine Ausblendung der jeweiligen Klassen erreicht werden.

Die hier vorgestellte, versuchsweise und exemplarische Modellierung hat einen starken Bezug zum Vorgehen der systemisch-funktionalen Grammatik (SFG, Halliday

1985) oder zu ihrer Anwendung in der Entwicklung von Modellen, die den Vergleich von Texten in Übersetzungsrelationen anstreben (vgl. Teich 2001). Wichtigstes Merkmal von SFG für die Thematik dieses Artikels ist die Behandlung von Form-Funktionsbeziehungen sprachlicher Einheiten. SFG beinhaltet zum einen eine paradigmatische Sicht, die sprachliche Funktionen in eine klassifikatorische Hierarchie einordnet. Demgegenüber steht die syntagmatische Sicht, die die Realisierungen der Funktionen mit bestimmten sprachlichen Mitteln beschreibt.

Sprachliche Realisierungsformen lassen sich in Form von Dokumentstrukturbeschreibungen (dokumentgrammatisch in Dokumentgrammatiken oder mittels Caterpillar-Ausdrücken in CSD-Dokumenten) spezifizieren. Diese Spezifikationen können zur Realisierung der Klassifikationshierarchie verwendet werden. Ein solches Verfahren vereinfacht die schnelle Überprüfung und Erweiterungen von sprachspezifischen und sprachübergreifenden Modellen der SFG, die anschließend im Bereich der Konsistenzanalyse und -wahrung von Übersetzungen eingesetzt werden könnten.

Bei einer linguistischen Konsistenzüberprüfungen kommen textuelle Auszeichnungen zum Einsatz, die eventuell viele verschiedene linguistische Beschreibungsebenen betreffen: syntaktische Funktionen, morphologische Kategorisierung, Wortreihenfolge, thematische Struktur etc. Die Konstruktion von Caterpillar-Ausdrücken und CSD-Dokumenten gerät zur nahezu unlösbaren Aufgabe, wenn zumindest die hierarchischen, d. h. die dokumentgrammatischen Beziehungen der Ebenen nicht in hohem Maße empirisch abgesichert sind. Zur Analyse der Ebenenbeziehungen bietet sich der Einsatz einer speziell dafür entwickelten Querysprache an, wie sie in Goecke et al. (in diesem Band) vorgestellt wird.

## 5 Zusammenfassung und Ausblick

Dieser Artikel stellte einen zu Schemasprachen komplementären Ansatz vor, der sich zur Konsistenzanalyse und -wahrung für annotierte Texte in Übersetzungsrelationen eignet. Mit CSD lassen sich strukturbezogene Klassifikationen für Informationseinheiten in Dokumentinstanzen erstellen und auf die Namensinventare verschiedener Dokumentgrammatiken abbilden. Klassifikationshierarchien können für verschiedene Arten der Konsistenzwahrung oder -analyse eingesetzt werden: Konsistenz der inneren Struktur einzelner Einheiten, Konsistenz von Einheiten hinsichtlich ihrer Position in der Dokumentstruktur, und Konsistenz von Annotationen linguistischer, form- und

funktionsbezogener Einheiten. Eine erste Implementierung der beschriebenen Funktionalität von CSD liegt vor[10].

Für die Zukunft bleibt auszuloten, wie die Beziehungen von generischen, dokumentgrammatischen Restriktionen für Klassen von Dokumenten einerseits und pfadzentristischen, strukturellen Bedingungen für ausgewählte Knoten in Dokumentinstanzen andererseits sind. Diese Frage wird im erwähnten Forschungsprojekt Sekimo zunächst nicht in der Domäne von Übersetzungen, sondern anhand von Dialog- und Textkorpora unterschiedlicher Sprachen angegangen. Für die Untersuchung von Beziehungen zwischen Texten in Übersetzungsrelation muss exploriert werden, wie weit sich linguistische Theorien in der Übersetzungsforschung (z. b. SFG) anhand von Dokumentgrammatiken und CSD-Dokumenten umsetzen, überprüfen und gegebenenfalls erweitern lassen. Ein großer Gewinn läge dabei in der engen Verbindung, die zwischen Theorie- und Hypothesenbildung einerseits und den textuell ausgezeichneten, sprachlichen Daten andererseits hergestellt werden kann. Die in diesem Artikel vorgestellten Beispiele haben bisher nur exemplarischen Charakter. Sie demonstrieren jedoch eine Bandbreite von Anwendungsmöglichkeiten, die eine strukturbezogene, klassifikatorische Sicht auf Informationseinheiten in Dokumenten für den Bereich der übersetzungsbezogenen Konsistenzwahrung und -analyse eröffnet.

[10] Beispieldokumente und Dokumentgrammatiken zur Erstellung von CSD-Dokumenten und Dokumenten zur Abbildung der Elementnamenstests finden sich unter http://www.text-technology.de/csd/ [letzter Zugriff: 26.1.2003] Eine Implementierung in Python ist auf Anfrage erhältlich. Ein Onlineprozessor für CSD-Dokumente und Dokumentinstanzen findet sich unter http://kalk.lili.uni-bielefeld.de/caterpillar/csd.html [letzter Zugriff: 26.1.2003]

## Literatur

Bray, T., J. Paoli, C. M. Sperberg-McQueen und E. Maler (2000): Extensible Markup Language (XML) 1.0 (Second Edition). W3C Empfehlung, 6. Oktober 2000. Siehe http://www.w3.org/TR/REC-xml/ [letzter Zugriff: 26.1.2003]

Brüggemann-Klein, A. und D. Wood (2000): Caterpillars: Caterpillars: A Context Specification Technique. In: Markup Languages 2(1), 81-106.

Clark, J. und S. deRose (1999): XML Path Language (XPath) Version 1.0. W3C Empfehlung, 16. November 1999. Siehe http://www.w3.org/TR/xpath/ [letzter Zugriff: 26.1.2003]

Clark, J. und M. Murata (2001): Relax NG Specification. Siehe http://www.oasis-open.org/committees/relax-ng/spec-20011203.html [letzter Zugriff: 26.1.2003]

Cowan, J. und R. Tobin (2001): XML Information Set. W3C-Empfehlung, 24. November 2001. Siehe http://www.w3.org/TR/xml-infoset/ [letzter Zugriff: 26.1.2003]

Goecke, D., D. Naber und A. Witt (in diesem Band): Query von Multiebenen-annotierten XML-Dokumenten mit Prolog.

Halliday, M. A. K. (1985): An introduction to functional grammar. London: Edward Arnold.

Hansen, S. und E. Teich (1999): Kontrastive Analyse von Übersetzungskorpora: Ein funktionales Modell. In: Gippert, J. (Hrg.): Multilinguale Corpora: Codierung, Strukturierung, Analyse. Prag: Enigma.

Jeliffe, R. (2002): The Schematron Assertion Language 1.5. Siehe. http://www.ascc.net/xml/resource/schematron/Schematron2000.html [letzter Zugriff: 26.1.2003]

Mehler, A. und H. Lobin (Hrg.) (im Druck): Werkzeuge zur automatischen Analyse und Verarbeitung von Texten: Formate, Tools, Softwaresysteme. Opladen: Westdeutscher Verlag.

Rehm, G. (im Druck): Ontologie-basierte Hypertextsorten-Klassifikation als Grundlage generischer Informationsextraktion. In: Mehler, A. und H. Lobin (Hrg.).

Sasaki, F. (im Druck): Strukturbezogene Klassifikation von Informationseinheiten in texttechnologischen Korpora. In: Mehler, A. und H. Lobin (Hrg.).

Sasaki, F. und J. Pönninghaus (2003): Testing structural properties in textual data: beyond document grammars. Erscheint in: Literary and Linguistic Computing, vol. 18.

Savourel, Y. und J. Reid (2002): XLIFF 1.0 Specification. Siehe http://www.oasis-open.org/committees/xliff/documents/xliff-specification.htm [letzter Zugriff: 26.1.2003]

Teich, E. (2001): Towards a model for the description of crosslinguistic divergence and commonality in translation. In: Steiner, E. und C. Yallop (Hrg.): Exploring Translation and Multilingual Text Production: Beyond Content. Berlin: de Gruyter.

Thompson, H., D. Beech, M. Maloney und Noah Mendelsohn (2001): XML Schema Part 1: Structures. W3C Empfehlung, 2. Mai 2001. Siehe http://www.w3.org/TR/xmlschema-1/ [letzter Zugriff: 26.1.2003]