

DIPLOMARBEIT

Mobile eLearning Content on Demand for hylOs

eingereicht von

Patricia Bartel

s0505485

Erstgutachterin: Prof. Dr. Debora Weber-Wulff

Zweitgutachter: Prof. Dr. Ing. Carsten Busch

Berlin, 12. Februar 2007

Fachbereich 4 – Wirtschaftswissenschaften II
Studiengang Internationale Medieninformatik
der Fachhochschule für Technik und Wirtschaft (FHTW) Berlin

Inhaltsverzeichnis

Kapitel 1	Einleitung	6
1.1	Motivation der Arbeit.....	6
1.2	Ziel der Arbeit.....	7
1.3	Aufbau der Arbeit.....	7
Kapitel 2	Mobiles Lernen – Mobile Learning	9
2.1	Kontext des Mobile Learnings.....	10
2.2	Abgrenzung eLearning – mLearning – Blended Learning.....	10
2.3	Rahmenbedingungen.....	11
2.3.1	Mobile Endgeräte.....	11
2.3.2	Datenübertragung.....	12
2.3.3	Inhaltsaufbereitung.....	13
2.3.4	Pädagogische Betrachtung.....	14
2.4	Mobiles Lernen – Akzeptanz und Interesse anhand einer Studie der Fernuniversität Hagen.....	16
2.5	Grenzen und Potenziale des mobilen Lernens.....	18
Kapitel 3	Technologische Grundlagen	19
3.1	Extensible Markup Language – XML	19
3.1.1	Extensible Stylesheet Language – XSL	22
3.1.2	XML Path Language – XPath	23
3.1.3	XML Linking Language – XLink	24
3.2	Der Webserver und Servlet Container Jetty	25
3.3	Das Cocoon-Framework.....	26
3.4	Hypermedia Learning Object System.....	28
3.4.1	Systemarchitektur und Dokumentengenerierung in hylOs.....	29
3.4.2	Erzeugung der Ausgabeformate.....	31
3.5	PlayStation Portable.....	32
Kapitel 4	Ist-Analyse	33
4.1	Hypermedia Learning Object System.....	33
4.1.1	Die Autorenumgebung	33
4.1.2	XML-Dokumenttypen in hylOs	37
4.1.3	Aufbau eines generierten XML-Dokumentes in hylOs.....	38
4.2	Funktionalitäten und Merkmale der PlayStation Portable	42
4.2.1	Basisfunktionalitäten der PlayStation Portable.....	42
4.2.2	Der Internetbrowser der PlayStation Portable.....	44
4.3	Beispiele für Lernsoftware.....	48
4.3.1	Lernsoftware für PlayStation Portable.....	48
4.3.2	Lernsoftware für Wirtschaftsingenieure.....	50

Kapitel 5	Anforderungen	53
5.1	Allgemeine Anforderungen.....	53
5.2	Inhaltliche Anforderungen.....	53
5.3	Designspezifische Anforderungen.....	53
5.4	Technische Anforderungen.....	54
Kapitel 6	Konzeption	56
6.1	Aufgabendefinition.....	56
6.2	Zielgruppendefinition.....	57
6.3	Anwendungsszenarien.....	57
6.3.1	Szenario 1: Schüler.....	57
6.3.2	Szenario 2: Student.....	58
6.3.3	Szenario 3: Handelsvertreter.....	59
6.4	Erstellung und Abruf von Lerninhalten.....	60
6.5	Navigationskonzept der Lerneinheit.....	61
6.5.2	Navigationsmöglichkeiten.....	63
6.6	Besondere Funktionalitäten der Anwendung.....	67
6.6.1	Wissenstest.....	68
6.6.2	Hilfefunktionen.....	68
6.6.3	Allgemeine Statusinformationen.....	69
Kapitel 7	Entwurf der Benutzeroberfläche	70
7.1	Layout der Lernumgebung.....	70
7.2	Sonderlayout für Hilfefunktionen.....	72
Kapitel 8	Implementierung und Test	74
8.1	Erstellung der Benutzeroberfläche.....	75
8.2	Technische Inhaltsaufbereitung.....	76
8.3	Lösungsansatz 1.....	76
8.3.1	Import der XML-Daten.....	77
8.3.2	Extraktion ausgewählter Elemente.....	79
8.3.3	Formatierung der Inhaltsobjekte.....	82
8.3.4	Test – Lösungsansatz 1.....	83
8.4	Lösungsansatz 2.....	86
8.4.1	Erstellung des XSLT-Stylesheets	86
8.4.2	Import der notwendigen XML-Dokumente.....	88
8.4.3	Extraktion ausgewählter Objekte.....	89
8.4.4	Formatierung der Inhaltsobjekte in Flash.....	89
8.4.5	Test Lösungsansatz 2.....	90
8.5	Anmerkungen und Komplikationen.....	90
Kapitel 9	Zusammenfassung und Ausblick	92
	Glossar und Abkürzungsverzeichnis	94

Abbildungsverzeichnis	98
Listingverzeichnis	99
Tabellenverzeichnis	100
Literaturverzeichnis	101
ANHANG – A	105
ANHANG – B	127
ANHANG – C	134

DANKSAGUNG

An dieser Stelle möchte ich allen Personen danken, die zum Gelingen dieser Arbeit beigetragen haben.

Im Besonderen danke ich ganz herzlich Arne Hildebrand und Prof. Dr. Deborah Weber-Wulff für die engagierte, freundliche und geduldige Betreuung während der Durchführung dieser Diplomarbeit.

Weiterhin möchte ich meiner Mama Gabriele Bartel, Fariha Ahmadzai und Nick Sebold für die liebevolle und geduldige Unterstützung, während des gesamten Studiums und besonders in der letzten Wochen, danken.

Konrad Jung und Ranty Islam möchte ich für die kritischen Anregungen danken, die zur Verständlichkeit des Inhaltes der Arbeit beigetragen haben.

Kapitel 1 Einleitung

1.1 Motivation der Arbeit

Mobiltelefone, digitale Musikplayer, PDAs¹, Smart Phones, tragbare Spielkonsolen, Tablet-PCs², Laptops etc. – die Liste der mobilen Endgeräte wird ständig erweitert und die Funktionalitäten der einzelnen Geräte wird immer vielfältiger. Aufgrund der Vielfalt an Geräten und Funktionalitäten lassen sich immer mehr Anwendergruppen erschließen, die zumindest eines der aufgezählten Geräte nutzen.

Hinzu kommt das stetige Wachstum der Mobilen Industrie, eine zunehmend flächendeckende Abdeckung durch Hochgeschwindigkeitsnetzwerke und eine ebenso schnelle Weiterentwicklung der kabellosen Netze. Damit kann man schon heute überall per Mobiltelefon erreichbar sein und in naher Zukunft orts- und zeitunabhängig mit seinem Mobilgerät ins Internet gehen, um sich Informationen zu beschaffen, zu interagieren, zu kommunizieren. All diese Weiterentwicklungen eröffnen neue Zugangsmöglichkeiten für den Bildungsmarkt und fordern und fördern neue Entwicklungen darin.

Mobile Endgeräte verändern unser Leben und begleiten uns täglich, somit kann nicht ausgeschlossen werden, dass auch die Wissensvermittlung in mobilen Kontexten stattfinden muss. Zeit- und ortseffizient und auch spielerisch sollen mobile Begleiter die Attraktivität und Alltagstauglichkeit von Bildungsprozessen steigern. Allerdings bedingt diese Vision eine spezialisierte – inhaltlich und technisch auf die Zielplattform abgestimmte – Aufbereitung von Lerninhalten ausgehend von bestehenden eLearning-Methoden. Denn nur wenige Inhalte lassen sich in identischer Form auf einem PC und einem mobilen Gerät darstellen und nutzen. Daher müssen eLearning-Angebote nicht nur technisch, sondern auch inhaltlich und didaktisch so überarbeitet werden, dass sie sinnvoll auf mobilen Endgeräten genutzt werden können.

Bezüglich ihrer relativ geringen Abmessungen und ihrer integrierten Steuerelemente, weisen Mobile Endgeräte häufig eine von gängigen Benutzeroberflächen abweichende Bedienoberfläche auf. Deshalb ist die Entwicklung von neuen technisch und didaktisch innovativen Methoden, die über reguläre Online-Usability³-Konzepte hinausgehen, erforderlich.

¹ Personal Digital Assistant,

² sehr flacher Computer, bei dem Eingaben über den Bildschirm per Stift oder Finger getätigt werden können

³ Usability bezeichnet Bedienfreundlichkeit von Webangeboten, also Web-Oberflächen, deren Bedienung ohne Hilfestellungen möglich ist

Die beiden persönlichen Herausforderungen der Autorin dieser Arbeit bestehen erstens in der Fragestellung, inwieweit sich die PlayStation Portable als ein mobiles Unterhaltungsmedium überhaupt für Lernzwecke eignet und wie sich die erwähnten innovativen Konzepte darauf übertragen lassen. Zweitens soll geprüft werden, ob sich hylOs als ein Lernobjektmanagementsystem¹ dazu eignet, Lerninhalte dynamisch auf verschiedene Plattformen zu überführen.

1.2 Ziel der Arbeit

Das Ziel dieser Diplomarbeit besteht in der plattformgerechten Überführung von eLearning-Inhalten aus hylOs auf die PlayStation Portable, um den Nutzern orts- und zeitunabhängiges – *mobiles* – Lernen zu ermöglichen.

Der Fokus liegt dabei auf der technischen Umsetzung: Dem Benutzer soll es möglich sein, mit seiner PlayStation Portable Lernkurse auf einer Webseite nach seinem Bedarf auszuwählen und auf der PlayStation Portable zu nutzen.

Der Mehrwert für den Nutzer der PlayStation Portable ergibt sich aus der erweiterten konstruktiven Nutzung eines Unterhaltungsmediums für die Wissensvermittlung.

1.3 Aufbau der Arbeit

Die vorliegende Diplomarbeit untergliedert sich in folgende Kapitel:

Kapitel 2 und 3 dienen zur Beschreibung der Grundlagen einerseits in Bezug auf Mobiles Lernen und andererseits werden die technischen Zusammenhänge die zum Verständnis dieser Arbeit notwendig sind erläutert.

In Kapitel 4 werden die zwei Hauptsysteme, die während der Arbeit genutzt wurden, genauer analysiert und anschließend wird bestehende Lernsoftware aus dem Bereich Mobile Learning und spielorientiertes Lernen vorgestellt.

In Kapitel 5 werden die Ergebnisse der Analyse zusammengefasst und entsprechende Anforderungskriterien für die zu realisierende Applikation definiert.

In Kapitel 6 wird das Konzept der Anwendung resultierend aus der Anforderungsdefinition vorgestellt und es werden Anwendungsfälle für mögliche Nutzergruppen der Applikation erarbeitet.

¹ bezeichnet eine Software, mit der Lerninhalte erstellt, verwaltet und angeboten werden können. Alternative Bezeichnung zu Lernplattform

Kapitel 7 erläutert den Entwurf der Benutzeroberfläche der Anwendung und die damit verknüpften Funktionalitäten in Hinblick auf die grafische Gestaltung.

In Kapitel 8 präsentiert die Realisierung der beiden Lösungsansätze und zeigt die Testergebnisse beider Ansätze und die jeweils aufgetretene Probleme.

Abschließend fasst Kapitel 9 die Arbeit zusammen und gibt einen Ausblick auf zukünftige Entwicklungen.

Kapitel 2 Mobiles Lernen – Mobile Learning

Die Errungenschaften der neuen mobilen Technologien halten mit dem computergestützten Lernen (eLearning¹) Einzug in die Lehre. Mobile Learning – Mobiles Lernen oder auch mLearning genannt – verfolgt die konsequente Entwicklung des eLearning in Bezug auf Ortsunabhängigkeit. Das heißt, Lerninhalte können jederzeit und überall abgerufen werden im Sinne eines „Learning on Demand“.

Mobile Learning ist noch ein recht junges Forschungsgebiet, das die Methoden des eLearning aufgreift und sie um orts- und zeitunabhängige Funktionalitäten erweitert. Somit hat Mobiles Lernen zum Ziel, dem Lernenden bedarfsgerechte und allgegenwärtig uneingeschränkt zugängliche Lerninhalte bereitzustellen.

Da die Entwicklungen im Bereich Mobile Learning noch sehr neu sind, gibt es noch keine standardisierte Definition des Begriffes Mobile Learning. Es gibt aber verschiedene „Visionen“ und Fragen rund um das Thema Mobile Learning, die darauf abzielen, inwieweit sich mLearning von eLearning in besonderer Weise abhebt und die Lehre um neue Methodiken revolutionieren kann. [FS04]

Einige dieser „Visionen“ werden im Folgenden aufgeführt:

1. Clark N. Quinn:

„mLearning is the intersection of mobile computing and eLearning: accessible resources wherever you are, strong search capabilities, rich interaction, powerful support for effective learning, and performance-based assessment. eLearning independent of location in time or space.“ [LIN]

2. Simon Geddes:

„mLearning is the acquisition of any knowledge and skill through using mobile technology, anywhere, anytime, that results in an alteration in behaviour.“ [KNO]

3. Giasemi N. Vavoula, Mike Sharples:

“Three ways learning can be considered mobile “learning” is mobile in terms of space; it is mobile in different areas of life; it is mobile with respect of time”. [VS02]

So unterschiedlich sich die Visionen auch darstellen, die gemeinsame Aussage lautet: Mobile Learning wird dem Anwender eine Umgebung schaffen, die es ihm erlaubt, Lernangebote nach seinen Bedürfnissen, unabhängig von Ort und Zeit, mobil zu nutzen.

¹ zu dt. electronic learning – elektronisch unterstütztes Lernen

2.1 Kontext des Mobile Learnings

Als Alternative zur klassischen Präsenzlehre haben sich viele Lernformen im Laufe der technologischen Entwicklungen herausgebildet, die ein weitestgehend ortsunabhängiges Lernen charakterisieren: diese Lernformen werden unter dem Begriff des „Distance Learning“ (Fernunterricht oder auch Heimunterricht) zusammengefasst. Hierzu zählen demnach grundsätzlich alle Lernformen, die nicht über einen physisch anwesenden Lehrkörper vermittelt werden, sondern über sämtliche Medien, digitale wie analoge. Diese Art der Lernvermittlung ist nicht nur wirtschaftlicher, sondern auch barrierefreier im Hinblick auf die unbeschränktere Verfügbarkeit von Lerninhalten bezüglich Ort und Zeit (vgl. [PAL]). In diesen Kontext fallen demnach nicht nur die computergestützte bzw. webbasierte Lehre, die unter dem Begriff des eLearning vereint sind, sondern auch „archaische“ Lernverfahren über Video (bspw. Bildungssendungen im Fernsehen) oder Rundfunk. Wenn die Methoden des eLearning unterstützend (simultan) oder ergänzend (zeitlich asynchron) zur klassischen Präsenzlehre genutzt werden, dann wird diese Lernform als „Blended Learning“ (Mischform) bezeichnet (vgl. [STA]). Da die Begriffe eLearning, Blended Learning und Mobile Learning sich zum Teil überschneiden und bisweilen synonym gebraucht werden, soll die folgende Abbildung deren Relationen skizzieren, bevor diese im Folgenden Abschnitt erklärt und zueinander abgegrenzt werden.

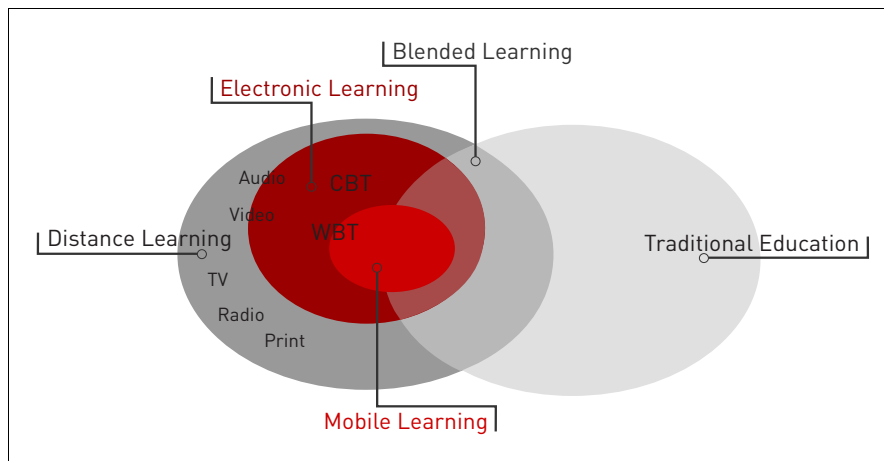


Abbildung 1: Kontext des Mobile Learnings

2.2 Abgrenzung eLearning – mLearning – Blended Learning

E-Learning ist das computergestützte (CBT – Computer Based Training) wie auch webbasierte Lernen (WBT – Webbased Training) mit multimedial und interaktiv aufbereiteten

Inhalten. Mobile Learning ist insofern als eine Nachfolgetechnologie des eLearning zu behandeln, als es seine Lernformen um portable und zusätzliche drahtlose Technologien und Geräte derart erweitert, dass entsprechende Anpassungen vonnöten sind. Ein wesentlicher Unterschied von eLearning und mLearning besteht darin, dass eLearning von einem Festrechner bedient wird und damit unflexibler als mLearning ist (vgl. [CMS] S.10).

Ferner bringt ein portables Gerät spezifische Funktionalitäten mit sich; darunter die Fähigkeit, ortssensitive Informationen anzubieten, sogenannte „Location Based Services“ oder standortbezogene Dienste. Diese Dienste stellen dem Nutzer Informationen bereit, welche mittels orts-, personen- und zeitabhängigen Daten ausgewählt werden. Darüber hinaus lassen durch diese Dienste Lerninhalte auf Mobilgeräten einfacher personalisieren, indem der Hauptnutzer ein persönliches Lernpräferenzprofil anlegen kann.

Sowohl eLearning als auch mLearning fallen insoweit in den Bereich des Blended Learning, als diese in Kombination mit der klassischen Präsenzlehre verwendet werden können. Jedoch können mit mLearning über die Lernsituationen des eLearning (z.B. Online-Diskussion im Anschluss an die Vorlesung) hinaus neue Lernszenarien des Blended Learning entworfen werden, z.B. das interaktive Trainieren von Vokabeln mit einem Handy.

2.3 Rahmenbedingungen

Um dem Benutzer mobiles Lernen zu ermöglichen, müssen verschiedene Rahmenbedingungen geschaffen werden. Da im Bereich Mobile Learning noch relativ viel Forschungsaufwand betrieben werden muss, werden nur die wichtigsten Schwerpunkte im Folgenden aufgelistet die sich bis dato herauskristallisiert haben:

- Mobile Endgeräte
- Datenübertragung
- Inhaltsaufbereitung
- Pädagogische Betrachtung

Um die Potentiale aber auch die Schwierigkeiten dieser Forschungsfelder vorzustellen, werden sie in den nächsten Abschnitten näher betrachtet.

2.3.1 Mobile Endgeräte

Mobilfunkgeräte (Handys), PDAs, MP3-Player, portable Spielekonsolen, Handhelds, nicht jedoch Notebooks zählen zu den mobilen Endgeräten, weil sie sich durch eine kleine, trag-

bare Größe und Gewicht auszeichnen und „spontan, auch in der Bewegung greif- und nutzbar“ sind (vgl. [RM+02]).

Mobile Endgeräte werden von einer breiten Masse der Bevölkerung genutzt, somit können neue Benutzergruppen in Bezug auf mobiles Lernen erschlossen werden. Vorteile liegen in der portablen Größe, d.h. der Benutzer kann das Gerät ständig bei sich tragen und kann selbst bestimmen, wann und wo er das Gerät einsetzt.

Neben den vielfältigen Möglichkeiten gilt es jedoch noch eine Vielzahl von Hindernissen und Schwächen zu überwinden, bevor ein flüssiges und benutzerfreundliches Lernen mit mobilen Endgeräten möglich sein wird.

Die Geräte verfügen meist über sehr begrenzten Speicherplatz und eignen sich aufgrund ihrer geringen Bildschirmgröße nicht zum Lesen langer Texte beziehungsweise für die Anzeige größerer oder detaillierter Grafiken. Zudem bestehen Limitationen hinsichtlich der Eingabe von Daten: beispielsweise gestaltet sich die Eingabe von Text als sehr mühsam, weil die meisten portablen Geräte aus Platzgründen über eingeschränkte und nicht-barrierefreie Tastaturfunktionen verfügen. Weitere kritische Punkte in Bezug auf die technischen Merkmale mobiler Endgeräte sind die Akkulaufzeit und die teilweise geringe Prozessorleistung.

Eine weitere Schwierigkeit der Softwareanwendungen besteht in der Abhängigkeit der Darstellung der Inhalte von der verwendeten Programmiersprache, beispielsweise WML¹, Java und HTML (vgl. [Kus05], S. 60). Laut Neil McLean ist das Übertragen zwischen den verschiedenen Browsern und Plattformen nahezu unmöglich ([McL03], S. 6). Aufgrund der heterogenen Anforderungen der Endgeräte können Anwendungen nicht plattformunabhängig erstellt werden, sondern müssen für jede Plattform entsprechend gesondert implementiert und angepasst werden.

2.3.2 Datenübertragung

Um mobiles Lernen orts- und zeitunabhängig zu realisieren, müssen die technologischen Grundlagen in Bezug auf drahtlose Datenübertragung gegeben sein. In größeren Städten gibt es bereits sogenannte WLAN²-Hot-Spots, innerhalb dessen die Benutzer nach Anmeldung das Internet nutzen können. Diese werden häufig von Unternehmen beziehungsweise von Privatpersonen als kostenloser Service bereitgestellt, sind jedoch nicht flächendeckend vorhanden. Daher müssen technische Lösungen entwickelt werden, die dem Benutzer je-

¹ Wireless Markup Language

² Wireless Local Area Network

derzeit unabhängig von seinem Aufenthaltsort Netzzugang gewähren. Allerdings fehlen bisher geeignete Business-Konzepte der Netzanbieter zur Abrechnung der Datenübertragungskosten. Zudem müssen geeignete Sicherheitskonzepte für die Datenübertragung geschaffen werden.

Neben der Datenübertragung über WLAN gibt es die Möglichkeit, mobile Datendienste über Mobilfunknetze (GSM, GPRS, UMTS)¹ zu nutzen, beispielsweise das Versenden von Textnachrichten per SMS² oder von Nachrichten mit Multimedia-Inhalten per MMS³. Das Problem dabei liegt jedoch noch an den zu hohen Übertragungskosten und den – von UMTS abgesehen – sehr langsamen Übertragungsraten. Momentan nutzen jedoch nur ein Bruchteil der Mobilfunkkunden den UMTS-Standard. ([DBR], Abbildung S. 1)

Erwähnenswert sind ebenso die Übertragungsmöglichkeiten für Kurzstrecken, Bluetooth und Infrarot, auf deren Basis neue Lernsituationen entwickelt werden können. Der Hauptvorteil dabei liegt in der kostenfreien Nutzung dieser Dienste.

2.3.3 Inhaltsaufbereitung

Wie bereits erwähnt, erfordern mobile Anwendungen aufgrund der beschriebenen hardware-spezifischen Limitationen entsprechend angepasste Inhalte. Dies gilt sowohl für die Darstellung, als auch für die didaktischen Konzepte. Dies soll im nächsten Abschnitt näher betrachtet werden.

Entweder müssen (mLearning-) Anwendungen für mobile Endgeräte eigens erstellt werden, oder es muss ein Standard für die Überführung von Inhalten existierender Weblernanwendungen in solche Inhalte, die sich für mobile Endgeräte eignen, definiert werden.

Wenn Inhalte speziell für mobile Anwendungen erstellt werden sollen, ist darauf zu achten, dass diese in einem darstellungsunabhängigen Format, beispielsweise in XML, abgelegt werden. XML gewährleistet eine Unabhängigkeit der Inhalte von ihrer Darstellung und vereinfacht die Transformation in verschiedene Ausgabeformate.

¹ Global System for Mobile Communications, General Packet Radio Service, Universal Mobile Telecommunications System

² Short Message Service

³ Multimedia Messaging Service

2.3.4 Pädagogische Betrachtung

Ein wesentlicher Forschungsschwerpunkt im Bereich Mobile Learning beschäftigt sich mit der Entwicklung von geeigneten Lernszenarien und den zugrundeliegenden pädagogischen Konzepten zur Steuerung des Lernverhaltens.

Kuszpa stellt in ([FER05], S. 9) fest, dass mobile Lernangebote immer dann genutzt werden, wenn keine anderen Formen des Lernens zur Verfügung stehen und sie sich als nützlich und sinnvoll erweisen, um „*ungeplante Leerzeiten spontan (..)*“ als „*Lernzeiten*“ ([FER05], S. 21) zu nutzen.

Jedoch sollen mobile Lernsituationen sich nicht einzig in den sogenannten „Leerzeiten“ oder Wartezeiten ergeben, sondern können durchaus auch in anderen situativen und ortsgebundenen „Kontexten“ entstehen. Frohberg und Schwabe definieren „Kontext“ als eine „*Lernumgebung inklusive ihrer Artefakte und Akteure*“. (vgl. [FS04]) Bei den Artefakten handelt es sich dabei um jene Objekte aus dem Lernumfeld bzw. der Lernumgebung, die den Lehrinhalt veranschaulichen.¹

Sie weisen dem Kontext, also der Lernumgebung, eine „*hohe Bedeutung (..) für sowohl das individuelle als auch das kooperative Lernen*“ zu, weil der Lernende auf diese Weise motiviert wird, sich „*in einem inhaltlich relevanten Umfeld*“ aktiv und interaktiv mit dem Inhalt auseinanderzusetzen (vgl. [FS04]).

Frohberg und Schwabe liefern weiterhin einen Ansatz, um Lernsituationen im mobilen Kontext zu identifizieren und zu kategorisieren, der im folgenden näher erläutert wird. Sie unterscheiden nach fünf Kontexten gemäß der Bedeutung der Lernumgebung und zeigen dabei jeweils den pädagogischen Mehrwert in Bezug auf Mobiltechnologien auf:

Freier Kontext:

Darunter fallen Lernsituationen, die keinen inhaltlichen Bezug zum aktuellen Lernort haben und somit überall, also ortsunabhängig und spontan, auftreten können.

Der Mehrwert besteht aus der konstruktiven Nutzung von Wartezeiten.

Formaler Kontext:

Kennzeichnend für einen formalen Kontext sind ein durchgeplantes und strukturiertes Umfeld, wo sowohl Lernort als auch Lernzeit bestimmt sind und durch eine Lehrkraft an-

¹ Bei einem Kochkurs wäre eine geeignete Lernumgebung eine Küche und die Kochutensilien und Werkzeuge stellten die Artefakte dar.

geleitet werden. Auch hier hat der Lernkontext keinerlei inhaltlichen Bezug zum Lerninhalt.

Der Mehrwert entsteht durch die Erweiterung der Interaktivität des Unterrichts durch mobile Geräte, beispielsweise über elektronische Kommunikation in Form von Chats, die anonym und unverbindlich ist und somit potentielle Hemmschwellen bei den Lernenden senkt.

Künstlicher Kontext:

Reale Situationen sollen hierbei anhand von Simulationen oder Rollenspielen erlebt und so trainiert werden. So kann der Lernende ohne Risiken und Konsequenzen die Realsituation durchleben und daraus Schlüsse ziehen, „wenn der natürliche Kontext zu risikobehaftet, zu teuer, zu wenig kontrollierbar oder schlicht nicht verfügbar ist.“

Mobile Technologien erlauben diesem Kontext die natürliche Integration der Simulation in den Alltag, dadurch dass der „Akteur“ jederzeit ortsunabhängig agieren und reagieren kann ohne an eine feste Umgebung gebunden zu sein. Ein anschauliches Beispiel hierfür liefert das Projekt: „Partizipative Virusverbreitungssimulation“¹.

Natürlicher elektronisch-passiver Kontext:

Hier spielt sich die Lernsituation in einem natürlichen Kontext ab, d.h. an einem Ort, der den Lerninhalt unmittelbar erleben lässt, z.B. Exkursionen. Es wird insofern als (elektronisch-)„passiv“ eingestuft, als das mobile Gerät lediglich bei Bedarf zum Sammeln und Erfassen von Daten genutzt wird. Der Mehrwert durch mobile Technologien ergibt sich aus der sofortigen digitalen Nutzung und Weiterverarbeitung der gesammelten Daten ohne das Risiko von fehlerbehafteten Medienbrüchen².

Natürlicher elektronisch-aktiver Kontext:

In diesem Kontext hingegen erkundet der Lernende seine natürliche Umgebung nicht nur, sondern interagiert mit ihren Artefakten und Akteuren, dadurch dass diese mit Sensoren und einer speziellen elektronischen Infrastruktur ausgestattet sind für die Kontexterfassung.

Dieser Kontext birgt insbesondere im Bereich mobiler Technologien erhebliche denkbare Potentiale: Als Beispiel sei das Museumsprojekt „Electronic Guidebook“ vom „Exploratori-

¹ MIT/USA, <http://xenia.media.mit.edu/~vanessa/part-sims/>

Ein mobiles Gerät zeigt den virtuellen Gesundheitszustand des Trägers an. Infizierung mit Virus findet statt, wenn er in Infrarotnähe eines "Infizierten" gerät. Ziel ist es, das Verbreitungsmuster von Krankheiten plastisch vor Augen zu führen.

² Bspw. durch die Digitalisierung von Mitschriften auf Papier

um Science Museum of San Francisco¹ aufgeführt, bei denen die Besucher über Lokalisierungsdienste automatisch mit Informationen zu den jeweiligen ausgestellten Exponaten versorgt werden. Zudem können sich die Besucher *„nach individuellen Interessen eine Tour empfehlen lassen und das Museumserlebnis durch Anlegen eines elektronischen Tagebuchs mit Notizen und Fotos auch nach Hause mitnehmen.“* [FS04]

2.4 Mobiles Lernen – Akzeptanz und Interesse anhand einer Studie der Fernuniversität Hagen

Eine Studie aus dem Jahr 2005 der Fernuniversität Hagen (vgl. [FER05]) beschäftigt sich mit Mobile Learning – vorrangig anhand von Mobilfunkgeräten – im deutschsprachigen Raum. Dabei wurden u.a. die Bereitschaft für Mobile Learning, typische Lernsituationen, mögliche Lernformen und -inhalte und Vor- und Nachteile aus der Sicht von potentiellen Anwendern untersucht. Als Probanden für diese Studie wurden Menschen ausgewählt, die *„einschlägige Erfahrungen und Expertenwissen mit Lernangeboten auf neuen Medien besitzen“* ([FER05] S. 2).

Die für diese Arbeit relevanten Ergebnisse sollen hier kurz vorgestellt werden und sind teilweise der Studie entnommen. Die angegebenen Prozentwerte beziehen sich auf insgesamt 168 Fragebögen.

Bereitschaft *„Können Sie sich vorstellen, selbst auf dem Handy zu lernen?“*

Lediglich 43% der Teilnehmer *„wären bereit, Mobile Learning selber zu nutzen“*, darüber hinaus gibt es noch 21,4% die *„unentschlossen, aber grundsätzlich nicht abgeneigt“* sind. Die Rest der Teilnehmer *„würde mobile Endgeräte nicht zum Lernen einsetzen“*.

Lernsituationen *„In welchen Situationen erscheint Ihnen ein Lernen auf dem Handy als sinnvoll?“*

Über drei Viertel der befragten Teilnehmer halten das Lernen *„auf Reisen (...) z.B. im Auto (als Beifahrer), Bus, Zug“* für nützlich. Auch das Nutzen von *„Wartezeiten beim Arzt oder Ämtern“* erscheint vielen (65,4%) ein als sinnvoller Einsatz für Mobiles Lernen.

Lernformen *„Welche Lernformen auf dem Handy erscheinen Ihnen als sinnvoll“*

¹ <http://www.exploratorium.edu/guidebook/>

Bei den Lernformen favorisieren die befragten Personen Richtig/Falsch-Fragen, Multiple-Choice-Aufgaben und Lückentexte. Als eher unpassend erscheint den Probanden „*das Lesen und Lernen umfangreicher Texte*“ sowie „*Textaufgaben für deren Lösung lange Texteingaben*“ notwendig sind. Als sehr nützlich empfinden die Probanden den „*Abruf von Glossarbegriffen aus Wissensdatenbanken*“ in aktuellen Situationen.

Lerninhalte „*Welche Lerninhalte erscheinen beim Lernen mit dem Handy als sinnvoll?*“

Über zwei Drittel der Befragten gaben an, dass sie „*fachspezifische Wissen – bspw. Produktinformationen*“ als sehr geeignet für das Lernen mit dem Handy halten. Fast genauso viele (71,4%) halten „*arbeitsprozessbegleitendes Lernen*“ für sehr hilfreich. Damit sind bspw. Lernszenarien für den technischen Außendienst vor Ort gemeint. Für 59,6% wären auch „*Vokabelprogramme mit kurzen Lernmodulen*“ zur Förderung der Sprachkompetenz vorstellbar. Im Gegenzug dazu halten einige der Teilnehmer Mobile Learning als ungeeignet für „*komplexe Wissen-/Anwendungsverknüpfungen*“ und für die „*Vermittlung von Methodenwissen*“.

Vorteile „*Das Lernen auf dem Handy ist insbesondere mit folgenden Vorteilen verbunden:*“

Als den größten Vorteil (90,4%) von Mobile Learning haben die Teilnehmer der Studie „*zeit- und ortsabhängiges Lernen*“ angeben. Darauf folgt mit 79,2% positiver Zustimmung die Steuerung des individuellen Lerntempos und immerhin 49,4% sehen das Lernen mit einem „*persönlichen und vertrauten Medium*“ als vorteilhaft.

Nachteile „*Mit welchen Nachteilen ist das Lernen auf dem Handy verbunden?*“

„*notwendige Selbstdisziplin*“ wird von 70,8% der Befragten als der größte Nachteil empfunden. Außerdem werden die „*unpersönliche Lernplattform*“ (53,6%) und die „*Gewöhnungsnotwendigkeit im Umgang mit dem Medium*“ (55,4%) als Nachteil empfunden. Andere erwähnte Nachteile sind:

- „*zu kleine Inhalte und eine schlechte Übersicht*“
- „*wenig komfortable in der Nutzung und (..) nur begrenzte Eingabemöglichkeiten*“
- „*Konzentrationsprobleme bei der Nutzung im öffentlichen Umfeld*“
- „*Durcharbeiten von komplexen Inhalten (Grafiken) kaum möglich*“
- „*hohe Kosten für die Nutzung von mobilen Lehrangeboten*“

2.5 Grenzen und Potenziale des mobilen Lernens

Die Einschränkung der Probanden auf eine Gruppe von Nutzern, die einschlägige Erfahrungen mit modernen Medien haben, erfordert eine vorsichtige Auswertung der Befragung der Fernuniversität Hagen [FER05]. Aus der Analyse nach Frohberg und Schwabe [FS04] sowie der Studie geht jedoch hervor, dass mobiles Lernen nicht allein Defizite, sondern durchaus erhebliche Potentiale birgt.

Laut Frohberg und Schwabe ist mLearning „*mehr als eLearning light*“. Es ist mehr als nur ein mangelbehafteter Ersatz für computergestütztes Lernen, bei welchem „*Lehrmaterialien (...) neben anytime auch anywhere verfügbar sind.*“ Dieser Eindruck entsteht oft dadurch, dass die Schwächen und Defizite gegenüber regulären eLearning Technologien herausgestellt werden. Ein Grund dafür ist, dass es bis dato nicht abzusehen ist, ob mobile Technologien die drahtgebundenen aufgrund ihrer Limitationen hinsichtlich der geringen Lesefläche der Bildschirme, fehlender Standards, mangelnder Sicherheitskonzepte und der nicht flächendeckenden und zum Teil inkompatiblen Datenübertragungstechnologien, ersetzen werden.

Es ist vorstellbar, dass die skeptischen Einschätzungen mobiler Lernanwendungen aus der Unkenntnis erwachsen, welche potentiellen Verwendungsgebiete mit mobilen Technologien über die ortsunabhängigen Charakteristika hinaus möglich sind: und zwar die Fähigkeit, kontextsensitives – also von der Lernumgebung abhängiges – Wissen mitzuteilen. Über den aktuellen Wissenstand der Bevölkerung zu den Potentialen mobilen Lernens liegen jedoch nach dem Kenntnisstand der Autorin keine erwähnenswerten Studien vor.

Frohberg und Schwabe haben verschiedene Szenarien in Bezug zum jeweiligen Kontext aufgeführt, die viel Gestaltungsraum für innovative Konzepte bieten. Dieser Gestaltungsraum stellt Entwickler und Pädagogen vor die neue Herausforderung, inwieweit sich mit mobilen Technologien die Lernwelt aber auch das Lernverhalten über die Konzepte des eLearning hinaus weiterentwickeln lassen.

Kapitel 3 Technologische Grundlagen

Zum besseren Verständnis dieser Arbeit werden in diesem Kapitel technologische Grundlagen der Dokumentbeschreibungssprache XML und ausgewählter Unterdialekte beschrieben. Es wird auf die Prinzipien eines Webservers und die Technologie des XML-Publishing Frameworks Cocoon näher eingegangen. Weiterhin wird das Hypermedia Learning Object System (hylOs), ein Contentmanagementsystem für Lerninhalte, vorgestellt und im letzten Abschnitt soll kurz auf die Eckdaten der Spielekonsole PlayStation Portable von Sony eingegangen werden.

3.1 Extensible Markup Language – XML

Mit der Extensible Markup Language¹, kurz XML, hat das World Wide Web Konsortium (W3C) im Februar 1998 einen Standard zur Strukturierung von Dokumenten und Daten definiert. Die Vielzahl der existierenden Formate machte es notwendig einen Standard zu entwickeln, der ein universelles Format hervorbringt, um strukturierte Dokumente plattform- und anwendungsübergreifend einsetzen zu können. XML hat im Gegensatz zu anderen Auszeichnungssprachen wie beispielsweise HTML den Vorteil, dass es Inhalte strukturiert und nicht layoutbasiert beschreibt. Somit können XML-Dokumente für den Datenaustausch zwischen beliebigen Anwendungen oder als Schnittstelle für unterschiedliche Publikationskanäle ohne Beeinflussung des Inhaltes genutzt werden (vgl. [Ste02]).

XML ist eine stark vereinfachte Teilmenge der Meta-Auszeichnungssprache SGML². SGML ist eine sehr mächtige Auszeichnungssprache und enthält eine Vielzahl von Befehlen. Aufgrund der sehr zahlreichen und zu komplexen Funktionen, die in SGML definiert waren, konnte es sich in der Praxis nicht durchsetzen (vgl. [Noa02] S.10). Mit XML hat das W3C versucht, die grundlegenden Vorteile von SGML aufzunehmen und dem Benutzer einen Standard zur Verfügung zu stellen, der schnell erlernbar, einfach anwendbar und von Menschen gelesen werden kann.

Die folgende Beschreibung des XML-Standards sind im Detail nachzulesen in [Noa02],[Ste02]. Speziell die Erläuterung der Parser wurde ([Nie07] S. 101ff) entnommen.

¹ dt. Erweiterbare Auszeichnungssprache

² Standard Generalized Markup Language, dt. Normierte Verallgemeinerte Auszeichnungssprache

Elemente und Attribute

XML-Dokumente bestehen im Wesentlichen aus frei definierbaren ineinander verschachtelten Elementen mit zugeordneten Attributen. Elemente beschreiben die Dokumentenstruktur und stellen in ihrer Anordnung eine Baumstruktur dar. Diese spiegelt die interne Struktur des Dokumentes wider. Jedes Element bildet einen Knoten, welcher wiederum Unterelemente enthalten kann. Ein vollständiges Element besitzt ein Start- und ein Endtag z.B. `<Elementname>Elementinhalt</Elementname>`, dabei ist der Name des Elementes frei wählbar. Es können ebenso Elemente vorkommen, die keinen Inhalt enthalten beispielsweise ein Zeilenumbruch oder ein Bild. Diese können ebenso durch ein Start- und Endtag dargestellt werden oder auch durch die verkürzte Form `<Elementname />`. XML ist beliebig erweiterbar und kann demnach eine unbegrenzte Anzahl von Elementen definieren. Um ein Element genauer zu beschreiben oder ihm bestimmte Werte zuzuordnen, können Attribute für des Element definiert werden. Die Anzahl der Attribute ist auch hier nicht begrenzt, jedoch darf ein Attribut in einem Element nur einmal auftreten.

Document Type Definition und XML Schema

Welche Elemente und Attribute zulässig sind und was für Eigenschaften und Werte Attribute annehmen können, ist in der Document Type Definition (DTD) oder in einem XML-Schema definiert. Im Gegensatz zu HTML enthält ein XML-Dokument keine vordefinierten DTDs, d.h. keinen vordefinierten strukturellen Aufbau. In XML werden eigene DTDs zur Festlegung der Regeln für die im XML-Dokument enthaltenen Elemente und Attribute definiert. Dabei kann festgelegt werden, welche Elementtypen und Elementinhalte, welche Attribute und Attributwerte enthalten sein können und wie Elemente in welcher Reihenfolge auftreten und ineinander verschachtelt sein dürfen. Das bedeutet, dass innerhalb des XML-Dokuments nur Elemente bzw. Attribute auftreten, die in der DTD definiert sind. Eine DTD kann intern im XML-Dokument definiert werden oder als externe Datei vorliegen, wobei das dazugehörige XML-Dokument mit einen Verweis auf die externe Datei versehen wird.

Neben der Benutzung von DTDs hat sich in den letzten Jahren der XML-Schema-Standard etabliert. XML-Schemata definieren wie DTDs den strukturellen Aufbau eines Dokumentes und erweitern die Möglichkeiten zum Beschreiben des Inhaltes von Elementen und Attributen. Dies geschieht durch die Verwendung von Datentypen wie beispielsweise Integer¹, String² oder Boolean³. Hinzu kommt die Möglichkeit der Definition von eigenen Da-

¹ Ganzzahl

² Zeichenkette

³ kann zwei Werte annehmen: true oder false

tentypen zum Einschränken des Inhaltes beispielsweise durch Festlegung des Wertebereichs den ein Element annehmen kann. Ein weiterer großer Vorteil ist, dass XML-Schemata im Gegensatz zu DTDs ebenfalls XML-Dokumente sind, somit ebenfalls mit XML-Technologien weiterverarbeitet werden können und der Entwickler keine neue Syntax erlernen muss.

Verarbeitung von XML-Dokumenten durch DOM oder SAX

Bevor XML-Dokumente in einer Anwendung verarbeitet werden können, müssen die Dokumente aufbereitet werden, um Zugriff auf die Elemente und ihre Inhalte zu erhalten. Dazu existieren sogenannte Parser¹ beispielsweise Simple API for XML (SAX) oder Document Object Model (DOM).

SAX ist kein offizieller Standard des W3C, wird jedoch von einer Vielzahl von Parser-Implementierungen unterstützt. SAX arbeitet ereignisorientiert und sequentiell. Der Parser durchläuft das XML-Dokument auf der Suche nach vom Anwender definierten Elementen. Wird dabei eines dieser Elemente gefunden, führt der Parser eine vorher festgelegte Operation aus. Anschließend fährt er mit dem nächsten gesuchten Element fort, ohne dass das vorherige Element im Speicher abgelegt wurde. Der Vorteil von SAX liegt in der schnellen Verarbeitungsgeschwindigkeit. Außerdem werden durch die sequentielle und ereignisorientierte Arbeitsweise die Ressourcen des Arbeitsspeicher geschont. Der SAX-Parser kommt häufig zum Einsatz, wenn nur einzelne Informationen benötigt werden.

Im Gegensatz zu SAX baut DOM zunächst die gesamte Baumstruktur als verzweigte Objekte im Arbeitsspeicher auf. Anschließend kann wahlfrei auf alle Elemente zugegriffen werden. Dies ist besonders bei Arbeiten, bei denen mehrfach durch den Dokumentenbaum navigiert werden muss, sehr nützlich. Es bedeutet jedoch gleichzeitig eine intensive Belastung des Arbeitsspeichers und somit eine verlangsamte Verarbeitung, was sich insbesondere bei sehr großen Dokumenten auswirkt.

Wohlgeformtheit und Gültigkeit

Ein Parser zerlegt das Dokument in seine Bestandteile, wobei er es auf Wohlgeformtheit und Gültigkeit prüft und die Elementstruktur mit ihren Eigenschaften ausgibt. Ein Dokument wird als wohlgeformt bezeichnet, wenn es gemäß der XML-Spezifikation [W3b] erstellt ist. Dazu gehört das Vorhandensein genau eines Wurzelementes, alle Elemente – sofern sie nicht leer sind – müssen ein Start- und ein Endtag aufweisen und die Verschachtelung der Elemente muss korrekt sein. Ein XML-Dokument muss wohlgeformt sein, sonst

¹ dt. Syntaxanalytiker. Hier sind speziell XML-Parser gemeint. Diese lesen XML-Dokumente aus, analysieren diese und bereiten sie für die Weiterverarbeitung auf.

kann es nicht weiterverarbeitet werden. Als gültig wird ein Dokument bezeichnet, wenn es wohlgeformt ist und über eine DTD oder ein XML-Schema verfügt. Darüber hinaus muss das Dokument allen – in der DTD oder in dem XML-Schema – definierten Regeln entsprechen und über eine Dokumenttypdeklaration verfügen. Eine Dokumenttypdeklaration – ein sogenannter Prolog – steht vor dem ersten Element im Dokument und definiert u.a. die XML-Version und den Zeichensatz. Ein XML-Dokument muss nicht zwangsläufig gültig sein, d.h. es könnte auch ohne DTD bzw. XML-Schema verwendet werden.

Namensräume

XML integriert zusätzlich das Konzept der Namensräume, welches hier der Vollständigkeit halber erläutert werden soll.

Das Konzept der Namensräume entstand aufgrund der durch den Anwender frei definierbaren Element- und Attributbezeichnungen. Da es dabei zu Doppelungen und Mehrdeutigkeiten kommen kann, wurde der XML-Standard um Namensräume erweitert. Sie beheben das Problem durch Definition einer eindeutigen Kennzeichnung der Element- und Attributbezeichnungen mittels Präfixen, die vor die Elementbezeichnungen eingefügt werden. Zum Verwenden von Namensräumen wird dieser am Anfang eines XML-Dokuments deklariert.

XML als universelles Format bildet die Grundlage für eine Vielzahl von Technologien. Einige für diese Arbeit wichtige Technologien sollen im Folgenden behandelt werden.

3.1.1 Extensible Stylesheet Language – XSL

XSL steht für Extensible Stylesheet Language und dient zur Transformation und Formatierung der auszugebenden Dokumente mithilfe von Stylesheets¹. Mit XSL kann durch Fallunterscheidungen und Schleifenanweisungen der Aufbau der Datenstruktur beeinflusst werden (vgl. [Noa02], S. 84). XSL arbeitet mit den XML-Dialekten XSLT², XPath³ und XSL-FO⁴, welche im Folgenden erläutert werden.

Mit XSLT⁵ erfolgt die Transformation der Baumstruktur des Quelldokumentes in die Baumstruktur des Zieldokumentes. Die Vorgaben für die Transformation sind in einem

¹ enthält Regeln zur Erstellung eines Dokumentes, es enthält dazu grafische und typografische Layoutmerkmale.(vgl. [ITW])

² XSL Transformation

³ XML Path Language

⁴ XSL – Formatting Objects

⁵ XSL Transformation

Stylesheet geregelt. Ein Stylesheet beinhaltet Muster zum Adressieren der Elemente im Quelldokument und Vorlagen zur Generierung der neuen Baumstruktur. Anhand der Muster werden die Elemente mit dem Quelldokument verglichen. Passt das Element zu dem definierten Muster sorgen die Vorlagen anschließend für die Transformation der gefundenen Elemente in die neue Baumstruktur. Ein Muster kann neben der Elementadressierung auch Layoutinformationen definieren, beispielsweise den Verweis auf eine CSS-Datei¹. Für den Mustervergleich und die Adressierung der Elemente ist XPath zuständig. Aufgrund der Komplexität ist XPath in einem eigenen Kapitel erläutert (siehe Kapitel 3.1.2).

XSL-FO ist eine Seitenbeschreibungssprache, die durch definierte Elemente, sogenannte Formatting Objects, beliebige XML-Dokumente in ein seitenbasiertes Format überführt, wie beispielsweise PDF oder RTF.

Zur Formatierung von Elementen hat das W3C zwei Standards verabschiedet, XSL und CSS. Cascading Style Sheets ist ein Standard zur Formatierung von strukturierten Dokumenten, verfügt jedoch im Unterschied zu XSL nicht über Funktionalitäten zur logischen Strukturierung der Daten. Nun stellt sich die Frage: Warum existieren zwei Standards zur Formatierung und welcher kommt wann zum Einsatz? Als Faustregel gilt laut W3C: „*Use CSS when you can, use XSL when you must.*“ [W3] Sollen Daten im Ausgabedokument sortiert, gefiltert oder ersetzt werden, also strukturell verändert werden, dann ist es sinnvoll XSL einzusetzen. Sollen die Daten lediglich grafisch aufbereitet werden, ist es ausreichend CSS einzusetzen. Die einfache Funktionsweise und die Befehlsreferenz für CSS reichen für die meisten Anforderungen aus. In der Praxis hat es sich etabliert, CSS vorwiegend für Ausgabeformate am Bildschirm und XSL zur Formatierung von Print-Publikationen einzusetzen. [XML],[HEI]

3.1.2 XML Path Language – XPath

XPath dient zur Adressierung von Knoten und wird zur Navigation innerhalb von XML-Dokumenten eingesetzt. Die Adressierung erfolgt durch Identifizierung der Knoten innerhalb eines Dokumentes. Als Knoten werden bei XPath einzelne Objekte wie Elemente, Attribute, Texte, Kommentare etc. betrachtet. Die Identifizierung des gesuchten Knotens erfolgt durch den Lokalisierungspfad. Der Lokalisierungspfad zeigt auf eine bestimmte Menge von Knoten und kann relativ oder absolut zum Wurzelement angegeben sein. Der Lokalisierungspfad besteht aus einzelnen Lokalisierungsstufen. Jede Lokalisierungsstufe enthält eine Achse, einen Knotentest und ein oder mehrere optionale Prädikate. Die Achse

¹ Cascading Style Sheet

drückt die Beziehung des Ausgangsknoten zur ausgewählten Knotenmenge aus und dient zum Navigieren in der XML-Struktur. Der anschließende Knotentest schränkt die Knotenmenge durch Bestimmung des Knotennamens und des Knotentyps weiter ein. Die optionale Verwendung von einem oder mehreren Prädikaten dient zur weiteren Filterung der bisherigen Knotenauswahl. Prädikate enthalten einen booleschen Ausdruck, welcher auf jeden Knoten der bisher selektierten Knotenmenge angewendet wird. Ist der Rückgabewert des booleschen Ausdrucks `false` – dies ist der Fall, wenn der getestete Knoten nicht dem geforderten Filterkriterium entspricht – wird der entsprechende Knoten aus der Knotenmenge entfernt.

Um bestimmte Bereiche in einem Knoten zu erhalten, gibt es die XPath-Erweiterung XPointer. XPointer¹ ermöglicht eine genauere Adressierung von Bereichen innerhalb eines Knotens, beispielsweise kann XPointer auf bestimmte Textstellen in einem Absatz zeigen.

XPath wurde bereits im Zusammenhang mit der XSL Transformation erwähnt. Bei der XSL Transformation werden in den Stylesheets Muster mithilfe von XPath festgelegt. XPath stellt dabei die notwendigen Abfragemechanismen beim Musterabgleich zur Verfügung und steuert die Lokalisierung der Knoten.

3.1.3 XML Linking Language – XLink

XLink ist ein Standard zur Definition dynamischer Verweise auf XML-Dokumente und andere Ressourcen und deren Beschreibung. Im Gegensatz zu XLink können mit HTML lediglich statische und unidirektionale Verweise erstellt werden, welche nur ein einzelnes Verweisziel referenzieren. Das Modifizieren dieser Verweise ist nur direkt im Quelldokument möglich. Aufbauend auf dieser Problemstellung hat das W3C mit XLink einen Standard definiert, der multidirektionale Verweise mit mehreren Zielen erlaubt. Zusätzlich können Verweise mit Metadaten versehen werden. Zur komfortableren Pflege der Verknüpfungen, werden diese unabhängig vom Dokument gespeichert.

Die XLink-Syntax erlaubt die Definition von einfachen und erweiterten multidirektionalen Link-Typen. Der einfache Link-Typ verknüpft, ähnlich wie in HTML Hyperlinks, Quell- und Zielressource miteinander, der Zielverweis kann eine lokale oder entfernte Ressource sein. Der erweiterte multidirektionale Link-Typ kann zwei oder mehrere Verweisziele enthalten. Dadurch können komplexe Verknüpfungen beliebig vieler Ressourcen erstellt werden. Verweise werden nicht als neues Element definiert, sondern als globales Attribut ei-

¹ XML Pointer Language

nem XML-Element in der DTD hinzugefügt. Somit kann jedes XML-Element, welches dieses globale Attribut enthält, auch ein XLink-Element darstellen. Den Link-Typen stehen zahlreiche optionale und obligatorische Attribute zur Verfügung, um Verweise näher zu spezifizieren. Neben dem obligatorischen Attribut Link-Typ können Attribute zur Adressangabe, zur Steuerung des Linkverhaltens, zur Angabe semantischer Beziehungen und zum Festlegen von Traversierungsregeln¹ genutzt werden.

3.2 Der Webserver und Servlet Container Jetty

Die Aufgabe eines Webservers² ist das Übermitteln von Ressourcen, die von einem Client³ beispielsweise einem Webbrowser angefordert werden. Dabei arbeitet der Webserver nach dem sogenannten Request⁴-Response⁵-Zyklus. Der Request-Response-Zyklus funktioniert folgendermaßen:

Ein Benutzer klickt in seinem Webbrowser auf einen Link einer Webseite und stellt damit eine Anfrage. Der Browser versieht die Anfrage mit Zusatzinformationen, generiert einen HTTP-Request und schickt diesen an den Webserver. Der HTTP-Request kann neben der Adresse und dem Dokumentenformat weitere Parameter enthalten wie beispielsweise die Request-Methode oder das benutzte Protokoll. Ist die angeforderte Ressource nicht statisch, sondern muss dynamisch erzeugt werden, entscheidet der Webserver aufgrund der im HTTP-Request enthaltenen Informationen wie die Weiterverarbeitung bzw. Erzeugung der Ressource erfolgt. Diese Aufgabe kann beispielsweise durch serverseitige Skriptsprachen wie PHP, JSP, ASP oder durch Servlets übernommen werden. Nach der Erzeugung schickt der Webserver die Antwort als HTTP-Response mit dem generierten Dokument zurück an den Webbrowser.

Es sind zahlreiche Webserver auf dem Markt zu finden. Die aktuell bedeutendsten [CON] sind Apache HTTP Server und Microsoft Internet Information Services. Das in dieser Arbeit genutzte Hypermedia Learning Object System setzt den Jetty Webserver ein.

Jetty ist vollständig in Java realisiert und kann somit plattformunabhängig eingesetzt werden. Die Vorteile gegenüber anderen Webservern neben der Plattformunabhängigkeit sind seine minimale Größe und die im Vergleich zu anderen Webservern einfache Konfiguration. Jetty stellt zusätzlich zu seiner Funktion als Webserver auch einen Container für Serv-

¹ traversieren = benutzen oder verfolgen eines Verweises

² ist eine Maschine bzw. eine Software auf einer Maschine, die Anfragen von einem Browser über das HTTP-Protokoll beantwortet.

³ dt. Kunde, Klient. Ist ein Computerprogramm, das Dienste eines Servers in Anspruch nimmt.

⁴ dt. Anfrage

⁵ dt. Antwort

lets und JSPs zur Verfügung. Ein Servlet-Container verwaltet die Servlets die auf einem Webserver vorhanden sind. Er ist verantwortlich für das Laden und Instantiieren des zuständigen Servlets. Servlets werden auch als Webapplikationen bezeichnet. Eine solche Webapplikation ist Cocoon, welche innerhalb des Jetty Servlet-Containers arbeitet.¹

Dabei kommuniziert Jetty über das Cocoon-Servlet mit der Cocoon-Applikation, um dieser die notwendigen Servlet-Objekte wie beispielsweise Request, Session oder Response zu übergeben. Daraufhin kann die Cocoon-Applikation diese Objekte entsprechend verwenden. Die Arbeitsweise der Cocoon-Applikation erfüllt alle Anforderungen an eine Webapplikation nach (vgl. [Nie07], S. 29) und im soll im nächsten Kapitel näher betrachtet werden.

3.3 Das Cocoon-Framework

Cocoon ist ein Publishing Framework auf Basis der XML-Technologie zur Veröffentlichung von Inhalten. Es erlaubt u.a. die dynamische Transformation von verschiedenen Eingabeformaten in unterschiedlichste Ausgabeformate. Cocoon ist Open Source Software und basiert auf der Apache Software License [COCb].

Der folgende Abschnitt beruht auf dem Buch „Cocoon 2 und Tomcat“ (vgl. [Nie07]) von Stephan Niedermeier.

Die allgemeine Arbeitsweise des Cocoon unterliegt ebenfalls dem Request-Response-Prinzip. Wird Cocoon als Webapplikation in einem Servlet-Container eingesetzt, so wird bei einem Request zunächst das Cocoon-Servlet aufgerufen und es werden die Parameter des jeweiligen Requests übergeben. Das Servlet löscht alle nicht benötigten Parameter bis auf die relative Pfadangabe auf das angeforderte Dokument. Dabei kann es sich um eine Datei beispielsweise im HTML- oder PDF-Format handeln. Die relative Pfadangabe wird in diesem Kontext als Sitemap-URI² bezeichnet. Sie dient zur Identifizierung einer Pipeline, die für die Bearbeitung des angeforderten Dokumentformats zuständig ist. Eine Pipeline enthält alle Komponenten, die nötig sind um das Dokument zu verarbeiten und abschließend in einem bestimmten Format auszugeben. Definiert ist die Pipeline in der Sitemap, der zentralen Datei in Cocoon. Die Sitemap enthält alle definierten Pipelines mit ihren Bedingungen und Komponenten, die zur Verarbeitung eines bestimmten Dokumentenformats benötigt werden. Cocoon vergleicht nun die Bedingungen aller in der Sitemap registrierten Pipelines auf Übereinstimmung mit der Sitemap-URI. Existiert eine solche Übereinstim-

¹ Cocoon kann nicht nur als Webapplikation betrieben werden, sondern ist ebenfalls fähig als Standalone-Applikation zu arbeiten oder kann als Modul in andere Java-Applikationen integriert werden.

² Uniform Resource Identifier, dt. einheitlicher Bezeichner von Ressourcen

mung mit einer Pipeline-Bedingung wird die entsprechende Pipeline ausgeführt und das angeforderte Dokument ausgegeben, anderenfalls erfolgt eine Fehlermeldung.

Pipeline in der Sitemap von Cocoon

Da Pipelines eines der zentralen Konzepte von Cocoon darstellen, soll der folgende Abschnitt dazu genutzt werden, die sogenannte Pipeline-Technik ([Nie07] S. 198-226) noch einmal im Detail zu erläutern.

Eine Pipeline kann verschiedene ineinander geschachtelte Komponenten enthalten. Diese Komponenten bearbeiten das Eingangsdokument entsprechend dem auszugebenden Zielformat bzw. Zielsystem. Eine Pipeline kann in der Regel in fünf unterschiedliche Prozesse eingeteilt werden:

1. Matching
2. Generierung (Erzeugung von SAX-Streams)
3. Transformation (Umwandlung durch SAX-Events) *optional*
4. Ausführung von Logik *optional*
5. Serialisierung (Ausgabe)

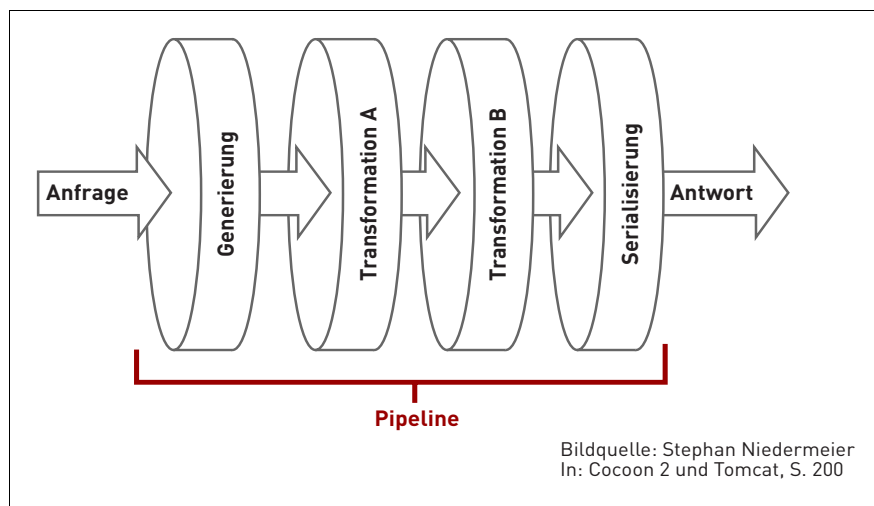


Abbildung 2: Pipeline in Cocoon

Im ersten Schritt wählt der Matcher beispielsweise anhand des in der Anfrage enthaltenen URI (Sitemap-URI) die Pipeline, die ausgeführt werden soll, aus. Abbildung 1 zeigt beispielhaft wie eine Pipeline aussehen könnte. Der Generator wandelt beliebig strukturierte

Daten in SAX-Ereignisse und gibt diese anschließend an die nächste Komponente weiter. Die Daten können dabei aus unterschiedlichsten Quellen kommen beispielsweise aus XML-Dokumenten, oder aus einer Datenbank. Um diese verschiedenen Daten zu verarbeiten, hält Cocoon diverse vorgefertigte Generatoren bereit, u.a. den FileGenerator [COC]. Der FileGenerator liest XML-Dokumente aus dem URI ein. Die nächste Komponente kann ein Transformator sein. Ein Transformator wandelt die SAX-Ereignisse mittels der definierten Anweisungen um und/oder gibt diese weiter an die nächste Komponente. Auch hier hält Cocoon bereits eine Vielzahl an vorgefertigten Transformatoren bereit, beispielsweise den XSLT-Transformator. Dieser transformiert das XML-Dokument mittels eines XSLT-Stylesheets. Die nächste Komponente kann wiederum ein Transformator sein oder ein Serialisierer. Der Serialisierer bildet das Ende der Pipeline und wandelt die SAX-Ereignisse in das angefragte Zielformat um, also beispielsweise XML, HTML oder PDF. Abschließend wird das Ergebnis des Serialisierers an den Client als Antwort(Response) geschickt.

3.4 Hypermedia Learning Object System

Das Hypermedia Learning Object System bezeichnet ein plattformunabhängiges lernobjektverarbeitendes Content Management System (Learning Management System). Es baut auf einem eLearning Information Object Model auf, welches Inhalte, Metadaten und Relationen entsprechend dem IEEE LOM (Learning Objects Meta Data) Standard [LTS] verarbeitet. hylOs bietet variable Zugangsstrukturen, es erlaubt dem Lernenden sowohl selbstentdeckendes Lernen als auch Lernen über einen vom Autor festgelegten Lernpfad. Eines der Basiskonzepte von hylOs ist die Unterstützung der XML-Technologien. Damit wird eine strikte Trennung von Inhalt, strukturierten Informationen, Anwendungslogik und Designelementen gewährleistet [EHL05]. hylOs wurde an der Fachhochschule für Technik und Wirtschaft Berlin entwickelt und im Jahr 2002 erstmals auf einem internationalen Workshop für Blended Learning in Villach vorgestellt.

hylOs baut auf dem allgemeinerem Media Information Repository (kurz MIR) [MIR] auf. Das MIR ist ein anwendungsneutrales, hypermediales Storage- und Laufzeitsystem zur Speicherung und Verarbeitung medialer Daten. MIR enthält das MIR adaptive Context Linking Environment (kurz MIRaCLE) – eine Verlinkungserweiterung. Da MIR Lerninhalte als XML-Struktur bereitstellt, ermöglicht es deren flexible Verarbeitung. Durch die

Nutzung von XML-Technologien ist die Präsentation des Inhalts in verschiedensten Ausgabeformaten möglich u.a. HTML, PDF¹ oder RSS-Feeds².

MIRaCLE [MIR] sorgt für die Verarbeitung multipler und kontextsensitiver Hyperreferenzen. Dabei werden Links und Anker unabhängig vom Inhalt als eigenständige Objekte gespeichert. Die Hyperreferenzen können mit frei definierbaren Kontexten versehen werden, welche in angeforderten Inhaltsbausteinen ein- oder ausgeblendet werden.

Neben der Erzeugung von Lerninhalten können die Hyperreferenzen zusätzlich mit Metadaten entsprechend dem LOM Metadaten Standard versehen werden. Als Metadaten bezeichnet man Daten die Informationen über andere Daten enthalten. LOM ermöglicht Lerninhalte durch semantische Beziehungen zu verknüpfen. Aus diesen Verknüpfungen bildet sich ein sogenanntes semantisches Netz, welches Lernobjekte sinngemäß miteinander verbindet. hylOs unterstützt den kompletten LOM Metadaten Standard. Gemäß diesem werden Lernobjekte in hylOs als selbstkonsistente, atomare Wissensseinheiten betrachtet. Die Lernobjekte enthalten sowohl den Inhalt als auch die Metadaten. Der Inhalt selbst kann dabei ebenfalls ein Lernobjekt darstellen [EHL05],[EKR+03].

3.4.1 Systemarchitektur und Dokumentengenerierung in hylOs

In diesem Kapitel wird die Arbeitsweise von hylOs bei der Generierung der Ausgabeformate beschrieben, gleichzeitig gibt dieser Abschnitt einen Einblick in die Systemarchitektur – in welche hylOs integriert ist. Abbildung 2 stellt den Prozess vom Aufruf durch den Benutzer im Webbrowser bis zur Auslieferung der gewünschten Webseite im Webbrowser dar.

Der Benutzer ruft auf der hylOs-Webseite einen Lernkurs auf (1) und erzeugt dadurch einen HTTP-Request(2). Der HTTP Request enthält u.a. den Pfad zum Zieldokument, dem Lernkurs, und dient zum Identifizieren der benötigten Objekte. Das System nutzt als Vermittler zwischen dem Webbrowser und dem Webserver den Apache Proxyserver. Der Apache Proxyserver sorgt für erhöhte Sicherheit vor unerlaubten Zugriffen, außerdem simuliert er virtuelle Webserver und leitet den HTTP-Request(3) an den Jetty Webserver(4) weiter. Dort sorgen Servlets für die Weiterverarbeitung der HTTP Anfrage – hylOs setzt unter anderem Cocoon und GetData als Webapplikation ein. Cocoon(5) erzeugt XML-Strukturen und transformiert diese in das angefragte Ausgabeformat. Je nach Art des Requests wird das XML-Dokument generiert bzw. werden die Binärdaten (beispielsweise Bil-

¹ Portable Document Format

² Nutzer können mithilfe eines RSS-Readers Informationen (RSS-Feeds) einer Webseite abonnieren. Informationen sind meist Nachrichten und Kurzinformationen.

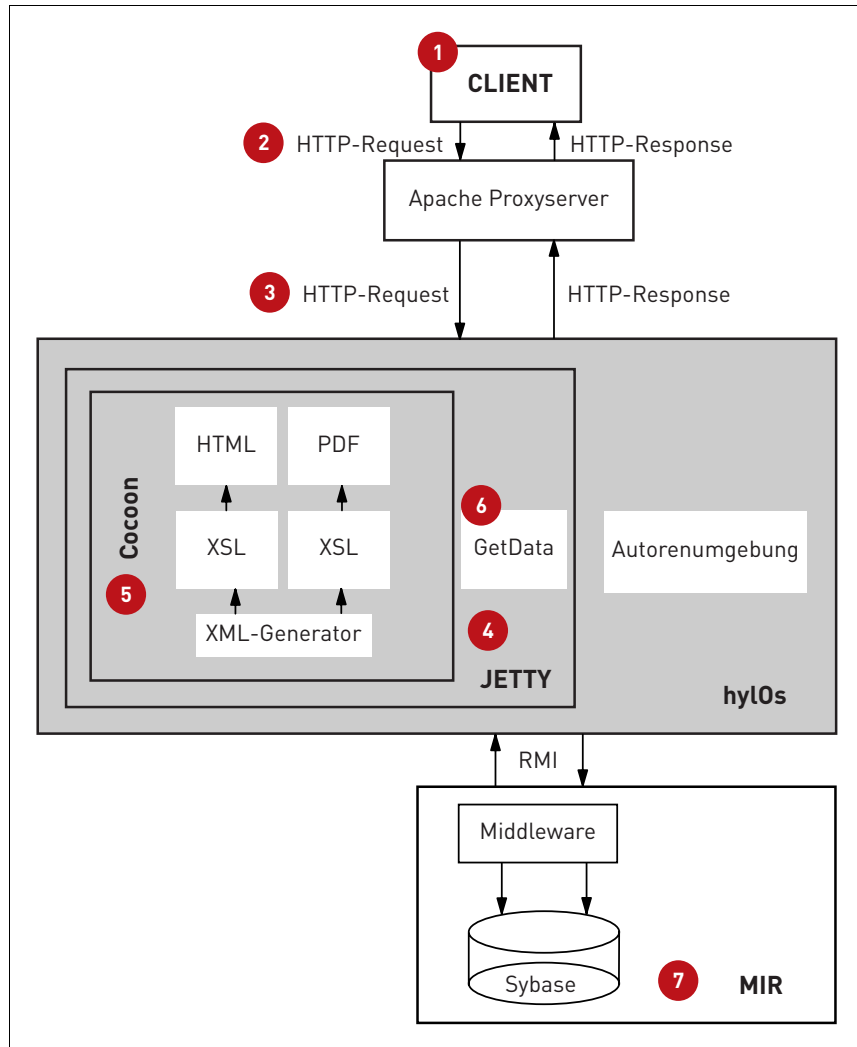


Abbildung 3: Systemarchitektur / Prozess der Dokumentengenerierung in hylOs

der und Videos) durch die separate Webapplikation GetData(6) über die Middleware aus der Datenbank ausgelesen. Die Datenbank und die Middleware bilden die Grundlage von hylOs und werden als das Media Information Repository (MIR)(7) bezeichnet. Die Datenbank unterhalb von MIR [MIRb] ist eine relationale Datenbank von Sybase, diese implementiert ein objektorientiertes Datenmodell und die Middlewarekomponenten sind mittels Java realisiert. Die Middleware koordiniert die Kommunikation mit der Datenbank und stellt Basisoperationen für die Sessionlogik bereit(vgl. [MIRb]). Sind die angefragten Ressourcen durch das GetData-Servlet bzw. das Cocoon-Servlet generiert worden, werden sie

zusammen mit dem HTTP-Response über den Apache Proxyserver an den Browser des Benutzers gesendet.

3.4.2 Erzeugung der Ausgabeformate

Im letzten Abschnitt wurde ein Überblick über die Generierung der Ausgabeformate von der Anfrage an den Server bis zur vollständigen Rückgabe des Dokumentes an den Browser gegeben. Der nun folgende Abschnitt beschäftigt sich genauer mit der Transformation der angefragten Ausgabeformate mit Cocoon. Dazu werden die benötigten Arbeitsschritte von Cocoon speziell mit hylOs näher erläutert.

Der HTTP-Request beinhaltet den Zielpfad und übergibt diesen an das Cocoon-Servlet. In hylOs besteht der Zielpfad aus Host¹, XML-Generator, Objektpfad und der Dateiendung, die das Ausgabeformat vorgibt. Wie bereits im Kapitel 3.3 zur allgemeinen Arbeitsweise von Cocoon erläutert, werden im ersten Schritt zur Erzeugung des Ausgabeformates XML-Generatoren benötigt. Bei der Entwicklung von hylOs wurden eigene XML-Generatoren implementiert. Die wichtigsten sind Inhalt, Navigation, Bibliographie und Glossar. Der Inhaltsgenerator ist zuständig für die Erzeugung des gesamten Inhaltsbereich, der Navigationsgenerator erstellt die komplette Navigation bzw. die Verknüpfungen des Lernobjektes, mit dem Bibliografie-Generator werden alle mit dem Lernobjekt verknüpften Bibliografieinträge erzeugt und der Glossar-Generator erstellt entsprechend alle verknüpften Glossareinträge. Anhand des Objektpfades kann das Lernobjekt identifiziert werden und über die Dateiendung wird das entsprechende Stylesheet für die Inhaltsaufbereitung des Dokumentes gewählt.

Soll nun beispielsweise der Inhalt eines Lernobjektes generiert werden, liest der Inhaltsgenerator die URL des Lernobjektes aus der Anfrage ein, generiert daraus eine Reihe von SAX-Ereignissen und gibt diese Ereignisse weiter an den Transformator. Der Transformator bereitet den Inhalt auf, indem er die SAX-Ereignisse verarbeitet und das XSL-Stylesheet für das angefragte Format auf den Inhalt anwendet. Für hylOs existieren XSL-Stylesheets beispielsweise für HTML, FO (Formatting Objects), SMIL² und RSS. Nach Anwendung der Stylesheets leitet der Transformator das Ereignis weiter an den Serialisierer, dieser wandelt die Ereignisse in das angefragte Ausgabeformat. Anschließend wird das generierte Dokument über den HTTP-Response zurück an den Browser gegeben.

¹ Rechnername

² Synchronized Multimedia Integration Language

3.5 PlayStation Portable

PlayStation Portable, kurz PSP genannt, bezeichnet die tragbare Multimedia-Konsole der Firma Sony Computer Entertainment. Im Dezember 2004 fand die Markteinführung in Japan statt, seit September 2005 ist die Konsole auch in Europa erhältlich. Mit der PlayStation Portable ist es möglich Musik zuhören, Filme anzuschauen, Spiele zuzuspielen und Fotos anzuschauen. Außerdem verfügt die PSP über eine WLAN-Funktion und ist somit in der Lage über den integrierten Webbrowser im Internet zu surfen.

Kapitel 4 Ist-Analyse

Im vorherigen Kapitel wurden die technischen Grundlagen für diese Arbeit kurz erläutert. Dieses Kapitel soll detaillierter die in der vorliegenden Arbeit verwendeten Systeme hylOs und PlayStation Portable analysieren, um genaue Informationen für das Konzept und die spätere Umsetzung zu erhalten. Im letzten Abschnitt dieses Kapitel werden zur Veranschaulichung von Lernsoftware zwei verschiedene Anwendungen vorgestellt.

4.1 Hypermedia Learning Object System

Im Folgenden sollen die in hylOs integrierten Werkzeuge und die XML-Datenstrukturen untersucht sowie deren Generierung aufgezeigt werden.

4.1.1 Die Autorenumgebung

Zur Arbeit mit den Lernobjekten stellt hylOs eine umfangreiche Autorenumgebung zur Verfügung. Die Hauptapplikationen zum Bearbeiten von Lernobjekten sind der Lernobjekt-Editor und der Instructional Designer. Außerdem stellt die Autorenumgebung noch kleinere Werkzeuge wie eine Bild- und Onlinematerialienverwaltung, ein Glossar- und ein Literaturverzeichnis bereit.

Der Lernobjekt-Editor

Der Lernobjekt-Editor ermöglicht es Autoren, Lernobjekte und Metadaten zu erstellen und zu editieren, sowie Relationstypen zum Verknüpfen von Lernobjekten anzulegen. Der Editor integriert zum Bearbeiten der Lernobjekte ein Modul zur Inhaltseingabe und ein weiteres Modul zum Bearbeiten der Metadaten.

Das Modul zur Inhaltseingabe präsentiert sich als WYSIWYG¹-Editor, wodurch dadurch die Arbeit für die Autoren stark vereinfacht wird. Nach Eingabe der Daten kann der Autor in einer Vorschauansicht das Endergebnis sofort im Browser kontrollieren. Innerhalb des Inhaltseditors werden Lernobjekte angelegt, denen Inhaltsobjekte zugefügt werden. Inhaltsobjekte können Textparagrafen, Tabellen, Mediencontainer oder Literaturverzeichnisse sein. Ein Textparagraf besteht aus strukturierten Daten wie: Überschriften verschiedenen Grades, Marginalien, Fließtexten, Bildern, Formeln etc. Der Inhaltseditor hält weiterhin

¹ „What You See Is What You Get“

ein Werkzeug zur genauen Klassifizierung von Textparagrafen, Tabellen oder Mediencontainern bereit. Damit kann beispielsweise ein Textparagraf abhängig vom Kontext gekennzeichnet werden. Ein Mediencontainer kann Audiodateien, Flash- und Videofilme beinhalten. Es ist nicht dokumentiert, welche Formate bei Audio- und Videodateien erlaubt sind.

Die erste Eingabemaske beim Öffnen eines Lernobjektes im Inhaltseditor erfragt notwendige Informationen über das Lernobjekt, die jeder Autor angeben sollte. Die Daten sind ein Auszug aus dem LOM Metadaten Standard. Zur Vereinfachung der Erfassung der Metadaten sind die manuell einzugebenden Daten auf das Nötigste reduziert und die Pflichtdaten teilweise bereits auf Standardwerte gesetzt. Pflichtdaten sind beispielsweise die Länge der Lerneinheit, Schwierigkeits- oder Interaktionsgrad.

Das Metadatenmodul innerhalb des Lernobjekt-Editors

Neben der reduzierten Eingabemaske für Metadaten gibt es zur detaillierteren Eingabe noch das Metadatenmodul. Das Metadatenmodul baut auf dem LOM Standard auf und enthält die nach diesem Standard definierten Kategorien. Diese Kategorien unterteilen sich wie folgt:

- Allgemein
enthält beispielsweise Titel des Lernobjektes, Beschreibung, Schlagworte etc.
- Lebenszyklus
gibt u.a. den Status der Bearbeitung an und enthält eine Liste der Autoren, die dieses Lernobjekt bearbeitet haben mit einer Datumsangabe
- Meta-Metadaten
enthält Informationen über die Metadaten selbst und eine Liste, welcher Autor die Metadaten wann bearbeitet hat.
- Pädagogisch
dient u.a. zur pädagogischen Einordnung des Lernobjektes, es kann beispielsweise die Zielgruppe, der Schwierigkeitsgrad, Lernzeit etc. angegeben werden.
- Rechte
dort können beispielsweise Rechte zur Endgelpflicht oder zum Urheberrecht eingetragen werden
- Beziehungen
dient zur Eingabe von Beziehungen des Lernobjektes zu anderen Lernobjekten

- Annotationen
Möglichkeit für den Autor, um Kommentare zu setzen
- Klassifikationen
Klassifizierung in bestimmten Fachgebieten bzw. Anwendungsbereichen (aktuell stehen die ACM CCS¹ und die DDC² Taxonomien zur Verfügung)

Durch Anwählen der einzelnen Kategorien hat der Autor die Möglichkeit in verschiedenen Textfeldern sein Lernobjekt näher zu beschreiben und einzuordnen. Metadaten wie Name des Autors oder Medientyp werden automatisch durch das System generiert. Alle Metadaten einschließlich der automatisch generierten können bei Bedarf verändert werden.

Die einzelnen Kategorien verfügen – wie bei der vorhergehenden Aufzählung kurz beschrieben – über weitere Unterfunktionalitäten. Die Kategorie Beziehung umfasst beispielsweise achtzehn unterschiedliche Arten von Relationen. Diese sind in der nachfolgenden Tabelle aufgelistet:

<i>Arten der Kategorie Beziehung in hylOs</i>		
Basiert auf	Ist Format von	Ist weniger detailliert als
Benötigt	Ist Grundlage	Referenziert
Hat Bestandteil	Ist Version von	Veranschaulicht/Erläutert
Hat Format	Ist allgemeiner als	Wird benötigt von
Ist Alternative zu	Ist detaillierter als	Wird referenziert von
Ist Bestandteil von	Ist spezieller als	Wird veranschaulicht/ erläutert von

Tabelle 4:1. Semantische Relationen in hylOs

Die Beziehungen – auch semantische³ Relationen genannt – bilden anhand sinngebender Bezeichnungen die Art der Relation zwischen den angegebenen Lernobjekten ab. Daraus ergibt sich unter den Lernobjekten ein Netz von Beziehungen, welches als semantisches Netz bezeichnet wird.

¹ Computer Classification System

² Dewey Decimal Classification

³ nach inhaltlicher Bedeutung miteinander verknüpfen

Der Instructional Designer

Das zweite Hauptwerkzeug in hylOs stellt der Instructional Designer dar. Dieses Werkzeug erlaubt dem Autor das Erstellen von Lerneinheiten aus bereits vorhandenen Lernobjekten. Demnach können neue Kurse auf Grundlage bereits vorhandener Lernobjekte erstellt und neu strukturiert werden. Nützlich kann die Neustrukturierung sein, wenn ein Kurs beispielsweise für eine andere Zielgruppe aufbereitet oder der Inhalt in einem anderen Kontext verwendet werden soll. Bei der Bearbeitung des neuen Kurses stehen dem Autor eine Reihe nützlicher Funktionen zur Verfügung. Über die Suchfunktion können bereits vorhandene Lernobjekte über verschiedene Parameter gesucht werden. Erleichtert wird die Suche durch eine Vorschaufunktion der gefunden Lernobjekte. Bei der Zusammenstellung von Lernobjekten können ein oder mehrere Lernobjekte auf einer Seite dargestellt werden. Zur didaktischen Zusammenführung der einzelnen Lernobjekte stehen dem Autor ein Prolog und ein Epilog zur Verfügung.

Weitere Werkzeuge

Neben den Hauptapplikationen Lernobjekteditor und Instructional Designer stellt hylOs noch eine Glossar-, Literatur-, Bilder- und Onlinematerialienverwaltung zur Verfügung.

Die Glossarverwaltung erlaubt das Anlegen eigener Glossare und das Hinzufügen einzelner Glossareinträge in bestehende Glossarverzeichnisse. Glossare dienen dem Lernenden zur Übersichtlichkeit und zum Nachschlagen von Schlagwörtern. Das Glossarverzeichnis entsteht durch Verknüpfungen innerhalb der Lerneinheit.

Die Literaturverwaltung dient zum Importieren von Bibliographien und erlaubt die manuelle Eingabe neuer Literatureinträge. Um ein Literaturverzeichnis nutzen zu können, werden zuerst die Literatureinträge in der Literaturverwaltung erstellt. Anschließend wird im Lernobjekt-Editor ein Literaturverzeichnis angelegt und die einzelnen Einträge, die das Verzeichnis enthalten soll, ausgewählt.

In der Bildverwaltung kann der Benutzer eigene Bilder einstellen und verwalten. Die Onlinematerialienverwaltung dient zum Einstellen und Verwalten von Materialien, beispielsweise PDFs, PowerPoint-Dokumenten, Filmen oder Animationen, die dem Lernenden zum Download zur Verfügung gestellt werden können. Dabei gibt es keine dokumentierten Beschränkungen was das Format bzw. die Größe der einzustellenden Dateien betrifft.

4.1.2 XML-Dokumenttypen in hylOs

Die Editoren und Werkzeuge von hylOs erlauben die Eingabe verschiedenartiger Inhalte. Dabei gibt es mehrere Typen von XML-Dokumenten, die die unterschiedlichen Inhalte abbilden. Die XML-Dokumente sind anhand eines speziellen Parameters gekennzeichnet. Die folgende Liste zeigt die wichtigsten XML-Dokumententypen in hylOs:

- **Inhalt-XML**
enthält den gesamten Inhalt der Lerneinheit
adressiert über: content
Beispielpfad: <http://hylosdev.fhtw-berlin.de/content/<Lernobjekt>.debug¹>
- **Baum-Navigation-XML**
enthält die Verknüpfung zu sämtlichen Unterlernobjekten des Lernobjektes
adressiert über: navtree
Beispielpfad: <http://hylosdev.fhtw-berlin.de/navtree/<Lernobjekt>.<debug>>
- **Lernpfad-Navigation-XML**
ist ein Lernobjekt über den Instructional Designer (siehe Kapitel 4.1.1, S. 36) erstellt worden, werden in hier die zusammengeführten Lernobjekte mit dem Prolog und dem Epilog abgebildet.
adressiert über: navinst
Beispielpfad: <http://hylosdev.fhtw-berlin.de/navinst/<Lernobjekt>.<debug>>
- **Semantische-Relationen-XML**
enthält die Verknüpfung zu sämtlichen semantischen Relationen des Lernobjektes
adressiert über: navcons
Beispielpfad: <http://hylosdev.fhtw-berlin.de/navcons/<Lernobjekt>.<debug>>
- **Bibliographie-XML**
enthält das mit dem Lernobjekt verknüpften Literaturverzeichnis
adressiert über: bib
Beispielpfad: <http://hylosdev.fhtw-berlin.de/bib/<Lernobjekt>.xml>
- **Glossar-XML**
enthält sämtliche im Lernobjekt enthaltenen Glossarbegriffe
adressiert über: glossary
Beispielpfad: <http://hylosdev.fhtw-berlin.de/glossary/<Lernobjekt>.xml>

¹ Die Endung, in diesem Fall „debug“, wird als Pipeline in der Sitemap von Cocoon definiert

4.1.3 Aufbau eines generierten XML-Dokumentes in hylOs

Für die Erzeugung des späteren Prototypen ist es notwendig alle Objekte mit denen hylOs arbeitet – speziell im Inhaltsbereich – zu identifizieren. In Kapitel 4.1.1 wurde der Lernobjekteditor und die möglichen Unterobjekte beschrieben. Diese Unterobjekte spiegeln sich in dem generierten Inhalt-XML-Dokument wieder. Das folgende Listing stellt einen beispielhaften XML-Code eines Lernobjektes dar:

```
1. <hylos:content>
2. <hylos:section>text</hylos:section>
3.   <paragraph marginalNote="Marginalie">text
4.     <br />
5.     <enumeration>
6.       <item>text</item>
7.       <item>text</item>
8.       <item>text</item>
9.     </enumeration>
10.    <image  caption="text" floating="false"
11.          inline="false" profile="standard"
12.          src="pfad"
13.          zoomable="false" />
14.  </paragraph>
15. </hylos:content>
```

Listing 1: Beispielhafte XML-Struktur in hylOs

Content gibt an, dass die folgenden Elemente zum Inhaltsbereich gehören. Danach folgt das Element section. Dieses markiert eine Überschrift ersten Grades. Das Element paragraph besitzt das Attribut marginalNote, d.h. neben dem Paragraphen wird eine Marginalie angezeigt. Der Paragraph enthält Fließtext, es folgen das Element enumeration - eine nummerierte Liste und image – ein Bild mit verschiedenen Attributen.

In hylOs können eine Reihe von weiteren Inhaltsobjekten angezeigt werden. In Tabelle 4:2 werden auszugsweise Inhaltsobjekte mit ihren dazugehörigen Elementnamen und möglichen Attributen aufgelistet.

<i>Elementname</i>	<i>Elementbeschreibung</i>	<i>Mögliche Attribute</i>
Strukturelemente		
Paragraph	Textparagraph	marginalNote=" " contentClass="addInfo" contentClass="animation" contentClass="definition" contentClass="experiment" contentClass="formula" contentClass="hint" contentClass="learningObjective" " contentClass="learningTime" contentClass="outline" contentClass="poem" contentClass="mouse-over" contentClass="calcex" contentClass="test" contentClass="tip" contentClass="audio" contentClass="video" contentClass="cite" contentClass="exercise"
table		align=" " border=" " marginalNote=" "
richmedium	andere Datenstruktur beispielsweise Video, Flashanimation etc.	src=" " base=" " mediaType=" " openNewWindow=" " height=" " width=" "
Inhaltselemente		
section	Überschrift ersten Grades	
subsection	Überschrift zweiten Grades	
br	Erzwungener Zeilenumbruch	
separator	Horizontale Linie	

4.1.3 AUFBAU EINES GENERIERTEN XML-DOKUMENTES IN HYL0S

<i>Elementname</i>	<i>Elementbeschreibung</i>	<i>Mögliche Attribute</i>
Inhaltselemente		
strong	Auszeichnung des Textes bold	
emph	Auszeichnung des Textes kursiv	
itemize	Nicht-Nummerierte Liste	
item	Ein Teil einer Liste	
enumeration	Nummerierte Liste	
description	Beschreibungsliste	
image	Bild	caption=" " floating=" " inline=" " profile=" " src=" " zoomable=" "
Hyperreferenzen		
mir:link	Hyperreferenz	mir:type=" "
mir:resource		xlink:type=" " mir:id=" "
mir:locator		mir:targetType=" " mir:targetType=" " mir:targetType=" " mir:extern=" " xlink:href=" " xlink:title=" " xlink:type=" "
Mathematische Formeln		
formula	Mathematische Formel	caption=" " src=" "
math	MathML ¹ Wurzelement	xmlns=" " xmlns:xlink=" "

¹ Mathematical Markup Language; W3C-Standard zur Darstellung von mathematischen Formeln und komplexen Ausdrücken

4.1.3 AUFBAU EINES GENERIERTEN XML-DOKUMENTES IN HYL0S

<i>Elementname</i>	<i>Elementbeschreibung</i>	<i>Mögliche Attribute</i>
		xmlns:pref=" " mode=" " overflow=" "

Tabelle 4:2. Auszug der möglichen Inhaltselemente in hyl0s

4.2 Funktionalitäten und Merkmale der PlayStation Portable

Die PlayStation Portable verfügt über ein – für ein mobiles Gerät – relativ großes Display mit den Maßen 95 x 54 mm. Die Auflösung des Bildschirms beträgt 480 x 272 Pixel bei 16.777.216 Farben. Um Daten auf der PlayStation zu speichern, benötigt man einen Memory Stick Pro Duo. Dieser ist erhältlich mit Speichergrößen ab 32 MB . Auf dem Memory Stick ist die Firmware gespeichert und der Nutzer kann beliebige Daten, beispielsweise Bilder, Musik, Videos etc. ablegen. Im Folgenden sind die für diese Arbeit relevanten Funktionen der PlayStation Portable aufgeführt. Diese Angaben beziehen sich auf die aktuelle Firmware 3.01:

- integrierter Webbrowser mit Flash Player 6.0 Unterstützung
- Bildformate: TIFF, GIF, PNG, BMP, JPG
- Audioformate: AAC, MP3, WMA (nur mit online-Aktivierung), WAV, MP4
- Videoformate: MPEG 4
- Unterstützung RSS-Funktion – Audio- und Videopodcasts, Bilder
- WLAN-Verschlüsselung WPA/WPA-PSK, WEP
- Integrierte Stereolautsprecher
- Laufwerk für Universal Media Disk (kurz UMD)

Um eine genaue Anforderungsanalyse für die Applikation zu erstellen, werden im Folgenden die verschiedenen Navigationsstrukturen der PlayStation Portable, die Tastenfunktionalitäten und die Eigenschaften des PlayStation Portable-Webrowsers genauer erläutert.

4.2.1 Basisfunktionalitäten der PlayStation Portable

Grundlegende Tastenfunktionen

In Abbildung 4 sind die grundlegenden Funktionen der Tasten der Portable PlayStation dargestellt. Dabei dienen die Richtungstasten zum Navigieren, um sich von Navigationspunkt zu Navigationspunkt zu bewegen und die Zeichentasten (Dreieck, Kreis, Quadrat, Kreuz) zur Arbeit mit Optionen. Mittels der HOME-Taste kann von jedem beliebigen Punkt zur Startnavigation der PlayStation Portable zurück navigiert werden.

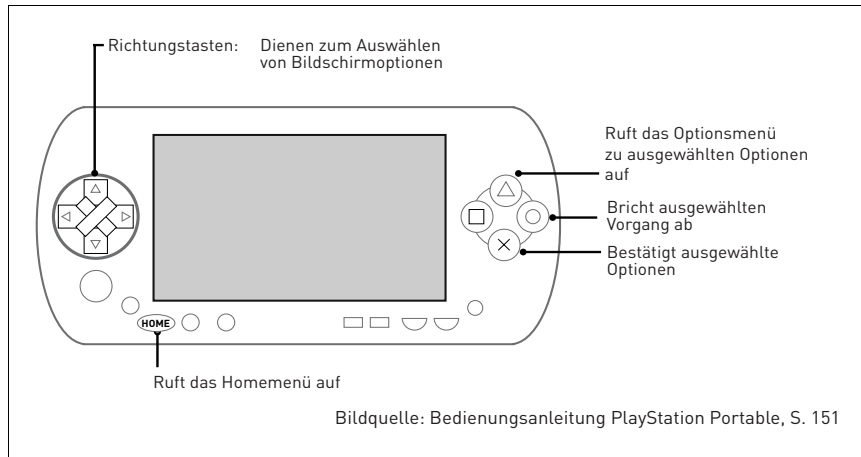


Abbildung 4: Grundlegende Tastenfunktionen

Die Basisnavigation

Die Basisnavigation bezieht sich auf das HOME-Menü der PlayStation Portable (vgl. Abbildung 5). Diese Navigation arbeitet mit drei Ebenen. Dabei kann in der ersten Ebene durch horizontales Navigieren (Richtungstasten rechts/links) eine Kategorie mithilfe der Kreuztaste angewählt werden. Es gibt sechs verschiedene Kategorien: Einstellungen, Musik, Bilder, Videos, Spiele, Netzwerk. Nach Auswahl der Kategorie kann durch vertikales Navigieren (Richtungstasten oben/unten) in der zweiten Ebene ein Unterpunkt mittels der

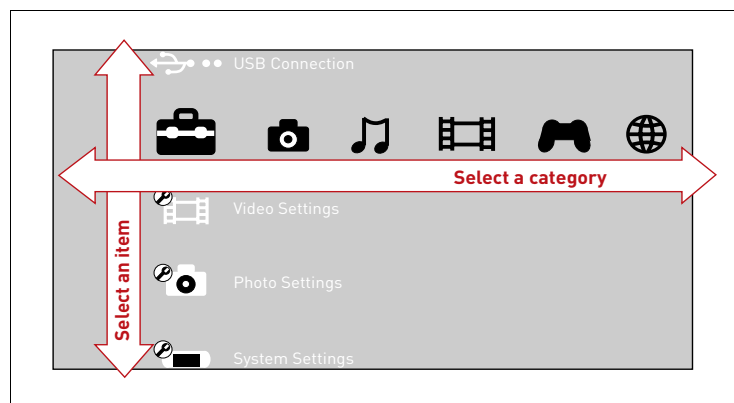


Abbildung 5: Startnavigation der PlayStation Portable

Kreuztaste ausgewählt werden. In Abbildung 5 ist beispielsweise die Kategorie Einstellungen angewählt und es sind einzelne Unterpunkte dargestellt. Nach Auswahl eines Kategorie-Unterpunktes folgt die dritte Ebene. Hier kann der Nutzer wiederum horizontal mittels

der Richtungstasten oben/unten durch die angezeigten Optionen navigieren und diese mit der Kreuztaste bestätigen.

4.2.2 Der Internetbrowser der PlayStation Portable

Im Folgenden wird die Navigation des Webbrowsers der PlayStation Portable beschrieben. Dabei bezieht sich die Beschreibung auf das europäische Modell der PlayStation Portable. Die detaillierte Beschreibung ist notwendig, um die Navigation für den späteren Prototypen festlegen zu können. Grundlegend ist der Webbrowser der PlayStation Portable in zwei Ebenen aufgeteilt. Die obere Ebene stellt die Browseroberfläche dar und die zweite Ebene die aktive Webseite. Entsprechend sind die Tastenfunktionen, die Navigationsarten und die Oberfläche an dieses 2-Ebenen-Konzept angepasst.

Grundlegende Tastenfunktionen und Navigationsarten

Abbildung 6 zeigt die Tasten, die im Browser verwendet werden können und gleichzeitig die damit verbundenen Funktionen. Die Zeichentasten (Quadrat, Dreieck, Kreis, Kreuz) sind im Webbrowser angepasst an die Basistastenfunktionen der PlayStation Portable. Dies erleichtert dem Benutzer die Navigation mit den Tasten. Kurz erläutert, haben die Zeichentasten folgende Funktionen: die Dreieckstaste aktiviert die Browseroberfläche, mithilfe der Richtungstasten können die einzelnen Funktionen ausgewählt werden, die Kreistaste schließt den Browser und die Kreuztaste dient zum Bestätigen der jeweiligen Option. Die Quadrattaste hat nur bei aktiver Webseite eine Funktion. Der Webbrowser verfügt dabei über drei Tabfenster und mit der Quadrattaste ist es möglich, eine Seite in einem anderen Tab zu öffnen. Durch die Kombination der Quadrattaste und der „Seite vorblättern“- bzw. „Seite zurückblättern“-Taste ist das Navigieren zwischen den Tabfenstern möglich.

Die Navigation auf einer Webseite kann auf drei Arten erfolgen: Durch die Richtungstasten, den Analog-Pad(8) oder die Quadrattaste(1) in Kombination mit den Richtungstasten. Beim Navigieren mit den Richtungstasten werden die jeweiligen Links auf der Webseite in der Reihenfolge ihres Auftretens „angesprungen“ und farblich markiert, um dem Nutzer anzuzeigen welcher Link gerade aktiv ist. Die Navigation mit dem Analog-Pad funktioniert ähnlich der Maus. Dabei kann mit dem Mauszeiger frei über den aktuell sichtbaren Seitenausschnitt navigiert werden und ein beliebiger Link ausgewählt werden. Das „Auswählen“ eines Links erfolgt bei beiden Navigationsarten mit der Kreuztaste. Bei der dritten Navigationsart – der Kombination aus Quadrattaste und Richtungstasten – kann über die gesamte



Abbildung 6: Navigation des Internetbrowsers

Länge der Webseite gescrollt werden, zum Auswählen eines Links wird dann allerdings eine der anderen beiden Navigationsarten genutzt. Das Zurückblättern(6) auf die vorhergehende Seite erfolgt mit der in Abbildung 6 links oben dargestellten und das Vorblättern(5) mit der rechts oben dargestellten Taste.

Beim Testen verschiedener Webseiten ist aufgefallen, dass der Mauszeiger nicht sofort auf das Loslassen des Analogsticks reagiert, was bei der Positionierung des Mauszeigers auf kleineren Buttons zu Problemen führen kann. Denn der Analogstick arbeitet nicht allzu präzise und damit ist eine exakte Steuerung der Maus nicht möglich.

PlayStation Portable-Browseroberfläche

Der Webbrowser der PlayStation Portable hat durch das 2-Ebenen-Konzept keine permanente Ansicht, sie kann ausgeblendet bzw. deaktiviert werden. Das bedeutet, um mit dem Browser zu arbeiten, wird die Browseroberfläche aktiviert. Möchte sich der Benutzer auf der Webseite bewegen, muss die Browseroberfläche deaktiviert und die Webseite somit in Vollansicht zu sehen sein. Die Browseroberfläche liegt als zweite Ebene über der Webseite und kann durch Einblenden genutzt werden. Soll beispielsweise eine neue Webseite in der Adressleiste aufgerufen werden, wird die Browseroberfläche durch die Dreieckstaste eingeschaltet.

Das Browsermenü besteht aus ähnlichen Elementen wie ein PC-Webbrowser, aber hält zur Vereinfachung der Bedienung und aus Platzgründen nur die wichtigsten Optionen bereit. Die Bildschirmoberfläche ist bei aktivierter Browseroberfläche in zwei Bereiche gegliedert. Am oberen Bildschirmrand befindet sich die Adressleiste und am unteren Bildschirmrand

befindet sich eine Optionsleiste mit folgenden Funktionen: Datei, Zurück, Vor, Aktualisieren, Home, Lesezeichen, Verlauf, Ansicht, Werkzeuge, Hilfe. Die Optionen sind anhand von kleinen Symbolen visualisiert und beim Aktivieren wird eine kurze Texterklärung über dem Symbol angezeigt (vgl. Abbildung 6 - Webseiten-Optionsleiste).

Zum Eingeben einer Webadresse in die Adressleiste, wird die Browseroberfläche aktiviert und die Adressleiste ausgewählt. Dabei öffnet sich ein Texteditor mit verschiedenen Optionen zum Eingeben von Internetadressen. Der Texteditor ist gegliedert in ein Tastaturfeld ähnlich wie das einer Handytastatur, das Tastatur-Optionsfeld, eine Navigationsleiste und das eigentliche Textfeld, in welchem der geschriebene Text angezeigt wird. Häufige Eingaben wie „http://“, „www.“, „.de“ etc. sind vordefiniert, sodass eine schnellere Eingabe möglich ist. Allgemein ist es jedoch sehr mühsam Text einzugeben, vor allem wenn es sich um eine längere URL handelt.

PlayStation Portable Browserfunktionalität

Für die spätere Implementierung ist neben der genauen Analyse der Basisfunktionalitäten der PlayStation Portable auch eine detaillierte Analyse der technischen Merkmale des Browsers notwendig.

Unterstützung Flash

Seit der Firmwareversion 2.70 unterstützt der Browser die Anzeige von Flashfilmen in der Version 6. Bevor die Option des Flash Players jedoch genutzt werden kann, muss der Benutzer diese erst in den Systemeinstellungen der PlayStation Portable online aktivieren. Nach der Aktivierung können sowohl in HTML eingebettete SWF-Dateien benutzt werden als auch direkt die SWF-Dateien über die Adressleiste angewählt werden (vgl. [BRO]).

Es werden jedoch nicht alle Funktionen des Flash Player 6 unterstützt. Dabei soll an dieser Stelle erwähnt werden, dass zu diesem Zeitpunkt keine offizielle Liste der unterstützten Funktionen - weder von Sony noch von Adobe - veröffentlicht ist. Die nachfolgenden Informationen beziehen sich auf die private Webseite [FLA] eines Entwicklers von Adobe Mobile.

Unterstützte Flash Player 6 – Funktionen:

- **Maus-Funktionen** – das Analog-Pad und die Kreuztaste arbeiten als eine „Ein-Tasten-Maus“ Bewegungs-, Klick- und Drag & Drop – Funktionen sind unterstützt.
- **Tastatur-Funktionen** – die Richtungstasten arbeiten wie die Pfeiltasten auf der PC-Tastatur

- **Texteingabe** – Texteingabe in Textfeldern wird durch die Tastatur des Texteditors des Browsers unterstützt
- **Font** – die Systemschrift der PlayStation Portable wird unterstützt.
- **Ton** – unterstützt PCM, ADPCM und MP3-Wiedergabe für Lautsprecher und Kopfhörer
- **Tastencodewerte** – Nach der Aussage von Alexander McCabe [CEN] können bestimmte Tasten der PlayStation Portable belegt werden, folgend sind diese mit den dazugehörigen Tastencodewerten aufgeführt:
Pfeil-Tasten links – Tastencode 37, rechts – Tastencode 39,
 oben – Tastencode 38, unten – Tastencode 40
Quadrat-Taste in Kombination mit den Pfeiltasten – Tastencode 27

Nichtunterstützte Flash Player 6 – Funktionen

- **Zwischenablage** – kein Kopieren, Ausschneiden und Einfügen von Text möglich
- **Video** – keine Wiedergabe von Videodaten als H.263, Sorenson Video oder Motion JPEG möglich
- **Kontext Menü** – keine Anzeige, Kontrolle und Eigenschaften des Kontext Menüs möglich
- **Druckfunktion** – nicht unterstützt
- **Live Verbindung** – keine Kommunikation zwischen JavaScript und Flash Player möglich
- **XMLSocket** – beständige Verbindung und Kommunikation zu einem Server mithilfe eines XMLSocket-Objektes nicht möglich

Bekannte Probleme

- Frame Rate variiert zwischen 18-24 fps (frames per second), abhängig vom Inhalt
- große Flashfilme können nicht abgespielt werden, die Größe des Inhalts einer Seite sollte 1,5 MB nicht überschreiten
- das Laden externer Daten unterscheidet sich von PCs, das Beenden des Ladens muss bestätigt werden.
- die Systemschrift ist in 5 unterschiedlichen Größen vorhanden

Ausgehend von der ausführlichen Analyse des hylOs-Systems und der PlayStation Portable können nun spezifische Anforderungen für das Konzept und somit für die spätere Umsetzung definiert werden.

4.3 Beispiele für Lernsoftware

Bevor jedoch eine Anforderungsdefinition vorgenommen wird, werden zwei unterschiedliche Beispiele von Lernsoftware genauer vorgestellt, um einen Eindruck von bereits umgesetzten Applikationen im Bereich von mobilem Lernen bzw. Edutainment zu gewinnen.

Einerseits wird in diesem Abschnitt eine kommerzielle Software von Sony Computer Edutainment für die PlayStation Portable, andererseits eine Lernsoftware, die von Studenten der Fachhochschule für Technik und Wirtschaft Berlin mit Macromedia Flash entwickelt wurde, vorgestellt.

Edutainment ist eine Kombination von Education¹ und Entertainment². Damit werden multimediale Anwendungen bezeichnet, die einerseits Wissen vermitteln und andererseits unterhalten sollen. Als Edutainment können beispielsweise Wissenssendungen wie „Die Sendung mit der Maus“ im Fernsehen bezeichnet werden oder Computerspiele, die didaktisch so aufbereitet sind, dass der Benutzer tatsächlich einen Lernnutzen daraus ziehen kann.

4.3.1 Lernsoftware für PlayStation Portable

Die Sprachsoftware „Talkman“ verfolgt die Ziele von Edutainment und ist gleichzeitig ein Beispiel für mobiles Lernen. Die Software wurde für die PlayStation Portable entwickelt und somit ist das Konzept auf orts- und zeitunabhängiges Lernen ausgerichtet. Im folgenden soll die Software näher erläutert werden.

„Talkman“ möchte grundlegende Kenntnisse in sechs verschiedenen Sprachen spielerisch vermitteln. Dabei verfolgt die Software nicht den Ansatz, dass der Benutzer nach Gebrauch der Software die Sprache perfekt beherrscht, sondern bietet eher die Möglichkeit, grundlegende Phrasen für einen Urlaub im jeweiligen Land zu vermitteln.

In der Produktbeschreibung (vgl. [Son06]) ist „Talkman“ als multilingualer Dolmetscher für die Westentasche beschrieben. Neben der Übersetzer-Funktion, stellt „Talkman“ zusätzlich ein Aussprache- und ein Zuhörspiel bereit. Außerdem sind hilfreiche Werkzeuge für den Urlaub, wie einen Einheitenrechner zum Umrechnen von Währungen, Längen, Temperaturen etc., eine Alarmfunktion für Termine und eine Sprachnotizfunktion zum Speichern persönlicher Anmerkungen in die Software integriert. Die Analyse von Talkman

¹ zu dt. Bildung

² zu dt. Unterhaltung

erfolgt ausschließlich für die drei großen Anwendungen Übersetzer, Aussprache- und Zuhörspiel, die Hilfswerkzeuge werden bei der Analyse vernachlässigt.

Als Moderator der Software dient ein großer blauer Vogel namens Max. Er begleitet den Benutzer durch alle Szenerien, fungiert als Sprachrohr, arbeitet als Dolmetscher, als Lehrer und Wegweiser und gibt zwischen den Szenen Tipps zum Lernen mit der Software und zum Erlernen von Fremdsprachen.

Die Dolmetscher-Funktion

Ziel der Dolmetscher-Funktion ist der Dialog mit einer fremdsprachigen Person über die PlayStation Portable. Die anfänglichen Einstellungen betreffen Sprache und Geschlecht des Benutzers und des Gesprächspartners. Nun kann der Dialog beginnen: der Benutzer wählt aus 28 typischen Urlaubsszenarien z.B. im Hotel, am Strand, am Flughafen oder im Restaurant, das gewünschte Szenario aus und spricht entweder in das mitgelieferte Mikrofon oder kann mittels eines Spickzettels vorgefertigte Phrasen auswählen und von Max übersetzen und sprechen lassen. Der Spickzettel ist ein Werkzeug zum Nachschlagen der in Talkman enthaltenen Phrasen. Im Anschluss kann der Gesprächspartner in seiner Sprache die Antworten ins Mikrofon sprechen oder wählt wiederum die Antwort aus dem Spickzettel aus. Der Vogel übersetzt und antwortet in der Benutzersprache.

Das Aussprachenspiel

Beim Aussprachenspiel fungiert Max als Lehrer, der Phrasen an die Tafel schreibt und anschließend ausspricht. Die Lerneinheiten sind in ein Puzzlespiel mit 5 x 5 Puzzleteilen eingebunden. Hinter jedem Puzzle verbergen sich 5 Phrasen. Je weiter rechts und unten sich ein Puzzleteil befindet desto schwieriger und länger sind Phrasen. Wird ein Puzzleteil ausgewählt, liest Max die erste Phrase vor. Daraufhin kann der Benutzer die Phrase ins Mikrofon sprechen oder sich die Ansage wiederholen lassen, zum besseren Verständnis sogar in einem verlangsamten Modus. Anschließend bewertet Max die Aussprache des Nutzers auf einer Skala von A – D, wobei A einem „sehr gut“ entspricht. Nach der Bewertung – welche gespeichert wird – hat der Benutzer die Wahl, die Phrase erneut einzugeben oder er kann zur nächsten Phrase übergehen. Die Lektion ist erfolgreich absolviert, wenn alle 5 Phrasen mit A bestanden sind, dann erscheint auf dem Puzzleteil ein Bild. Die Lektionen können beliebig oft wiederholt werden, selbst wenn bereits das Bild auf dem Puzzleteil zu sehen ist.

Das Zuhörspiel

Auch das Zuhörspiel ist nach dem Puzzle-Prinzip aufgebaut, doch in den Lektionen werden dem Benutzer jetzt vier Phrasen angesagt, deren Übersetzung auf dem Display angezeigt

wird. Die Phrasen kann der Benutzer sich beliebig oft anhören. Im Anschluss wählt Max eine der vier Phrasen und der Benutzer soll über Multiple Choice die richtige Übersetzung wählen. Eine Lektion enthält fünf Multiple Choice Fragen und die Lektion ist bestanden, wenn alle Fragen richtig beantwortet sind. Auch hier ist die „Belohnung“ das Aufdecken eines Bildes im Puzzle.

Das Fazit für „Talkman“

Aussprache- und Zuhörspiel sind didaktisch sehr gut aufbereitet und der Benutzer kann sich tatsächlich Grundkenntnisse der verschiedenen Fremdsprachen aneignen. Der Lernende hat jederzeit die Möglichkeit, die Lektionen zu wiederholen oder abubrechen, Zwischenstände werden auf dem Memory-Stick der PlayStation Portable abgespeichert, somit können abgebrochene Lektionen dort wieder aufgenommen werden, wo sie zuvor unterbrochen wurden. Max als Begleiter und Lehrer hat aufmunternde und lobende Worte parat und motiviert den Benutzer zum Weiterlernen.

Eine Schwäche der Software zeigt sich bei den langen Ladezeiten zwischen den einzelnen Schritten, wenn der Zwischenstand gespeichert oder die nächste Phrase angezeigt wird. Eine weitere Schwachstelle zeigt sich bei der freien Übersetzung. Es ist sehr mühevoll, einen Dialog zu führen, da bekannt sein muss, in welchem Szenario die gesuchten Phrasen enthalten sind. Sinnvoller wäre zusätzlich zu den Szenarien einen freien Bereich bereitzuhalten, der alle Phrasen enthält. Wahrscheinlich gibt es die Szenarien aus Gründen der Performance, damit die Suche nach der gewünschten Phrase beschleunigt wird. Die Sprachnotizfunktion, welche bei den Werkzeugen enthalten ist, wäre sinnvoller in den Lektionen, zum Einfügen eigener Anmerkungen, zu nutzen. Doch aus den Lektionen kann die Sprachnotizfunktion leider nicht angewählt werden. WO sind die und warum ungünstig

4.3.2 Lernsoftware für Wirtschaftsingenieure

Unter <http://wilo.fhtw-berlin.de/> wurde von Studenten der FHTW ein Lernportal für Wirtschaftsingenieure errichtet. Dort haben die Studenten für Wirtschaftsingenieurwesen die Möglichkeit einen Theorie- und einen Praxisteil zum Thema „Erstellung eines Arbeitsplans am Beispiel der Antriebswelle“ als E-Learning-Einheit zu absolvieren. Der Theorie-Teil ist mit hylOs aufbereitet und der Übungsteil ist mit Macromedia Flash realisiert und wird als Flash-Anwendung zur Verfügung gestellt.

Ursprünglich sollte der Übungsteil direkt auf der PlayStation Portable getestet werden, aber aus anzeigetechnischen Gründen konnte die Anwendung dort nicht angezeigt werden. Le-

diglich die Navigation wird angezeigt, aber es ist nicht möglich zu navigieren. Somit wurde der Test im Webbrowser auf dem PC durchgeführt.

In der Übungsanwendung wird der Lernende in die Rolle eines Praktikanten in einem Unternehmen versetzt und bekommt von seiner Vorgesetzten eine Aufgabe im Bereich Arbeitsplanung, die es zu lösen gilt. Während der Übung durchläuft der Lernende die unterschiedlichen Abteilungen des Unternehmens und trifft auf Personen, mit denen er gemeinsam die Aufgabe löst.

Die Übung ist in sieben Module gegliedert und der Praktikant arbeitet diese hintereinander ab. Dabei durchläuft er den gesamten Prozess für die Grobplanung einer Antriebswelle – von der technischen Zeichnung bis zur Produktion – und kann in praktischen Übungen sein Wissen aus dem theoretischen Teil des Kurses anwenden.

Modul 1: Arbeitsanweisung – Die Vorgesetzte gibt dem Praktikanten per Dialog die Arbeitsanweisung und schickt ihm diese anschließend per Email mit einer technischen Zeichnung des Endproduktes.

Modul 2: Technische Zeichnung – Der Praktikant soll die technische Zeichnung mit allen Eigenschaften und Beschriftungen studieren. Die Zeichnung kann als PDF abgespeichert werden und ist jederzeit über die allgemeine Navigation erneut abrufbar.

Modul 3: Konstruktionsabteilung – Hierbei erfolgt die Abbildung zweier Auszüge der technischen Zeichnung aus Modul 2, allerdings ohne Beschriftung. Der Praktikant soll jeweils sechs vorgegebene Teile der Zeichnung zuordnen.

Modul 4: Materialwirtschaft – Auswahl des Materials für das Produkt. Dem Praktikant stehen zwei Materialarten zur Auswahl, er soll sich für die richtige entscheiden. Trifft er die falsche Entscheidung, erklärt der Kollege der Abteilung, warum diese Entscheidung falsch ist.

Modul 5: Fertigung – Auswahl der Fertigungsart. Der Praktikant hat zwei Fertigungsarten zur Auswahl. Auch hier soll er sich für die richtige Art entscheiden und bekommt wiederum bei der falschen Entscheidung eine Erklärung des Kollegen.

Modul 6: Arbeitsvorbereitung – in 3 Schritte gegliedert:

Prozesszuordnung: Der Praktikant hat die richtige Reihenfolge der Prozesse zu bestimmen. Hierbei sollen die Prozesse auf einem Zeitstrahl per Drag and Drop angeordnet werden.

Auswahl der Betriebsmittel: Bilder von Maschinen soll den einzelnen Prozessen per Drag and Drop zugeordnet werden.

Erstellung des Arbeitsplans: Abbildung eines Standardformulars mit allen Prozessen als Zusammenfassung der Aufgabe.

Modul 7: Produktion – Abschlussbesprechung der Aufgabe mit der Vorgesetzten.

Fazit: Die Übungseinheit gibt dem Lernenden einen sehr guten Einblick in die laufenden Prozesse eines Unternehmens, außerdem ist die Anwendung sehr gut auf den theoretischen Teil abgestimmt und sehr abwechslungsreich gestaltet. Der Lernende kann den Kurs online bewältigen oder sich die Anwendung herunterladen und offline bearbeiten. Bei der Gestaltung der Anwendung wurde eine Navigation im unteren Bereich des Bildschirms angelegt. Dort kann der Lernende unterschiedliche Funktionen nutzen. Dargestellt sind u.a. die Navigationselemente zum Navigieren durch die Module, die Anzeige der Aufgabenstellung und der Zeichnung, sowie ein Fortschrittsbalken anhand dessen er sein Vorankommen in der Lerneinheit kontrollieren kann. Im letzten Modul findet eine Auswertung der bewältigten Aufgabe durch die Vorgesetzte statt. Dabei ist es jedoch irrelevant, wie die tatsächliche Leistung des Lernenden war, die Vorgesetzte lobt in jedem Fall.

Kapitel 5 Anforderungen

In diesem Kapitel werden die Anforderungen an die zu realisierende Anwendung basierend auf den Ergebnissen der Ist-Analyse der beiden Systeme – PlayStation Portable und hylOs definiert. Dabei werden die Anforderungen in allgemeine, inhaltliche, designspezifische und technische Anforderungen unterteilt. Die allgemeinen Anforderungen beschreiben die nötigen Systemvoraussetzungen und -einstellungen. Die inhaltlichen beschreiben die Anforderungen an die Inhalte der Lerneinheiten, die technischen Anforderungen definieren spezifische Voraussetzungen zur fehlerfreien Funktionsweise des Prototyps auf der PlayStation Portable und in den designspezifischen Anforderungen werden die Erfordernisse für die Darstellung bzw. an das Layout für den Prototypen erläutert.

5.1 Allgemeine Anforderungen

Um die mobilen Lerneinheiten nutzen zu können, müssen einige Systemvoraussetzungen und -einstellungen erfüllt sein. Grundvoraussetzung ist das Vorhandensein einer PlayStation Portable mit der Firmwareversion 2.7. Auf der PlayStation Portable muss zunächst in den Systemeinstellungen das Flash-Plugin aktiviert werden. Außerdem ist es notwendig, dass die PlayStation Portable über eine bestehende WLAN-Verbindung mit dem Internet verfügt. Darüber hinaus sollten dem Nutzer die Webseiten bekannt sein, auf denen Lerneinheiten zu finden sind, die mit hylOs publiziert worden sind.

5.2 Inhaltliche Anforderungen

Eine Anforderungsanalyse nach semantischen und didaktischen Kriterien findet im Rahmen dieser Arbeit nicht statt. Die Inhalte der Lerneinheiten werden unverändert aus dem hylOs-System extrahiert und lediglich transformiert.

5.3 Designspezifische Anforderungen

Um dem Benutzer eine klare und einfache Bedienung der Lerneinheit zu ermöglichen, wird die Struktur und damit die Navigation durch die Lerneinheit übersichtlich und benutzerfreundlich gestaltet. Aufgrund der in der Ist-Analyse vorgestellten Tastenfunktionen der PlayStation Portable – insbesondere die Tastenfunktionen für den Webbrowser – ist ein besonderes Navigationskonzept zu entwickeln.

Die Gestaltung des Layouts ist an das Corporate Design des hylOs-Systems angelehnt und wird entsprechend an die Erfordernisse der PlayStation Portable angepasst. Dies betrifft im

Besonderen die Bildschirmauflösung und die Tastenfunktionen zur Navigation durch die Lerneinheit. Die Navigation durch die Lerneinheit wird durch die Richtungstasten in Kombination mit der Quadrattaste und dem Analog-Pad erfolgen. Die Gestaltung der Links bzw. Buttons darf daher nicht zu klein ausfallen, da das Analog-Pad nicht allzu präzise arbeitet und damit eine exakte Maussteuerung nicht möglich ist (siehe Kapitel 4.2.2, S. 45).

5.4 Technische Anforderungen

In der hylOs-Autorenumgebung kann der Autor beliebige Medienformate in die Lerneinheiten einbetten. Diese sollten ebenfalls auf der PlayStation Portable verfügbar sein. Aufgrund der in der Ist-Analyse recherchierten Ergebnisse sind nur bestimmte Medienformate für die PlayStation Portable zulässig (siehe Kapitel 4.2, S.42). Dies muss bei der Umsetzung mit berücksichtigt werden. Das hylOs-System hingegen setzt keine Beschränkungen für Bild- und Videomaterial. Das bedeutet, der Autor kann beliebige Materialien in beliebiger Dateigröße in das System laden.

Weiterhin gibt es Einschränkungen für Datei- und Formatgrößen von Bild, Video- und Animationsformaten angesichts der Bildschirmabmessung sowie der beschränkten Speicherkapazität (ausgehend von einer maximalen Speichergröße der Memory-Card von 32 MB und des Webbrowsercache von max. 2048 KB). Das bedeutet, dass Bilder für den Prototypen auf Format und Dateigröße getestet und aussortiert werden müssen.

Auch für Videos gibt es Einschränkungen: Um Videos in dem Prototypen dynamisch laden zu können, müssen diese in eines der Medienformate, die die PlayStation Portable unterstützt, konvertiert werden (siehe Kapitel 4.2, S. 42). Diese automatische Konvertierung sollte durch hylOs vorgenommen werden. Bis zu dem jetzigen Zeitpunkt liegt ein solches Werkzeug noch nicht im hylOs-System vor, deshalb werden Videos vorerst nicht in den Prototypen eingebunden.

Ein Merkmal des mobilen Lernens ist die plötzliche Unterbrechung des Studierens durch äußere Umwelteinflüsse, bspw. Erreichung des Reisezieles oder der Akku der PlayStation Portable ist leer. Deshalb sollten die mobilen Lernangebote so aufbereitet sein, dass der Studierende jederzeit wieder an das vorher Bearbeitete anknüpfen kann, dass also sein letzter Lernstand gespeichert wird.

Die Lerneinheiten können Verknüpfungen zu externen Webseiten enthalten. Dabei sollte beachtet werden, dass die wenigsten Webseiten für die PlayStation Portable optimiert werden. Es kann also zu Problemen bei der Ansicht der externen Verknüpfungen kommen.

Deshalb sollte der Prototyp alle externen Verknüpfungen in einer Datei auf der Memory-Card speichern, um dem Benutzer eine Möglichkeit zu geben, sich die Verknüpfungen später im PC-Webbrowser anzuschauen. Zusätzlich zu den externen Verknüpfungen sollten auch die Links zu allen Bildern, die in der Lerneinheit vorhanden sind, gespeichert werden. Der Grund ist, dass es trotz der vorherigen Anpassung der Bilder auf die Abmessungen der Bildschirmgröße, zu einer schlechten Ansicht (etwa bei detailreichen Bildern) kommen kann und Einzelheiten nicht mehr erkennbar sind. Durch Speichern der Bildverknüpfungen kann der Benutzer sich die Bilder im größeren Format auf dem PC anschauen.

Kapitel 6 Konzeption

Dem Lernenden sollen unterstützend zur Wiederholung und Vorbereitung des Lernstoffes mobile Lerninhalte für die PlayStation Portable auf einer Webseite zur Verfügung gestellt werden.

Zur Umsetzung des definierten Zieles soll eine prototypische Anwendung entwickelt werden, die Lernkurse aus hylOs auf der PlayStation Portable abbildet. Die Anwendung soll mit Macromedia Flash realisiert werden, da die PlayStation Portable über ein Flash-Plugin verfügt und Flash sämtliche Anforderungen an die Anwendung erfüllen kann. Eine Realisierung als PlayStationPortable-Anwendung würde den zeitlichen Rahmen dieser Arbeit aufgrund der langen Einarbeitungszeit in die PSP-spezifische Anwendungsentwicklung erheblich überschreiten. Außerdem wäre eine Lizenzierung als PSP-Softwareentwickler notwendig, die ihrerseits mit einem größeren bürokratischen Aufwand verbunden wäre.

6.1 Aufgabendefinition

Um Lernkurse aus hylOs auf der PlayStation Portable abzubilden, sind folgende Schritte für Entwicklung der Anwendung notwendig:

Technische Aufgaben

- Import der XML-Dokumente aus hylOs in Flash
- Extraktion ausgewählter Objekte aus den importierten XML-Dokumenten
- Formatierung der Inhaltsobjekte

Die technischen Aufgaben werden ausführlich im Kapitel 8, S. 74 erläutert.

Designspezifische Aufgaben

- Definition eines PSP-spezifischen Navigationskonzeptes
- grafische Gestaltung der Benutzeroberfläche
- grafische Formatierung der XML-Objekte für die Zielplattform PSP

Zusätzliche Sonderfunktionalitäten (Wissenstest, Hilfefunktion etc.)

Die designspezifischen Aufgaben legen die Rahmenbedingungen für die Benutzeroberfläche und die Funktionalitäten der Anwendung fest. Das Navigationskonzept baut auf den in der Ist-Analyse von hylOs und der PlayStation Portable festgestellten Eigenschaften und Funktionsweisen auf. Die Aspekte der grafischen Gestaltung der Benutzeroberfläche und der ein-

zelen Inhaltsobjekte werden in Kapitel 7, S. 70 detaillierter beschrieben. Unter Sonderfunktionalitäten werden zusätzliche Funktionalitäten, die dem Benutzer zur Orientierung innerhalb der Lernumgebung und zur Kontrolle seines Lernfortschritts dienen, konzipiert.

6.2 Zielgruppendefinition

Da das hylOs-System bisher ausschließlich im Hochschulbereich eingesetzt wird, sind die vorrangige Zielgruppe Studenten aller Semester. Es ist nicht auszuschließen, dass sich die Zielgruppe in naher Zukunft um beispielsweise Schüler der Altersklassen 12-18 Jahre – besonders im Hinblick auf die Inhaltsaufbereitung von Lerneinheiten für mobile und spielorientierte Medien. Auch Berufsgruppen, die mobil unterwegs sind und sich ständig neues Wissen aneignen kommen als neue Zielgruppe in Frage.

Im Rahmen dieser Arbeit wurde keine genaue Zielgruppenanalyse erstellt, jedoch können anhand der bisherigen Zielgruppe und der im folgenden Abschnitt erstellten Anwendungsszenarien mögliche Benutzergruppen festgelegt werden.

6.3 Anwendungsszenarien

Zur Demonstration der möglichen Einsatzgebiete der zu erstellenden Anwendung, sollen im Folgenden drei Anwendungsszenarien mit verschiedenen Zielgruppen aufgezeigt werden.

6.3.1 Szenario 1: Schüler

Tom ist 14 Jahre alt und geht in die 8. Klasse einer Gesamtschule. Tom hat einen langen Weg zur Schule und er braucht ca. eine Stunde mit der Bahn. Auf dem Weg zur Schule liest er oft Comics, schaut aus dem Fenster oder spielt mit seiner PlayStation Portable. Toms Mutter hat auf der Webseite der Schule entdeckt, dass dort ausgewählte PSP-Lernkurse für die 8. Klasse zur Verfügung stehen; u.a. gibt es ein Hausaufgabenmodul für Geschichte und Toms Leistungen in Geschichte sind sehr schlecht. Seine Mutter sagt ihm, dass er PSP-Verbot hat, wenn er sich bis zum Ende des Schuljahres nicht verbessert. Daraufhin schaut sich Tom das Hausaufgabenmodul an und merkt, dass Hausaufgaben auch „Spaß“ machen können.

In diesem Hausaufgabenmodul ist das Grundwissen der 8. Klasse in verschiedene Schulfächer unterteilt. Ein Schulfach, z.B. Geschichte, ist in verschiedene Rubriken unterteilt, in diesem Fall in historische Epochen z.B. „Französische Revolution“ oder „Der Dreißigjähri-

ge Krieg“. Tom wählt nun die Rubrik „Französische Revolution“ aus und gelangt direkt in die 1. Lektion.

Die Lektionen sind kurz und abwechslungsreich durch Texte, Bilder und Aufgaben gestaltet. Eine Lektion besteht aus dem Theorieteil mit einer kurzen Erläuterung anhand von Texten und Bildern und anschließendem Übungsbereich mit jeweils drei zu lösenden Aufgaben. Der Theorieteil enthält bspw. signifikante Begriffe und Jahreszahlen zur Französischen Revolution. Hat Tom sich die Erläuterung durchgelesen, gelangt er anschließend in den Übungsbereich.

Die erste Aufgabe ist eine Multiple-Choice Frage nach einer Jahreszahl, d.h. eine Jahreszahl ist gegeben und drei verschiedene Ereignisse stehen der Jahreszahl gegenüber. Tom soll nun entscheiden, welches die richtige Antwort ist und markiert diese. Hat er Schwierigkeiten, sich für die richtige Antwort zu entscheiden, kann er jederzeit zurück in den Theorieteil wechseln, um sich die Erläuterungen noch einmal anzuschauen.

Hat er nun alle Aufgaben gelöst, kann er das Ergebnis auf dem Server in einer „Highscore“-Liste speichern (optional) und dort gleichzeitig seine Ergebnisse mit denen seiner Mitschüler vergleichen. Tom kann mithilfe des Hausaufgabenmoduls seine Leistungen in Geschichte erheblich verbessern, schreibt in der letzten Geschichtsarbeit eine gute Note und wird versetzt.

6.3.2 Szenario 2: Student

Ein Student des Wirtschaftsingenieurwesens möchte auf der Heimfahrt von der Vorlesung in der Universität die Vorlesung zum Thema „Arbeitsplan Antriebswelle“ wiederholen. Der Studierende verfügt über eine PlayStation Portable und hat entdeckt, dass er sich auf der Webseite seines Professors Lerninhalte zu verschiedenen Themen auf seine PSP herunterladen kann.

Dazu steht ihm auf der Webseite eine Übersicht aller vom Professor publizierten Kurse – darunter auch ein Kurs zum Thema „Arbeitsplan Antriebswelle“ – zur Verfügung. Nun wählt der Student diesen Kurs aus und kann anschließend aus zwei Optionen wählen. Er kann entweder die gesamte Lerneinheit betrachten oder herunterladen oder aber sich einen schnellen Überblick der Lerneinheit über eine sogenannte „Vorschauansicht“ verschaffen.

Die Vorschau besteht aus einem Inhaltsverzeichnis und einer Kurzbeschreibung des gesamten Inhalts sowie einer Kurzbeschreibung der Kapitel. Auch die Lernziele und die benötigte

Lernzeit zum Bearbeiten der kompletten Lerneinheit sind in der Vorschauansicht aufgeführt.¹ Zudem wird ein Link zur gesamten Lerneinheit bereitgestellt, sodass der Student die Lerneinheit direkt aufrufen kann.

In der eigentlichen Lerneinheit wird die Vorlesung ausführlich behandelt. Zur Veranschaulichung der Informationen werden zusätzlich zu den Texten, Bilder und Flashanimationen bereitgestellt. Durch eine semantische Navigation kann der Student zusätzliche Informationen aufrufen und sich somit interessantes Hintergrundwissen aneignen. Am Ende eines jeden Kapitels kann der Studierende optional einen Test zum Prüfen seines Wissens absolvieren. Die Testergebnisse werden in Liste gespeichert und sind beliebig wiederholbar, sodass er seine Testergebnisse jederzeit verbessern kann und seinen Lernfortschritt ablesen kann. Dabei gibt es verschiedene Aufgabentypen, wie etwa Multiple Choice-Fragen, Lückentexte zum Vervollständigen oder Abbildungen mit zu vervollständigenden Beschriftungen. Der Testbereich ist nicht nur sinnvoll zur Wissensprüfung der einzelnen Kapitel, sondern kann dem Studierenden auch sehr nützlich zur Klausurvorbereitung sein und ist deshalb jederzeit über die Hauptnavigation erreichbar.

6.3.3 Szenario 3: Handelsvertreter

Herr K. , 35 Jahre, Handelsvertreter, ist beruflich sehr viel unterwegs und besucht Kunden, um die neuen Produkte seines Unternehmens vorzustellen. Nach einem anstrengenden Arbeitstag entspannt er sich gern mit seiner PlayStation Portable.

Herr K. muss immer auf dem laufenden sein und sich ständig mit den neuesten Produkten seines Unternehmens vertraut machen, da er alle Fragen seiner Kunden zu deren vollster Zufriedenheit beantworten möchte. Nun hat K. entdeckt, dass seine Arbeitgeber die Beschreibung der neuesten Produkte in Form eines Lernmoduls pro Produkt für die unterschiedlichsten Plattformen aufbereitet haben, so auch für die PlayStation Portable.

Er lädt sich ein Modul zu einem Produkt herunter. Das Lernmodul ist in drei Teilbereiche untergliedert. Eine allgemeine Produktbeschreibung, eine anschauliche 3D-Videoanimation und eine detaillierte Funktionsbeschreibung.

In der allgemeinen Produktbeschreibung kann sich K. eine Übersicht zu Produktfunktionen, Einsatzbereichen und besonderen Merkmalen des neuen Produktes verschaffen. Dieser Bereich wird ihm anhand von Texten und Bildern präsentiert. Die 3D-Videoanimation

¹ Kurzbeschreibung zur gesamten Lerneinheit sowie für Unterkapitel, Lernziel und die Bearbeitungszeit sind als optionale Metadaten bereits in hylOs definiert.

zeigt K. das neue Produkt aus allen Perspektiven und demonstriert die einzelnen Funktionen des neuen Produktes. Im dritten Bereich – dem Funktionsbereich hat K. Die Gelegenheit sich detailliert mit den Funktionen des neuen Produktes vertraut zu machen. Dieser Bereich enthält Texte und Bilder der einzelnen Funktionen und ist in verschiedene Lektionen unterteilt. Der Funktionsbereich schließt zusätzlich ein Prüfungsmodul ein, mit welchem K. sein Wissen zu dem neuen Produkt testen kann.

Das Prüfungsmodul testet das Wissen des gesamten Lernmoduls und kann deshalb auch von jeder Navigationsebene des Lernmoduls angewählt werden. Das Wissen über das neue Produkt wird durch verschiedene Aufgabentypen abgefragt, beispielsweise über Multiple-Choice-Fragen oder über Text-Bild-Zuordnungen. Bei der Text-Bild-Zuordnung wird das Produktbild angezeigt. Darauf sind spezifische Bereiche markiert und nummeriert. Neben dem Bild ist eine Liste mit Bezeichnungen dieser Bereiche aufgeführt. K. soll nun in einem Textfeld die korrekten Nummern den Bezeichnungen zuweisen.

Das Prüfungsmodul kann K. immer wieder zur Hand nehmen und zur Wiederholung und Auffrischung seines Wissen über das Produkt nutzen.

6.4 Erstellung und Abruf von Lerninhalten

Im Folgenden ist der typische Ablauf aus Sicht des Autors (bzw. des Lehrenden) und des Lernenden aufgeführt, um die Besonderheit der Offline-Nutzung zu erläutern.

Für den Autor beziehungsweise Lehrenden ändert sich an der bisherigen Arbeitsweise nichts. Er gibt die Lerninhalte über die Autorenumgebung in hylOs ein. Der Inhalt wird entsprechend für die PlayStation Portable generiert, sobald der Lernende diese auf der Webseite abrufen.

Dem Lernenden stehen zwei Möglichkeiten zur Nutzung des Lernangebotes zur Verfügung. In der sind die beiden Optionen abgebildet.

Um die Lerneinheit online zu nutzen, ruft der Benutzer – nach der Aktivierung der WLAN-Funktion seiner PlayStation Portable – die Webseite mit dem Lernangebot auf, wählt einen Kurs aus und bearbeitet diesen. Dazu ist es jedoch notwendig, dass die Verbindung zum Internet bestehen bleiben. Möchte der Benutzer den Kurs offline nutzen, schließt er die Memory-Card aus der PlayStation Portable an seinen PC an, sucht über seinen PC-Webbrowser die Webseite mit dem Lernangebot auf, wählt den entsprechenden Kurs aus, lädt diesen Kurs auf die Memory-Card herunter und legt diese wieder in die PlayStation Portable ein. Anschließend ruft er im PSP-Webbrowser durch Eingabe von

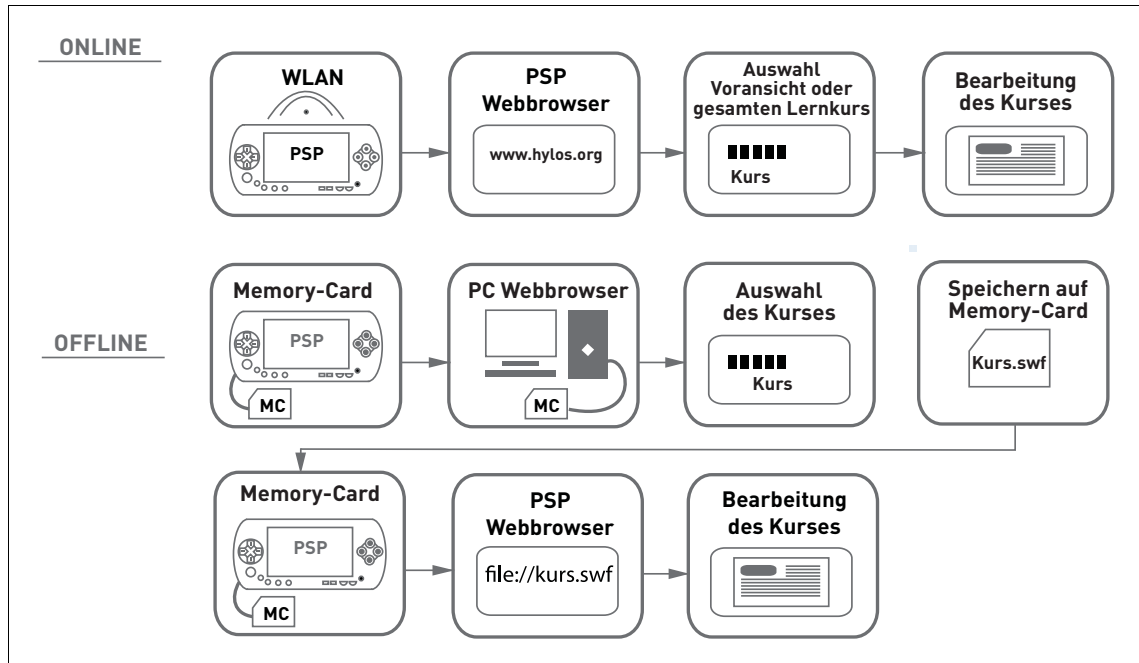


Abbildung 7: Ablauf zur Online- bzw. Offline-Bearbeitung des Kurses

beispielsweise `file://Pfad/kurs.swf` den Kurs auf und kann diesen dann auf der PSP offline bearbeiten.

6.5 Navigationskonzept der Lerneinheit

In Kapitel 6.3.2 wurde ein Anwendungsszenario anhand der Zielgruppe Studenten vorgestellt. Da hylOs bisher ausschließlich Hochschulinhalt für Studenten bereitstellt, soll das Navigationskonzept der Lerneinheit aufbauend auf diesem Anwendungsfall ausgearbeitet werden.

Bevor die Navigationsmöglichkeiten festgelegt werden sollen, ist es sinnvoll den gesamten Ablauf innerhalb der Anwendung zu verdeutlichen. Dazu soll vorerst die Übersicht in Abbildung 8 dienen. Ausgehend von dieser Abbildung leiten sich folgende Schritte ab:



Abbildung 8: Mögliche Präsentationsansichten für Kursinhalte

Kursübersicht auf der Webseite

Der Benutzer bekommt nach Aufruf der Webseite eine Übersicht aller publizierten Lernkurse. Dort ist jeweils eine Kurzbeschreibung der einzelnen Kurse sowie ein Link zur Vorschauansicht bzw. zum gesamten Lernkurs.

Vorschauansicht einer Lerneinheit

Die Vorschauansicht enthält eine Kurzbeschreibung des gesamten Kurses und das Inhaltsverzeichnis. Außerdem sind allgemeine Informationen wie die Zielgruppe, Lernziele und benötigte Bearbeitungszeit aufgeführt. Das Inhaltsverzeichnis enthält sämtliche Kapitel des Kurses und eine Kurzbeschreibung der Kapitel auf der ersten Ebene (Hauptkapitel). Die Aufgabe der Voransicht ist es dem Benutzer einen Einblick in die Lerneinheit zu geben.

Falls er beispielsweise den Kurs offline bearbeiten möchte, kann er anhand der Voransicht sicherstellen, dass die Lerneinheit alle für ihn interessanten Informationen enthält.¹

Ansicht einer Lerneinheit

Die Lerneinheit selbst enthält alle Inhalte unterteilt nach Kapiteln, die diesem Lernkurs zugeordnet sind. In der Lerneinheit soll sich der Benutzer anhand zweier Navigationen durch den Kurs bewegen können – hierarchisch und anhand der semantischen Beziehungen (siehe Kapitel 4.1.1, S. 34). Dieses Konzept soll nun näher beschrieben werden.

6.5.2 Navigationsmöglichkeiten

Aufgrund der Tatsache, dass hylOs drei Arten von XML-Dokumenten generiert, die spezielle Navigationsarten abbilden², sollen diese dem Benutzer innerhalb der Anwendung zur Verfügung stehen. Die Lernpfade in der Lernpfad-Navigation-XML gleichen strukturell der Baum-Navigation-XML und werden nicht gesondert betrachtet. Das Konzept für hierarchisch strukturierte Daten und Wissensnetze wird im Folgenden beschrieben.

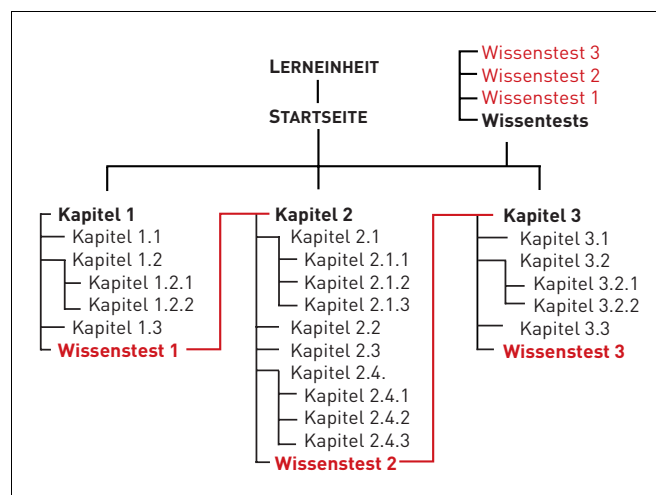


Abbildung 9: Beispielstruktur einer Lerneinheit

¹ Zukünftig könnte sich die Vorschauansicht sehr gut dazu eignen, um kostenpflichtige Lernkurse einzusehen und dem Benutzer so vor dem Kauf einen Einblick in die Inhalte zu gewähren.

² Baum-Navigation-XML oder hierarchische Navigation genannt, Lernpfad-Navigation-XML, Semantische-Relationen-XML, siehe Kapitel 4.1.2, S. 37

Hierarchischer Aufbau einer Lerneinheit

Die Abbildung 9 stellt die hierarchische Struktur eines Beispiel-Lernkurses dar. Die Startseite der Lerneinheit enthält zum Beispiel einen Überblick über die im Kurs enthaltenen Kapitel. Weitere Informationen sind: Zielgruppe der Lerneinheit, Lernziele und benötigte Zeit zum Bearbeiten des gesamten Kurses. Der Beispiel-Lernkurs enthält insgesamt drei Kapitel. Die jeweilige Startseite eines Kapitels (bspw. Kapitel 1) enthält wiederum Informationen zum Inhalt des Kapitels, Lernziel und Dauer der Bearbeitung. In den anschließenden Unterkapiteln sind die Informationen in Form von Text, Bildern, Links, Tabellen etc. aufbereitet. Das Ende eines jeden Kapitels bildet ein optionaler Wissenstest zu dem zuvor bearbeiteten Material. Nach Bearbeitung bzw. Überspringen des Wissenstests folgt das nächste Kapitel. Damit der Benutzer die Wissenstests unabhängig von der Bearbeitung der Kurse absolvieren kann, ist es möglich, über ein Symbol auf der Benutzeroberfläche jederzeit auf den Punkt Wissenstests mit sämtlichen Tests der Lerneinheit zuzugreifen.

Hierarchische Navigation

Die hierarchische Navigation orientiert sich an der Struktur, wie sie der Autor in hylOs erstellt hat. Abbildung 10 zeigt die hierarchische Navigation der Kapitel mit den entsprechenden Seitenzahlen. Anhand der hierarchischen Navigation kann der Benutzer nacheinander durch die einzelnen Seiten navigieren. Zusätzlich hat er die Möglichkeit, von jeder beliebigen Seite aus zum nächsten bzw. vorherigen Kapitel zu navigieren. Die einzelnen Funktionen der hierarchischen Navigation sind demzufolge :

Seite	Kapitel
1	Kapitel 1
2	Kapitel 1.1
3	Kapitel 1.1.1
4	Kapitel 1.1.2
5	Kapitel 1.2
6	Wissenstest 1
7	Kapitel 2
8	Kapitel 2.1
9	Kapitel 2.2
10	Kapitel 2.2.1
11	Kapitel 2.2.2
12	Wissenstest 2
13	Kapitel 3
...	...

zunächst zur nächsten/vorherigen Seite blättern
 • zur nächsten/vorherigen Kapitel blättern

Die hierarchische Navigation wird in der Anwendung durch eine Pfeilnavigation abgebildet, näheres dazu folgt in Kapitel 7.1, S. 70.

Abbildung 10: Beispielstruktur einer Lerneinheit

Semantische Bezüge einer Lerneinheit

Nachdem in der Ist-Analyse von hylOs (siehe Kapitel 4.1.1, S. 34) das Metadatenmodul beschrieben wurde, wird nun alternativ zur klassischen hierarchischen Navigation eine weitere Navigationsart vorgestellt. Dem Benutzer soll ermöglicht werden anhand der semantischen Beziehungen eines Lernobjektes (eines Kapitels) durch die Lerneinheit zu navigieren.

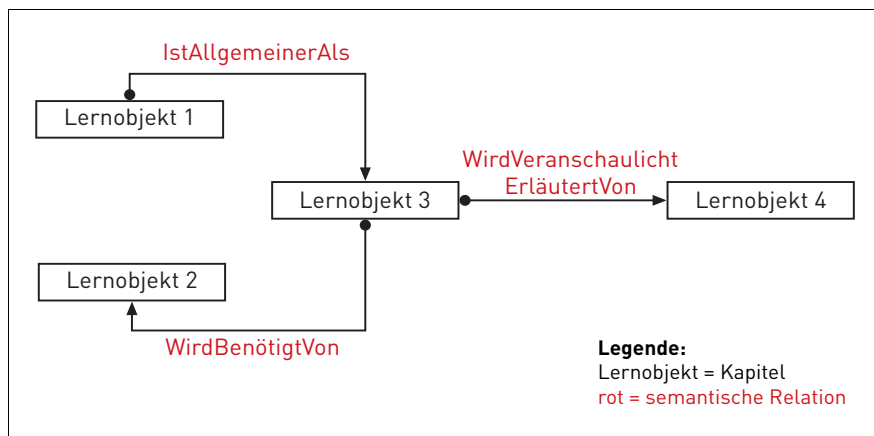


Abbildung 11: Beispielhafte semantische Struktur einer Lerneinheit

Dazu sollen die Daten der Kategorie Beziehungen in der Anwendung genutzt werden.

Abbildung 11 zeigt beispielhaft die semantischen Beziehungen eines Lernobjektes zu anderen Lernobjekten. Dabei sind die Beziehungen des Lernobjekt 3 dargestellt. Lernobjekt 3 könnte bspw. das Kapitel 3 aus Abbildung 9 (S. 63) und das Lernobjekt 4 könnte bspw. Kapitel 1.2 sein. Diese beiden Kapitel stehen in der hierarchischen Struktur in keinerlei Beziehung, außer das sie vielleicht ein Glied der gesamten Lerneinheit darstellen. Durch das Zuweisen einer Beziehung zwischen den beiden Lernobjekten, werden die beiden Kapitel direkt zueinander in Beziehung gesetzt. Das heißt für das Beispiel, Kapitel 3 wird durch Kapitel 1.2 veranschaulicht und diese Information soll dem Benutzer zusätzlich im Lernprozess als Navigationsmöglichkeit zwischen unterschiedlichen Lernobjekten zur Verfügung stehen.

Semantische Navigation

In [EHL+06]¹ wurden semantische Relationen in Kategorien zusammengefasst, um hylOs-Lernkurs-Inhalte für den iPod² aufzubereiten. Eine ähnliche Kategorisierung und damit Zusammenfassung der achtzehn vorhandenen Relationen soll ebenfalls für die semantische Navigation erfolgen, allerdings werden die Kategorien an die speziellen Erfordernisse dieser Arbeit angepasst und somit wie folgt eingeteilt:

- Allgemeinere Informationen
- Detailliertere Informationen
- Alternativen
- Beispiele

Diesen Kategorien werden die folgenden Relationen – aus dem Metadatenmodul der Autorenumgebung in hylOs – zugeordnet:

- Allgemeinere Informationen – Relationen:
Benötigt, ist detaillierter als, ist spezieller als, basiert auf, ist Bestandteil von
- Detailliertere Informationen – Relationen:
ist allgemeiner als, ist weniger detailliert als, wird benötigt von, ist Grundlage, veranschaulicht/erläutert
- Alternativen bzw. Ähnliches
ist Alternative zu, wird referenziert von, referenziert, hat Format, ist Format von
- Beispiele
wird veranschaulicht/erläutert von

Die Relationen 'hat Bestandteil' und 'ist Version von' werden nicht mit in die Kategorien aufgenommen, da sie thematisch nicht einer einzigen Kategorie zugewiesen werden können.

Aus der Ist-Analyse der PlayStation Portable (siehe Kapitel 4.2.2, S. 47) geht hervor, dass es möglich ist, die Richtungstasten der PlayStation Portable in Kombination mit der Quadrat-taste zu nutzen. Diese Kombination soll für die semantische Navigation genutzt werden und zwar in folgender Weise (siehe Abbildung 12): die vier Richtungstasten sollen die vier

¹ Diese Publikation stellt Mobile Learning Szenarien mit hylOs-Lerninhalten u.a. für den iPod vor.

² portabler MP3-Player der Firma Apple

Kategorien repräsentieren, vorausgesetzt, dass das Lernobjekt über die entsprechenden Relationen verfügt.

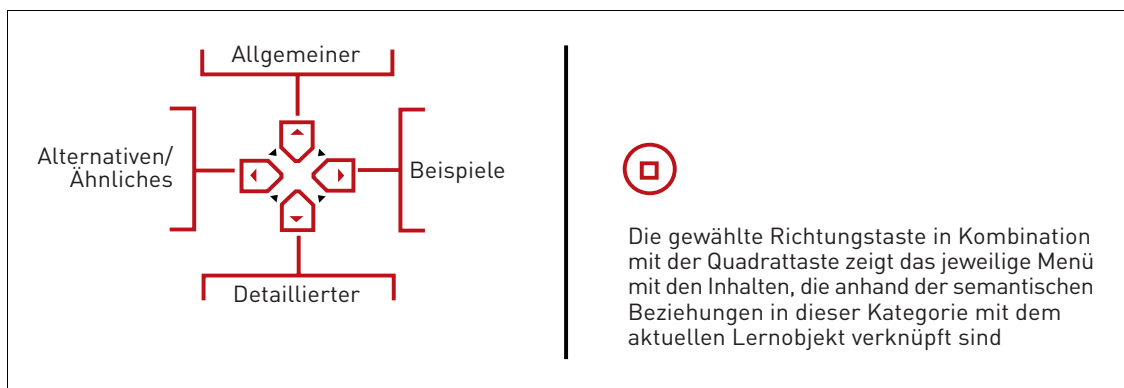


Abbildung 12: Beispielhafte semantische Struktur einer Lerneinheit

Sind semantische Relationen angegeben, wird das durch Einblenden einer zusätzlichen Navigationsleiste signalisiert. Möchte der Benutzer beispielsweise wissen, in welchen Gesamtkontext sich ein Kapitel eingliedert, dann drückt er die Quadrattaste in Verbindung mit der Richtungstaste „oben“ und erhält daraufhin eine Linkübersicht mit Verweisen auf allgemeinere Informationen und Oberkategorien zu dem aktuellen Kapitel.

Die Tastenbelegung wurde so gewählt, dass die Navigation angelehnt ist an die erprobten Muster typisch menschlicher¹ Assoziationen der Begriffe „oben“ mit Allgemein, „unten“ mit detailliert“. „Gleichstufige“ Informationen stellen oft etwas Ähnliches (Alternativen) oder Weiterführendes (Beispiele) dar und so werden die Richtungstasten „links“ (Alternativen/Ähnliches) und „rechts“ entsprechend belegt. Auf diese Weise gestaltet sich die Bedienung so intuitiv wie möglich, ohne den Nutzer zum Erlernen eines schwer zugänglichem Navigationsmuster zu zwingen.

6.6 Besondere Funktionalitäten der Anwendung

Neben der Bearbeitung der Lerneinheit stehen dem Benutzer noch andere Funktionalitäten innerhalb der Anwendung zur Verfügung. Diese werden unter den folgenden Punkten erläutert.

¹ abendländischer – da Usability kulturspezifisch orientiert ist, werden sich unter Umständen in anderen Kulturen andere Muster bewährt haben

6.6.1 Wissenstest

Zur Kontrolle des Lernerfolges wird nach jedem Hauptkapitel ein Wissenstest mit Inhalten aus dem vorangehenden Kapitel angeboten. Der Wissenstest ist optional und kann vom Anwender übersprungen werden.

Ausgehend von den Ergebnissen der Studie der Fernuniversität Hagen (siehe Kapitel 2.4, S. 16) sind innerhalb des Wissenstests verschiedene Aufgabentypen denkbar, bspw. :

- Multiple Choice Fragen
- Richtig oder Falsch Fragen
- Lückentexte
- Zuordnung der Beschriftungen von Bildern

Ein solcher Wissenstest muss jedoch bereits in der Autorenumgebung hylOs durch die Autoren vorbereitet werden. Aktuell verfügt hylOs noch nicht über eine Eingabemaske für die Autoren, um Wissenstests für die Lernenden zur Verfügung zu stellen. Angesichts dieser Tatsache wird diese Funktionalität in der prototypischen Anwendung noch nicht umgesetzt und auch im weiteren Verlauf dieser Arbeit nicht weiter ausgearbeitet.

6.6.2 Hilfsfunktionen

Zur besseren Übersicht über die gesamte Lerneinheit und zur Orientierung innerhalb der Lerneinheit werden folgende Hilfsfunktionen implementiert:

- **Anzeige des Inhaltsverzeichnis**

Beim Anzeigen des Inhaltsverzeichnis werden die Kapitel mit allen Unterkapiteln jeweils als Verweis angezeigt. Innerhalb des Inhaltsverzeichnis hat der Nutzer die Möglichkeit, ein Kapitel bzw. Unterkapitel anzuwählen, damit wechselt er zur angewählten Seite in die Ziellernumgebung (das Fenster mit dem Inhaltsverzeichnis schließt sich) oder er kann die Seite wieder schließen und gelangt auf diese Weise zurück zur Ausgangslernumgebung.

- **Anzeige des Glossars**

Das Glossar dient dem Benutzer zum Nachschlagen von Fachbegriffen. Es enthält alle vom Autor definierten Glossarbegriffe für die gesamte Lerneinheit. Im Text in der Lernumgebung sind diese Begriffe besonders hervorgehoben. Das gesamte Glossar der Lerneinheit wird dargestellt. Im oberen Bereich wird eine alphabetische Liste zu sehen sein

und der Nutzer kann durch Auswahl eines Buchstaben sämtliche Begriffe nachschlagen, die unter diesem Buchstaben vorhanden sind. Dazu wird der Begriff grafisch hervorgehoben und es folgt eine kurze Begriffserklärung.

- **Anzeige des Literaturverzeichnisses**

Das Literaturverzeichnis dient dem Benutzer zum Nachschlagen von benutzter Literatur innerhalb des Lernkurses. Das Literaturverzeichnis soll ähnlich dem Glossar mit einer alphabetischen Liste im oberen Bereich angezeigt und die einzelnen Einträge durch das Auswählen eines Buchstabens angeschaut werden können. Die Einträge werden nach dem Nachnamen des Autors sortiert.

- **Anzeige von Bedienungshinweisen**

Innerhalb der Bedienungshinweise wird eine Liste erstellt, die dem Benutzer die Bedienung der Benutzeroberfläche erklärt.

- **Ein- und Ausschalten von Tooltips**

Tooltips sind kurze Informationen die eingeblendet werden, wenn der Benutzer mit dem Mauszeiger über einen Button fährt. Dabei werden in der Anwendung sämtliche Buttons mit einem Tooltip versehen, um eine Hilfestellung zur Funktion der Buttons zu geben, bevor diese benutzt werden. Da der Bildschirm der PlayStation Portable relativ klein ist, kann es für den Benutzer nützlich sein, dass er die Tooltips nach Belieben ein- und ausschalten kann.

6.6.3 Allgemeine Statusinformationen

Lernobjektinformationen dienen dem Lernenden einerseits zur Orientierung innerhalb des gesamten Lernkurses und andererseits zur Einordnung seines aktuellen Fortschritts im Lernprozess. Dazu werden die Unterfunktionen zur Anzeige des Hauptkapitels, die Bezeichnung des Kapitels und die Seitenzahl permanent auf der Benutzeroberfläche sichtbar sein.

Kapitel 7 Entwurf der Benutzeroberfläche

Im Folgenden werden auf Basis der Konzeption sämtliche Eigenschaften der Elemente für die Anwendung festgelegt, die später implementiert werden sollen. Dabei wird das Aussehen der Benutzeroberfläche mit den Funktionen im Detail erläutert. Dieser Teil der Arbeit bildet die Grundlage für die spätere Implementierung der Anwendung.

Die Basis der Anwendung bildet die Lernumgebung, bestehend aus der Benutzeroberfläche innerhalb der PlayStation Portable. Auf dieser interagiert der Benutzer mit dem Inhalt des Lernobjektes. Aufgrund der in der Ist-Analyse beschriebenen Tatsache, dass in der Anwendung nur Flashvideos dynamisch geladen werden können, die aber die wenigsten Autoren tatsächlich bereitstellen, konzentriert sich die Arbeit bei den Inhaltsobjekten ausschließlich auf Texte und Bilder. Die Inhaltsobjekte müssen entsprechend grafisch aufbereitet werden. Diese Anforderung wird dynamisch mithilfe von CSS umgesetzt.

7.1 Layout der Lernumgebung

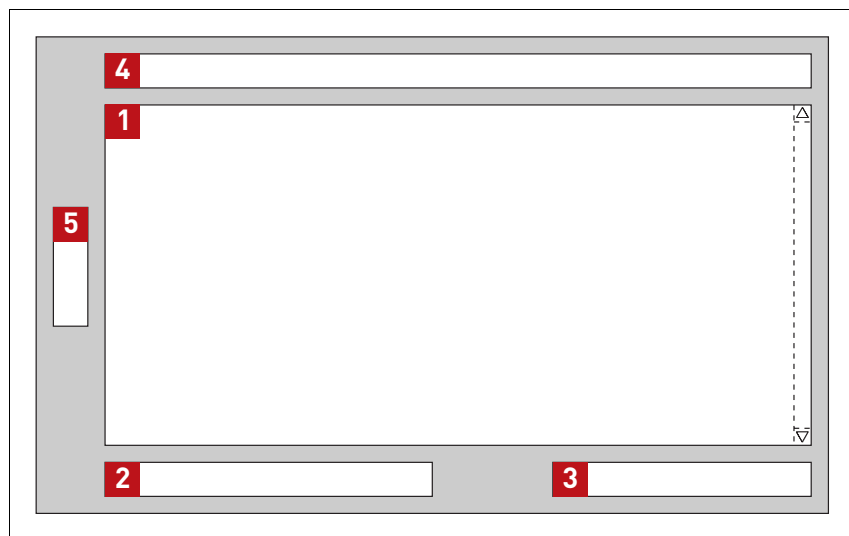


Abbildung 13: Layout der Lernumgebung

Die Lernumgebung ist der wichtigste Bereich der Anwendung, es ist der Bereich mit dem der Anwender hauptsächlich arbeiten wird. Dieser Bereich ist der Ausgangspunkt für alle anderen Funktionalitäten der Anwendung.

Das grafische Layout der Lernumgebung wird in Abbildung 13 gezeigt. Wie die Grafik verdeutlicht ist die Lernumgebung in 5 Bereiche unterteilt:

1. Inhaltsbereich
2. Hilfreiche Funktionen
3. Hierarchische Navigation
4. Informationen zum Lernobjekt
5. Semantische Navigation

Im Inhaltsbereich (1.) werden die Inhalte der jeweiligen Lernobjekte angezeigt. Ist der Inhalt länger als der sichtbare Bereich, wird ein Scrollbalken angezeigt, mithilfe dessen sich der Nutzer den gesamten Inhalt durch hoch- und runterscrollen anschauen kann.

Unter Hilfreiche Funktionen (2.) findet der Benutzer fünf Buttons. Der erste Button dient zum Anzeigen des Inhaltsverzeichnis des gesamten Lernkurses, der zweite zum Anzeigen des Glossars, der dritte zum Anzeigen des Literaturverzeichnisses, der vierte dient zum Anzeigen der FAQ-Liste und über den fünften Button können die Tooltips ein- oder ausgeschaltet werden.

Mittels der Hierarchischen Funktion (3.) navigiert der Benutzer durch die Anwendung. Das Layout der Buttons ist in Abbildung 14 dargestellt. Die vier Buttons haben folgende Funktionen:

- vorheriges Kapitel (1)
- vorherige Seite (2)
- nächste Seite (3)
- nächstes Kapitel (4)

Im oberen Bereich unter Informationen zum Lernobjekt (4.) werden jeweils das Kapitel, die Überschrift des aktuellen Lernobjektes und die Seitenzahl angezeigt. Diese Information sollen dem Benutzer zur Orientierung innerhalb des Lernkurses dienen.

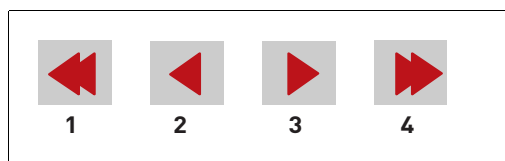


Abbildung 14: Layout der Lernumgebung

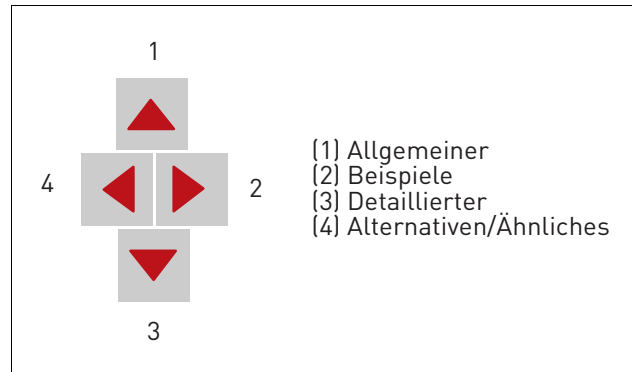


Abbildung 15: Buttons Semantik

Der Bereich der Semantischen Navigation (5.) wird nur bei Vorhandensein von semantischen Beziehungen des Lernobjektes eingeblendet. In diesem Bereich werden sich vier Buttons befinden (siehe Abbildung 15), ähnlich gestaltet wie die Richtungstasten der PlayStation Portable. Es wird nur der Richtungsbutton angezeigt, in dessen Kategorie zusätzliche Informationen zur Verfügung stehen. Sind bspw. nur zusätzliche Informationen in der Kategorie Allgemein vorhanden, wird nur der Button (1) angezeigt.

7.2 Sonderlayout für Hilfsfunktionen

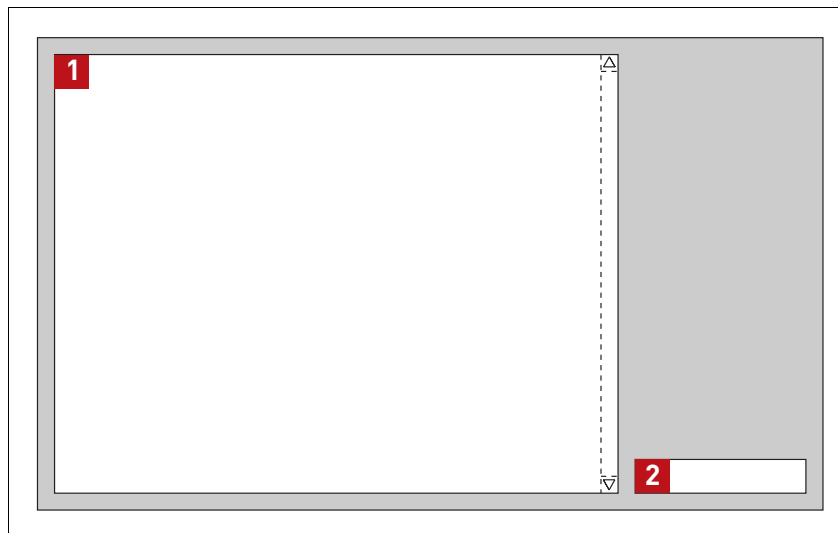


Abbildung 16: Layout Hilfsfunktionen

Die Hilfsfunktionen Inhaltsverzeichnis, Glossar, Literaturverzeichnis und FAQ-Liste werden aufgrund ihrer Inhalte in einem besonderen Layout dargestellt. Abbildung 16 zeigt das

Layout für die Hilfefunktionen. Dieses Layout legt sich wie eine zweite Ebene über die Lernumgebung und füllt den gesamten Bildschirm aus. Der mit (1) markierte Bereich stellt jeweils die Inhalte der Verzeichnisse bzw. der FAQ-Liste dar und bei längerem Inhalt wird auch hier eine Scrollleiste erscheinen. Bereich (2) markiert einen Button mit dem der Benutzer dieses Fenster schließen kann.

Kapitel 8 Implementierung und Test

Dieses Kapitel dokumentiert die Umsetzung der Anwendung mit den im Konzept und im Entwurf festgelegten Funktionalitäten. Allerdings konnten aufgrund des begrenzten Rahmens dieser Arbeit nicht sämtliche Funktionalitäten umgesetzt werden. Die Realisierung der Anwendung erfolgte mit Macromedia Flash 8 [ADO] und parallel mit dem SEPY ActionScript-Editor [SEP] zur Programmierung der ActionScript-Dateien. Nach diesem ersten Realisierungsschritt haben sich beim ersten Testdurchlauf Probleme bei der Darstellung der Inhaltsobjekte abgezeichnet. Daraufhin wurde ein zweiter Lösungsansatz entwickelt, bei welchem anhand eines XSLT-Stylesheets die Probleme weitestgehend behoben werden konnten. Beide Testdurchläufe werden jeweils innerhalb der Beschreibung der Lösungsansätze dokumentiert.

Bevor die einzelnen Schritte zur Realisierung erläutert werden, sollen kurz die wesentliche Grundkonzepte von Macromedia Flash vorgestellt werden, die innerhalb der Applikation Verwendung finden. Darauf folgt die Beschreibung der Realisierung der Benutzeroberfläche.

Die Grundkonzepte von Macromedia Flash

Macromedia Flash ist ein Authoring-Tool zum Erstellen von Animationen, skalierbaren Vektorgrafiken, Navigationselementen und interaktiven On- und Offline-Anwendungen mit integrierten Bild, Sound- und Videodateien.

Die Oberfläche des Programms beinhaltet eine „Bühne“, diverse Werkzeugpaletten und eine Zeitleiste. Auf der *Bühne* werden Symbole abgelegt. Symbole bilden die Grundlage eines Flash-Films und können beispielsweise Vektorgrafiken, *Buttons* oder *Movieclips* sein. Flash nutzt eine *Bibliothek* zum Ablegen der Symbole, um anschließend die Instanzen der Symbole innerhalb des Films zu verwenden. Somit sind Symbole wiederverwendbare Objekte. Das Konzept der Wiederverwendung und die Möglichkeit der Benutzung von Vektorgrafiken sorgt für eine relative kleine Dateigröße der Flash-Dateien.

In der Zeitleiste wird der zeitliche Ablauf des Films gesteuert. Diese ist in einzelne Frames aufgeteilt, wobei jeder Frame ein Einzelbild des Films darstellt. Das Einzelbild kann unterschiedliche Symbole enthalten. Die Zeitleiste beinhaltet neben den Frames zur zeitlichen Steuerung des Films zusätzlich ein Ebenenkonzept. Das erleichtert die Arbeit mit graphischen Objekten, da diese auf übereinanderliegenden Ebenen angeordnet und unabhängig voneinander gesteuert werden können.

Die bereits erwähnten *Movieclips* sind ebenfalls kleine Filmsequenzen, die sich in einem Frame im Hauptfilm befinden. Diese besitzen eine eigene Zeitleiste und können somit unabhängig von der Hauptzeitleiste abgespielt werden. Außerdem haben *Movieclips* die Eigenschaft, dass sie auf Ereignisse reagieren und untergeordnete Symbole jeglicher Art verwalten, steuern und überwachen können.

Mit Flash können zeitleistenbasierte Filme als Präsentationsfilme ohne jegliche Interaktion seitens des Nutzers durch die verschiedenen Werkzeuge und Funktionen der Flash-Oberfläche erstellt werden. Das ist sehr nützlich für Anwender ohne Programmierkenntnisse, jedoch wird so bei weitem nicht die volle Bandbreite von Flash ausgenutzt. Die eigentliche Leistungsfähigkeit von Flash besteht in der Entwicklung interaktiver Anwendungen, die über die ereignisgesteuerte Skriptsprache ActionScript realisiert wird.

8.1 Erstellung der Benutzeroberfläche

Bevor der Anwendung die Funktionalitäten hinzugefügt werden können, muss die Benutzeroberfläche erstellt werden. Dazu wurde im Kapitel 7.1, S. 70 das Layout der Benutzeroberfläche festgelegt. Die Benutzeroberfläche wird bewusst einfach und übersichtlich gestaltet, damit der Lernende nicht von den Inhalten abgelenkt wird. Die gesamte Oberfläche

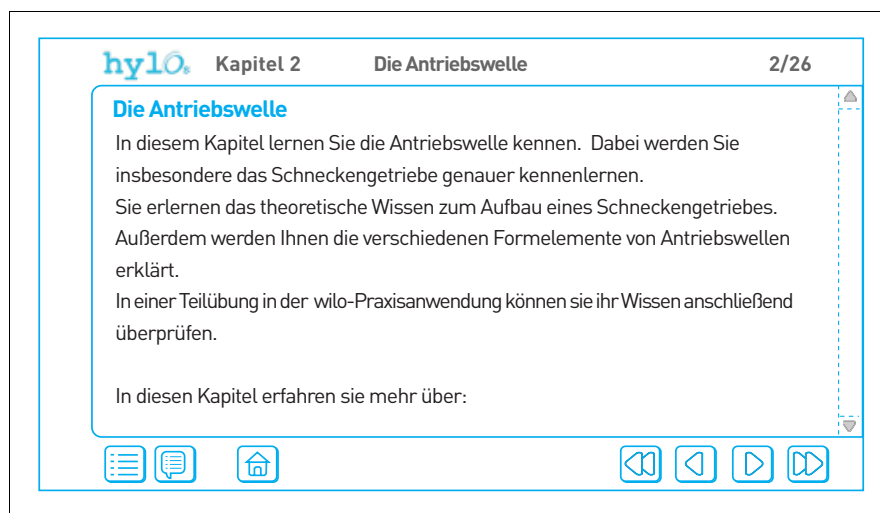


Abbildung 17: Benutzeroberfläche mit Beispielinhalt

wird in Macromedia Flash anhand der Zeichenfunktionen umgesetzt. Alle graphischen Elemente der Oberfläche bestehen aus Vektorgrafiken, die auf unterschiedlichen Ebenen angelegt sind und im späteren Verlauf der Umsetzung anhand von ActionScript mit Funktionalitäten

litäten belegt werden. Abbildung 17 zeigt die erstellte Benutzeroberfläche mit Beispielinhalt. Die einzelnen Vektorgrafiken werden, um über ActionScript adressierbar zu sein, in Movieclips umgewandelt. Die oberste Ebene enthält sämtliche Skripte der Anwendung in Form von Verweisen auf die ActionScript-Dateien die über den Befehl `include` eingebunden werden.

8.2 Technische Inhaltsaufbereitung

Während der Umsetzung entwickelte sich ein zweiter Lösungsansatz, der abweichend von den in Kapitel 6, S.56 aufgelisteten Aufgaben, einen zusätzlichen Arbeitsschritt zur Umsetzung der Anforderungen erforderte. Unabhängig von den Lösungsansätzen ist der erste Schritt der Realisierung die Erstellung der Benutzeroberfläche.

Die weiteren Arbeitsschritte des ersten Lösungsansatzes sind (siehe Kapitel 6, S.56):

- Import folgender XML-Dokumente aus hylOs in Flash (content - für den Inhaltsbereich, navtree - für die hierarchische Navigation (siehe Kapitel 4.1.2, S. 37))
- Extraktion ausgewählter Objekte aus den importierten XML-Dokumenten
- Formatierung ausgewählter Elemente

Die Arbeitsschritte nach der Erstellung des zweiten Lösungsansatzes sind:

- Erstellung eines XSLT-Stylesheets für die Transformation der XML-Dokumente vom Dokumenten-Typ content aus hylOs.
- Import der notwendigen XML-Dokumente aus hylOs in Flash (content, navtree)
- Extraktion ausgewählter Objekte aus den importierten XML-Dokumenten.
- Formatierung ausgewählter Elemente

Im Folgenden wird die Umsetzung beider Lösungsansätze auszugsweise erläutert.

8.3 Lösungsansatz 1

Bei diesem Ansatz werden die XML-Dokumente ohne vorherige Transformation mithilfe von ActionScript in Flash eingelesen und anschließend verarbeitet und formatiert. Im Folgenden werden die wichtigsten Arbeitsschritte, die für die Erstellung der Anwendung und damit für die Anzeige der Inhalte sowie für die hierarchische Navigation erforderlich sind, näher beschrieben.

8.3.1 Import der XML-Daten

In Kapitel 4.1.2, S. 37 wurden die verschiedenen XML-Dokumenttypen erläutert. Innerhalb der Anwendung werden lediglich das Inhalt-XML- und das Baum-Navigation-XML-Dokument ausgelesen.

Die Inhalt-XML-Datei enthält die Metadaten aus den Kategorien Allgemein, Pädagogisch, Rechte, Lebenszyklus und den Inhaltsbereich `<hylos:content>` mit sämtlichen Inhaltselementen des Lernobjektes. Für die Anwendung werden lediglich die Inhaltselemente benötigt, nicht jedoch die Metadaten und somit werden alle Elemente innerhalb des Elementes `<hylos:content>` ausgelesen.

Die Baum-Navigation-XML-Datei enthält alle für die hierarchische Navigation notwendigen Informationen. Darunter fallen alle Unterelemente des Lernobjektes, wobei diese Unterelemente jeweils dieselben Metadatenkategorien wie die Inhalt-XML-Datei enthalten. Diese Metadaten werden wie bei der Inhalt-XML-Datei ebenfalls nicht berücksichtigt.

Laden des Inhalt-XML-Dokuments

Flash stellt eine XML-Schnittstelle zum Laden und Verarbeiten von XML-Daten bereit. Die Lerninhalte werden dynamisch aus hylOs in die Anwendung geladen.

Jedes Mal, wenn der Benutzer eine neue Seite innerhalb des Lernkurses aufruft, wird die in Listing 2 abgebildete Funktion `loadContentXML()`, die von ActionScript bereitgestellt wird, aufgerufen, um ein neues Lernobjekt zu laden.

Um in Flash XML-Dokumente zu laden, muss zunächst ein XML-Objekt durch `var XMLcontent = new XML()` erzeugt werden. Im Listing 2, Zeile 2 wird die URL für das aktuelle Lernobjekt zusammengesetzt. Die Zeile 3 dient zum Löschen aller Textknoten, die ausschließlich Leerräume enthalten. Durch die Funktion `load()` wird der Inhalt eines lokalisierten XML-Dokuments in das XML-Objekt `XMLcontent` geladen. Anschließend wird die Methode `onLoad()` aufgerufen. Nur wenn das XML-Dokument vollständig geladen wurde, wird über `XMLcontent.firstChild.lastChild` auf das Element `<hylos:content>`, also auf den Inhalt zugegriffen (Zeile 7) und dieser im XML-Objekt gespeichert. Anschließend werden die einzelnen Inhaltselemente aus dem XML-Objekt ausgelesen.

```
1. function loadContentXML (XMLcontent : XML){
2.     XMLcontentPath = host+contentTypeContent+learningObject+".xml";
3.     XMLcontent.ignoreWhite = true;
4.     XMLcontent.load (XMLcontentPath);
5.     XMLcontent.onLoad = function (success){
6.         if (success){
7.             this = XMLcontent.firstChild.lastChild;
8.             readContentData (this, 0);
9.         }
10.    };
11. }
```

Listing 2: Laden der Inhalte eines Lernobjektes aus der Inhalt-XML-Datei

Beim Auslesen des XML-Dokumentes traten Probleme in Flash bei der Verarbeitung von Namespaces auf. Diese werden erst ab Flash Player 8 unterstützt. Die Element- und Attributnamen im XML-Dokument sind durch Namespaces eindeutig gekennzeichnet, bspw. `hylos:name` (`hylos` stellt dabei den Namespace dar und `name` das XML-Element). Wird in Flash darauf zugegriffen, gibt es Probleme wegen dem Doppelpunkt hinter dem Namensraum-Präfix. Selbst wenn der Element- und Attributname mit dem Namespace vorher als String gespeichert und übergeben wurde, kann das Element nicht gefunden werden. Um dieses Problem zu lösen, wurden aus den XML-Dokumenten die Namespaces vorerst manuell gelöscht. Mithilfe eines XSLT-Stylesheets kann diese Aufgabe automatisch bewältigt werden, was in Lösungsansatz 2 näher erläutert wird.

Laden des Baum-Navigation-XML-Dokuments

Das Laden der Baum-Navigation-XML-Datei erfolgt im Grunde wie das Laden der Inhalt-XML-Datei, abgesehen von Zeile 6 - 9 wie Listing 3 zeigt.

In das XML-Objekt `XMLnavTree` wird der Inhalt der URL, welcher in `XMLnavTreePath` abgelegt ist, gespeichert. Anschließend wird überprüft, ob das XML-Dokument über die URL geladen wurde. Wenn das XML-Dokument vollständig geladen wurde, werden die benötigten Objekte aus dem XML-Dokument ausgelesen.

Innerhalb der Ladefunktion werden verschiedene Methoden aufgerufen, in denen die ausgelesenen Objekte verarbeitet werden, u.a. um das aktuelle Kapitel auf der Benutzeroberfläche anzuzeigen. Auf Details wird im folgenden Kapitel eingegangen.

```
1. function loadNavTreeXML (XMLnavTree : XML){
2.     XMLnavTree.ignoreWhite = true;
3.     XMLnavTree.load (XMLnavTreePath);
4.     XMLnavTree.onLoad = function (success){
5.         if (success){
6.             this = readNavTreeData (this, 0);
7.             setChapter (layerNav);
8.             setNav (navTree, chapterNav, layerNav);
9.             setTableOfContent ();
10.        }
11.    };
12. }
```

Listing 3: Laden des XML-Dokumentes für die hierarchische Navigation

8.3.2 Extraktion ausgewählter Elemente

Die Extraktion ausgewählter Elemente aus dem XML-Objekt dient zum Identifizieren der benötigten Objekte für die Lerneinheit. Hierbei werden sowohl die Objekte für den Inhaltsbereich der Benutzeroberfläche also beispielsweise, Überschriften, Texte, Links etc. extrahiert als auch alle Elemente für die hierarchische Navigation, um durch die Lerneinheit navigieren zu können.

Extraktion der Elemente für den Inhaltsbereich

Das Extrahieren der Elemente für den Inhaltsbereich erfolgt mittels der in Listing 4 abgebildeten Methode `readContentData (node : XML, depth : Number)`. Diese hat als Parameter das XML-Objekt und den Integer-Wert für die Ebenentiefe.

Listing 4 zeigt dies nur auszugsweise, um die grundlegende Funktionsweise der Methode zu veranschaulichen. Die komplette Methode ist im *Anhang, S. 107* aufgeführt. Der erste Knoten aus dem XML-Objekt wird innerhalb der Schleife auf den Knotentyp über die `NodeType`-Eigenschaft der XML-Klasse von ActionScript daraufhin überprüft, ob es sich um ein Bild oder einen Text handelt. Die `if`-Bedingung in Zeile 4 prüft, ob das aktuelle Element ein Text ist. Der numerische Wert 3 ist eine in ActionScript definierte Konstante, die einen Textknoten kennzeichnet.

```
1. function readContentData (node : XML, depth : Number){
2.     var child = node.firstChild;
3.     while (child)      {
4.         if ( child.nodeType == 3 ){
5.             showContent(child.nodeValue,i,child.ParentNode.nodeName);
6.             i ++;
7.         }
8.         /* (...) */
9.         if (child.nodeName == "image"){
10.            showContent (child.nodeName, i, child.attributes.src);
11.            i ++;
12.        }
13.        readContentData (child, depth + 1);
14.        child = child.nextSibling;
15.    }
16. }
```

Listing 4: Auslesen der benötigten Elemente für den Inhaltsbereich

Ein Textelement könnte beispielsweise wie folgt aussehen: `<hylos:section>Willkommen zur Lernheit - Wilo Arbeitsplan</hylos:section>`. Trifft der Parser auf die Stelle „Willkommen zur Lernheit - Wilo Arbeitsplan“, stellt er fest, dass dies ein Textelement ist und führt entsprechend die in der if-Bedingung angegebene Methode `showContent()` aus. Dabei werden die Parameter `child.nodeValue`, `i`, `child.ParentNode.nodeName` mitgegeben. `child.nodeValue` enthält in diesem Fall den Wert „Willkommen zur Lernheit - Wilo Arbeitsplan“, `i` enthält den Integer-Wert (wird zur Weiterverarbeitung mit den Textfeldern benötigt) und `child.ParentNode.nodeName` enthält den Namen des Eltern-Elementes, in diesem Fall „section“. Die Methode `showContent(child.nodeValue, i, child.ParentNode.nodeName)` formatiert die jeweiligen Elemente und gibt diese als einzelnes Textfeld aus. Wurde die if-Bedingung ausgeführt, erfolgt ein rekursiver Aufruf der Methode und die Variable `child` wird auf das nächste Element gesetzt, um diese auf ihren Knotentyp zu überprüfen.

Extraktion der Elemente für die hierarchische Navigation

Hierbei werden sämtliche Pfade, die als Verweise zu Kapiteln, Unterkapiteln und Seiten des gesamten Lernkurses führen, ausgehend vom Wurzelement des Lernkurses extrahiert.

Der folgende Code-Ausschnitt zeigt das Wurzelement eines Beispielkurses:

`<hylos:lo hylos:path="/HyLOS/content/Data/wilo/ArbeiAntri/ArbeiAntri">`. `Hylos` gibt den Namespace an, `lo` ist das Wurzelement, `path` ist ein Attribut von `lo` und gibt den Pfad des Elementes – in diesem Fall des Wurzelementes an. Im Anhang, S. 107 ist die komplette Beispieldatei abgebildet und listet alle enthaltenen Elemente des Wurzelementes auf. Für jedes Lernobjekt existiert ein Attribut `path`, das für die Anwendung zu extrahieren ist (siehe Listing 5).

```
1. function readNavTreeData(xmlroot_:XMLNode,depthNav:Number):Array{
2.     var node : XMLNode = xmlroot_.firstChild;
3.     while (node){
4.         if (node.attributes.path != undefined){
5.             var elementTitle = node.firstChild.firstChild.firstChild;
6.             elementTitles.push (elementTitle);
7.             var elementNav = node.attributes.path;
8.             navTree.push (elementNav);
9.             layerNav.push (depthNav);
10.        }
11.        readNavTreeData (node, depthNav + 1);
12.        node = node.nextSibling;
13.    }
14.    return navTree;
15. }
```

Listing 5: Auslesen der Elemente für die hierarchische Navigation

Die Methode besitzt zum einen ein XML-Objekt vom Typ `XMLNode` als Parameter für die einzelnen Knoten, zum anderen einen numerischen Wert vom Typ `Number` als Parameter für die Tiefe der Ebenen. Als Rückgabewert liefert die Methode ein Array (`navTree`) mit dem Pfad zu allen Unterelementen zurück. In `node` wird zuerst das Wurzelement gespeichert und dient damit als Ausgangspunkt für die darauf folgende `While`-Schleife. Die Methode wird solange rekursiv ausgeführt, bis alle Unterelemente des Wurzelementes identifiziert wurden.

Es werden nur diejenigen Knoten extrahiert, die das Attribut `path` enthalten. Zusätzlich wird das Element `<hylos:title>`, welches wiederum ein Unterelement des aktuellen Elementes ist (siehe Anhang, `ArbeiAntri.xml`, S. 128), extrahiert und in dem `elementNav`-Ar-

ray gespeichert. Gleichzeitig wird die Ebenentiefe des jeweiligen Elementes innerhalb der Baumstruktur in `depthNav`-Array gespeichert.

8.3.3 Formatierung der Inhaltsobjekte

Nachdem in Kapitel 8.3.2, S. 79 u.a. die Extraktion der Inhaltselemente für den Inhaltsbereich beschrieben wurde, folgt nun die Beschreibung wie diese Elemente formatiert werden.

Angesichts des relativ kleinen Bildschirms der PlayStation Portable werden die Bilder zu den Texten separat angezeigt. Der Inhaltsbereich der Anwendung enthält nur Textelemente. Ist innerhalb des Textes ein Bild enthalten, wird statt des Bildes ein Verweis zum Bild angezeigt, über den sich das gesamte Bild in Vollbildansicht anzeigen lässt (siehe Abbildung 18).

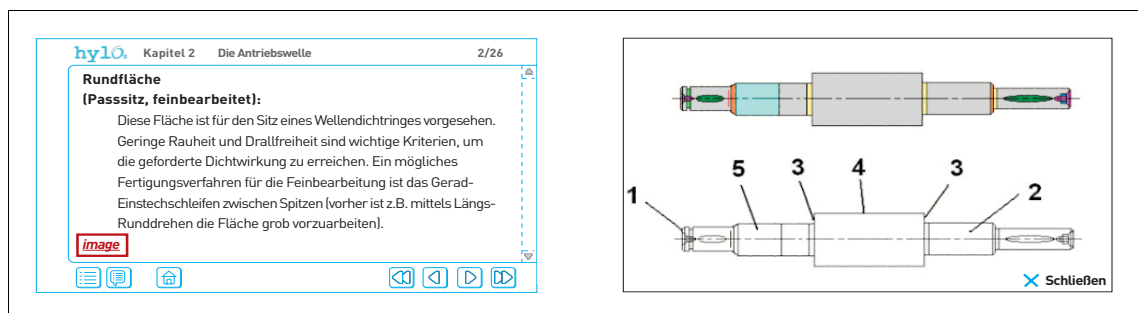


Abbildung 18: Sonderansicht für Bilder

In diesem Lösungsansatz wird zur eindeutigen Formatierungsvorschrift jedem Knoten eines Inhaltselementes ein eigenes Textfeld mit entsprechenden Formateigenschaften zugewiesen. Die linke Abbildung in Abbildung 18 zeigt beispielsweise die vier Inhaltselemente (Überschrift, Unterüberschrift, Absatz, Bildfeld), die jeweils in einem gesonderten Textfeld platziert wurden.

Flash hält für die Formatierung von HTML- und XML-Texten zwei Möglichkeiten bereit: CSS-Stylesheets und die `TextFormat`-Klasse. Da die Anwendung innerhalb des Flash Player 6 laufen soll, ist es nicht möglich, die Formateigenschaften der Inhaltselemente zentral in einem CSS-Stylesheet zu definieren, denn diese Funktion wird ab Flash Player 7 bereitgestellt. Somit wurden die Eigenschaften der Inhaltselemente über die `TextFormat`-Klasse definiert. In Listing 6 werden die für die Formatierung zuständigen Code-Fragmente aufgeführt, die auszugsweise der bereits erwähnten Methode `showContent()` entnommen wurden. Dazu wird zuerst ein `TextFormat`-Objekt erstellt und diesem verschiedene Eigenschaf-

ten wie Schriftart, -farbe und -größe zugewiesen. In Zeile 5 werden einem zuvor erzeugten Textfeldnamens holder ["text_txt" + i] mit enthaltenem Inhaltsobjekt diese Formateigenschaften zugewiesen.

Ist das Inhaltselement vom Autor beispielsweise ausgezeichnet durch ein XML-Tag , wird dem Inhaltselement die entsprechende Formatanweisung der TextFormat-Klasse zugewiesen, in diesem Fall Eigenschaft bold.

```
1. formatText = new TextFormat ();
2. formatText.font = "arial";
3. formatText.color = 0x333333;
4. formatText.size = 13;
5. holder ["text_txt" + i].setTextFormat (formatText);
6. if (parentname == "strong") {
7.     formatText = new TextFormat ();
8.     formatText.bold = true;
9.     holder ["text_txt" + i].setTextFormat (formatText);
10. }
```

Listing 6: Formatierung der einzelnen Inhaltselemente mithilfe von TextFormat()

8.3.4 Test – Lösungsansatz 1

Da die erstellte Anwendung in erster Linie die plattformgerechte Darstellung der Inhaltsobjekte auf der PlayStation Portable sicherstellen soll, liegt der Fokus beim Testen der Flash-Anwendung ausschließlich auf der Überprüfung der korrekten Formatierung.

Als Testkurs wurde der komplette hylOs-Lernkurs „Arbeitsplan Antriebswelle“¹ eingesehen und auf die Anzeige der Inhaltsobjekte hin analysiert.

**Problem der
-Elemente**

Innerhalb des gesamten Inhaltsbereiches fallen beim Testen der Anwendung die großen Abstände zwischen den einzelnen Textabschnitten auf. Es werden Zeilenumbrüche an Stellen vorgenommen, an denen kein Zeilenumbruch notwendig ist. Als Beispiel wird in Abbildung 19 die Ansicht in der Benutzeroberfläche und rechts das dazugehörige XML-Dokument des Lernobjektes „Die Antriebswelle“ (siehe Anhang, Antri.xml, S. 130) gezeigt.

¹ Publiziert unter <http://hylosdev.fhtw-berlin.de/index.xhtml>

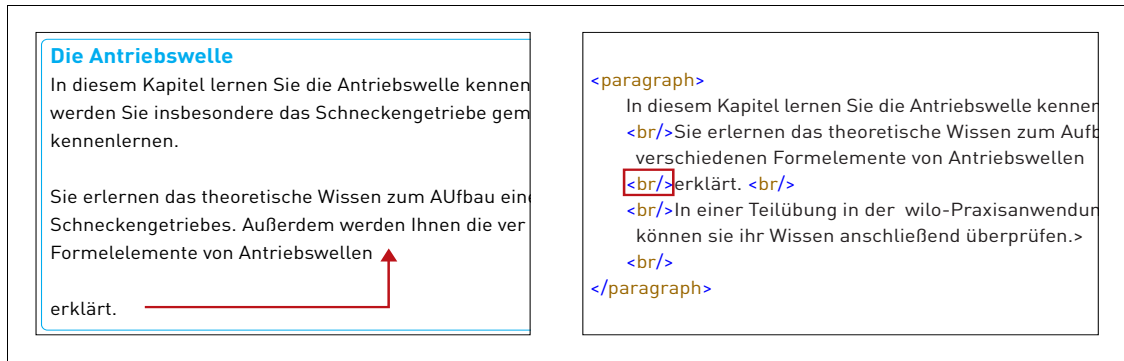


Abbildung 19: Interpretation der
-Elemente innerhalb der Benutzeroberfläche (links) und dazugehöriges XML-Dokument (rechts)

Dieses Problem resultiert aus der Tatsache, dass die Inhalte pro Element in einem Textfeld gespeichert und ausgegeben werden, denn der Text innerhalb des
-Tags bzw. Knotens wird in der Anwendung als ein Inhaltselement interpretiert. Entsprechend Listing 4 (S. 80) werden auch
-Elemente zunächst extrahiert, weil sie vorgesehen sind und benötigt werden für einen regulären Zeilenumbruch. Allerdings gibt es bisweilen auch „erzwungene“ Zeilenumbrüche, die vom Autor webbrowserspezifisch vorgenommen wurden, wie Abbildung 19 demonstriert.

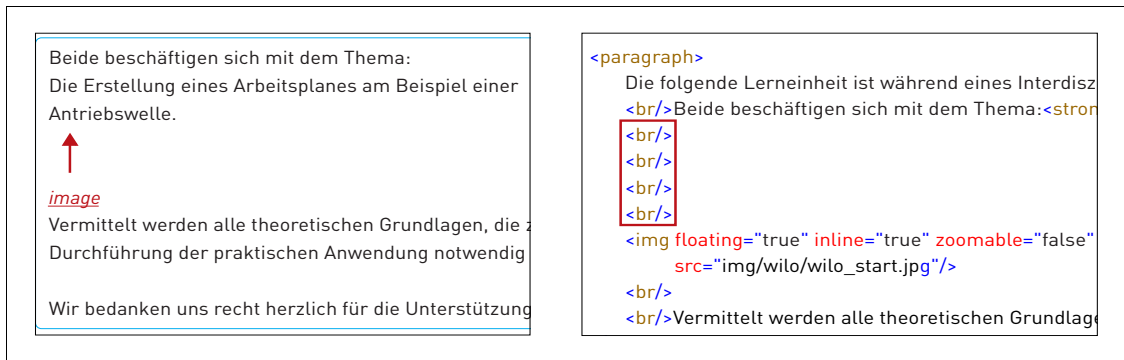


Abbildung 20: ungewollt große Abstände durch mehrere Zeilenumbrüche

Ein anderes Beispiel, bei welchem die Zeilenumbrüche optisch unschöne Effekte auslösen, sind in Abbildung 20 zu sehen. Um diesen Effekt zu vermeiden, wird innerhalb der Extraktion des
-Elements auf den folgenden Nachbarn des Elementes geprüft, ob es sich ebenfalls um ein
 handelt. Ist dies der Fall, wird der Folge-
 gelöscht.

Problem durch Text hervorhebungen mittels `<emph>` und ``

Ähnlich verhält es sich bei den Text hervorhebungen `<emph>` und ``. Da in der Anwendung keine Unterscheidung nach Inhaltselementen innerhalb eines Paragraphen gemacht wird, wird dem Inhalt dieser Elemente ebenfalls ein Textfeld zugewiesen, wodurch ein Zeilenumbruch entsteht. Um dieses Problem zu vermeiden, müssten diese innerhalb eines `<paragraph>`-Elementes ausgegeben werden.

Ein anderer Effekt im Zusammenhang mit den Hervorhebungen ist in Abbildung 21 im grünmarkierten Bereich zu sehen. Dieser Effekt könnte bei der Erstellung des Lernkurses in der Autorenumgebung hylOs entstanden sein, beispielsweise wenn der Autor zuerst das Wort `Nute` markiert und kursiv setzt und anschließend getrennt davon `nfräsen` als kursiv markiert.

Langlochnuten:

- zusammen mit Passfedern können Drehmomente übertragen werden. Dies geschieht durch eine formschlüssige Verbindung. Ein mögliches Fertigungsverfahren ist

Nute `<emph>`

nfräsen

Fasen ``

- oder auch abgerundete Außenkanten

```

<paragraph>
  <enumeration>
    <item>
      <strong>Langlochnuten:</strong> zusammen
      ist <emph>Nute</emph><emph>nfräsen</emph>
    </item>
    <item>
      <strong>Fasen</strong> oder auch abgerundete
      <emph>Formdrehen</emph>.
    </item>
  </enumeration>

```

Abbildung 21: ungewollter Zeilenumbruch bei Hervorhebungen

Dieses Problem kann entsprechend dem zweiten `
`-Lösungsansatz behoben werden.

Problem der nummerierten Listen

Beim Programmieren der Anwendung fiel auf, dass nummerierte Listen über die `TextFormat`-Klasse in Flash nicht umgesetzt werden können. Die Klasse stellt für Listen lediglich die Eigenschaft `bullet` bereit, die sich nur für unsortierte Listen eignet. Nummerierte Listen können somit nur als unsortierte Listen angezeigt werden.

8.4 Lösungsansatz 2

Dieser zusätzliche Lösungsansatz wurde gewählt, um die bei dem ersten Lösungsansatz aufgetretenen Probleme u.a. mit den Namespaces und die daraus resultierende manuelle Bearbeitung der XML-Dokumente zu lösen.

Der wesentliche Unterschied dieses Ansatzes gegenüber des vorherigen, besteht in der Selektion der Inhaltselemente über ein XSLT-Stylesheet anstatt deren programmatische Extraktion innerhalb von Flash. Bei dem ersten Ansatz wurden die XML-Dokumente mit allen Inhaltselementen, die vom Autor eingetragen wurden, eingelesen und jeweils in einem Textfeld ausgegeben. Nun werden nur die durch die XSLT-Transformation selektierten Elemente in Flash eingelesen. Da die einzelnen Inhaltselemente bei der XSL-Transformation zu HTML-Tags umgewandelt und in Flash nicht einzeln extrahiert und formatiert werden müssen, erfolgt die Ausgabe sämtlicher Inhaltselemente in einem Textfeld. Beispielsweise wird das XML-Element `<paragraph>` durch das HTML-Tag `<p>` ersetzt. Der Vorteil daran ist, dass bei der Verarbeitung in Flash die eingelesenen HTML-Elemente aus dem XML-Dokument innerhalb eines Textfeldes interpretiert, also entsprechend formatiert werden.

Die Erzeugung der Benutzeroberfläche sowie die Erstellung der hierarchischen Navigation wurden bei diesem Ansatz nicht modifiziert.

8.4.1 Erstellung des XSLT-Stylesheets

Ausgehend von den Problemen, die beim ersten Lösungsansatz bei den Namespaces (siehe Kapitel 8.3.1, S. 77) aufgetreten sind, wurde ein Stylesheet zur Transformation der XML-Dokumente vom Dokumenten-Typ `content` erstellt (siehe Kapitel 4.1.2, S. 37).

Das Stylesheet basiert zum Teil auf einem in hylOs existierenden Stylesheet und wurde um neue Funktionen ergänzt. Die übernommenen Passagen sind im Stylesheet entsprechend gekennzeichnet. Das gesamte Stylesheet ist im Anhang (`hylos-content-ppsp.xml`, S. 135) aufgelistet.

Transformation der Namespaces

In Lösungsansatz 1 wurde das Problem mit der Verarbeitung der Namespaces in Flash vorerst durch manuelles Löschen der Namespaces behoben. Dies ist jedoch für die dynamische Verarbeitung keine Lösung. Durch die Erzeugung eines Templates innerhalb des Stylesheets können die Namespace-Präfixe noch vor der Verarbeitung der XML-Dokumente be-

seitigt werden. In Listing 7 ist der Programmcode für die beispielhafte Transformation der Namespaces für das Attribut `path` abgebildet. Dafür werden alle Attribute mit dem Namen `hylos:path` ersetzt durch `path`.

```
1. <xsl:attribute name="path">
2.     <xsl:value-of select="@hylos:path"/>
3. </xsl:attribute>
```

Listing 7: Transformation der Namespaces im XSL-Stylesheet

Formatierung der Inhaltselemente durch das Stylesheet

In Lösungsansatz 1 wurden sämtliche Inhaltselemente in Flash über die Textformat-Klasse formatiert. Mittels des Stylesheets kann die Formatierung der Inhaltselemente in hylOs bereits vor dem Einlesen der XML-Dokumente erfolgen. Dazu werden dem jeweiligen Element HTML-Elemente

```
1. <xsl:template match="paragraph">
2.   <p>
3.     <font color="#333333" size="13" face="Arial">
4.       <xsl:apply-templates/>
5.     </font>
6.   </p>
7. </xsl:template>
```

Listing 8: Formatierung des Inhaltselements „paragraph“ im Stylesheet

zugeordnet. In Listing 8 ist als Beispiel dafür ein Template für das Inhaltselement `paragraph` aufgeführt. Das `paragraph`-Element wird durch die HTML-Elemente `<p>` und `` ersetzt und zugleich wird diese Formatierungsanweisung auf sämtliche Unterelemente übertragen (Zeile 4). Zum besseren Verständnis ist der entsprechende Code-Ausschnitt aus dem XML-Dokument in Listing 9 und Listing 10 vor und nach der Transformation dargestellt.

```
1. <paragraph>
2.     Die folgende Lerneinheit ist im Sommersemester 2006 entstanden.
3. </paragraph/>
```

Listing 9: paragraph-Element vor der Transformation

```
1. <p><font color="#333333" size="13" face="Arial">
2.     Die folgende Lerneinheit ist im Sommersemester 2006 entstanden.
3. </font>
4. </p>
```

Listing 10: paragraph-Element nach der Transformation

Pipeline in der Sitemap von Cocoon

Um die XML-Dokumente, die mit dem Stylesheet transformiert werden, in der Flash-Anwendung nutzen zu können, muss die Sitemap von Cocoon (Kapitel 3.3, S. 27) um eine neue Pipeline erweitert werden. Dazu wird ein XPath-Pattern erzeugt (siehe Listing 11, Zeile 2). Nun können innerhalb der Flashanwendung XML-Dokumente vom Dokumenten-Typ `content`, mit der Endung `.psp`, ausgelesen werden. Zeile 4 zeigt die Verknüpfung des Stylesheets mit der Dateiendung `.psp`.

```
5. <!-- Autor: Arne Hildebrand, http://www.link-lab.net -->
6. <map:pipeline>
7.   <map:match pattern="content/**/*.psp">
8.     <map:generate src="hylos://cpages/{1}" type="content" />
9.     <map:transform src="stylesheets/hylos-content-psp.xsl" />
10.    <map:serialize type="xml" />
11.  </map:match>
12.</map:pipeline>
```

Listing 11: Pipeline für das Stylesheet in der Sitemap von Cocoon

8.4.2 Import der notwendigen XML-Dokumente

Der Import der XML-Dokumente vom Dokumenten-Typ `content` und `navtree` erfolgt grundsätzlich wie das Laden der XML-Dokument in Lösungsansatz 1 (siehe Kapitel 8.3.1, S. 77). Der Unterschied besteht darin, dass nun alle Inhaltsobjekte in einem Textfeld ausgegeben und daher nicht gesondert extrahiert werden müssen – bis auf das ``-Element, welches während der Extraktion modifiziert wird.

8.4.3 Extraktion ausgewählter Objekte

Im Gegensatz zu Lösungsansatz 1 wird bei der Extraktion der XML-Objekte lediglich auf das `img`-Element getestet. Dieses wird durch einen Verweistext ersetzt, da auch bei diesem Ansatz die Bilder in einem gesonderten „Fenster“ angezeigt werden sollen. Listing 12 zeigt die Abfrage auf das `img`-Element. Zunächst wird ein XML-Objekt namens `neu` erzeugt, das den Verweis auf das Bild erhalten wird. Die Flash-Methode `parseXML()` übergibt den Pfad des Bildes und erzeugt einen „klickbaren“, also funktionsfähigen Link darauf, der über HTML Code sichtbar gemacht wird.

Dieses so umgeparste `img`-Element wird als XML-Knoten in das XML-Objekt an der Stelle eingefügt, an der sich das ursprüngliche `img`-Element befunden hat, welches stattdessen gelöscht wird.

```
1.  if (child.nodeName == "img") {
2.      var neu = new XML();
3.      neu.parseXML("<a href=\"\" asfunction:_root.link_click, \"
4.          +child.attributes.src+\"\"><font color=\"\#992233\">
5.          <i>Bild bitte hier klicken</i></font></a>");
6.      child.parentNode.insertBefore(neu, child);
7.      child.removeNode();
8.  }
```

Listing 12: Extraktion des `img`-Inhaltselementes

8.4.4 Formatierung der Inhaltsobjekte in Flash

Die Formatierung der Inhaltsobjekte erfolgt wie bereits erwähnt vorwiegend innerhalb des XSLT-Stylesheets entsprechend Listing 8. Dort werden dem jeweiligen Element bestimmte HTML-Tags zugewiesen. Die Ausgabe aller Inhaltsobjekte, die sich in dem XML-Dokument befinden, erfolgt in einem Textfeld. Über ActionScript werden dem Textfeld über die `TextFormat`-Klasse in ActionScript zusätzliche Eigenschaften zugewiesen, die im Stylesheet so nicht möglich sind. Diese Eigenschaften sind beispielsweise der Zeilenabstand und der Abstand zum Rand. Anschließend wird das Textfeld im Inhaltsbereich der Benutzeroberfläche ausgegeben.

8.4.5 Test Lösungsansatz 2

Beim Testen des zweiten Lösungsansatzes hat sich herausgestellt, dass das `<emph>` und ``-Anzeige Problem (siehe Kapitel 8.3.4, S. 85) durch das Stylesheet behoben werden konnte. Denn auch diese Elemente werden zuvor zu `<i>` und ``-HTML-Tags transformiert und mit den übrigen Inhaltselementen in einem Textfeld ausgegeben. Auch das Problem mit nummerierten Listen konnte durch ein Stylesheet gelöst werden, indem die Listeneinträge hochgezählt und die Ziffern textuell ausgegeben werden.

Entsprechend können auch die `
`-Probleme gelöst werden, der Aufwand dafür übersteigt jedoch die Möglichkeiten dieser Arbeit.

8.5 Anmerkungen und Komplikationen

In der Ist-Analyse wurde festgestellt, dass Autoren beliebige Bildformate einstellen können und es innerhalb von hylOs keine Routine zu Umwandlung der Bildformate auf ein einheitliches Format wie zum Beispiel das „jpeg“-Format gibt. Das Laden von Bildern innerhalb der Flash-Anwendung ist lediglich mit nicht-progressiven JPEGs möglich, da die Flash Player Version 6 nur diese unterstützt [vgl. [SCH]]. Dasselbe gilt auch für andere Formate, die von Flash nicht unterstützt werden. Daher müssten nicht oder unzureichend darstellbare Inhaltselemente wie externe Links, PDFs, nicht-unterstützte Bildformate mit Rückmeldung für den Nutzer abgefangen werden.

Da die Bildanzeige des Prototypen in einem separaten Fenster erfolgt, besteht das Problem, dass der Nutzer eventuell vorhandene textuelle Bezüge oder Legenden nicht zusammen mit dem Bild überblicken kann, sondern vor- und zurücknavigieren muss. Hierfür müssten innerhalb von hylOs neue Inhaltselemente definiert werden, um Legenden mit dem Bild zusammenzufassen.

Wie bereits in der Analyse beschrieben, hat sich beim Testen der Anwendung mit der PlayStation Portable herausgestellt, dass der Analogstick sich als Barriere für die Navigation erweist, insbesondere auf kleinen Flächen wie auf den Pfeilen der Scrollbalken. Außerdem soll hier noch bemerkt werden, dass das Laden der Inhalte mit der PlayStation Portable genauso wie das Nutzen der Scrollbalken relativ lange dauert, dazu wurden jedoch keine weiteren genaueren Untersuchungen durchgeführt. Beim Testen der Anwendung in einem PC-Webbrowser gab es keine Auffälligkeit die Geschwindigkeit betreffend.

Aus zeitlichen Gründen wurden weiterhin interne Links zu anderen Kapiteln bzw. Seiten im Text wie im Inhaltsverzeichnis nur als Text umgesetzt, so dass die hierarchische Navigation nur beschränkt umgesetzt werden konnte.

Der Prototyp erhebt nicht den Anspruch auf Vollständigkeit und Mangelfreiheit. Zu Demonstrationszwecken wurden daher nur die dominierenden, nicht jedoch spezielle Inhaltsobjekte, wie Tabellen, mathematische Formeln, etc. entsprechend formatiert.

Abgesehen von den hier aufgeführten Problemen wurde mit dem Prototypen eine navigationsfähige Applikation erstellt, die demonstriert, dass Inhalte aus hylOs dynamisch in Flash geladen und auf der PlayStation Portable abgebildet werden können.

Kapitel 9 Zusammenfassung und Ausblick

Entsprechend der formulierten Zielsetzung wurde im Rahmen dieser Diplomarbeit ein lauffähiger Prototyp einer mobilen Lernanwendung für die PlayStation Portable erstellt, nachdem die zugrundeliegenden Rahmenbedingungen eingehend diskutiert wurden.

Sämtliche Untersuchungen im Rahmen dieser Diplomarbeit haben ergeben, dass es durchaus möglich ist, mobile Inhalte aus hylOs auf die PlayStation Portable abzubilden und dem Nutzer somit neue Möglichkeiten zu eröffnen, sich orts- und zeitunabhängig weiterzubilden.

Da die Lerninhalte aus hylOs XML-Technologien zur Strukturierung und Formatierung der Dokumente verwenden, können Inhalte leicht in andere Formate überführt und in andere Anwendungen ohne zusätzliche Schnittstellen oder aufwändige Konvertierungsmechanismen eingebunden werden. Wie in den Kapiteln Konzept und mobile Learning beschrieben, sind beim mLearning gegenüber eLearning vielfältige und neuartige Lernszenarien denkbar. Das Schlagwort „Lebenslanges Lernen“ erfährt im mobilen Kontext insoweit eine Reformation, als der örtliche Bezug sowohl das Wissen selbst, als auch die Informationserfassung und -aufnahme mitbestimmt.

Da es sich beim mLearning – im Vergleich zum eLearning – um ein relativ neues Forschungsfeld handelt, wurde eine Studie zum mobile Learning vorgestellt, um einen thematischen Einstieg zu geben und die Möglichkeiten und Grenzen des Fachgebietes einschätzen und evaluieren zu können.

Im Rahmen der Analyse wurden die technischen Hintergründe des hylOs-Systems erörtert und die technischen Leistungsmerkmale der PlayStation beschrieben, um Grundlagen für die Konzeption der Anwendung zu schaffen.

Um einen Eindruck von spielebasierter Lernsoftware innerhalb des fachlichen Kontextes dieser Arbeit zu erhalten, wurden zum einen eine kommerzielle Lernsoftware für die PlayStation Portable und zum anderen eine freie, spielbasierte Lernsoftware zu einen hylOs-Lernkurs betrachtet.

Ausgehend von den Ergebnissen der Systemanalyse wurden die Anforderungen an die zu realisierende Anwendung festgelegt und im Konzept aufgegriffen und um neue Ideen ergänzt. Der lauffähige Prototyp beinhaltet hierbei lediglich grundlegende Funktionalitäten, um die Inhalte plattformgerecht anzuzeigen und hierarchisch zu navigieren. Die Umset-

zung ergab zwei lauffähige Versionen. Da die Inhaltsobjekte zunächst einzeln formatiert werden mussten, wurde im zweiten Ansatz die Formatierung durch XSLT-Stylesheets vereinfacht und optimiert.

Die festgesetzten Ziele wurden weitestgehend realisiert. Allerdings haben sich im Zuge der Realisierung und des anschließenden Tests zusätzliche Optimierungsansätze zur Erweiterung der Funktionalitäten für hylOs sowie für den Prototypen selbst herauskristallisiert.

Für eine mögliche Umsetzung des im Konzept aufgenommenen Wissenstests sowie zur Extraktion von Bildlegenden wäre die Definition neuer Inhaltstypen innerhalb der Autorenumgebung in hylOs denkbar. Weiterhin wäre ein automatischer Rich-Media-Konverter nützlich, um solche Medien für die anfragende Anwendung plattformgerecht bereitzustellen.

Grundlegend wären über XSLT-Anweisungen weitere Szenarien denkbar, wie beispielsweise eine nutzergerechte Filterung von Inhalten, etwa nach Schwierigkeitseinstufung.

Die Umsetzung der semantischen Relationen entsprechend meines Konzeptes würden dem Nutzer den Lernprozess unterstützende kontextsensitive Informationen zur Verfügung stellen. Dies wäre ein interessantes und herausforderndes Aufgabengebiet.

Weiterhin wünschenswert wären lernproduktive Funktionalitäten für die Texteingabe und zur Orientierung, welche die Bedienfreundlichkeit unterstützen, wie bspw. das Setzen von Lesezeichen, Textmarkern oder Kommentarfunktionen.

Da beim mLearning aufgrund der nicht flächendeckenden Verbreitung von drahtlosen Netzen die Gefahr besteht, dass die Verbindung abrupt unterbrochen wird, wäre eine automatische Zwischenspeicherung der dynamisch geladenen Inhalte auf der PlayStation Portable sinnvoll.

Da mobile Geräte neben ihrem Kommunikationszweck vorwiegend als Unterhaltungsmedium genutzt werden, sollten spielorientierte Lernkonzepte speziell auf Mobilgeräte zugeschnitten und angewandt werden.

GLOSSAR UND ABKÜRZUNGSVERZEICHNIS

AAC	Advanced Audio Coding
ADPCM	Adaptive Differential Pulse Code Modulation
Apache Software License	Freie Software-Lizenz der Apache Software Foundation
ASP	Active Server Pages
Blended Learning	zu dt. Integriertes Lernen, ist ein Lernkonzept, das die klassische Präsenzlehre und moderne eLearning-Methoden didaktisch sinnvoll miteinander verbindet
Bluetooth	Funktechnologie zur Übertragung von Daten über kurze Distanzen (bis zu 10 Meter)
BMP	Windows Bitmap
CMS	Content-Management-System, System zur Erstellung, Verwaltung und Publikation dynamischer Inhalte von Webseiten. Ein CMS verfolgt den Ansatz der Trennung von Inhalt und Struktur und erleichtert somit die Arbeit für Autoren ohne Programmierkenntnisse.
CSS	Cascading Style Sheets, Standard Beschreibung von Formateigenschaften eines strukturierten Dokumentes (bspw. HTML,XML). Die Definition der Eigenschaften kann hierbei innerhalb des Dokumentes oder in einer separaten Datei erfolgen.
DDC	Dewey Decimal Classification, bezeichnet ein internationales Klassifikationsschema zur Katalogisierung von Büchern in Bibliotheken
DTD	Document Type Definition, definiert die Dokumentenstruktur eines XML-Dokumentes. Dabei können zulässige Elementtypen, Elementinhalte, Attribute und Attributwerte definiert werden. Außerdem wird in der DTD die zulässige Reihenfolge des Auftretens der Elemente und die Verschachtelung dieser festgelegt. Eine Definition kann innerhalb oder in einer externen Datei erfolgen.

Edutainment	ist eine Kombination von Education und Entertainment. Damit werden multimediale Anwendungen bezeichnet, die einerseits Wissen vermitteln und andererseits unterhalten sollen.
Framework	zu dt. Rahmenwerk, genutzt bei der Softwareprogrammierung, stellt eine Sammlung verschiedener Werkzeuge und Klassenbibliotheken für verschiedene Funktionen zur Verfügung und kann beliebig erweitert werden
GIF	Graphics Interchange Format
GPRS	General Packet Radio Service, erlaubt Übertragungsgeschwindigkeiten von bis zu 115 KBit pro Sekunde, GSM-Erweiterung (siehe GSM)
GSM	Global System for Mobile Communications, Mobilfunkstandard der sogenannten zweiten Generation, erlaubt eine relativ geringe Übertragungsgeschwindigkeit von 9,6 KBit pro Sekunde
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
hylOs	Hypermedia Learning Object System
JPG	Joint Photographic Experts Group
JSP	Java Server Pages
Lernobjekt	bezeichnet die kleinste sinnvolle Lerneinheit in die ein Lernangebot geteilt werden kann
Lernobjektmanagementsystem	bezeichnet eine Software, mit der Lerninhalte erstellt, verwaltet und angeboten werden können. Alternative Bezeichnung zu Lernplattform
LOM	Learning Objects Metadata, Standard zur Beschreibung von Lernobjekten.
MathML	Mathematical Markup Language, W3C-Standard zur Darstellung von mathematischen Formeln und komplexen Ausdrücken
MIR	Media Information Repository
MMS	Multimedia Messaging Service
MPEG	Moving Picture Experts Group

Open Source	zu dt. Quelloffenheit, Quellcode einer Software, welcher der Allgemeinheit frei zugänglich und kostenlos zur Verfügung gestellt wird und der verändert werden darf
PCM	Puls-Code-Modulation
PDF	Portable Document Format
PHP	Hypertext Preprocessor, ursprünglich Personal Home Page Tools
PNG	Portable Network Graphics
Proxyserver	Ein Proxyserver erhöht einerseits die Sicherheit, da er als Filter für ungewollte Anfragen fungieren kann und andererseits sorgt er für einen effizienteren Datentransfer durch einen integrierten Zwischenspeicher für bereits angeforderte Webseiten. Durch den Zwischenspeicher verringert der Proxyserver den Datentransfer und sorgt für eine schnellere Bearbeitung der HTTP Anfragen.
PSP	PlayStation Portable
RSS	Really Simple Syndication
RSS-Feed	werden bspw. von Newsseiten angeboten, können von Nutzern abonniert werden und mittels eines Feed-Readers gelesen werden. Die Inhalte bestehen aus XML und stellen Kurzinformationen oder Artikel der Webseite dar. Der Feed-Reader lädt die neuesten abonnierten RSS-Feeds automatisch in regelmäßigen Abständen herunter. Inhalte können neben den Text ebenso Podcasts also Audio- oder Videodaten enthalten
SGML	Standard Generalized Markup Language, zu dt. Normierte Verallgemeinerte Auszeichnungssprache
SMIL	Synchronized Multimedia Integration Language
SMS	Short Message Service
SWF	ShockWave Flash oder Small Web Format
TIFF	Tagged Image File Format
UMD	Universal Media Disc, ist ein von Sony entwickeltes optisches Speichermedium speziell für die PlayStation Portable zum Abspielen von Filmen, Musik und Spielen. Die UMD hat eine Speicherkapazität von 1,8 GB.

UMTS	Universal Mobile Telecommunications System, Mobilfunkstandard der sogenannten dritten Generation, erlaubt Datenübertragung bis zu 2 MBit pro Sekunde, Nachfolger von GSM (siehe GSM)
URI	Uniform Resource Identifier
Usability	bezeichnet Bedienfreundlichkeit von Webangeboten, wichtige Kriterien sind u.a. eine einfache Handhabung und damit ein barrierefreier Zugang zu den Inhalten für den Benutzer.
WAV	Waveform Audio File Format
WEP	Wired Equivalent Privacy
WLAN	Wireless Local Area Network, drahtloses lokales Netzwerk
WMA	Windows Media Audio
WML	Wireless Markup Language
WPA	WiFi Protected Access
WPA-PSK	WPA Pre-Shared-Key
WYSIWYG	What You See Is What You Get
XML	Extensible Markup Language, dt. erweiterbare Auszeichnungs-Sprache
XPath	XML Path Language
XPointer	XML Pointer Language
XSL	Extensible Stylesheet Language
XSL-FO	XSL Formatting Objects
XSLT	XSL Transformations

ABBILDUNGSVERZEICHNIS

Abbildung 1: Kontext des Mobile Learnings.....	10
Abbildung 2: Pipeline in Cocoon.....	27
Abbildung 3: Systemarchitektur / Prozess der Dokumentengenerierung in hylOs.....	30
Abbildung 4: Grundlegende Tastenfunktionen	43
Abbildung 5: Startnavigation der PlayStation Portable.....	43
Abbildung 6: Navigation des Internetbrowsers.....	45
Abbildung 7: Ablauf zur Online- bzw. Offline-Bearbeitung des Kurses.....	61
Abbildung 8: Mögliche Präsentationsansichten für Kursinhalte.....	62
Abbildung 9: Beispielstruktur einer Lerneinheit.....	63
Abbildung 10: Beispielstruktur einer Lerneinheit.....	64
Abbildung 11: Beispielhafte semantische Struktur einer Lerneinheit.....	65
Abbildung 12: Beispielhafte semantische Struktur einer Lerneinheit.....	67
Abbildung 13: Layout der Lernumgebung.....	70
Abbildung 14: Layout der Lernumgebung.....	71
Abbildung 15: Buttons Semantik.....	72
Abbildung 16: Layout Hilfsfunktionen.....	72
Abbildung 17: Benutzeroberfläche mit Beispielinhalt.....	75
Abbildung 18: Sonderansicht für Bilder.....	82
Abbildung 19: Interpretation der -Elemente innerhalb der Benutzeroberfläche (links) und dazugehöriges XML-Dokument (rechts).....	84
Abbildung 20: ungewollt große Abstände durch mehrere Zeilenumbrüche.....	84
Abbildung 21: ungewollter Zeilenumbruch bei Hervorhebungen.....	85

LISTINGVERZEICHNIS

Listing 1: Beispielhafte XML-Struktur in hylOs.....	38
Listing 2: Laden der Inhalte eines Lernobjektes aus der Inhalt-XML-Datei.....	78
Listing 3: Laden des XML-Dokumentes für die hierarchische Navigation.....	79
Listing 4: Auslesen der benötigten Elemente für den Inhaltsbereich.....	80
Listing 5: Auslesen der Elemente für die hierarchische Navigation.....	81
Listing 6: Formatierung der einzelnen Inhaltselemente mithilfe von TextFormat().....	83
Listing 7: Transformation der Namespaces im XSL-Stylesheet.....	87
Listing 8: Formatierung des Inhaltselements „paragraph“ im Stylesheet.....	87
Listing 9: paragraph-Element vor der Transformation.....	87
Listing 10: paragraph-Element nach der Transformation.....	88
Listing 11: Pipeline für das Stylesheet in der Sitemap von Cocoon.....	88
Listing 12: Extraktion des img-Inhaltselementes	89

TABELLENVERZEICHNIS

Tabelle 4:1. Semantische Relationen in hylOs.....35

Tabelle 4:2. Auszug der möglichen Inhaltselemente in hylOs.....41

LITERATURVERZEICHNIS

- [ADO] **Flash Professional 8**, *Online*:
<http://www.adobe.com/de/products/flash/flashpro/>, *Letzter Zugriff*: 02.02.2007
- [BRO] **Web design for the Sony PSP**, *Online*:
<http://www.brothercake.com/site/resources/reference/pspl/>, *Letzter Zugriff*:
25.11.2006
- [CEN] **PSP Buttons Map to 'Escape' Key (27)**, *Online*:
http://www.centralquestion.com/archives/2006/04/pspl_start_button_maps_to_escape_key_27.html, *Letzter Zugriff*: 08.02.2007
- [CMS] **Unplugged learning: A report on the rise of mobile technology in learning.**,
Online: <http://cms.steo.govt.nz/NR/rdonlyres/6C03CDFA-70E7-4179-90E5-7637BA267D7C/0/UnpluggedLearningFinalNovember04.doc>, *Letzter Zugriff*:
22.01.2007
- [COC] **Generators in Cocoon**, *Online*:
<http://cocoon.apache.org/2.1/userdocs/generators.html>, *Letzter Zugriff*:
02.12.2006
- [COCb] **The Apache Cocoon Project**, *Online*: <http://cocoon.apache.org/>, *Letzter Zugriff*: 21.01.2007
- [CON] **Aktuelle Webserver Statistik nach Verwendung und Performance**, *Online*:
<http://www.contentmetrics.de/WebSurveillant/Web-Server-Statistik.jsp>, *Letzter Zugriff*: 19.12.2006
- [DBR] **Auslaufmodell GSM**, *Online*:
http://www.dbresearch.de/PROD/DBR_INTERNET_DE-PROD/PROD0000000000198071.pdf, *Letzter Zugriff*: 04.02.2007
- [EHL+06] **Michael Engelhardt, Arne Hildebrand, Dagmar Lange, Thomas C. Schmidt, Mathias Werlitz** : Mobile eLearning Content on Demand, In: *Proceedings of the International Conference Interactive Computer aided Learning ICL 2006*, 2006. Kassel University Press

- [EHL05] **Michael Engelhardt, Arne Hildebrand, Thomas C. Schmidt** : Automatisierte Augmentierung von Lernobjekten in einer semantischen Interpretationsschicht der HyLOS Plattform, In: *DeLFI 2005 3. Deutsche e-Learning Fachtagung Informatik*, 2005.
- [EKR+03] **M. Engelhardt, A. Karpanti, T. Rack, I. Schmidt, T. C. Schmidt** : Hypermedia Learning Objects System -- On the Way to a Semantic Educational Web, In: *Proceedings of the International Workshop "Interactive Computer aided Learning" ICL 2003. Learning Objects and Reusability of Content*, 2003.
- [FER05] **Maciej Kuszpa, Ewald Scherm** : Mobile Learning - Modetrend oder wesentlicher Bestandteil lebenslangen Lernens?, In: *Diskussionsbeitrag des Fachbereichs Wirtschaftswissenschaft der FernUniversität in Hagen, Nr. 380*, 2005.
- [FLA] **Sony PSP & Flash Player 6: Supported Features**, *Online*:
<http://www.flashdevices.net/2006/04/sony-ppsp-flash-player-6-supported.html>,
Letzter Zugriff: 30.11.2006
- [FS04] **Dirk Frohberg, Gerhard Schwabe** : M-Learning - Kooperatives Lernen im Kontext, In: *WISU - das Wirtschaftsstudium*, (Heft 8/9, S. 1071-1077), 2004.
- [HEI] **XSL: XML-Formatierungssprache verabschiedet**, *Online*:
<http://www.heise.de/newsticker/meldung/21871>, *Letzter Zugriff*: 05.12.2006
- [ITW] **Stylesheet**, *Online*:
http://www.itwissen.info/definition/lexikon/__/stylesheet_stylesheet.html,
Letzter Zugriff: 23.01.2007
- [KNO] **M-learning in the C21st: benefits for learners**, *Online*:
<http://knowledgetree.flexiblelearning.net.au/edition06/html/prasimongeddes.html>, *Letzter Zugriff*: 23.01.2007
- [Kus05] **Maciej Kuszpa** : Mobile Learning - Studieren mit dem Handy, In: *Jahrbuch 2004 der Gesellschaft der Freunde der Fernuniversität (S. 59-71)*, 2005.
- [LIN] **M-Learning. Mobile, Wireless, In-Your-Pocket Learning.**, *Online*:
<http://www.linezine.com/2.1/features/cqmmwiyp.htm>, *Letzter Zugriff*:
 23.01.2007

- [LTS] **WG12: Learning Object Metadata**, *Online:* <http://ltsc.ieee.org/wg12/>, *Letzter Zugriff:* 11.12.2006
- [McL03] **Neil McLean** : The M-learning paradigm: An overview., In: *A report for the Royal Academy of Engineering and the Vodaphone group foundation.*, 2003.
- [MIR] **Einführung in das MIR-System**, *Online:* <http://www.rz.fhtw-berlin.de/content/Projekte/MIR>, *Letzter Zugriff:* 11.12.2006
- [MIRb] **MIR white paper**, *Online:* <http://www.rz.fhtw-berlin.de/content/Projekte/MIR/publications.html>, *Letzter Zugriff:* 19.12.2006
- [Nie07] **Stephan Niedermeier** : , In: *Cocoon 2 und Tomcat*, 2007. Galileo Computing
- [Noa02] **W. Noack** : , In: *XML 1.0 Grundlagen*, 2002. RRZN / Universität Hannover
- [PAL] **When e-learning becomes m-learning**, *Online:* <http://www.palmpowerenterprise.com/issues/issue200106/elearning002.html>, *Letzter Zugriff:* 23.01.2007
- [RM+02] **R. Reichwald, R. Meier, et al.** : Die mobile Ökonomie - Definition und Spezifika., In: *Mobile Kommunikation - Wertschöpfung, Technologien, neue Dienste (S. 3-16)*, 2002.
- [SCH] **PHP-FlashImageConverter**, *Online:* <http://www.schallschlucker.de/open-source/macromedia-flash/flash-image-proxy.html>, *Letzter Zugriff:* 08.02.2007
- [SEP] **SE|PY ActionScript Editor**, *Online:* <http://www.sephiroth.it/python/sepy.php>, *Letzter Zugriff:* 02.02.2007
- [Son06] **Sony Computing Entertainment Inc.** : , In: *Talkman - Produktbeschreibung*, 2006.
- [STA] **eLearning, E-Learning, Blended Learning**, *Online:* <http://www.stangl-taller.at/ARBEITSBLAETTER/LERNEN/Elearning.shtml>, *Letzter Zugriff:* 03.02.2007
- [Ste02] **Magnus Stein** : , In: *Workshop XML*, 2002. Addison Wesley
- [VS02] **Giasemi N. Vavoula, Mike Sharples** : KLeOS: A Personal, Mobile, Knowledge and Learning Organisation System, In: *IEEE International Workshop on Wireless and Mobile Technologies in Education (S. 152 - 156)*, 2002.
- [W3] **CSS & XSL**, *Online:* <http://www.w3.org/Style/CSS-vs-XSL.html>, *Letzter Zugriff:* 12.12.2006

- [W3b] **Extensible Markup Language (XML) 1.0 (Fourth Edition)**, *Online:*
<http://www.w3.org/TR/2006/REC-xml-20060816/>, *Letzter Zugriff:*
23.01.2007
- [XML] **Printing XML: Why CSS Is Better than XSL**, *Online:*
<http://www.xml.com/pub/a/2005/01/19/print.html>, *Letzter Zugriff:*
05.12.2006

ANHANG – A

ActionScript – Lösungsansatz 1

Im Folgenden sind sämtliche ActionScript der Anwendung aufgelistet, diesen befinden sich ebenfalls auf der beiliegenden CD unter prototyp/LA_1.

Datei: Main.as

```
1) var XMLcontentPath : String;
2) var XMLnavTreePath : String;
3)
4) //var x = 0;
5) var navElementNext : String;
6) var navElementNextChapter : String;
7) var pageNumber : Number;
8)
9) var host : String;
10) var contentType : String;
11) var learningObject : String;
12)
13) host = "http://www.f4.fhtw-berlin.de/~s0505485/test/LA_1/";
14) //host = "http://localhost/test/LA_1/";
15) contentTypeContent = "content/";
16) contentTypeNavTree = "navtree/";
17) learningObject = "ArbeiAntri";
18)
19) XMLnavTreePath = host + contentTypeNavTree + learningObject + ".xml";
20) var XMLcontent = new XML ();
21) var XMLnavTree = new XML ();
22) loadContentXML (XMLcontent);
23) loadNavTreeXML (XMLnavTree);
24) var chapterNav : Array = new Array ();
25) var navTree : Array = new Array ();
26) var layerNav : Array = new Array ();
27) var elementTitles : Array = new Array ();
28)
29) // Laden des XML-Dokumentes
30) function loadContentXML (XMLcontent : XML){
31) XMLcontentPath = host + contentTypeContent + learningObject + ".xml";
32) XMLcontent.ignoreWhite = true;
33) XMLcontent.load (XMLcontentPath);
34) XMLcontent.onLoad = function (success){
35)     if (success){
36)         this = XMLcontent.firstChild.lastChild;
37)         readContentData (this, 0);
38)     } else {
39)         trace ("Content XML nicht geladen");
```

```

40)     }
41) };
42) }
43) // Parsen des XML-Dokumentes das den Lernobjektinhalt enthält
44) function readContentData (node : XML, depth : Number){
45)     var child = node.firstChild;
46)     while (child){
47)         if (child.nodeType == 3 && child.parentNode.parentNode.nodeName
48)             != "richmedium" && child.parentNode.nodeName != "title"){
49)             showContent (child.nodeValue, i, child.ParentNode.nodeName);
50)             i ++;
51)         }
52)         if (child.nodeName == "image"){
53)             showContent (child.nodeName, i, child.attributes.src);
54)             i ++;
55)         }
56)         if (child.nodeName == "br"){
57)             if (child.nextSibling.nodeName == "br"){
58)                 child.nextSibling.removeNode ();
59)             }
60)             child.nodeValue = "&nbsp;";
61)             showContent (child.nodeValue, i, child.ParentNode.nodeName);
62)             i ++;
63)         }
64)         if (child.nodeName == "resource"){
65)             showContent (child,i,child.ParentNode.ParentNode.attributes.contentClass,
66)                 child.nextSibling.attributes.href);
67)             i ++;
68)         }
69)         if (child.parentNode.parentNode.nodeName == "richmedium"
70)             && child.parentNode.nodeName == "title"){
71)             showContent (child, i, child.parentNode.parentNode.nodeName,
72)                 child.parentNode.parentNode.attributes.src);
73)             i ++;
74)         }
75)         readContentData (child, depth + 1);
76)         child = child.nextSibling;
77)     }
78) }
79)
80) /* Laden des XML-Dokumentes mit allen Unterelementen der Root-Objektes
81) * für die Navigation durch die Lerneinheit */
82)
83) function loadNavTreeXML (XMLnavTree : XML){
84)     XMLnavTree.ignoreWhite = true;
85)     XMLnavTree.load (XMLnavTreePath);
86)     XMLnavTree.onLoad = function (success){
87)         if (success){
88)             this = readNavTreeData (this, 0);
89)             setChapter (layerNav);

```

```

90)         setNav (navTree, chapterNav, layerNav);
91)         setTableOfContent ();
92)     }
93)     else
94)     {
95)         trace ("Nav XML nicht geladen");
96)     }
97) };
98) }
99)
100) /* Auslesen aller Verknüpfungen (zu den Unterelementen)
101) *  aus dem der Navigations-XML */
102)
103) function readNavTreeData (xmlroot_ : XMLNode, depthNav : Number) : Array {
104)     var node : XMLNode = xmlroot_.firstChild;
105)     while (node) {
106)         // node enthält Attribut path
107)         if (node.attributes.path != undefined){
108)             var elementTitle = node.firstChild.firstChild.firstChild;
109)             elementTitles.push (elementTitle);
110)             var elementNav = node.attributes.path;
111)             navTree.push (elementNav);
112)             layerNav.push (depthNav);
113)         }
114)         readNavTreeData (node, depthNav + 1);
115)         node = node.nextSibling;
116)     }
117)     return navTree;
118) }

```

Datei: ContentFormat.as

```

1) sc.setStyle ("borderStyle", "none");
2) sc.setStyle ("themeColor", "0xD63529");
3) sc.contentPath = "content_mc";
4) _root.tocBG.sc_toc.setStyle ("borderStyle", "none");
5) _root.tocBG.sc_toc.setStyle ("themeColor", "0xD63529");
6) _root.tocBG.sc_toc.contentPath = "tocsc_mc";
7)
8) var textHeight : Number = 0;
9)
10) /* formatiert das aktuelle Element im XML-Dokument Formatierung durch
11) *  TextFormat innerhalb eines ScrollPane.MovieClip.TextField */
12)
13) sc.content.createEmptyMovieClip ("holder_mc", 1);
14) function showContent(node : XML, i : Number, parentname : XML, attribut : XML){
15)     var holder : MovieClip;
16)     holder = sc.content.holder_mc.createEmptyMovieClip ("con" + i, i + 1);
17)     holder.createTextField ("text_txt" + i, i, 0, textHeight, 390, 4);

```

```

18) holder ["text_txt" + i].autoSize = true;
19) holder ["text_txt" + i].wordWrap = true;
20) holder ["text_txt" + i].html = true;
21)
22) if (parentname != "resource"){
23)     holder ["text_txt" + i].htmlText = node;
24) }
25) formatText = new TextFormat ();
26) formatText.font = "arial";
27) formatText.color = 0x333333;
28) formatText.leading = 3;
29) formatText.leftMargin = 3;
30) formatText.rightMargin = 5;
31) formatText.size = 13;
32) holder ["text_txt" + i].setTextFormat (formatText);
33) if (parentname == "section"){
34)     formatText = new TextFormat ();
35)     formatText.color = 0x00adef;
36)     formatText.size = 14;
37)     formatText.leading = 10;
38)     formatText.bold = true;
39)     holder ["text_txt" + i].setTextFormat (formatText);
40) }
41) if (parentname == "subsection"){
42)     formatText = new TextFormat ();
43)     formatText.color = 0x00adef;
44)     formatText.bold = true;
45)     holder ["text_txt" + i].setTextFormat (formatText);
46) }
47) if (parentname == "emph"){
48)     formatText = new TextFormat ();
49)     formatText.italic = true;
50)     holder ["text_txt" + i].setTextFormat (formatText);
51) }
52) if (parentname == "strong"){
53)     formatText = new TextFormat ();
54)     formatText.bold = true;
55)     holder ["text_txt" + i].setTextFormat (formatText);
56) }
57) if (node.nodeName == "resource" && node.nextSibling.attributes.download
58)     == undefined && parentname == "outline"){
59)     formatText = new TextFormat ();
60)     formatText.font = "arial";
61)     formatText.color = 0x00adef;
62)     formatText.underline = true;
63)     formatText.bullet = true;
64)     holder ["text_txt" + i].htmlText = node.childNodes;
65)     holder ["text_txt" + i]._parent.onPress = function () {
66)         ermiddlePosElement (attribut);
67)         setNewChapter (attribut);

```

```

68)         setPageNumber (pos);
69)         setChapterTitle (pos);
70)     }
71)     holder ["text_txt" + i].setTextFormat (formatText);
72) }
73) if (parentname == "item"){
74)     formatText = new TextFormat ();
75)     formatText.bullet = true;
76)     holder ["text_txt" + i].setTextFormat (formatText);
77) }
78) if (node == "image"){
79)     formatText = new TextFormat ();
80)     formatText.color = 0x992233;
81)     formatText.url = + parentname;
82)     formatText.underline = true;
83)     formatText.italic = true;
84)     holder ["text_txt" + i]._parent.onPress = function ()
85)     {
86)         setImage (parentname);
87)         _root.imageBG._alpha = 100;
88)     }
89)     holder ["text_txt" + i].setTextFormat (formatText);
90) }
91) if (parentname == "richmedium")
92) {
93)     formatText = new TextFormat ();
94)     formatText.color = 0x992233;
95)     formatText.italic = true;
96)     holder ["text_txt" + i].setTextFormat (formatText);
97) }
98) textHeight = holder ["text_txt" + i].textHeight + textHeight;
99) //scrollbalken auf position 0;
100) sc.redraw ();
101) sc.invalidate ();
102) }
103) /* Anzeige der Bilder innerhalb eines, über der Benutzeroberfläche liegenden,
104) * MovieClips Skalierung, wenn diese größer als der PSP-Bildschirm */
105)
106) function setImage (imgUrl){
107)     _root.navi_mc.inhalt.enabled = false;
108)     _root.navi_mc.home.enabled = false;
109)     _root.navi_mc.previousItem.enabled = false;
110)     _root.navi_mc.previousChapter.enabled = false;
111)     _root.navi_mc.nextItem.enabled = false;
112)     _root.navi_mc.nextChapter.enabled = false;
113)     _root.attachMovie ("img_mc", "image_mc", 1);
114)     _root.image_mc._x = 0;
115)     _root.image_mc._y = 0;
116)     _root.image_mc.createEmptyMovieClip ("image", - 16400);
117)     _root.image_mc.image.loadMovie (imgUrl);

```

```

118)   this.onEnterFrame = function () {
119)       total = _root.image_mc.getBytesTotal ();
120)       geladen = _root.image_mc.getBytesLoaded ();
121)       prozent = int (geladen * 100 / total);
122)       if (prozent >= 100){
123)           delete this.onEnterFrame;
124)           this._visible = true;
125)           var verhaeltnis = 470 / _root.image_mc.image._width;
126)           if (_root.image_mc.image._width > 470 ||
127)               _root.image_mc.image._height > 255){
128)               if (_root.image_mc.image._width > _root.image_mc.image._height){
129)                   _root.image_mc.image._width = 460;
130)                   _root.image_mc.image._height
131)                       = _root.image_mc.image._height * verhaeltnis;
132)               }
133)               if (_root.image_mc.image._width < _root.image_mc.image._height){
134)                   _root.image_mc.image._width
135)                       = _root.image_mc.image._width * verhaeltnis;
136)                   _root.image_mc.image._height = 250;
137)               }
138)           }
139)           //image_mc._x = (480 - image_mc._width) / 2;
140)           _root.image_mc.image._y = 0
141)           //image_mc._y = (272 - image_mc._height)/2;
142)
143)       }
144)   };
145)   //Zuweisung der Tooltip-Funktionalität
146)   _root.image_mc.close_btn.onRollOver = function (){
147)       this._visible = true;
148)       cont = _root.image_mc;
149)       toolTipText = "Bild schließen";
150)       setToolTip (toolTipText, cont);
151)   }
152)   _root.image_mc.close_btn.onRollOut = function (){
153)       _root.image_mc.tooltip_mc.tooltip.removeTextField ();
154)   }
155)   //Button zum Schließen des MovieClips, in dem das Bild enthalten ist
156)   _root.image_mc.close_btn.onPress = function (){
157)       _root.navi_mc.inhalt.enabled = true;
158)       _root.navi_mc.home.enabled = true;
159)       _root.navi_mc.previousItem.enabled = true;
160)       _root.navi_mc.previousChapter.enabled = true;
161)       _root.navi_mc.nextItem.enabled = true;
162)       _root.navi_mc.nextChapter.enabled = true;
163)       _root.image_mc.image.unloadMovie ();
164)       _root.image_mc.tooltip_mc.tooltip.removeTextField ();
165)       _root.image_mc.close_btn._visible = false;
166)       _root.imageBG._alpha = 0;
167)   }

```

168) }

Datei: Navigation.as

```

1) var currentPageTitle : String;
2) _root.tocBG.close_btn._visible = false;
3)
4)
5) /* erstellt Array, das alle Elemente enthält, welche Unterelemente enthalten
6) * genutzt zur Kapitelanwahl über die Navigationstasten "nächstes Kapitel"
7) * und "vorhergehendes Kapitel" */
8)
9) function setChapter (layerNav) : Array {
10)  gesamtAnzahl = layerNav.length;
11)  var pos : Number = 0;
12)  while (pos <= gesamtAnzahl) {
13)    if (layerNav [pos] < layerNav [pos + 1] || layerNav [pos]
14)      == 1 && layerNav [pos + 1] == 1){
15)      chapterNav.push (pos);
16)    }
17)    pos ++;
18)  }
19)  return chapterNav;
20) }
21)
22) // belegt die Navigationstasten mit dem jeweils aktuellen Element
23)
24) function setNav (navTree, chapterNav, layerNav){
25)  var pos : Number = 0;
26)  _root.navi_mc.nextItem.ID = 1;
27)  _root.navi_mc.previousItem.ID = 2;
28)  _root.navi_mc.nextChapter.ID = 3;
29)  _root.navi_mc.previousChapter.ID = 4;
30)  _root.navi_mc.home.ID = 5;
31)  if (navTree.length > 0){
32)    _root.navi_mc.nextItem._alpha = 100;
33)    _root.navi_mc.nextChapter._alpha = 100;
34)    _root.navi_mc.previousItem.enabled = false;
35)    _root.navi_mc.previousChapter.enabled = false;
36)    _root.navi_mc.home._alpha = 50;
37)    _root.navi_mc.home.enabled = false;
38)    _root.navi_mc.nextItem.onRelease =
39)    _root.navi_mc.previousItem.onRelease =
40)    _root.navi_mc.nextChapter.onRelease =
41)    _root.navi_mc.previousChapter.onRelease =
42)    _root.navi_mc.home.onRelease = function (){
43)      var cl : Number = chapterNav.length;
44)      if (this.ID == 1 && pos >= 0){
45)        navElementNext = navTree [pos + 1];

```



```

46)         pos ++;
47)         if (pos == 1){
48)             _root.navi_mc.previousItem._alpha = 100;
49)             _root.navi_mc.previousItem.enabled = true;
50)             _root.navi_mc.home._alpha = 100;
51)             _root.navi_mc.home.enabled = true;
52)         }
53)         if (pos == navTree.length - 1){
54)             this._alpha = 50;
55)             this.enabled = false;
56)         }
57)         if (pos == 1){
58)             _root.navi_mc.previousItem._alpha = 100;
59)             _root.navi_mc.previousItem.enabled = true;
60)             _root.navi_mc.previousChapter._alpha = 100;
61)             _root.navi_mc.previousChapter.enabled = true;
62)             _root.navi_mc.home.enabled = true;
63)             _root.navi_mc.home._alpha = 100;
64)         }
65)         if (pos == chapterNav [cl - 1]){
66)             _root.navi_mc.nextChapter._alpha = 50;
67)             _root.navi_mc.nextChapter.enabled = false;
68)         }
69)     }
70)     if (this.ID == 2 && pos >= 0){
71)         navElementNext = navTree [pos - 1];
72)         pos --;
73)         if (pos == 0){
74)             this._alpha = 50;
75)             this.enabled = false;
76)             _root.navi_mc.previousChapter._alpha = 50;
77)             _root.navi_mc.previousChapter.enabled = false;
78)             _root.navi_mc.home.enabled = false;
79)             _root.navi_mc.home._alpha = 50;
80)         }
81)         if (pos == navTree.length - 2){
82)             _root.navi_mc.nextItem._alpha = 100;
83)             _root.navi_mc.nextItem.enabled = true;
84)         }
85)         if (pos == chapterNav [cl - 1] - 1){
86)             _root.navi_mc.nextChapter._alpha = 100;
87)             _root.navi_mc.nextChapter.enabled = true;
88)         }
89)     }
90)     if (this.ID == 3 && pos >= 0){
91)         var v : Number = 0;
92)         while (v <= cl - 2){
93)             if (chapterNav [v] <= pos && pos <= chapterNav [v + 1]){
94)                 if (pos == chapterNav [v + 1]){
95)                     pos = chapterNav [v + 2];

```

```
96)         }
97)         else{
98)             pos = chapterNav [v + 1];
99)         }
100)         if (pos == chapterNav [cl - 1]) {
101)             _root.navi_mc.nextChapter._alpha = 50;
102)             _root.navi_mc.nextChapter.enabled = false;
103)         }
104)         if (pos >= 1){
105)             _root.navi_mc.previousItem._alpha = 100;
106)             _root.navi_mc.previousItem.enabled = true;
107)             _root.navi_mc.previousChapter._alpha = 100;
108)             _root.navi_mc.previousChapter.enabled = true;
109)             _root.navi_mc.home._alpha = 100;
110)             _root.navi_mc.home.enabled = true;
111)         }
112)         if (pos == 0){
113)         }
114)         break;
115)     }
116)     v ++;
117) }
118)     navElementNext = navTree [pos];
119) }
120) if (this.ID == 4 && pos >= 0){
121)     var v : Number = 0;
122)     while (v <= navTree.length){
123)         if (pos >= chapterNav [v - 1] && pos <= chapterNav [v]){
124)             pos = chapterNav [v - 1];
125)             if (pos == 0){
126)                 this._alpha = 50;
127)                 this.enabled = false;
128)                 _root.navi_mc.previousItem._alpha = 50;
129)                 _root.navi_mc.previousItem.enabled = false;
130)                 _root.navi_mc.home.enabled = false;
131)                 _root.navi_mc.home._alpha = 50;
132)             }
133)             if (pos == 1){
134)                 _root.navi_mc.previousItem._alpha = 100;
135)                 _root.navi_mc.previousItem.enabled = true;
136)             }
137)             if (pos <= chapterNav [cl - 1]){
138)                 _root.navi_mc.nextChapter._alpha = 100;
139)                 _root.navi_mc.nextChapter.enabled = true;
140)                 _root.navi_mc.nextItem._alpha = 100;
141)                 _root.navi_mc.nextItem.enabled = true;
142)             }
143)             break;
144)         }
145)         if (pos > chapterNav [cl - 1]){
```

```

146)         pos = chapterNav [cl - 1];
147)         _root.navi_mc.nextItem._alpha = 100;
148)         _root.navi_mc.nextItem.enabled = true;
149)         break;
150)     }
151)     v ++;
152) }
153)     navElementNext = navTree [pos];
154) }
155) //Startseite - Button
156) if (this.ID == 5 && pos >= 0){
157)     if (pos == 0){
158)         _root.navi_mc.home.enabled = false;
159)         _root.navi_mc.home._alpha = 50;
160)         _root.navi_mc.previousChapter._alpha = 50;
161)         _root.navi_mc.previousChapter.enabled = false;
162)         _root.navi_mc.previousItem._alpha = 50;
163)         _root.navi_mc.previousItem.enabled = false;
164)     }
165)     else{
166)         pos = 0;
167)         navElementNext = "ArbeiAntri";
168)         _root.navi_mc.home.enabled = false;
169)         _root.navi_mc.home._alpha = 50;
170)         _root.navi_mc.previousChapter._alpha = 50;
171)         _root.navi_mc.previousChapter.enabled = false;
172)         _root.navi_mc.previousItem._alpha = 50;
173)         _root.navi_mc.previousItem.enabled = false;
174)         _root.navi_mc.nextItem._alpha = 100;
175)         _root.navi_mc.nextItem.enabled = true;
176)         _root.navi_mc.nextChapter._alpha = 100;
177)         _root.navi_mc.nextChapter.enabled = true;
178)     }
179) }
180)     currentPageTitle = elementTitles [pos];
181)     setPageNumber (pos);
182)     setChapterTitle (pos, currentPageTitle);
183)     setNewChapter (navElementNext);
184) }
185) }
186) else{
187)     _root.navi_mc.previousItem.enabled = false;
188)     _root.navi_mc.previousChapter.enabled = false;
189)     _root.navi_mc.home.enabled = false;
190) }
191) }
192) /* erzeugt Inhaltsverzeichnis mit allen einzelnen Kapiteln
193) * und Seiten der Lerneinheit */
194)
195) var hoehe_toc : Number = 0;

```

```

196) function setTableOfContent (){
197)     _root.tocBG.sc_toc.refresh ();
198)     _root.tocBG.sc_toc.redraw ();
199)     _root.navi_mc.inhalt.enabled = true;
200)     _root.navi_mc.inhalt.onRelease = function (){
201)         _root.tooltip_mc.tooltip.removeTextField ();
202)         _root.tocBG._alpha = 100;
203)         _root.navi_mc.inhalt.enabled = false;
204)         _root.navi_mc.home.enabled = false;
205)         _root.navi_mc.previousItem.enabled = false;
206)         _root.navi_mc.previousChapter.enabled = false;
207)         _root.navi_mc.nextItem.enabled = false;
208)         _root.navi_mc.nextChapter.enabled = false;
209)         _root.tocBG.sc_toc.content.toc_mc._visible = true;
210)         _root.tocBG.enabled = true;
211)         _root.tocBG.close_btn._visible = true;
212)         if (_root.tocBG.sc_toc.content.toc_mc == undefined){
213)             _root.tocBG.sc_toc.content.attachMovie ("toc", "toc_mc", 1);
214)             _root.tocBG.sc_toc.content.toc_mc._visible = true;
215)             for (var f = 0; f < elementTitles.length; f++){
216)                 _root.tocBG.sc_toc.content.toc_mc.createTextField ("toc_txt" + f,
217)                     f, 0, hoehe_toc, 340, 4);
218)                 _root.tocBG.sc_toc.content.toc_mc ["toc_txt" + f].autoSize = true;
219)                 _root.tocBG.sc_toc.content.toc_mc ["toc_txt" + f].wordWrap = true;
220)                 _root.tocBG.sc_toc.content.toc_mc ["toc_txt" + f].html = true;
221)                 _root.tocBG.sc_toc.content.toc_mc ["toc_txt" + f].htmlText
222)                 = elementTitles [f];
223)                 _root.tocBG.sc_toc.content.toc_mc ["toc_txt" + f]._y = hoehe_toc;
224)                 hoehe_toc += _root.tocBG.sc_toc.content.toc_mc ["toc_txt"
225)                 + f]._height;
226)                 if (layerNav [f] == 0){
227)                     formatText = new TextFormat ();
228)                     formatText.font = "arial";
229)                     formatText.color = 0x00adef;
230)                     formatText.bold = true;
231)                     formatText.underline = true;
232)                     formatText.leading = 5;
233)                     formatText.leftMargin = 3;
234)                     formatText.size = 12;
235)                     _root.tocBG.sc_toc.content.toc_mc ["toc_txt"
236)                     + f].setTextFormat (formatText);
237)                 }
238)                 if (layerNav [f] == 1){
239)                     formatText = new TextFormat ();
240)                     formatText.font = "arial";
241)                     formatText.color = 0x333333;
242)                     formatText.leading = 5;
243)                     formatText.underline = true;
244)                     formatText.bold = true;
245)                     formatText.leftMargin = 15;

```

```

246)         formatText.size = 12;
247)         _root.tocBG.sc_toc.content.toc_mc ["toc_txt"
248)         + f].setTextFormat (formatText);
249)     }
250)     if (layerNav [f] == 2){
251)         formatText = new TextFormat ();
252)         formatText.font = "arial";
253)         formatText.color = 0x333333;
254)         formatText.leading = 5;
255)         formatText.underline = true;
256)         formatText.leftMargin = 25;
257)         formatText.size = 10;
258)         _root.tocBG.sc_toc.content.toc_mc ["toc_txt"
259)         + f].setTextFormat (formatText);
260)     }
261)     if (layerNav [f] >= 3){
262)         formatText = new TextFormat ();
263)         formatText.font = "arial";
264)         formatText.color = 0x333333;
265)         formatText.leading = 5;
266)         formatText.leftMargin = 35;
267)         formatText.underline = true;
268)         formatText.size = 10;
269)         _root.tocBG.sc_toc.content.toc_mc ["toc_txt"
270)         + f].setTextFormat (formatText);
271)     }
272) }
273) }
274) _root.tocBG.sc_toc.invalidate ();
275) _root.tocBG.close_btn.onRollOver = function (){
276)     this._visible = true;
277)     cont = _root.tocBG;
278)     tooltipText = "Schließen";
279)     setToolTip (tooltipText, cont);
280) }
281) _root.tocBG.close_btn.onRollOut = function (){
282)     _root.tocBG.tooltip_mc.tooltip.removeTextField ();
283) }
284) _root.tocBG.close_btn.onPress = function (){
285)     _root.tocBG.tooltip_mc.tooltip.removeTextField ();
286)     _root.tocBG.close_btn._visible = false;
287)     _root.tocBG._alpha = 0;
288)     _root.tocBG.enabled = false;
289)     _root.navi_mc.inhalt.enabled = true;
290)     _root.navi_mc.home.enabled = true;
291)     _root.navi_mc.previousItem.enabled = true;
292)     _root.navi_mc.previousChapter.enabled = true;
293)     _root.navi_mc.nextItem.enabled = true;
294)     _root.navi_mc.nextChapter.enabled = true;
295)     _root.tocBG.sc_toc.content.toc_mc._visible = false;

```

```

296)     }
297)   }
298) }
299)
300) // erzeugt Kapiteltitel des aktuellen Lernobjektes
301)
302) function setChapterTitle (pos) {
303)   //textfield Kapiteltitel
304)   this.currentPageTitle = currentPageTitle;
305)   if (pos == 0){
306)     _root.chapterTitle_txt._visible = false;
307)   }
308)   else{
309)     _root.chapterTitle_txt._visible = true;
310)     _root.chapterTitle_txt.text = currentPageTitle;
311)   }
312)   var cNum : Number;
313)   var chapterNum : Array = new Array ();
314)   //Kapitelnummerierung
315)   for (cNum = 0; cNum < layerNav.length; cNum++){
316)     if (layerNav [cNum] == 1){
317)       chapterNum.push (cNum);
318)     }
319)   }
320)   var chapterLength : Number = chapterNum.length;
321)   for (cNum = 0; cNum < chapterNum.length; cNum++){
322)     if (pos >= chapterNum [cNum] && pos < chapterNum [cNum + 1]){
323)       cNum++;
324)       _root.chapter_txt._visible = true;
325)       _root.chapter_txt.text = "Kapitel " + cNum;
326)       break;
327)     }
328)     if (pos >= chapterNum [chapterLength - 1]){
329)       _root.chapter_txt._visible = true;
330)       _root.chapter_txt.text = "Kapitel " + chapterLength;
331)       break;
332)     }
333)     if (pos == 0){
334)       _root.chapter_txt._visible = false;
335)       break;
336)     }
337)   }
338) }
339)
340) // erzeugt Seitenzahl des aktuellen Lernobjektes
341)
342) function setPageNumber (pageNumber){
343)   if (pageNumber == 0){
344)     _root.page_txt.text = "Start";
345)   }

```

```

346)     else{
347)         var gesamtSeiten : Number = navTree.length;
348)         gesamtSeiten -= 1;
349)         _root.page_txt.text = pageNumber + "/" + gesamtSeiten;
350)     }
351) }
352)
353) // erzeugt Seiteninhalt des aktuellen Lernobjektes
354)
355) function setNewChapter (navElementNext){
356)     learningObject = navElementNext;
357)     var XMLcontent : XML = new XML (navElementNext);
358)     loadContentXML (XMLcontent);
359)     textHeight = 0;
360)     _root.sc.content.holder_mc.unloadMovie (con);
361) }
362) function ermittlePosElement (navElementNext) : Number {
363)     for (var s = 0; s < navTree.length; s ++){
364)         if (navTree [s] == navElementNext){
365)             pos = s;
366)             break;
367)         }
368)     }
369)     return pos;
370) }

```

Datei: Tooltipps.as

```

1) var onOrOff : Boolean = true;
2) _root.navi_mc.tooltip_on.onRelease = function (){
3)     _root.navi_mc.tooltip_on._visible = false;
4)     _root.navi_mc.tooltip_on._enabled = false;
5)     onOrOff = false;
6) }
7) //Button Ein/Aus
8) _root.navi_mc.tooltip_off.onRelease = function (){
9)     _root.navi_mc.tooltip_on._visible = true;
10)    _root.navi_mc.tooltip_on._enabled = true;
11)    onOrOff = true;
12) }
13)
14) function setToolTip (toolTipText, cont){
15)     if (onOrOff == true){
16)         cont.createEmptyMovieClip ("tooltip_mc", 1);
17)         if (_root._xmouse >= 400){
18)             xmouse = 395;
19)         }
20)     } else{
21)         xmouse = _root._xmouse;

```

```

22)     }
23)     cont.tooltip_mc.createTextField ("tooltip", 1, xmouse,
24)         _root._ymouse - 20 , 30, 30);
25)     cont.tooltip_mc.tooltip.autoSize = "left";
26)     cont.tooltip_mc.tooltip.borderColor = 0x666666;
27)     cont.tooltip_mc.tooltip.background = true;
28)     cont.tooltip_mc.tooltip.backgroundColor = 0xB3EAF;
29)     cont.tooltip_mc.tooltip.text = tooltipText;
30)     formatTooltip = new TextFormat ();
31)     formatTooltip.color = 0x333333;
32)     formatTooltip.size = 10;
33)     formatTooltip.font = "arial";
34)     formatTooltip.align = "center";
35)     cont.tooltip_mc.tooltip.setTextFormat (formatTooltip);
36) }
37) }
38) _root.navi_mc.tooltip_on.onRollOver = function (){
39)     cont = _root;
40)     tooltipText = "Tooltips aus";
41)     setToolTip (tooltipText, cont);
42) }
43) _root.navi_mc.home.onRollOver = function (){
44)     cont = _root;
45)     tooltipText = "Zur Startseite";
46)     setToolTip (tooltipText, cont);
47) }
48) _root.navi_mc.previousChapter.onRollOver = function (){
49)     cont = _root;
50)     tooltipText = "vorheriges Kapitel";
51)     setToolTip (tooltipText, cont);
52) }
53) _root.navi_mc.previousItem.onRollOver = function (){
54)     cont = _root;
55)     tooltipText = "vorherige Seite";
56)     setToolTip (tooltipText, cont);
57) }
58) _root.navi_mc.nextChapter.onRollOver = function (){
59)     cont = _root;
60)     tooltipText = "nächstes Kapitel";
61)     setToolTip (tooltipText, cont);
62) }
63) _root.navi_mc.nextItem.onRollOver = function (){
64)     cont = _root;
65)     tooltipText = "nächste Seite";
66)     setToolTip (tooltipText, cont);
67) }
68) _root.navi_mc.inhalt.onRollOver = function (){
69)     cont = _root;
70)     tooltipText = "Inhaltsverzeichnis";
71)     setToolTip (tooltipText, cont);

```



```
72) }
73) _root.navi_mc.inhalt.onRollOut =
74) _root.navi_mc.nextItem.onRollOut =
75) _root.navi_mc.nextChapter.onRollOut =
76) _root.navi_mc.previousChapter.onRollOut =
77) _root.navi_mc.home.onRollOut =
78) _root.navi_mc.previousItem.onRollOut =
79) _root.navi_mc.tooltip_on.onRollOut= function (){
80) _root.tooltip_mc.tooltip.removeTextField ();
81) }
```

ActionScript – Lösungsansatz 2

Im Folgenden sind sämtliche ActionScript der Anwendung aufgelistet, diesen befinden sich ebenfalls auf der beiliegenden CD unter prototyp/LA_2.

Datei: Main.as

```
1) var XMLcontentPath:String;
2) var XMLnavTreePath:String;
3) var navElementNext:String;
4) var navElementNextChapter:String;
5) var pageNumber:Number;
6) var host:String;
7) var contentType:String;
8) var learningObject:String;
9) host = "http://www.f4.fhtw-berlin.de/~s0505485/test/LA_2/";
10) //host = "http://localhost/test/LA_2/";
11) contentTypeContent = "content_new/";
12) contentTypeNavTree = "navtree/";
13) learningObject = "ArbeiAntri";
14) XMLnavTreePath = host+contentTypeNavTree+learningObject+".xml";
15) var XMLcontent = new XML();
16) var XMLnavTree = new XML();
17) loadContentXML(XMLcontent);
18) loadNavTreeXML(XMLnavTree);
19) var chapterNav:Array = new Array();
20) var navTree:Array = new Array();
21) var layerNav:Array = new Array();
22) var elementTitles:Array = new Array();
23)
24) // Laden des XML-Dokumentes
25)
26) function loadContentXML(XMLcontent:XML){
27)     XMLcontentPath = host+contentTypeContent+learningObject+".xml";
28)     XMLcontent.ignoreWhite = true;
29)     XMLcontent.load(XMLcontentPath);
30)     XMLcontent.onLoad = function(success){
31)         if (success) {
32)             this = XMLcontent.firstChild.firstChild;
33)             readContentData(this, 0);
34)             showContent(this);
35)         } else {
36)             trace("Content XML nicht geladen");
37)         }
38)     };
39) }
```

```

40) function link_click(imgPath:String) {
41)     setImage(imgPath);
42)     _root.imageBG._alpha = 100;
43) }
44)
45) // Parsen des XML-Dokumentes das den Lernobjektinhalt enthält
46)
47) function readContentData(node:XML, depth:Number) {
48)     var child = node.firstChild;
49)     while (child) {
50)         if (child.nodeName == "img") {
51)             var neu = new XML();
52)             neu.parseXML("<a href=\"\"asfunction:_root.link_click,\"
53)             +child.attributes.src+\"\"><font color=\"#992233\"size=\"13\"
54)             face=\"Arial\"><i>Bild bitte hier klicken</i></font></a>");
55)             child.parentNode.insertBefore(neu, child);
56)             child.removeNode();
57)         }
58)         i++;
59)         readContentData(child, depth+1);
60)         child = child.nextSibling;
61)     }
62) }
63)
64) /* Laden des XML-Dokumentes mit allen Unterelementen der Root-Objektes
65) * für die Navigation durch die Lerneinheit*/
66)
67) function loadNavTreeXML(XMLnavTree:XML) {
68)     XMLnavTree.ignoreWhite = true;
69)     XMLnavTree.load(XMLnavTreePath);
70)     XMLnavTree.onLoad = function(success) {
71)         if (success) {
72)             this = readNavTreeData(this, 0);
73)             setChapter(layerNav);
74)             setNav(navTree, chapterNav, layerNav);
75)             setTableOfContent();
76)         } else {
77)             trace("Nav XML nicht geladen");
78)         }
79)     };
80) }
81)
82) /* Auslesen aller Verknüpfungen (zu den Unterelementen) aus dem
83) * der Navigations-XML */
84)
85) function readNavTreeData(xmlroot_:XMLNode, depthNav:Number):Array {
86)     var node:XMLNode = xmlroot_.firstChild;
87)     while (node) {
88)         // node enthält Attribut path
89)         if (node.attributes.path != undefined) {

```

```

90)     var elementTitle = node.firstChild.firstChild.firstChild;
91)     elementTitles.push(elementTitle);
92)     var elementNav = node.attributes.path;
93)     navTree.push(elementNav);
94)     layerNav.push(depthNav);
95)   }
96)   readNavTreeData(node, depthNav+1);
97)   node = node.nextSibling;
98) }
99) return navTree;
100) }

```

Datei: ContentFormat.as

```

1) sc.setStyle ("borderStyle", "none");
2) sc.setStyle ("themeColor", "0xD63529");
3) sc.contentPath = "content_mc";
4) _root.tocBG.sc_toc.setStyle ("borderStyle", "none");
5) _root.tocBG.sc_toc.setStyle ("themeColor", "0xD63529");
6) _root.tocBG.sc_toc.contentPath = "tocsc_mc";
7) var textHeight : Number = 0;
8)
9) /* formatiert das aktuelle Element im XML-Dokument
10) * Formatierung durch TextFormat
11) * innerhalb eines ScrollPane.MovieClip.TextField */
12)
13) sc.content.createEmptyMovieClip ("holder_mc", 1);
14) function showContent (node : XML){
15)   var holder : MovieClip;
16)   holder = sc.content.holder_mc.createEmptyMovieClip ("con" + i, i + 1);
17)   holder.createTextField ("text_txt" + i, i, 0, textHeight, 390, 4);
18)   holder ["text_txt" + i].autoSize = true;
19)   holder ["text_txt" + i].wordWrap = true;
20)   holder ["text_txt" + i].html = true;
21)   holder ["text_txt" + i].htmlText = node;
22)   formatText = new TextFormat ();
23)   formatText.font = "arial";
24)   formatText.leading = 3;
25)   formatText.leftMargin = 10;
26)   formatText.rightMargin = 10;
27)   formatText.size = 13;
28)   holder ["text_txt" + i].setTextFormat (formatText);
29)   textHeight = holder ["text_txt" + i].textHeight + textHeight;
30)   //scrollbalken auf position 0;
31)   sc.redraw ();
32)   sc.invalidate ();
33) }
34)
35) /* Anzeige der Bilder innerhalb eines, über der Benutzeroberfläche liegenden

```

```

36) * MovieClips. Skalierung d. Bilder, wenn diese größer als der PSP-Bildschirm*/
37) function setImage (imgUrl){
38)  _root.navi_mc.inhalt.enabled = false;
39)  _root.navi_mc.home.enabled = false;
40)  _root.navi_mc.previousItem.enabled = false;
41)  _root.navi_mc.previousChapter.enabled = false;
42)  _root.navi_mc.nextItem.enabled = false;
43)  _root.navi_mc.nextChapter.enabled = false;
44)  _root.attachMovie ("img_mc", "image_mc", 1);
45)  _root.image_mc._x = 0;
46)  _root.image_mc._y = 0;
47)  _root.image_mc.createEmptyMovieClip ("image", - 16400);
48)  _root.image_mc.image.loadMovie (imgUrl);
49)  this.onEnterFrame = function (){
50)    total = _root.image_mc.getBytesTotal ();
51)    geladen = _root.image_mc.getBytesLoaded ();
52)    prozent = int (geladen * 100 / total);
53)    if (prozent >= 100){
54)      delete this.onEnterFrame;
55)      this._visible = true;
56)      var verhaeltnis = 470 / _root.image_mc.image._width;
57)      if (_root.image_mc.image._width > 470
58)        || _root.image_mc.image._height > 255) {
59)        if (_root.image_mc.image._width > _root.image_mc.image._height){
60)          _root.image_mc.image._width = 460;
61)          _root.image_mc.image._height
62)            = _root.image_mc.image._height * verhaeltnis;
63)        }
64)        if (_root.image_mc.image._width < _root.image_mc.image._height){
65)          _root.image_mc.image._width
66)            = _root.image_mc.image._width * verhaeltnis;
67)          _root.image_mc.image._height = 250;
68)        }
69)      }
70)      //image_mc._x = (480 - image_mc._width) / 2;
71)      _root.image_mc.image._y = 0
72)      //image_mc._y = (272 - image_mc._height)/2;
73)    }
74)  };
75) //Zuweisung der Tooltip-Funktionalität
76) _root.image_mc.close_btn.onRollOver = function (){
77)   this._visible = true;
78)   cont = _root.image_mc;
79)   tooltipText = "Bild schließen";
80)   setToolTip (tooltipText, cont);
81) }
82) _root.image_mc.close_btn.onRollOut = function (){
83)   _root.image_mc.tooltip_mc.tooltip.removeTextField ();
84) }
85) //Button zum Schließen des MovieClips, in dem das Bild enthalten ist

```

```
86) _root.image_mc.close_btn.onPress = function (){
87)     _root.navi_mc.inhalt.enabled = true;
88)     _root.navi_mc.home.enabled = true;
89)     _root.navi_mc.previousItem.enabled = true;
90)     _root.navi_mc.previousChapter.enabled = true;
91)     _root.navi_mc.nextItem.enabled = true;
92)     _root.navi_mc.nextChapter.enabled = true;
93)     _root.image_mc.image.unloadMovie ();
94)     _root.image_mc.tooltip_mc.tooltip.removeTextField ();
95)     _root.image_mc.close_btn._visible = false;
96)     _root.imageBG._alpha = 0;
97) }
98) }
```

Datei: Navigation.as

```
1) // gleicher Programmcode wie LA_1 / Navigation.as siehe S. 112
```

Datei: ToolTipps.as

```
1) // gleicher Programmcode wie LA_1 / ToolTipps.as siehe S. 119
```

ANHANG – B

Beispieldatei für die hierarchische Navigation:

Hinweis: Der folgende Code stellt beispielhaft eine Baum-Navigation-XML dar, dabei sind die Metadaten lediglich beim ersten Element (Zeile 4 – 34) aufgelistet. Die Metadaten sind in der Originaldatei für jedes lo-Element vorhanden, werden jedoch hier aus nur auszugsweise dargestellt. Die vollständige Datei befindet sich auf der beiliegenden CD unter Beispieldokumente/.

Datei: ArbeiAntri.xml

```
1) <?xml version="1.0" encoding="UTF-8"?>
2) <lo xmlns:hylos="http://hylos.fhtw-berlin.de" path="ArbeiAntri"
3)   lastmod="2006-11-22T20:05+0100">
4)   <general>
5)     <title>Start</title>
6)     <language>de</language>
7)     <description>
8)       Die folgende Lerneinheit ist während eines Interdisziplinären Projektes
9)       im Sommersemester 2006 an der FHTW Berlin entstanden. Studenten des
10)      Wirtschaftsingenieurwesens erarbeiteten gemeinsam mit den Studenten der
11)      Internationalen Medieninformatik ein Ko
12)    </description>
13)    <coverage>Einleitung </coverage>
14)    <structure>Parceled</structure>
15)    <aggregationLevel>1</aggregationLevel>
16)  </general>
17)  <educational>
18)    <interactivityType>Undefined</interactivityType>
19)    <learningResourceType>guide</learningResourceType>
20)    <interactivityLevel>low</interactivityLevel>
21)    <semanticDensity>veryLow</semanticDensity>
22)    <intendedEndUserRole>Learner</intendedEndUserRole>
23)    <context>TechnicalSchoolFirstCycle</context>
24)    <typicalAgeRange>20_30</typicalAgeRange>
25)    <difficulty>veryEasy</difficulty>
26)    <typicalLearningTime>0</typicalLearningTime>
27)    <description/>
28)    <language/>
29)  </educational>
30)  <rights>
31)    <cost>yes</cost>
32)    <copyrightAndOtherRestrictions>yes</copyrightAndOtherRestrictions>
33)    <description>FHTW, 2006</description>
34) </rights>
35) <lo path="GlieD" lastmod="2006-10-12T16:49+0200"></lo>
36) <lo path="Antri" lastmod="2006-11-22T20:00+0100">
37)   <lo path="DasSchne" lastmod="2006-11-22T20:01+0100"></lo>
38)   <lo path="AufbaSchne" lastmod="2006-10-13T16:18+0200"></lo>
39)   <lo path="LagerSchmi" lastmod="2006-07-12T00:13+0200"></lo>
```



```

40) <lo path="FormederAn" lastmod="2006-11-26T14:05+0100"></lo>
41) <lo path="FormeAntriII" lastmod="2006-11-26T18:40+0100"></lo>
42) </lo>
43) <lo path="DieFerti" lastmod="2006-11-30T17:24+0100">
44) <lo path="Rohma" lastmod="2006-11-22T13:33+0100"></lo>
45) <lo path="Ferti" lastmod="2006-11-30T17:55+0100"></lo>
46) <lo path="EinleFertiFilme" lastmod="2006-11-22T13:51+0100"></lo>
47) <lo path="Saege" lastmod="2006-07-12T21:23+0200">
48) <lo path="Rohma" lastmod="2006-11-22T13:33+0100">
49) <lo path="Ferti" lastmod="2006-11-30T17:55+0100"></lo>
50) </lo>
51) </lo>
52) <lo path="Pland" lastmod="2006-07-12T21:23+0200"></lo>
53) <lo path="Zentr" lastmod="2006-07-12T21:23+0200"></lo>
54) <lo path="DreheSchup" lastmod="2006-10-12T14:59+0200"></lo>
55) <lo path="Schli" lastmod="2006-11-30T17:24+0100"></lo>
56) <lo path="Umspa" lastmod="2006-07-12T21:23+0200"></lo>
57) <lo path="Fraes" lastmod="2006-07-12T21:23+0200"></lo>
58) <lo path="Fertil" lastmod="2006-10-12T14:57+0200"></lo>
59) <lo path="Ferti2" lastmod="2006-10-12T15:01+0200"></lo>
60) </lo>
61) <lo path="DerArbei" lastmod="2006-11-22T19:53+0100">
62) <lo path="Arbei" lastmod="2006-11-30T17:23+0100"></lo>
63) <lo path="GrobpFeinp" lastmod="2006-11-30T17:24+0100"></lo>
64) <lo path="ZwangArbei" lastmod="2006-11-22T14:12+0100"></lo>
65) </lo>
66) </lo>

```

Beispieldatei für eine Inhalt-XML-Datei in hylOs für Lösungsansatz 1:

Hinweis: Der folgende Code stellt beispielhaft eine Inhalt-XML-Datei aus Lösungsansatz 1 dar, dabei sind die Metadaten komplett ausgelassen, da diese hier nicht von Bedeutung sind. Die Metadaten sind in der Originaldatei vorhanden. Die vollständige Datei befindet sich auf der beiliegenden CD unter prototyp/LA_1/content/.

Datei: Antri.xml

```
67) <?xml version="1.0" encoding="UTF-8"?>
68) <lo xmlns:hylos="http://hylos.fhtw-berlin.de" path="Antri"
69)     lastmod="2006-11-22T20:00+0100">
70)   <content>
71)     <section>Die Antriebswelle</section>
72)     <paragraph>In diesem Kapitel lernen Sie die Antriebswelle kennen.Dabei
73)       werden Sie insbesondere das Schneckengetriebe genauer kennenlernen.
74)     <br/>Sie erlernen das theoretische Wissen zum Aufbau eines
75)     Schneckengetriebes. Außerdem werden Ihnen die verschiedenen Formelemente
76)     von Antriebswellen <br/>erklärt. <br/>
77)     <br/>In einer Teilübung in der wilo-Praxisanwendung können sie ihr
78)     Wissen anschließend überprüfen.
79)   </paragraph>
80)   <paragraph contentClass="outline">In diesen Kapitel erfahren sie mehr über:
81)     <br/>
82)     <link xmlns:mir="http://www.rz.fhtw-berlin.de/MIR" type="text">
83)       <resource xmlns:xlink="http://www.w3.org/1999/xlink"
84)         id="Das_Schneckengetriebe" type="resource">Das Schneckengetriebe
85)       </resource>
86)       <locator xmlns:xlink="http://www.w3.org/1999/xlink"
87)         targetType="GenericClass.GenericObject.GenericMob.GenericHob
88)           .GenericLob.LearningObject" href="AufbaSchne" title="Verweis zum
89)           Schneckengetriebe" type="locator"/>
90)     </link>
91)     <br/>
92)     <link xmlns:mir="http://www.rz.fhtw-berlin.de/MIR" type="text">
93)       <resource xmlns:xlink="http://www.w3.org/1999/xlink"
94)         id="Aufbau_des_Schn_2" type="resource">Aufbau des
95)         Schneckengetriebes</resource>
96)       <locator xmlns:xlink="http://www.w3.org/1999/xlink"
97)         targetType="GenericClass.GenericObject.GenericMob
98)           .GenericHob.GenericLob.LearningObject" href="AufbaSchne"
99)         title="Verweis zum Aufbau des Schneckengetriebes" type="locator"/>
100)     </link>
101)     <br/>
102)     <link xmlns:mir="http://www.rz.fhtw-berlin.de/MIR" type="text">
103)       <resource xmlns:xlink="http://www.w3.org/1999/xlink"
104)         id="Lagerung_und_Sc" type="resource">Lagerung und Schmierung
105)       </resource>
```

```

106)         <locator xmlns:xlink="http://www.w3.org/1999/xlink"
107)             targetType="GenericClass.GenericObject.GenericMob
108)             .GenericHob.GenericLob.LearningObject" href="LagerSchmi"
109)             title="Verweis zur Lagerung und Schmierung" type="locator"/>
110)     </link>
111)     <br/>
112)     <link xmlns:mir="http://www.rz.fhtw-berlin.de/MIR" type="text">
113)         <resource xmlns:xlink="http://www.w3.org/1999/xlink"
114)             id="Formelemente_de" type="resource">Formelemente der
115)             Antriebswelle I
116)         </resource>
117)         <locator xmlns:xlink="http://www.w3.org/1999/xlink"
118)             targetType="GenericClass.GenericObject.GenericMob
119)             .GenericHob.GenericLob.LearningObject" href="FormederAn"
120)             title="Verweis zu den Formelementen der Antriebswelle I "
121)             type="locator"/>
122)     </link>
123)     <br/>
124)     <link xmlns:mir="http://www.rz.fhtw-berlin.de/MIR" type="text">
125)         <resource xmlns:xlink="http://www.w3.org/1999/xlink"
126)             id="Formelemente_de_2" type="resource">Formelemente der
127)             Antriebswelle II
128)         </resource>
129)         <locator xmlns:xlink="http://www.w3.org/1999/xlink"
130)             targetType="GenericClass.GenericObject.GenericMob
131)             .GenericHob.GenericLob.LearningObject" href="FormeAntriII"
132)             title="Verweis zu den Formelementen der Antriebswelle II"
133)             type="locator"/>
134)     </link>
135) </paragraph>
136) <paragraph contentClass="learningObjective">Wenn sie dieses Kapitel
137)     bearbeitet haben wissen Sie:
138)     <itemize>
139)         <item>
140)             was ein Schneckengetriebe ist und wie es aufgebaut ist,
141)         </item>
142)         <item>
143)             welche Formelemente man bei Antriebswellen unterscheidet.
144)         </item>
145)     </itemize>
146) </paragraph>
147) <paragraph contentClass="learningTime">
148)     Für das Durcharbeiten dieser Lerneinheit benötigen Sie etwa 1 Stunde.
149) </paragraph>
150) </content>
151) </lo>

```

Beispieldatei für eine Inhalt-XML-Datei in hylOs für Lösungsansatz 2:

Hinweis: Der folgende Code stellt beispielhaft eine Inhalt-XML-Datei aus Lösungsansatz 2 dar, dabei sind die Metadaten komplett ausgelassen, da diese hier nicht von Bedeutung sind. Die Metadaten sind in der Originaldatei vorhanden. Die vollständige Datei befindet sich auf der beiliegenden CD unter `prototyp/LA_2/content/`.

Datei: Antri.xml

```
1) <?xml version="1.0" encoding="UTF-8"?>
2) <lo title="Die Antriebswelle"
3)   path="/HyLOs/content/Data/wilo/ArbeiAntri/Antri/Antri">
4)   <content>
5)     <p><font color="#00adef" size="14" face="Arial">
6)       <b>Die Antriebswelle</b>
7)     </font>
8)   </p>
9)   <p><font color="#333333" size="13" face="Arial">
10)    In diesem Kapitel lernen Sie die Antriebswelle kennen. Dabei werden Sie
11)    insbesondere das Schneckengetriebe genauer kennenlernen. <br/>Sie erlernen
12)    das theoretische Wissen zum Aufbau eines Schneckengetriebes. Außerdem werden
13)    Ihnen die verschiedenen Formelemente von Antriebswellen <br/>erklärt. <br/>
14)    <br/>In einer Teilübung in der wilo-Praxisanwendung können sie ihr Wissen
15)    anschließend überprüfen.
16)   </font>
17) <br/>
18) </p>
19) <p><font color="#333333" size="13" face="Arial">
20) In diesen Kapitel erfahren sie mehr über: <br/>
21)   <u><i><font color="#992233">
22)     <a>Das Schneckengetriebe </a>
23)   </font>
24)   </i>
25) </u>
26) <br/>
27) <u><i><font color="#992233">
28)   <a>Aufbau des Schneckengetriebes</a>
29) </font>
30) </i>
31) </u>
32) <br/>
33) <u><i><font color="#992233">
34)   <a>Lagerung und Schmierung </a>
35) </font>
36) </i>
37) </u>
38) <br/>
39) <u><i><font color="#992233">
```

```

40)           <a>Formelemente der Antriebswelle I </a>
41)           </font>
42)           </i>
43)           </u>
44)           <br/>
45)           <u><i><font color="#992233">
46)               <a>Formelemente der Antriebswelle II</a>
47)               </font>
48)               </i>
49)           </u>
50)           </font>
51)           <br/>
52)           </p>
53)           <p><font color="#333333" size="13" face="Arial">
54)           Wenn sie dieses Kapitel bearbeitet haben wissen Sie: <br/>• was ein
55)           Schneckengetriebe ist und wie es aufgebaut ist,<br/>• welche Formelemente
56)           man bei Antriebswellen unterscheidet.<br/>
57)           </font>
58)           <br/>
59)           </p>
60)           <p><font color="#333333" size="13" face="Arial">
61)           Für das Durcharbeiten dieser Lerneinheit benötigen Sie etwa 1 Stunde.</font>
62)           <br/>
63)           </p>
64)           </content>
65)           </lo>

```

ANHANG – C

XSLT-Stylesheet

Hinweis: Der folgende Code bildet das erstellte XSLT-Stylesheet zur Transformation der XML-Dokumente vom XML-Dokumententyp `content` ab. Das Dokument befindet sich ebenso auf der beiliegenden CD unter `prototyp/LA_2/`.

Datei: `hylos-content-psp.xsl`

```
1) <?xml version="1.0" encoding="UTF-8"?>
2) <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3)   xmlns:mir="http://www.rz.fhtw-berlin.de/MIR"
4)   xmlns:xlink="http://www.w3.org/1999/xlink"
5)   xmlns:hylos="http://hylos.fhtw-berlin.de"
6)   exclude-result-prefixes="mir xlink hylos">
7)   <!-- strip-space löscht Whitespaces -->
8)   <xsl:strip-space elements="itemize/item or list/item or enumeration/item"/>
9)   <xsl:template match="/hylos:lo">
10)     <xsl:element name="lo">
11)       <xsl:attribute name="title">
12)         <xsl:value-of select="hylos:general/hylos:title"/>
13)       </xsl:attribute>
14)       <xsl:attribute name="path">
15)         <xsl:value-of select="@hylos:path"/>
16)       </xsl:attribute>
17)       <xsl:element name="content">
18)         <xsl:apply-templates select="hylos:content"/>
19)       </xsl:element>
20)     </xsl:element>
21) </xsl:template>
22) <xsl:template match="hylos:content">
23)   <xsl:apply-templates/>
24) </xsl:template>
25) <!-- Template für SECTION -->
26) <xsl:template match="hylos:section">
27)   <p>
28)     <font color="#00adef" size="14" face="Arial">
29)       <b>
30)         <xsl:value-of select="."/>
31)       </b>
32)     </font>
33)   </p>
34) </xsl:template>
35) <!-- Template für SUBSECTION -->
36) <xsl:template match="hylos:subsection">
37)   <p>
38)     <font color="#00adef" size="13" face="Arial">
39)       <b>
40)         <xsl:value-of select="."/>
```

```

41)         </b>
42)         </font>
43)     </p>
44) </xsl:template>
45) <!-- Template für PARAGRAPH -->
46) <xsl:template match="paragraph">
47)     <p>
48)         <font color="#333333" size="13" face="Arial">
49)             <xsl:apply-templates/>
50)         </font>
51)         <xsl:if test="not(following-sibling::*[position() = 1 and self::table])">
52)             <br/>
53)         </xsl:if>
54)     </p>
55) </xsl:template>
56) <!-- Template für Zeileumbrüche -->
57) <xsl:template match="br">
58)     <br/>
59) </xsl:template>
60) <!-- Template für kursive Hervorhebung innerhalb des Textes -->
61) <xsl:template match="emph">
62)     <i>
63)         <xsl:apply-templates/>
64)     </i>
65) </xsl:template>
66) <!-- Template für fettgedruckte Hervorhebung innerhalb des Textes -->
67) <xsl:template match="strong">
68)     <b>
69)         <xsl:apply-templates/>
70)     </b>
71) </xsl:template>
72) <!-- Template für Bilder -->
73) <xsl:template match="image">
74)     <img align="left" border="0">
75)         <xsl:attribute name="src">
76)             <xsl:value-of select="@src"/>
77)         </xsl:attribute>
78)     </img>
79)     <br/>
80)     <br/>
81) </xsl:template>
82) <!-- Template für Links -->
83) <xsl:template match="mir:link">
84)     <u>
85)         <i>
86)             <font color="#992233">
87)                 <a>
88)                     <!--
89)                         <xsl:attribute name="href"/>
90)                     <xsl:value-of select="mir:locator/@xlink:href"/>

```



```

91)             </xsl:attribute>-->
92)             <xsl:apply-templates/>
93)             </a>
94)             </font>
95)             </i>
96)         </u>
97) </xsl:template>
98) <!-- Template für Listen -->
99) <xsl:template match="itemize">
100)     <br/>
101)     <xsl:for-each select="item">
102)         <xsl:text>&#8226;&#160;</xsl:text>
103)         <xsl:apply-templates/>
104)     <br/>
105)     </xsl:for-each>
106) </xsl:template>
107) <!-- Template für richmedium-Elemente -->
108) <xsl:template match="richmedium">
109)     <p>
110)         <font color="#992233" size="13" face="Arial">
111)             <xsl:value-of select="@src"/>
112)         </font>
113)     <br/>
114)     </p>
115) </xsl:template>
116) <!-- Template für nummerierte Listen -->
117) <xsl:template match="enumeration">
118)     <xsl:for-each select="item">
119)         <xsl:number count="item"/>
120)         <xsl:text>.&#160;</xsl:text>
121)         <xsl:apply-templates/>
122)     <br/>
123)     <br/>
124)     </xsl:for-each>
125) </xsl:template>
126) <!-- Ausblenden des theorem-Elementes -->
127) <xsl:template match="theorem"/>
128) <!-- Ausblenden des theorem-Elementes -->
129) <xsl:template match="table"/>
130) </xsl:stylesheet>

```

Hinweise zur beiliegenden CD

Auf der beiliegenden CD ist die erstellte Anwendung mit den zugehörigen ActionScript-Dateien beider Lösungsansätze sowie XML-Dokumente eines Beispielkurses zu finden. Außerdem sind alle Online-Quellen sowie diese Diplomarbeit als PDF beigefügt.

Steht zum Testen der Applikation eine PlayStation Portable zur Verfügung. Dann sollten folgende Schritte vorbereitet werden:

Zunächst die Memory-Card der PlayStation Portable an einen PC anschließen, danach den Ordner `prototyp_psp` auf die Memory-Card kopieren. Diese nach Abschluß des Kopierens wieder in die PlayStation Portable einlegen. Anschließend den Browser der PSP öffnen und folgenden Pfad in die Adressleiste eingeben: `file://prototyp_psp/LA_1/psp.swf` (für Lösungsansatz 1) oder `file://prototyp_psp/LA_2/psp.swf` (für Lösungsansatz 2)

Falls zum Testen der Applikation keine PlayStation Portable zur Verfügung steht, kann die Anwendung innerhalb eines normalen Webbrowsers getestet werden. Dazu wird empfohlen Mozilla Firefox zu verwenden, da die Applikation beim Testen dort fehlerfrei gearbeitet hat. Zum Testen in Mozilla sollte `prototyp/LA_1/psp.html` bzw. `prototyp/LA_2/psp.html` von der CD aufgerufen werden.

<i>Verzeichnisse auf der CD</i>	<i>Beschreibung der Inhalte</i>
onlinequellen	Online-Quellen dieser Arbeit als PDF
prototyp	Sämtliche Dateien der beiden Lösungsansätze zum Testen im Webbrowser Mozilla Firefox
prototyp_psp	Sämtliche Dateien der beiden Lösungsansätze zum Testen auf der PlayStation Portable
diplomarbeit_bartel.pdf	Diplomarbeit als PDF
readme.txt	Informationen zur CD und zum Testen der Anwendung

Inhaltsverzeichnis der CD

EIGENHÄNDIGKEITSERKLÄRUNG

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel verfasst habe. Die Arbeit wurde bisher nicht veröffentlicht und keiner anderen Prüfungsbehörde in ähnlicher oder gleicher Form vorgelegt.

Berlin, 12.02.2007

Patricia Bartel