



WESTFÄLISCHE  
WILHELMS-UNIVERSITÄT  
MÜNSTER

Arbeitsbereich Linguistik, FB 9  
PROF. DR. W. PAPROTTÉ

## Sachlicher Abschlußbericht

Autor: Hendrik Feddes, Lea Cyrus

<b>Projekt</b>	RomaniS – Romanische Sprachen für die Sekundarstufe I
<b>Teilprojekt</b>	Multifunktionale Annotation von Lehrmaterialien für die romanischen Sprachen
<b>FKZ</b>	08NM232
<b>Laufzeit</b>	September 2002–August 2005

GEFÖRDERT VOM



Bundesministerium  
für Bildung  
und Forschung

## Gliederung (gemäß BNBest-BMBF 98)

<b>1</b>	<b>Kurzdarstellungen</b>	<b>4</b>
1.1	Aufgabenstellung . . . . .	4
1.2	Voraussetzungen . . . . .	4
1.3	Planung und Ablauf des Vorhabens . . . . .	5
1.4	Wissenschaftlicher und technischer Stand . . . . .	7
1.5	Zusammenarbeit mit anderen Stellen . . . . .	7
<b>2</b>	<b>Eingehende Darstellungen</b>	<b>8</b>
2.1	Erzieltes Ergebnis . . . . .	8
2.1.1	Konkretes Datenmodell . . . . .	8
2.1.1.1	Zentrale Aspekte des Datenmodells . . . . .	9
2.1.1.2	Lernzielkataloge . . . . .	10
2.1.1.3	Visualisierung . . . . .	11
2.1.1.4	Werkzeuge . . . . .	13
2.1.2	Datenmodell als Musterlösung . . . . .	13
2.1.2.1	Typisierung . . . . .	13
2.1.2.2	Strukturierte Dokumentation . . . . .	14
2.1.2.3	Metadaten . . . . .	15
2.1.3	Datenmodell als Fallstudie . . . . .	16
2.2	Voraussichtlicher Nutzens, Verwertbarkeit . . . . .	17
2.2.1	Konkretes Datenmodell . . . . .	17
2.2.2	Datenmodell als Musterlösung . . . . .	18
2.2.3	Datenmodell als Fallstudie . . . . .	18
2.3	Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen . . . . .	19

2.4 Erfolgte oder geplante Veröffentlichungen des Ergebnisses . . . . . 19

# 1 Kurzdarstellungen

## 1.1 Aufgabenstellung

Das Projekt „Multifunktionale Annotation von Lehrmaterialien für die romanischen Sprachen“ ist ein ergänzendes Teilprojekt zum Verbundprojekt „RomaniS – Romanische Sprachen für die Sekundarstufe I“, das vom Ernst Klett Verlag, Stuttgart, koordiniert wurde.

Im Rahmen des Hauptprojekts „RomaniS“ war es Ziel des Klett Verlags, eine modulare Unterrichtssoftware zu entwickeln, die den handlungs- und prozessorientierten Fremdsprachenunterricht Französisch in der Sek. I unterstützt, aber auch auf weitere Fremdsprachen ausgeweitet werden kann. Aufgabe dieses Teilprojekts war es, für diese Software ein multifunktionales Datenmodell auf XML-Basis zu entwickeln.

Aus Sicht der Datenmodellierung waren vor allem folgende Komponenten der Software von Belang:

**Präsentationsmodul** multimediale Stoffpräsentation zur eigenständigen Erarbeitung

**Trainingsmodul** multimediale Übungen zu Grammatik, Wortschatz und Textverständnis

**Werkzeugmodul** allgemeine Lerntechniken, Lexikonarbeit, interaktive Lernkartei und Grammatik

Jede dieser Komponenten stellt unterschiedliche Anforderungen an das zugrundeliegende Datenmodell. So ist es z. B. für die Übungen im Trainingsmodul zentral, Metainformationen über Lernziele der Übungen bereitzustellen, um auf dieser Basis die Benutzerführung dynamisch aufgrund von individuellen Lernerleistungen anpassen zu können. Auch die Strukturierung und Modellierung solcher Metainformationen gehörte zu den Aufgaben des Teilprojekts.

Eine weitere Aufgabe war die Annotation von XML-Daten auf der Grundlage des erstellten Datenmodells. Dies diente zum einen der Evaluation des Datenmodells, zum anderen als Machbarkeitsstudie für das im Laufe des Projekts entstandene Redaktionssystem, das den Redakteuren die direkte Annotation von XML-Daten abnimmt.

## 1.2 Voraussetzungen

Der Arbeitsbereich Linguistik kann auf Erfahrungen in zwei einschlägigen Projekten verweisen, die die Erstellung von multimedialen Lehr- und Lernsystemen zum Inhalt

hatten bzw. haben. Im zusammen mit der Universität Bielefeld durchgeführten Projekt „Linguistik und Neue Medien“, das im Rahmen der NRW-Initiative „Neue Medien in der Hochschullehre“ zwei Jahre lang gefördert wurde und im Jahr 2000 zum Abschluß kam, wurde die Lehr- und Lernplattform BabelOn (<http://luna.lili.uni-bielefeld.de/babelon/>) für den Einsatz an Universitäten sowohl inhaltlich konzipiert als auch technisch implementiert.

Dieses Projekt ist in einem ebenfalls vom BMBF im Rahmen des Programms „Neue Medien in der Bildung“ geförderten Verbundprojekt mit dem Titel „Hochschulverbund zum Einsatz curricularer Bausteine in der Sprach- und Kommunikationswissenschaft – multimediale Lehr- und Lernressourcen für Präsenzlehre, Selbststudium und Weiterbildung“ fortgeführt worden. In diesem Projekt, an dem neun Institutionen beteiligt sind, hat die Universität Münster – wiederum zusammen mit der Universität Bielefeld – eine java-basierte Lehr- und Lernumgebung für Linguistik entwickelt.

Darüber hinaus verfügt der Arbeitsbereich Linguistik über eine Reihe weiterer projektrelevanter Kompetenzen und Ressourcen in den Bereichen der Computer- und Korpuslinguistik, lexikalischen Semantik und Lexikographie. Insbesondere bestehen langjährige Erfahrungen bei der Akquisition, elektronischen Aufbereitung und morphologischen und syntaktischen Annotation großer Textkorpora, auf deren Basis lexikalische Daten generiert wurden, die die Grundlage für die Herstellung elektronischer Wörterbücher bilden.

### 1.3 Planung und Ablauf des Vorhabens

Im folgenden wird der Projektablauf knapp und in Anlehnung an die ausführlichere Beschreibung in den Zwischenberichten dargestellt. Für weitere Details wird auf diese verwiesen.

**Vorbereitende Arbeiten** (1)<sup>1</sup> Kick-off-Meeting; Abstimmung mit dem Software-Partner M.I.T; Aufbau einer verschlüsselten Verbindung zum Klett Verlag; Einrichtung eines Versionsverwaltungssystems (CVS); Einarbeitung der Mitarbeiter in XML; Pflichtenheftphase

**Beginn der Datenmodellentwicklung** (1) Erstellung einer DTD für den sogenannten Übungskatalog (Verzeichnis aller für die Lernsoftware vorgesehenen Übungstypen)

**Umstellung auf XML Schema** (2) Überführung des bisherigen DTD-basierten Datenmodells auf ein XML Schema-basiertes Datenmodell

---

<sup>1</sup>Die Zahlen in Klammern beziehen sich auf die Berichtshalbjahre: **1** 09.2002–02.2003; **2** 03.2003–08.2003; **3** 09.2003–02.2004; **4** 03.2004–08.2004; **5** 09.2004–02.2005; **6** 03.2005–08.2005.

**Beginn der Dokumentation des Datenmodells** (2) Dokumentation des Übungskatalogs; Integration der Dokumentation in das Datenmodell unter Ausnutzung der Möglichkeiten von XML Schema

**Datenvisualisierung** (2) Entwicklung von Web-basierten Werkzeugen zur Visualisierung der annotierten Daten; dadurch zusätzliche Validierung der annotierten Daten möglich

**Werkzeuge für das Projektmanagement** (2) In-House-Entwicklung von auf das Datenmodell angepaßten, Web-basierten Werkzeugen zum Projektmanagement:

- „Xref“ verwaltet Querverweise zwischen Pflichtenheft und Datenmodell
- „Changelog“ verzeichnet Änderungen am Datenmodell
- „Todo“ verwaltet eine mit Pflichtenheft/Datenmodell verknüpfte Aufgabenliste

**Webseite** (2) Bereitsstellung aller Projektergebnisse über eine teilweise geschützte Webseite (<http://romanis.uni-muenster.de/>)

**Modularisierung des Datenmodells** (3) Umstellung von einem monolithischen auf ein modulares Datenmodell im Rahmen der Weiterentwicklung des Redaktionssystems (Hauptprojekt)

**Lernzielkatalog** (3) Beginn der Entwicklung von Lernzielkatalogen für die Adaptive Lernerinformation (ALi); Integration der Lernzielkataloge in das Datenmodell

**Annotation auf modularer Basis** (4) weitere Annotation von Lernmaterialien auf der Basis des modularen Datenmodells

**Weitere Lernzielkataloge** (4) Entwicklung weiterer Lernzielkataloge für verschiedene Lehrwerke des Klett Verlags

**Dokumentation des modularen Datenmodells** (5) Beginn der ausführlichen Dokumentation des vollständigen Datenmodells

**Konkordanzwerkzeuge** (5–6) Entwicklung der Konkordanzwerkzeuge; Bereitstellung über die Projektwebseite

**Dokumentation in Schema integriert** (6) Abschluß der Integration der Dokumentation in das Schema-basierte Datenmodell

**Evaluation des Datenmodells** (6) Abschluß der Evaluation des Datenmodells im Rahmen einer Übergabedokumentation für den Koordinator

## 1.4 Wissenschaftlicher und technischer Stand

Die für dieses Teilprojekt zentrale Technologie ist XML mit all seinen verwandten Technologien für Validierung und Visualisierung der Daten, ferner Standardtechnologie für Darstellung von Daten und Bereitstellung von Werkzeugen im World Wide Web. Im Detail seien genannt:

- Extensible Markup Language (XML) und Document Type Definitions (DTDs)<sup>2</sup>
- XML Schema<sup>3</sup>
- Extensible Stylesheet Language for Transformations (XSLT)<sup>4</sup>
- XPath<sup>5</sup>
- Hypertext Markup Language (HTML)<sup>6</sup>
- Common Gateway Interface (CGI)<sup>7</sup>
- Practical Extraction and Report Language (Perl)<sup>8</sup>

Als Fachliteratur wurde Standardliteratur zu den genannten Technologien benutzt, vor allem aus dem Verlag O'Reilly.<sup>9</sup>

## 1.5 Zusammenarbeit mit anderen Stellen

Die Arbeit ist in enger Abstimmung mit dem Klett Verlag geschehen, der das Hauptprojekt koordiniert hat. Für die technische Umsetzung des Softwareprodukts hat Klett die Firma M.I.T beauftragt. Insbesondere bei Themen technischer Natur gab es eine direkte Zusammenarbeit mit M.I.T.

---

<sup>2</sup>URL <http://www.w3.org/TR/2004/REC-xml-20040204>

<sup>3</sup>URL <http://www.w3.org/XML/Schema>

<sup>4</sup>URL <http://www.w3.org/TR/xslt>

<sup>5</sup>URL <http://www.w3.org/TR/xpath>

<sup>6</sup>URL <http://www.w3.org/TR/xhtml1>

<sup>7</sup>URL <http://www.w3.org/CGI>

<sup>8</sup>Wall, Larry *et al.*: Programming Perl. Sebastopol *et al.*: O'Reilly, 2000.

<sup>9</sup>URL <http://www.oreilly.com>

## 2 Eingehende Darstellungen

### 2.1 Erzieltes Ergebnis

Das von dem Teilprojekt erzielte Ergebnis läßt sich unter drei Aspekten beschreiben und bewerten:

- als konkretes, in XML implementiertes Datenmodell für Lehrwerke des Klett Schulbuchverlags, unter der Berücksichtigung der besonderen Gegebenheiten des Verlags (Workflow in der Redaktion) und der Lernsoftware, einschließlich der Lernzielkataloge für die Adaptive Lernerinformation und einiger zusätzlicher Werkzeuge,
- als Musterlösung (*best practice*) für einen allgemeinen Ansatz zur Datenmodellierung von Lehrwerken unter Verwendung zahlreicher mit XML verbundener Technologien und unter Ausnutzung spezifischer Eigenschaften dieser Technologien,
- als Fallstudie für Maßnahmen, um eben diese „ideale“ Datenmodellierung mit den Einschränkungen, die aus den technischen Rahmenbedingungen der Implementation des Modells in ein konkretes Softwareprodukt erwachsen, in Einklang zu bringen.

Auf diese drei Aspekte gehen wir im folgenden differenziert ein.

#### 2.1.1 Konkretes Datenmodell

Die konkreten Arbeiten für das Projekt umfassen die Entwicklung des Datenmodells, die inhaltliche Entwicklung und XML-Implementierung von Lernzielkatalogen für die Adaptive Lernerinformation und die Entwicklung von Werkzeugen zur Visualisierung und zum Projektmanagement.

Sämtliche genannten Ergebnisse werden im folgenden kurz skizziert. Sie stehen außerdem über die Webseite des Teilprojekts zur Verfügung:

**URL** <http://romanis.uni-muenster.de>



**2.1.1.1 Zentrale Aspekte des Datenmodells** Das entwickelte Datenmodell zeichnet sich vor allem durch die folgenden Aspekte aus:

**Plattformunabhängigkeit** XML ist ein offener Standard und ein Klartextformat, so daß maximale Plattformunabhängigkeit gewährleistet ist. Über für praktisch alle Plattformen zur Verfügung stehenden Werkzeuge und APIs kann leicht auf die Daten zugegriffen werden. Die mit dem Modell erstellten Daten sind also nicht an eine bestimmte Rechner-/Betriebssystemkonfiguration gebunden.

**Wiederverwertbarkeit** XML ist ein *transformierbares* Format, d. h., daß die Daten über (wiederum offene und standardisierte) Transformationsmechanismen (z. B. XSLT) in andere Formate überführt werden können. Diese anderen Formate sind zum einen die Darstellung in der Schüler-Software selbst, zum anderen die Darstellung im Rahmen des Redaktionssystems. Als Fallstudie wurde vom Teilprojekt eine HTML-Darstellung der Daten für den Webbrowser entwickelt. Denkbar sind aber auch stärker inhaltlich orientierte Transformationen, beispielsweise auch die Transformation von Übungspooldaten in konkrete Übungssequenzen für bestimmte Lehrwerke.

**Mediennutralität** Mit der Wiederverwertbarkeit geht die Mediennutralität von XML einher: Da logische Struktur der Daten und ihre Darstellung strikt voneinander getrennt sind und die Darstellung erst durch die *Transformation* (s. o.) der Daten in ein Darstellungsformat geschieht, ist die Datenhaltung in XML medienneutral. Die auf der Grundlage des Datenmodells erstellten Daten sind also leicht auf andere Medien (Webseiten, Druckwerke) zu übertragen. Vgl. hierzu beispielhaft Abschnitt 2.1.1.3.

**Validierbarkeit** XML ist ein *validierbares* Format, d. h. es ist über standardisierte Mechanismen möglich, zu überprüfen, ob gewisse Daten strukturell mit dem zu Grunde liegenden Datenmodell konform sind. Somit ist es eine wesentliche Eigenschaft des entwickelten Datenmodells, daß Lehrmaterialien, die mit diesem Modell erstellt worden sind, unabhängig von einem Redaktionssystem oder einer Lernsoftware auf Korrektheit überprüfbar sind.

**XML Schema** Die dem Datenmodell zu Grunde liegende Dokumentstrukturbeschreibungssprache ist XML Schema, die den bisherigen Standard DTD (*document type definitions*) teilweise abgelöst hat. Der Vorteil von XML Schema gegenüber DTDs ist vor allem die wesentlich stärkere Ausdruckskraft, von der im Projekt ausgiebig Gebrauch gemacht wurde. Auf diesen Aspekt gehen wir in Abschnitt 2.1.2.1 näher ein.

**Integration vom Metadaten** Ein weiterer und für das Datenmodell entscheidender Vorteil von XML Schema ist die Tatsache, daß XML Schema selbst ein XML-Format ist und so leicht die Einbettung von Metadaten erlaubt, auf die mit den

gleichen Mechanismen zugegriffen werden kann wie auf die übrigen Inhalte des Modells. Dies wurde zum einen für menschenlesbare Metadaten ausgenutzt, um die Dokumentation des Datenmodells in das Modell selbst zu integrieren; zum anderen wurden maschinenlesbare Metadaten in das Modell integriert. Auf diese Aspekte gehen wir in den Abschnitten 2.1.2.2 und 2.1.2.3 näher ein.

**Kompaktheit** Mit diesem Punkt schließlich sollen die zuvor genannten Aspekte der Validierbarkeit und der Integration von Metadaten noch einmal zusammengefaßt und betont werden: Durch die Verwendung von XML und XML Schema wurde es möglich, zwei wesentliche Bestandteile von Datenmodellen, nämlich die Mechanismen zur Konsistenzprüfung der Daten und die Dokumentation des Modells, im Modell selbst kompakt zu integrieren.

**2.1.1.2 Lernzielkataloge** Für drei Lehrwerke wurden Lernzielkataloge entwickelt, die sich an die in den Lehrwerken behandelte Grammatik anlehnen. Zweck dieser Lernzielkataloge ist Klassifikation von Übungen. So ist es mit dieser Klassifikation für einen Redakteur ein leichtes, eine passende Übung aus einem Übungspool zu wählen.

Der Lernzielkatalog findet auch Verwendung in der sogenannten Adaptiven Lernerinformation: Wenn die Software den Fehlern, die die Schüler bei einer bestimmten Übung machen, eine Fehlerklasse (d. h. ein Eintrag im Lernzielkatalog) zuordnen kann, können für den betreffenden Schüler weitere (einfachere oder vertiefende) Übungen gewählt werden, die mit dem gleichen Lernziel klassifiziert sind. Durch eine Hierarchisierung der Lernziele ist hier eine flexible Auswahl möglich.

Der folgende Ausschnitt zum grammatischen Bereich „Negation“ aus dem Lernzielkatalog für das Lehrwerk *Découvertes* soll der Illustration dienen:<sup>10</sup>

- **Negation** lzk\_523801\_sb\_neg
  - Negation allgemein lz\_523801\_sb\_neg\_allg
  - **Negationspartikeln** lzk\_523801\_sb\_neg\_part
    - \* Negationspartikeln allgemein lz\_523801\_sb\_neg\_part\_allg
    - \* ne...pas lz\_523801\_sb\_neg\_part\_nepas
    - \* ne...plus lz\_523801\_sb\_neg\_part\_neplus
    - \* ne...jamais lz\_523801\_sb\_neg\_part\_nejam
    - \* ne...rien lz\_523801\_sb\_neg\_part\_nerien
  - **Kombination mit Begleiter** lzk\_523801\_sb\_neg\_bgl

<sup>10</sup>Typographische Legende: **Kategorie** – Lernziel – Id – Erläuterung – (Beispiele)

- \* Kombination mit Begleiter allgemein lz\_523801\_sb\_neg\_bgl\_allg
  - \* Verwendung von "de" lz\_523801\_sb\_neg\_bgl\_de *Negation als Mengenangabe: "de" statt unbest. Artikel* (z. B. Elle n'a pas de chiens.)
  - \* Sonderregelungen lz\_523801\_sb\_neg\_bgl\_sond mit "être"; mit "aimer"; mit *Redewendungen* (z. B. Ce n'est pas un chien. Je n'aime pas les chiens.)
- Bei zusammengesetzten Verbformen lz\_523801\_sb\_neg\_zus (z. B. Je ne vais pas dormir./Je n'ai pas dormi.)

Wie alle übrigen Projektergebnisse sind auch die Lernzielkataloge in XML implementiert worden. Die Darstellung sowohl für diesen Abschlußbericht (Satzsystem  $\text{\LaTeX}$ ) als auch für die Webseite (HTML), auf der die vollständigen Kataloge zur Verfügung stehen, wurde aus XML-Quellen transformiert und zeigt somit nebenbei die Medienneutralität des gesamten Ansatzes.

**2.1.1.3 Visualisierung** Um die Medienneutralität zu illustrieren und für eine schnelle visuelle Kontrolle der Annotation zusätzlich zur Validierung, wurde im Teilprojekt eine prototypische Visualisierung implementiert, die die XML-Daten in ein einfaches HTML-Format für die Betrachtung in einem Web-Browser transformiert.

Die Visualisierung für den folgenden (gekürzten) Ausschnitt aus den XML-Daten für einen sogenannten Integrierten Hörtext findet sich in Abbildung 1.

```

<iht id="pk_54721_U1_Text">
  <ihtLektionsTeilTitel>
    <pkSatz soundRef="sounds/au01990521_pk54721_0259.mp3">
      <pkTextDaten>The <pkWortform
5         id="wf_54721_1_000060">Frog</pkWortform> (I)</pkTextDaten>
      </pkSatz>
    </ihtLektionsTeilTitel>
    <pkScreen id="pk_54721_U1_Text_3" template="PK1020_0001">
      <bild bildRef="images/SE46547210_032_03-2.jpg"/>
10     <pkTextBlock>
      <pkFliesstext>
        <pkSatz soundRef="sounds/au01990521_pk54721_0291.mp3">
          <pkTextDaten>It&quot;s
15           <pkWortform id="wf_54721_1_000075">teatime</pkWortform>.
          </pkTextDaten>
        </pkSatz>
        <pkSatz soundRef="sounds/au01990521_pk54721_0292.mp3">
          <pkTextDaten>It's_Simon's turn to set the
20           table.</pkTextDaten>
        </pkSatz>
      </pkFliesstext>
    </pkTextBlock>
  </pkScreen>
</iht>

```

```

25 </pkTextBlock>
    <!-- [...] -->
    <pkNeueVok matRef="matid_65bdb11fdq"
        wortFormRefs="wf_54721_1_000075" />
    <!-- [...] -->
</pkScreen>
<!-- [...] -->
</iht>

```

#### HTML-Export GLNBy

[Unit 01 The Burtons and the Dixons](#) [Unit 05 A weekend at the seaside](#)

[Intro: The Burtons and the Dixons](#) [Language A: We're busy](#) [Language B: Breakfast at the Dixons' house!](#) [Text: The Frog \(1\)](#)

[Screen 1](#) [Screen 2](#) [Screen 3](#)



It's **teatime**. It's Simon's turn to set the table.

**Becky** : What about the plates, Simon?

**Simon** : Oh, yes. They're in the **cupboard**

**Mrs B** : Becky, can you **call** Dad? **Tea**'s ready.

**Becky** : OK, Mum.

**Simon** : Aaah! Mum! Mum!

**Mrs B** : What's the matter, Simon? Are you OK?

**Simon** : A frog! In the cupboard! Look!

**Mrs B** : That's silly.

**Simon** : Really! Look!

**Mrs B** : But how can a frog get into our cupboard? Becky?

**Becky** : Ask Simon.

[Grammatik einblenden](#) [Vokabeln/Grammatik ausblenden](#)

[Zurück zur Band-Auswahl](#) [Zurück zu RomaniS Intern](#)

Abbildung 1: Screenshot eines prototypischen HTML-Exports eines Integrierten Hör-texts (aus einem testweise annotierten englischen Lehrwerk); die neuen Vokabeln dieses Texts sind bereits eingblendet (gelbe Hervorhebungen)

Es ist in dieser Visualisierung auch gut zu sehen, wie ein großer Teil der späteren Programmfunktionalität (hier die Möglichkeit des Einblendens neuer Vokabeln (im Beispiel Element `pkNeueVok`, Zeile 24f.) und neuer Grammatik in den Text) bereits in den XML-Daten angelegt ist und leicht auch in einer einfachen HTML-Visualisierung implementiert werden kann (Buttons „Vokabeln einblenden“ bzw. „Grammatik einblenden“ in Abbildung 1).

**2.1.1.4 Werkzeuge** Für das Projektmanagement wurden folgende Werkzeuge entwickelt (sämtlich auf XML-Datenbasis mit einer Web-Schnittstelle):

**Xref** Dieses Werkzeug verwaltet die Querverweise vom Pflichtenheft auf das Datenmodell. Während die Querverweise aus dem Datenmodell auf das Pflichtenheft in das Datenmodell integriert werden konnten (vgl. hierzu detailliert Abschnitt 2.1.2.2), mußten die Verweise vom Pflichtenheft separat verwaltet werden, da das Pflichtenheft nicht in XML vorlag. Das Werkzeug ermöglichte so die einfache Abarbeitung der „Pflichten“ und eine leichte Kontrolle des Fortschritts der Implementierung.

**Changelog** Mit diesem Werkzeug wurden Änderungen am Datenmodell dokumentiert. Hier wurde der Weg einer Eigenentwicklung gewählt, um eine Verknüpfung mit dem Datenmodell zu ermöglichen. So sind sämtliche dokumentierten Änderungen, wenn möglich, direkt mit den entsprechenden Elementen und Attributen des Datenmodells verknüpft, so daß die Änderungen im Team leicht verfolgt werden konnten.

**Todo** Eine Liste zu erledigender Arbeiten am Datenmodell wurde ebenfalls mit einer Eigenentwicklung verwaltet, aus ähnlichen Gründen: Eine direkte Verknüpfung der Aufgaben mit Pflichtenheft, Datenmodell und Changelog wurde so ermöglicht.

## 2.1.2 Datenmodell als Musterlösung

Bei der konkreten Modellierung der Daten der Klett-Lehrwerke wurden auf Basis einiger Eigenschaften des zugrundeliegenden Standards XML Schema Techniken angewandt, die über das konkrete Datenmodell hinausgehen und somit allgemein bei der Annotation von vergleichbaren Daten nützlich sein können. Auf diese „Musterlösungen“ soll im folgenden eingegangen werden.

**2.1.2.1 Typisierung** Gegenüber DTDs, die keine Datentypen kennen, hat XML Schema ein reichhaltiges und erweiterbares Typsystem. Schon die fest integrierten Typen wie `xs:string` oder `xs:integer` erlauben eine größere Präzision bei der Validierung der Daten, gerade im Zusammenspiel mit der exakteren Modellierung, die XML Schema generell erlaubt.

Eine über diese Basistypen hinausgehende Typisierung bietet aber gerade im Zusammenspiel mit einer Software, die die Daten zu interpretieren hat, große Vorteile. So kann das Vorhandensein eines bestimmten Datentyps, z. B. „Vokabel-Übung mit Sprechblasen“, mit einer bestimmten Klasse des Programm-Codes verknüpft werden,

die für diese Daten zuständig ist. Diese Information kann wiederum als Metadatum im Datenmodell selbst festgehalten werden (vgl. hierzu Abschnitt 2.1.2.3).

Auch mit einfachen Datentypen kann eine Verknüpfung mit der Software sinnvoll genutzt werden. So kann beispielsweise das Vorhandensein eines Datentyps `xs:boolean` dahingehend interpretiert werden, daß das Redaktionssystem, also die Software, die für diese Daten eine graphische Eingabemöglichkeit bereitstellt, an dieser Stelle eine Ja-/Nein-Antwort mit entsprechenden Buttons vorsieht. Ein Datentyp `xs:enumeration`, also eine Aufzählung diskreter Werte, kann als Aufklapp-Menü realisiert werden. Auf diese Weise wird der Programmieraufwand erheblich reduziert und ein großer Teil der Programmlogik bereits im Datenmodell abgebildet.

**2.1.2.2 Strukturierte Dokumentation** Ähnliches gilt für die in das Datenmodell integrierte strukturierte Dokumentation, denn auch diese kann direkt von einer sie interpretierenden Software genutzt werden. Ein allgemeiner Hilfe-Knopf kann im Redaktionssystem an jedem Eingabeelement vorgesehen werden und gegebenenfalls die direkt aus dem Datenmodell extrahierte Information dem Benutzer darstellen. Ein Vorteil von XML Schema, daß nämlich die Strukturbeschreibungssprache selbst XML und somit parsbar ist, kommt hier voll zum Tragen.

Der folgende Ausschnitt aus der Erweiterung der XML-Schema-DTD erläutert die Struktur der integrierten Dokumentation und zeigt, daß es ein leichtes wäre, aus dem Datenmodell selbst eine lesbare Dokumentation zu generieren, z. B. per XSLT:

```

<!ELEMENT docRomanis
                                ((docElement,docAttribut*,docComment?) |
                                (docType,docComment?)) >
<!ELEMENT docElement          (docName?,docDesc) >
5 <!-- docRefs                 Verweise (Pflichtenheft-IDs) auf die
                                relevanten Teile des Pflichtenhefts -->
<!ATTLIST docElement
          docRefs             CDATA #IMPLIED >
<!ELEMENT docName              (#PCDATA) >
10 <!ELEMENT docDesc            (#PCDATA) >
<!ELEMENT docAttribut         (docName,docDesc) >
<!ATTLIST docAttribut
          docRefs             CDATA #IMPLIED >
<!ELEMENT docComment          (#PCDATA) >
15 <!ELEMENT docType           (docName?,docDesc) >

```

Auf einen weiteren Aspekt soll noch hingewiesen werden, nämlich auf das in der Dokumentation verwandte Attribut `docRefs` (Zeilen 8 und 13). An dieser Stelle werden Verweise auf das dem Datenmodell zugrundeliegende Pflichtenheft registriert. Somit wird Zugriff auf eine noch umfangreichere Dokumentation geboten.

Wenn das Pflichtenheft ebenfalls in XML vorläge, könnte es natürlich direkt in die Do-

kumentationsstruktur integriert werden. So muß aber dieser Weg gewählt werden, und vor allem für Querverweise vom Pflichtenheft auf das Datenmodell muß ein gänzlich externes Verfahren genutzt werden. Hierzu wurde ein eigener Index definiert, der in einer eigenen XML-Datei gepflegt wird. Zu den Stellen im Pflichtenheft, wo es sinnvoll erscheint, werden Verweise auf das Datenmodell gegeben. Der Verweis auf das Datenmodell wird dabei in der standardisierten XPath-Syntax angegeben, mit der auf Stellen in einem XML-Dokument verwiesen werden kann.

**2.1.2.3 Metadaten** Mit „Metadaten“ sind hier in das Modell integrierte Daten über das Modell gemeint. Hierzu gehört natürlich auch die eben besprochene Dokumentation, die jedoch, wenn auch über den Weg einer automatisierten Transformation, für den Menschen gedacht ist. In diesem Abschnitt soll es nun um rein maschinell interpretierte Daten gehen.

Der folgende Ausschnitt aus der Erweiterung der XML-Schema-DTD zeigt die im Datenmodell verwendeten Metadaten:

```

<!ELEMENT appInfoRomanis
                                EMPTY                                >
<!--      class      Class
          mediatype  vom Type benoetigtes Medienformat
5         xsd        vom Type benoetigtes XML-Schema
          toplevel   vom Type benoetigtes Top-Level-
                                Element des Schemas                -->
<!ATTLIST appInfoRomanis
10      class      CDATA      #IMPLIED
          mediatype CDATA      #IMPLIED
          xsd       CDATA      #IMPLIED
          toplevel  CDATA      #IMPLIED >

```

Das Attribut `class` (Zeile 9) ist der Ort, an dem Verknüpfungen des Datenmodells mit konkreten Implementierungen, sei es in Form der Schülersoftware oder in Form eines Redaktionssystems, ihren Platz haben. Dieser Mechanismus ist im vorhergehenden Abschnitt 2.1.2.1 bereits beschrieben worden.

Weiterhin hervorzuheben sind die Attribute `xsd` und `toplevel` (Zeilen 11 und 12), die zusammengenommen einen bestimmten Elementtyp einer Schema-Spezifikation bezeichnen. Auf diese Weise kann der Verzicht auf einen ID/IDREF-Mechanismus, wie er im Zuge der Modularisierung des Datenmodells (vgl. dazu den folgenden Abschnitt 2.1.3) nötig wurde, ausgeglichen werden: Mithilfe dieser Metadaten-Spezifikation kann ein externer Validierungsmechanismus, z. B. als Teil eines Redaktionssystems, sicherstellen, daß bestimmte Verweise, z. B. auf eine bestimmte Übung vom Typ „Vokabel-Übung mit Sprechblasen“, auch tatsächlich XML-Daten referenzieren, die genau dieser bestimmten Struktur entsprechen.

### 2.1.3 Datenmodell als Fallstudie

Als letzter Aspekt soll auf die Entwicklung des Datenmodells im Sinne einer Fallstudie eingegangen werden. Während der Entwicklung des Modells ist es nämlich zu einer grundlegenden Änderung gekommen, die aufgrund ihrer Ursachen erwähnenswert ist. Diese Erfahrungen können für vergleichbare Projekte herangezogen werden, um unnötige Irrwege zu vermeiden.

XML ist grundsätzlich ein dokumentenzentriertes Format: Die zu modellierenden Daten werden als ein Dokument angesehen, dessen Struktur mit einer Dokumentenbeschreibungssprache (DTDs oder Schema-Sprachen) spezifiziert wird. XML stellt Mechanismen bereit, um die strukturelle Integrität eines solchen Dokuments sicherzustellen (Validierung). Diese Sichtweise bietet sich auch für ein Lehrwerk an, sei es ein Buch oder eine Software: Eine Lektion hat eine wiederkehrende Struktur, besteht vielleicht aus Abschnitten, Übungen und Vokabular, und niemals wird z. B. eine Übung eine neue Vokabel einführen, oder in der Vokabelliste nicht plötzlich eine Übung auftauchen.

XML kann aber auch als datenzentriertes Format genutzt werden. In diesem Fall beschreibt es die Struktur von einzelnen „Objekten“ einer „Sammlung“. Wie diese Objekte aber zu einem „Dokument“ zusammengesetzt oder genutzt werden, wird über zusätzliche Mechanismen geregelt, z. B. über ein XML-Dokument, das diese Objekte einbindet. Diese Sichtweise bot sich für den sogenannten „Übungskatalog“ an, in dem sämtliche im Lehrwerk vorkommenden Übungstypen beschrieben wurden, nicht aber eine konkrete Abfolge von Übungen in einer Lektion. Dies würde z. B. erst in dem Übungsabschnitt einer (dokumentzentrierten) Lektion geschehen.

Das gesamte Lehrwerk mit seinen „Objekten“ auf diese Weise zu einem Dokument zusammenzuführen, erschien vorteilhaft. Zum einen ist hier die einfache Validierbarkeit zu nennen. Zum anderen kann auf diese Weise der gerade für komplexe Daten wichtige ID/IDREF-Mechanismus von XML genutzt werden, der es erlaubt, sicherzustellen, daß Verweise auch tatsächlich auf vorhandene Elemente verweisen, und daß die „Namen“ von Elementen, auf die verwiesen wird, eindeutig sind.

Zwei Gründe haben dazu geführt, daß unser Datenmodell von einem monolithischen in ein modulares überführt werden mußte, das aus mehreren XML-Dokumenten besteht. Zum einen war dies die mangelnde Handhabbarkeit sehr großer Dokumente durch die Software, im Redaktionssystem, vor allem aber auch zur Laufzeit der Lernsoftware. Zum anderen hat sich, begründet durch den Workflow in einer Redaktion mit zahlreichen externen Mitarbeitern, die Notwendigkeit ergeben, flexibel und mit einer granularen Rechteverwaltung auf Teile der Daten zugreifen und diese Teile jederzeit auslagern zu können. Ein „Herausschneiden“ von Teildaten aus dem Gesamtdokument hätte hier keine Lösung gebracht, da diese Dokumentteile auf Grund von nicht mehr



auflösbaren Referenzen potentiell nicht mehr valide gewesen wären.

Ein völliger Verzicht auf diese XML-internen ID/IDREF-Mechanismen durch eine weitgehende Auslagerung der Referenzvalidierung auf externe Mechanismen wäre unbefriedigend gewesen. Durch die Nutzung von XML Schema konnte jedoch ein zufriedenstellender Kompromiß gefunden werden, der die Validierung zwar auslagert, jedoch dabei auf im Datenmodell in Form von Schema-Metadaten vorhandene Informationen zurückgreift. Dies geschieht durch die Elemente `xsd` und `toplevel`, die in Abschnitt 2.1.2.3 beschrieben worden sind.

## 2.2 Voraussichtlicher Nutzens, Verwertbarkeit

Wie in Abschnitt 2.1 möchten wir auch bezüglich des Nutzens und der Verwertbarkeit des Ergebnisses des Teilprojekts differenzieren zwischen dem konkreten Datenmodell, dem Datenmodell als Musterlösung und der Entwicklung des Datenmodells als Fallstudie.

### 2.2.1 Konkretes Datenmodell

Es liegt in der Natur der Sache, daß das konkrete, für die Lehrwerke entwickelte Datenmodell (vgl. Abschnitt 2.1.1.1) sich stark an deren Struktur und Besonderheiten orientiert. Zwar wurde, wann immer möglich, versucht, möglichst allgemein zu modellieren, jedoch sind dieser Abstraktion stets Grenzen gesetzt, die vor allem darin begründet sind, daß die Lernsoftware die im Datenmodell vorliegende Strukturierung umsetzen können muß.

Für Lehrwerke des Klett-Verlags ist eine Wiederverwertbarkeit gegeben und wurde auch belegt, indem das Datenmodell für drei recht unterschiedliche Lehrwerke<sup>11</sup> zum Einsatz kam. Für Lehrwerke anderer Verlage müßte eine Übertragbarkeit zunächst geprüft werden. Es ist allerdings anzunehmen, daß eine Portierung mit nicht geringem Aufwand verbunden ist, insbesondere, was den zentralen Teil des Datenmodells, den sogenannten Übungskatalog, angeht, da die zahlreichen vorliegenden Übungstypen nicht standardisiert sind.

Was die entwickelten Werkzeuge (vgl. Abschnitt 2.1.1.4) angeht, ist eine Portierung auf vergleichbare Projekte leicht denkbar. Durch die Verwendung der Skriptsprache Perl und die Nutzung einer plattform-neutralen Web-Oberfläche ist eine große Flexibilität gegeben. Anpassungsbedarf bestünde nur in den für die Querverweise genutzten

---

<sup>11</sup>In der Frühphase des Projekts für das englische Lehrwerk *Green Line New*, dann für die beiden Französisch-Lehrwerke *Découvertes* und *Tous ensemble*.

XML-Attributen des jeweiligen Datenmodells. Dies sollte jedoch nur einen geringen Aufwand darstellen.

Ebenfalls gut für weitere Projekte nutzbar sind die entwickelten Lernzielkataloge (vgl. Abschnitt 2.1.1.2), die den grammatischen Lernstoff einer bestimmten Lernstufe strukturiert repräsentieren. Hier ist eine Adaption für vergleichbare Lehrwerke leicht möglich.

### 2.2.2 Datenmodell als Musterlösung

Unter dem Aspekt der Musterlösung sehen wir den größten Nutzen des entwickelten Datenmodells für vergleichbare Projekte. Es sind die in Abschnitt 2.1.2 genannten allgemeinen Modellierungsprinzipien, die an dieser Stelle noch einmal knapp zusammengefaßt werden sollen:

**Typisierung** starke Ausnutzung des Typensystems von XML Schema → exaktere Modellierung (bessere Validierbarkeit) und Möglichkeit der Integration eines Großteils der Programmlogik in das Datenmodell

**strukturierte Dokumentation** Integration dokumentierender Metadaten in das Datenmodell (XML Schema) → Kompaktheit des Datenmodells

**Metadaten** Integration von Metadaten, die von Programmen auszuwerten sind, in das Datenmodell (XML Schema) → Kompaktheit und wiederum Integration eines Teils der Programmlogik in das Datenmodell

### 2.2.3 Datenmodell als Fallstudie

Auch unter diesem Aspekt sehen wir einen Nutzen für vergleichbare Projekte. Es ist dies die „Lektion“, auf die wir in Abschnitt 2.1.3 eingegangen sind und die zur Entwicklung des modularen Datenmodells geführt hat. In Kürze zusammengefaßt bedeutet dies, daß der Workflow derjenigen, die mittelbar oder direkt mit dem Datenmodell arbeiten (z. B. eine Redaktion), einen großen Einfluß auf die Modellierung haben kann. In unserem Fall ergab sich die Notwendigkeit der Modularisierung auf XML-Ebene, um ein granulares Rechteemanagement zu ermöglichen. Diese Erfahrung kann vergleichbaren Projekten von Nutzen sein, um mögliche Irrwege zu vermeiden.

### **2.3 Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen**

Bezug nehmend auf den letzten Zwischenbericht, ist an dieser Stelle allenfalls zu erwähnen die im Projektzeitraum stark gestiegene Verbreitung von XML Schema und die damit einhergehende Verfügbarkeit verlässlicher Validationswerkzeuge, was die frühzeitige Entscheidung, von DTDs auf XML Schema zu wechseln, positiv befördert hat.

### **2.4 Erfolgte oder geplante Veröffentlichungen des Ergebnisses**

Die Projektergebnisse stehen über die Projekt-Webseite zur Verfügung:

**URL** `http://romanis.uni-muenster.de`

Desweiteren wird der öffentliche Teil dieses Abschlußberichts auf dem offiziellen Publikationsserver der Universität Münster veröffentlicht:

**URL** `http://miami.uni-muenster.de`

Dies sichert die dauerhafte Verfügbarkeit der Projektergebnisse. Weitere Veröffentlichungen sind zur Zeit nicht geplant.