

Hochschule für Technik und Wirtschaft(FH)
Fachbereich: Informatik/Mathematik

Diplomarbeit

im Studiengang Medieninformatik

Thema: Design und Entwicklung eines Lernmoduls für das Bildungsportal Sachsen. Untersuchung des Scorm 1.3 – Navigation Model.

Eingereicht von: Jana Krüger

Eingereicht am: 4. Juli 2007

Betreuer: Prof. Dr. Teresa Merino, Hochschule für Technik und Wirtschaft Dresden

Inhaltsverzeichnis

Einleitung	3
1 Lernen am Computer	5
1.1 Grundlegende Begriffe.....	5
1.1.1 E-Learning.....	5
1.1.2 LMS.....	5
1.1.3 Standardisierungen.....	6
1.1.4 Standard/Norm	6
1.1.4.1 Standard.....	7
1.1.4.2 Norm.....	7
1.2 CBT.....	7
1.3 WBT	8
1.4 Blended Learning.....	8
2 Projektentwicklung	9
2.1 Vorgehensweise bei der Entwicklung eines Lernmoduls	9
2.2 Konzeption des Lernmoduls.....	10
2.2.1 Das Grobkonzept.....	11
2.2.2 Das Feinkonzept	11
2.3 Gestaltung der Benutzeroberfläche.....	13
2.4 Evaluation des Lernmoduls	14
2.5 Entwicklung des Lernmoduls	15
2.5.1 Flash MX 2004 als Autorenwerkzeug.....	15
2.5.1.1 Begriffsklärung Autorenwerkzeug.....	15
2.5.1.2 Auswahlkriterien bei der Entscheidung für ein Autorenwerkzeug	16
2.5.1.3 Überblick über das Autorenwerkzeug Flash MX 2004.....	16
2.5.2 Entwicklung des Lernmoduls mit Flash MX 2004.....	18
2.5.2.1 Grundlagen der Entwicklung.....	18
2.5.2.2 Klassenstruktur nach Rülke	18
2.5.2.3 Die Generelle Struktur des Lernmoduls	23
2.5.2.4 Anpassung des Lernmoduls.....	25
2.5.2.5 Die wichtigsten Funktionalitäten	31
2.5.2.6 Verwendung von SCORM 1.2.....	35
2.5.3 Technischer Test	36
3. SCORM	39
3.1 Überblick.....	39
3.2 Overview.....	41
3.3 Content Aggregation Model (CAM).....	41
3.3.1 Content Model.....	42

3.3.2 Content Packaging	44
3.3.3 Metadaten	45
3.3.4 Sequencing and Navigation	46
3.4 Run-Time Environment (RTE)	46
3.4.1 Launch.....	47
3.4.2 API – Application Programming Interface	47
3.4.3 Datenmodell	47
3.5 Sequencing and Navigation (SN).....	48
3.5.1 Sequencing Konzept und Terminologie.....	50
3.5.1.1 Activity	50
3.5.1.2 Activity Tree	52
3.5.1.3 Objectives	53
3.5.2 Sequencing Definition Model	53
3.5.2.1 Sequencing Control Modes	54
3.5.2.2 Sequencing Rules	55
3.5.2.3 Rollup Rules.....	55
3.5.3 Sequencing Behavior Model.....	55
3.5.3.1 Tracking Model.....	55
3.5.3.2 Activity State Model	56
3.5.3.3 Sequencing Definition Model	57
4. Verwendung des Navigation-Models.....	58
4.1 Theoretische Betrachtung des Navigation-Models	58
4.1.1 Einführung	59
4.1.1.1 Abarbeiten des Navigation Requests.....	61
4.1.1.2 Beenden eines Content Objects	62
4.1.1.3 Navigation und Hilfsmittel	63
4.1.2 UI-Devices für die Navigation	63
4.1.2.1 UI-Devices für die Navigation vorbereiten	63
4.1.2.2 Nutzung des isVisible Attributes	64
4.1.2.3 Presentation Information Model	65
4.1.2.4 Kommunikation zwischen SCO und LMS bei einem Navigation Request ..	66
4.1.2.5 Run-Time Navigation Data Model	68
4.2 Ideen zur Optimierung des Lernmoduls.....	71
4.2.1 Weiter- bzw. Zurück Button	71
4.2.2 Link innerhalb eines Textes.....	71
4.2.3 Veränderung der rechten Navigation	71
4.3 Prototypische Umsetzung der Ideen	72
4.3.1 Vorbetrachtungen.....	72
4.3.1.1 Reload Editor	72

4.3.1.2 ADL Sample Runtime Environment.....	73
4.3.2 Beschreibung des Lernmodulauszugs.....	75
4.3.3 Beispielumsetzung: Allgemeines	77
4.3.3.1 Kursstruktur	77
4.3.3.2 Sequencing.....	78
4.3.4 Beispielumsetzung: Weiter-Button	79
4.3.5 Beispielumsetzung: Verlinkung	81
4.3.6 Beispielumsetzung: Rechte Navigation.....	81
4.4 Fazit	82
Abbildungsverzeichnis.....	84
Tabellenverzeichnis	86
Quellcodeverzeichnis.....	87
Literaturverzeichnis	88
Anhang A.....	89
Anhang B.....	96
Anhang C.....	97
Anhang D.....	100
Anhang E.....	102
Selbständigkeitserklärung.....	105

Einleitung

In der heutigen Zeit müssen wir eine ungeheure Menge an Wissen vermitteln, da versteht es sich, dass Experten auf der Suche nach effektiven Methoden sind.

[Ris02 S. 51]

Ein guter Ansatzpunkt, um die Vermittlung von Wissen voran zu treiben, ist die Nutzung elektronischer Medien. Schon in den sechziger Jahren gab es erste Entwicklungen in dieser Richtung. Die Erwartungen an die neuen Technologien waren hoch. Einige glaubten Hochschulveranstaltungen durch selbstständiges virtuelles Lernen ersetzen zu können. Doch auch heute findet E-Learning nicht den richtigen Zugang zur Lehre, die Einsatzgebiete sind marginal. In den neunziger Jahren gab es einen Entwicklungsaufschwung der neuen Technologien. In diesem Zuge haben Bund und Länder begonnen, Projekte zu unterstützen, die den Einsatz neuer Medien fördern. Ein solches Projekt war und ist das Bildungsportal Sachsen. In der Euphorie der neuen Möglichkeiten wurde allerdings nicht beachtet, dass es nicht möglich ist, Lehrende komplett zu ersetzen. Projekte, die darauf abzielen, scheitern.

Der neue Weg ist die Kombination von Lehrendem und neuer Technologie. Das Bildungsportal Sachsen geht diesen Weg der Kombination. Es schreibt auf der Website:

Es wird angestrebt, mit den Möglichkeiten moderner Informations- und Kommunikationstechnologien Erfolgssteigernde Bedingungen sowohl für ein selbstständiges, zeitlich und örtlich flexibles Lernen als auch für die Präsenzlehre zu schaffen [Ris02, S. 51]

Das Bildungsportal Sachsen ist ein Internetportal der sächsischen Hochschulen, das der Unterstützung der Aus- und Weiterbildung dient. In diesem Portal stehen den Studierenden des Freistaates Sachsen Lernmodule, die sie privat oder auch in Präsenzveranstaltungen nutzen können, zur Verfügung. Die Lernmodule werden im Rahmen des Verbundprojektes Bildungsportal Sachsen gefördert und erstellt. Lernmodule werden aber nicht nur für das Bildungsportal Sachsen erstellt, sie finden auch Zugang zu anderen Lernmanagementsystemen¹. Weltweit werden für die verschiedensten Lernmanagementsysteme und für die offline Verwendung Lerninhalte entwickelt. Um die Wiederverwendbarkeit von Lerninhalten zu gewährleisten bildeten sich verschiedene Standardisierungsinitiativen und begannen mit der Entwicklung von Spezifikationen und Standards. Ein Teil dieser Richtlinien hat Eingang in das Sharable Content Object

¹ Begriffsklärung siehe 1.1.2 LMS Seite 5

Reference Model (SCORM) der ADL Initiative² gefunden. SCORM gilt als quasi Standard und findet eine breite Unterstützung bei Herstellern im E-Learning Bereich.

Diese Arbeit beschreibt die Konzeption und Entwicklung des Lernmoduls „Digitale Bildbearbeitung mit Adobe Photoshop CS2“, welches an der Hochschule für Technik und Wirtschaft (HTW Dresden) entwickelt wurde. Dieses Modul soll im Praktikum der Lehrveranstaltung „Digitale Bildbearbeitung“ im Studiengang Medieninformatik begleitend zur Präsenzveranstaltung zum Einsatz kommen. Außerdem soll es den Studenten im Bildungsportal Sachsen zur Verfügung stehen.

Das Erste Kapitel gibt eine Einführung in das Thema „Lernen am Computer“; dabei werden wichtige Begriffe zum Thema geklärt. Des Weiteren gibt es einen Überblick über die verschiedenen Möglichkeiten im Bereich E-Learning.

Das zweite Kapitel beschreibt die Entwicklung des Lernmoduls „Digitale Bildbearbeitung mit Adobe Photoshop CS2“. Das Hauptaugenmerk soll dabei auf die technische Entwicklung gelegt werden. Die einzelnen Entwicklungsphasen, wie Konzeptions- und Designphase, werden kurz erläutert.

Die letzten zwei Kapitel beschäftigen sich mit SCORM. Das Kapitel 3 gibt dabei einen allgemeinen Überblick und erläutert die 4 SCORM-Bücher und ihre Inhalte.

Das vierte Kapitel widmet sich dem SCORM Navigation Model des vierten SCORM Buches. Im ersten Teil des Kapitels wird ausführlich auf die Theorie dieses Modells eingegangen. Anschließend soll es eine praktische Betrachtung der Möglichkeiten geben, die aus der Nutzung des Navigation Modells für das Lernmodul „Digitale Bildbearbeitung mit SCORM“ resultieren. Am Ende dieses Kapitels werden diese Ideen und Betrachtungen prototypisch umgesetzt.

² Begriffsklärung siehe 3.1 Überblick S. 39

1 Lernen am Computer

Lernen am Computer ist ein weitläufiges Gebiet, es umfasst unter anderem das Blended Learning, das Computer Based Training und das Web Based Training. Nach dem die wichtigsten Begriffe auf dem Gebiet E-Learning geklärt wurden, soll kurz auf alle drei Formen des „Lernens am Computer“ eingegangen werden.

1.1 Grundlegende Begriffe

Im Folgenden werden wichtige Begriffe zum Thema E-Learning erläutert.

1.1.1 E-Learning

E-Learning ist die Kurzform von Electronic Learning und umfasst genau genommen die volle Bandbreite des elektronischen Lernens. Der Begriff wird heute verwendet, wenn man vom Lernen mit digital gespeicherten Lerninhalten und softwareunterstützten Lernumgebungen spricht. Für den Begriff E-Learning sind verschiedene Schreibweisen verbreitet, z. B. e-Learning, eLearning, elearning. In dieser Arbeit wird die Schreibweise E-Learning verwendet.

1.1.2 LMS

Ein Lernmanagementsystem ist eine serverseitige Software, die es ermöglicht E-Learning in seiner gesamten Bandbreite zu nutzen. Zur Bandbreite gehören die administrativen Aufgaben ebenso, wie die Kommunikativen. Das LMS bietet dazu eine Nutzungsoberfläche. Über ein LMS ist die Administration von Lernern, Lerninhalten und Übungsaufgaben, sowie Kommunikationstools möglich. Außerdem können dem Lernenden Lerninhalte zur Verfügung gestellt werden, wobei der Lernprozess über das LMS verfolgt und protokolliert werden kann. Es ist möglich den Bearbeitungsstatus eines Kurses anzeigen zu lassen, verschiedene Kurse aufzurufen und auszuwählen. Weiterhin kann die Bearbeitungszeit ausgegeben werden, sowie die Ergebnisse von Tests.

Werkzeuge, mit denen zwar die Lerninhalte erstellt werden können, die aber nicht in der Lage sind Lerninhalte zu präsentieren, zählen nicht zu den Lernplattformen, sondern heißen „Autorensysteme“. LMS bzw. Lernplattformen bieten oft einfache Autorensysteme, z. B. zum Erzeugen von Testaufgaben.

1.1.3 Standardisierungen

Im täglichen Leben dienen Standardisierungen und Normen der Qualitätssicherung sowie die Möglichkeit des Austausches von Produkten und Technologien. Außerdem sichern sie die Wettbewerbsfähigkeit von Unternehmen. Auch auf dem Gebiet von E-Learning soll es möglich sein, mit Hilfe von Standards, Inhalte auszutauschen und zu kombinieren. Um dies zu erreichen müssen Systeme, Inhalte und Plattformen kompatibel sein. Es muss also ein festes „Vokabular“ definiert sein, welches Hersteller und Nutzer gleichermaßen verwenden.

Für den Bereich E-Learning gibt es eine Reihe von Standardisierungsgremien, die sich vor allem in den USA und in Europa gebildet haben. Alle haben das Ziel technische Standards für E-Learning Produkte zu entwickeln, diese Standards sind zu meist „Quasi-Standards“, da nur IEEE³ (Institute of Electrical and Electronic Engineers) berechtigt ist echte Standards zu verabschieden.

Zu den wichtigsten Initiativen gehören:

- AICC (Aviation Industry Computer Based Training Committee)
- IEEE LTSC (Learning Technology Standards Committee eine Einrichtung des Institute of Electrical and Electronic Engineers)
- IMS (Instructional Management Systems)
- ADL (Advanced Distributed Learning – Initiative des amerikanischen Verteidigungsministeriums)

Auf die Initiative ADL und ihrer Spezifikation SCORM soll im Laufe dieser Arbeit noch genauer eingegangen werden. (siehe 3. SCORM S. 39)

Da E-Learning das Zusammenspiel mehrerer unterschiedlicher Elemente – Lernender, LMS, Lernobjekt – sowie den Austausch vieler Daten beinhaltet, setzen die verschiedenen Standards unterschiedliche Schwerpunkte.

1.1.4 Standard/Norm

Standard ist allgemein bekannt als Regel oder Norm. Das British Standards Institute definiert Standard folgender Maßen: „Ein Standard ist ein öffentlich zugängliches technisches Dokument, dass unter Beteiligung aller interessierter Parteien entwickelt wird und deren Zustimmung findet. Der Standard beruht auf Ergebnissen aus Wissenschaft und Technik und zielt darauf ab das Gemeinwohl zu fördern.“ Diese Definition zeigt, dass bestimmte Dinge, wie zum Beispiel Microsoft Word, obwohl weit verbreitet, kein Standard sind. Verbreitung hat nichts mit Standard zu tun.

³ IEEE ist der größte technische Berufsverband der Welt. Er beschäftigt sich mit der Normung von Techniken, Hardware und Software. Standards von IEEE sind unter anderem Bluetooth und WLAN.

1.1.4.1 Standard

Im Bereich von Technik und Naturwissenschaften wird der Begriff Standard als Überbegriff für technische Normen (im Sinne von Übereinkünften und Verordnungen) verwendet. Diese Normen haben sich in der Praxis eine breite Akzeptanz verschafft. Außerdem ist es ein Begriff für Vereinheitlichungen, die sich ungeplant infolge gesellschaftlicher Prozesse und Erfahrungen der Praxis ergeben, entwickelt und als eine Art stillschweigende Übereinkunft etabliert haben. Beispiele sind Industriestandards und „herstellerspezifische (proprietäre) Standards“. Setzen sich Methoden oder Regeln nicht infolge von Vereinbarungen, Gesetzen oder Verordnungen durch, sondern in der Praxis, spricht man auch von de facto oder Quasi-Standards. SCORM von ADL ist solch ein de facto Standard.

1.1.4.2 Norm

Eine Norm ist eine allseits rechtlich anerkannte, allgemeingültige und veröffentlichte Regel zur Lösung eines Sachverhaltes, die durch ein Normungsverfahren beschlossen wurde. Das heißt diese Regel hat alle Instanzen eines Normungsverfahrens durchlaufen und wurde anschließend beschlossen und veröffentlicht. Eine Norm muss einen Nutzen für Anwender haben und sie muss technisch ausgereift sein. Internationale Organisationen zur Normung sind ISO und IEC, nationale Organisationen sind das DIN und die DKE (Deutsche Kommission Elektrotechnik).

1.2 CBT

Computer Based Training (CBT) bedeutet auf Deutsch computergestützter Unterricht. Man spricht auch von „Lernprogrammen“. Diese werden auf einer CD-ROM oder ähnlichem verteilt. Der Lernende hat das Lernprogramm vor Ort auf einem Computer gespeichert. CBT ist die meist genutzte Form des elektronischen Lernens. Durch die direkte Speicherung vor Ort hat man die Möglichkeit datenintensive, aufwendige und große Programme zu verwenden. Bandbreitenlimits entfallen.

Bei CBT unterscheidet man drei verschiedene Arten von Systemen:

- Simulationssysteme – Bei diesen Programmen wird der Sachverhalt spielerisch gelöst. Planspiele sind ein Beispiel für diese Form des CBT.
- Tutorielle Systeme – Bei dieser Form des CBT reagiert das System auf Eingaben des Lernenden und greift unterstützend ein.
- Präsentationssystem – Hier wird dem Nutzer ein vorgefertigtes Programm zur Verfügung gestellt, bei dem er nicht viele Möglichkeiten hat aktiv einzugreifen. Ein Beispiel für Präsentationssysteme sind digitale Bücher.

Der Nachteil des CBT ist die mögliche Isolation des Lernenden und die fehlende Motivation, die diese mit sich bringt.

1.3 WBT

Web Based Training (WBT) beschreibt Lernprogramme, die auf Internet-Technologien basieren. Es ist eine Weiterentwicklung des CBT. Beim WBT werden die Inhalte nicht auf Datenträgern verteilt, sondern von Webservern online abgerufen. Durch diese Art der Nutzung ergeben sich für den Lernenden vielfältige Möglichkeiten der Kommunikation und Interaktion mit seinen Mitlernenden. Die Isolation, wie sie beim CBT der Fall ist, tritt bei dieser Form des elektronischen Lernens nicht so stark auf. Es ist möglich das WBT zu moderieren oder die Lernenden beim Nutzen der Programme anzuleiten.

WBT spielt in der betrieblichen Weiterbildung großer Firmen eine wichtige Rolle, weil damit die Lerninhalte auch den weit entfernten Betriebsteilen zeitgleich unter Anleitung zur Verfügung gestellt werden können. Die Voraussetzung für die Nutzung von WBT ist ein leistungsfähiges Intranet. Die Lerninhalte im WBT sind relativ wartungsfreundlich, da sie nicht auf ein festes Medium fixiert sind.

Lernprogramme, die als WBT genutzt werden, dürfen nicht sehr datenintensiv sein, da auf die Datenübertragungsrate des Internets bzw. Intranets geachtet werden muss. WBT Lernprogramme werden in einem Browser ausgeführt. WBT ist on-demand Training, bei dem das Lernprogramm auf einem Server gespeichert wird und im ganzen Netzwerk zur Verfügung steht.

1.4 Blended Learning

Blended Learning bedeutet integriertes bzw. vermischtes Lernen. Beim Blended Learning werden verschiedene Ansätze der Lernorganisation kombiniert. So wird zum Beispiel Präsenzveranstaltung und E-Learning kombiniert, um die Vorteile von beiden zu verstärken und die Nachteile zu minimieren. Die Face-to-Face Kommunikation der Präsenzveranstaltung verbindet sich mit der Effektivität und der Flexibilität der elektronischen Medien. Das heißt klassisches Klassenzimmerlernen und die moderne Form des E-Learning bilden ein Ganzes, wobei das eine nicht ohne das andere funktioniert und die online Phasen optimal aufeinander abgestimmt werden sollten. Das Lernprogramm „Digitale Bildbearbeitung mit Photoshop CS2“ wurde für Blended Learning entwickelt und soll als Unterstützung des an der HTW-Dresden durchgeführten Praktikums dienen. Darüber hinaus kann es für das Selbststudium genutzt oder in anderen Lehrveranstaltungen in sächsischen Hochschulen integriert werden.

2 Projektentwicklung

Das Kapitel beschreibt, wie unser Projekt entwickelt wurde. Auf die Konzeption des Lernmoduls, auf die Gestaltung, sowie die Evaluation des Lernmoduls soll nur kurz eingegangen werden. Der Hauptteil dieses Kapitels beschäftigt sich mit der Entwicklung des Lernmoduls.

Das Projekt „Digitale Bildbearbeitung mit Adobe Photoshop CS 2“ wurde unter Leitung von Professor Merino in Zusammenarbeit mit Jana Krüger (Verfasserin der vorliegenden Arbeit), Kati Petrowitz, Jana und Ralf Halgasch sowie Wolfgang Schilling entwickelt.

2.1 Vorgehensweise bei der Entwicklung eines Lernmoduls

Bei der Produktion von Lernmodulen bzw. CBT/WBT-Produktionen, gibt es drei verschiedene Phasen, wobei die beiden großen Abschnitte des Projektablaufs die Konzeption und die Realisierung darstellen. Über alle Phasen erstrecken sich Projektleitung und -controlling, sowie die Qualitätssicherung.

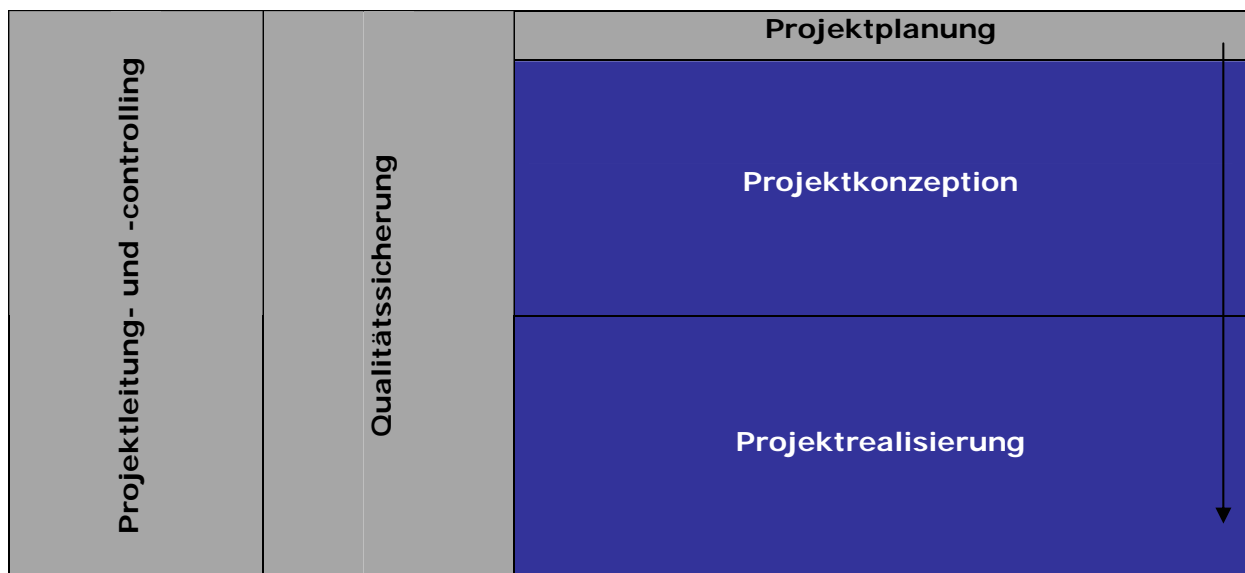


Abbildung 1 - Phasen der Produktion [Ris06, S. 98]

Die Projektkonzeption und Projektrealisierung gliedern sich wiederum in einzelne Phasen, die in Abbildung 2 zu sehen sind. Qualitätssicherung und Projektmanagement werden selbstverständlich immer am Rand mitgeführt.



Abbildung 2 - Unterteilung Konzeptions- und Realisierungsphase

2.2 Konzeption des Lernmoduls

Zu Beginn der Konzeption unseres Lernmoduls wurde zunächst eine technische Analyse durchgeführt. Dabei wurden weitestgehend die technischen Vorgaben vom Lernmodul „Flash interaktiv“ übernommen. Das Lernmodul sollte in Flash/ActionScript (Flash MX 2004) entwickelt und als SCORM Paket realisiert werden. Ob SCORM 1.2 oder SCORM 2004 war zu diesem Zeitpunkt noch nicht eindeutig geklärt. Als Browser wurden IE 6.0 und Mozilla Firefox 1.07 sowie Safari 2.0 angegeben, wobei das im Nachhinein beim technischen Test korrigiert wurde. Als Internetverbindung wurde DSL festgelegt, da heutzutage die meisten Studenten mit einer DSL-Verbindung arbeiten und das Lernmodul in erster Linie für Studenten gemacht ist.

Für die funktionelle Gestaltung wurden folgende Vorgaben gemacht: Es sollte verschiedene Lernmodi geben – einen Theorieteil, einen Test und einen Praxisteil mit Workshop. Bei der funktionellen Gestaltung galt es die Ergebnisse der Summativen Evaluation des Lernmoduls „Flash interaktiv“ zu berücksichtigen. Aus der Evaluation wurde klar, dass Workshops sehr gut angenommen werden, besonders wenn sie etwas anspruchsvoller sind. Daher sollen die Workshops eine wesentliche Rolle im Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ spielen. Dabei darf die Theorie natürlich nicht völlig vernachlässigt werden, da nur in Kombination mit der Theorie die zu vermittelnden Sachverhalte völlig klar werden. Animationen sind beim Lernmodul „Flash interaktiv“ sehr gut angekommen, deshalb sollen sie Eingang in das neue Lernmodul finden.

Aus den Vorgaben der Analysephase wurde das Grobkonzept entwickelt.

2.2.1 Das Grobkonzept

Das Grobkonzept bildet die Ausgangsbasis für alle späteren Schritte. Für das Grobkonzept wurden zunächst die Lernziele festgelegt. Aus den Lernzielen wurde dann der Basaltext für das Lernmodul entwickelt. Der Basaltext enthält alle Informationen, die für den Inhalt bedeutsam sind, und dient später als Grundlage für die Erstellung des Drehbuches.

Der Basaltext für „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ ist eine Tabelle und beinhaltet zu jedem Lernschritt eine kurze Beschreibung des Inhaltes. Abbildung 3 zeigt einen Screenshot des Basaltexes des Lernmoduls.

	A	B	C	D	E	F
64						
65						
66						
67	1	Kurze Vorstellung des nächsten Kapitels und seiner Schwerpunkt			05	
68	Grundlagen	1.1	1.1.1	Definition Pixelgrafik, erklären der Eigenschaften und Beispielbilder zur Darstellung des Auflösungsverhalten bei nahem Zoom	06	
69		Grundbegriffe	Pixel- und Vektorgrafik	Definition Vektorgrafik, erklären der Eigenschaften sowie Darstellung zum Auflösungsverhalten bei nahem Zoom	07	
70				Gegenüberstellung beider Grafikarten und Vergleich (ohne Wertung) anhand einer Tabelle	08	
71				Erläuterung der Vor- und Nachteile von Pixel- und Vektorgrafiken (Speicherbedarf etc.)	09	Vektorgrafiken
72				Vorstellen der bevorzugten Einsatzgebiete der jeweiligen Grafikart und Verwendungshinweise anführen (evtl. Ratschläge wann Pixel-, wann Vektorgrafik)	10	
73			1.1.2	Erklären der Bildauflösung und deren spezifischen Eigenarten anhand von Beispielbildern	11	
74			Bildauflösung			
75			1.1.3	Erklären der Druckauflösung anhand sich drastisch unterscheidender Beispielbilder	12	
76			Druckauflösung			
77				Darstellen des Unterschieds zwischen Bild- und Druckauflösung, sowie Anführen einiger Rechenbeispiele	13	
78			1.1.4	Erklären des Begriffs "Farbtiefe" anhand von Grafiken	14	
79			Farbtiefe			
80				Vorstellen der 1 Bit, 8 Bit, 16 Bit und 24 Bit Farbtiefe mit Beispielbildern sowie Erklärung des Begriffs "indiziert"	15	
81			1.1.5	Vorstellen des RGB Farbmodus und des RGB Farbmodells sowie zeigen der beiden Quotierungsarten (0,0,0 bzw. #000000)	16	
82			Farbmodus			
83				Vorstellen des CMYK Farbmodus und des CMYK Farbmodells (evtl. zum Vergleich: die Zusammensetzung primitiver Farben im CMYK und RGB Farbmodus)	17	
84						
85						
86						
87						
88						
89						
90						

Abbildung 3 - Screenshot Basaltext Lernmodul "Digitale Bildbearbeitung mit Adobe Photoshop CS2"

Nachdem der Basaltext geschrieben und abgenommen wurde, konnte er als Vorgabe für die Erstellung des Feinkonzeptes, des Drehbuchs genutzt werden.

2.2.2 Das Feinkonzept

Das Feinkonzept des Lernmodules „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ ist im Großen und Ganzen das Drehbuch. Das Drehbuch einer Lernanwendung ist eine detaillierte Beschreibung des Inhalts und des Aufbaus der Bildschirmseiten. Ein Drehbuch beinhaltet folgende Elemente:

- Projektbezeichnung
- Modulbezeichnung
- Version
- Datum
- Autor

-
- Historie (Wer hat am Drehbuch gearbeitet, welche Version wurde bearbeitet und was wurde geändert?)
 - Inhaltsverzeichnis
 - Evtl. Metadaten
 - Evtl. Zeitangabe (Wie lange braucht der Lernende für eine Seite)
 - Bildschirmtexte (Vollständig ausformuliert, mit Angaben zu welchem Inhalt welche Grafik gehört, Platzierung und Formatierung des Textes)
 - Grafik (vollständige Beschreibung des Bildinhaltes, Angabe wozu welche Grafik gehört und wo die Grafik im Bildschirm erscheinen soll, sowie Quellenangaben, woher die Grafik stammt)
 - Sprechertexte (vollständig ausformuliert mit Angaben, wozu er gehört und evtl. Bemerkungen falls eine besondere Aussprach nötig ist)
 - Geräusche und Sound (welcher Art die Geräusche sind und wo bzw. wann sie ertönen)
 - Effekte und Animationen (Art der Animationen, zu welchem Zeitpunkt sie erscheinen und eine Erläuterung des Ablaufs einer Animation)

Um all diese Informationen unter zu bekommen, wurde eine Vorlage erstellt, in welche die Inhalte des Drehbuchs eingetragen wurden. Ein Beispiel für eine Drehbuchseite ist im Anhang B zu finden.

Da die Bildschirmtexte, wie sie im Drehbuch stehen genauso im Lernmodul verwendet werden sollten, wurden im Vorfeld wichtige Hinweise für die Erstellung von Bildschirmtexten erarbeitet. Diese Hinweise sollen folgend kurz aufgezählt werden.

- Ein Text mit vielen ungeläufigen oder unbekanntem Wörtern z. B. Fremdwörter oder Fachbegriffe ist schwer verständlich. Daher sollten solche Begriffe erläutert oder vermieden werden.
- Das Verständnis für längere Texte verbessert sich, wenn man z. B. Wörter mehrmals verwendet und Konjunktionen oder Proformen bildet.
- Texte sollen dazu anregen, sich Gedanken zum Lesestoff zu machen, d. h. man stellt eine Beziehung zwischen Vorwissen und neuen Informationen her.
- Dem Lernenden sollte das Filtern der Texte nach wichtigen Informationen erleichtert werden. Dies geschieht mit Hilfe von:
 - einer Zusammenfassung am Anfang oder Ende eines Textes
 - Zwischenüberschriften
 - Absätzen oder
 - Aufzählungen

Leicht zeitversetzt zur Erstellung des Drehbuches wurde die Benutzeroberfläche entworfen.

2.3 Gestaltung der Benutzeroberfläche

Bei der Gestaltung einer Benutzeroberfläche gibt es aus Benutzersicht eine Menge zu beachten. Im Folgenden soll nur kurz auf die wichtigsten Dinge eingegangen werden, da dieses Thema nicht Hauptbestandteil der Diplomarbeit sein soll.

Bilder und Texte sind essentieller Bestandteil von Lernmaterialien und spielen eine wesentliche Rolle im Rahmen der Gestaltung und Konzeption. Bei der Gestaltung von Seiten für den Bildschirm gibt es ein paar Besonderheiten im Vergleich zur Gestaltung von Seiten für ein Buch oder Arbeitsheft. So lassen sich zum Beispiel keine guten Lernerfolge erzielen, wenn man Lernumgebungen entwickelt die ausschließlich Text oder Bilder beinhalten. Eine Kombination von Beiden ist ideal.

Text am Bildschirm

Länge Texte sind für eine Bildschirmseite ungeeignet, weil das Lesen am Bildschirm mühsamer und anstrengender ist. Außerdem sollte horizontales Scrollen in jedem Fall vermieden werden, da die Lesbarkeit mit einem horizontalen Scroller um ein Vielfaches sinkt. Um Text möglichst sinnvoll am Bildschirm zu gestalten, ist eine Strukturierung sowie Gliederung enorm wichtig. Ein angepasster Sprachstil, lernerbezogene Wortwahl, einfacher Satzbau sowie Orientierungsmarken unterstützen die Lesbarkeit eines Textes. Zur Unterstützung der Lesbarkeit sollten auch typografische Aspekte beachtet werden:

Wahl der Druckschrift	Ohne Serifen
Schriftgrad	12 – 14
Schriftstärke	Schmallaufende Schriften
Auszeichnung	Fett besser als unterstrichen, weniger ist mehr
Buchstaben + Wortabstände	Normale Laufweite
Schriftlage bzw. -stile	Kursiv schwerer zu lesen
Zeilenabstand	Muss mühelos ohne ablenkende Seitensprünge zu durchmustern sein
Kontrast	Farbe zur Unterlegung wählen, ideal ist ein schwarz/weiß Kontrast. Weißer Hintergrund und schwarze Schrift
Seitenaufteilung	Nicht überladen

Tabelle 1 - Typografische Aspekte der Textgestaltung

Bilder im E-Learning

Bilder werden manchmal oberflächlicher als Text wahrgenommen. Da das Lesen am Bildschirm eher schlecht ist, kommt Bildern im E-Learning eine besondere Bedeutung zu. Ideal ist die Kombination von Bild und Text, dabei müssen sich Bild und Text allerdings

aufeinander beziehen. Ein Bild ohne Bezug zum Text ist nicht sinnvoll. Bilder unterstützen im besten Fall das Textverstehen bzw. Text unterstützt die Bildanalyse positiv.

Bei jeglicher Verwendung von Bild, Text und auch Ton am Bildschirm gilt: „Weniger ist mehr!“ Nur eine ausgewogene Kombination bringt den gewünschten Erfolg.

Unter Berücksichtigung der genannten Aspekte wurde das Design für das Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ entwickelt – und ein Style Guide erstellt. Der StyleGuide ist im Anhang C der Diplomarbeit zu finden.



Abbildung 4 - Finales Layout des Lernmoduls

2.4 Evaluation des Lernmoduls

Nach der Entwicklung der ersten Seiten des Lernmoduls wurden die erstellten Seiten evaluiert. Wichtig war es dabei heraus zu finden, wie die Nutzer mit unserem Lernmodul klarkommen, ob sie damit umgehen können und ob der Inhalt für sie verständlich ist. Die Einzelheiten der Evaluation sollen an dieser Stelle nicht erläutert werden, dazu soll es später eine Diplomarbeit von Kati Petrowitz geben.

2.5 Entwicklung des Lernmoduls

Wurde das Drehbuch für ein Lernprogramm entwickelt, soll es im nächsten Schritt alleine oder im Team umgesetzt werden. Daher ergibt sich die Frage welches Autorenwerkzeug eingesetzt werden soll und wie die Umsetzung funktioniert. In diesem Kapitel soll geklärt werden, was Autorenwerkzeuge sind, welche Unterschiede es gibt und welches Autorenwerkzeug wir für die Entwicklung des Lernmoduls „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ eingesetzt haben. In einem Weiteren Punkt soll die Entwicklung des Lernmoduls sowie die Klassenstruktur erläutert werden.

2.5.1 Flash MX 2004 als Autorenwerkzeug

2.5.1.1 Begriffsklärung Autorenwerkzeug⁴

Autorenwerkzeuge dienen der Erstellung von Lerninhalten. Sie werden in der Literatur auch als Autorensoftware bezeichnet. In der vorliegenden Arbeit, soll der Begriff Autorenwerkzeug verwendet werden. Ziel dieser Werkzeuge ist es, die Entwicklung so einfach und komfortable wie möglich zu halten. Die einzelnen Autorenwerkzeuge unterscheiden sich besonders in ihrem Einarbeitungsaufwand. Sie lassen sich folgendermaßen unterteilen:

Professionelle Autorensysteme: Derartige Werkzeuge verfügen über eine integrierte Programmiersprache und erfordern im Allgemeinen einen hohen Einarbeitungsaufwand. Sie bieten den Entwicklern jedoch im Gegenzug häufig ein hohes Maß an kreativen Freiheiten.

WYSIWYG-HTML-Editoren: WYSIWYG-Editoren („What You See Is What You Get“) unterstützen die Autoren digitaler Lerninhalte mittels einer grafischen Schnittstelle, über die sie wie gewohnt Texte schreiben und Animationen oder Bilder einfügen können. Die Arbeit erfolgt auf einer abstrahierten Ebene und hat den Vorteil, dass sich die Entwickler nicht mit den komplexen Details den zugrunde liegenden Aufzeichnungssprachen (HTML, XML usw.) befassen müssen.

Rapid Content Development Tools: Hierbei handelt es sich um Autorenwerkzeuge ohne hohen Einarbeitungsaufwand, mit deren Hilfe sich Lerninhalte auf schnelle und unkomplizierte Weise generieren lassen.

[Rül06, S. 10]

⁴ Autorenwerkzeug wird in der Literatur auch als Autorensoftware bezeichnet. In der Diplomarbeit soll der Begriff Autorenwerkzeug verwendet werden.

2.5.1.2 Auswahlkriterien bei der Entscheidung für ein Autorenwerkzeug

Bei der Auswahl eines Autorenwerkzeugs sollte berücksichtigt werden, welche Ansprüche die Nutzer an Interaktivität und Grafikperformance stellen, sowie ob sich jemand im Team befindet der Programmierkenntnisse hat. Ist letzteres der Fall, sollte auf ein Autorenwerkzeug zurückgegriffen werden, dass für professionelle Entwicklung vorgesehen ist. Weiterhin sollten sie bei der Auswahl auf folgende Aspekte achten.

- Protokollfunktion
- mehrere interaktive Optionen
- Nutzbarkeit von allen gängigen multimedialen Dateien
- individuelle Auswertung von Fehlertypen
- vordefinierte Elemente
- ausreichender Support durch den Hersteller
- Möglichkeit der Einbindung von Programmiersprache und Zusatztools
- zukünftig Lauffähigkeit der erstellten Applikationen on- und offline

[Ris02, S. 145]

Ein wichtiges Auswahlkriterium bei der Entscheidung für ein Autorenwerkzeug ist weiterhin, dass das Autorentool auch in Zukunft vom Hersteller weiterentwickelt wird. Ebenso ist es wichtig zu bedenken, dass die Zielgruppen vom Medienkonsum immer mehr verwöhnt sind und daher die Ansprüche an die Optik und Funktionalität eines Lernmoduls wachsen. Schon heute belächelt man die schlichten 16-Farb-Offline Lernprogramme, wie sie in den frühen Neunzigerjahren entstanden sind. Animationsfiles und Videos halten immer stärker Einzug in Lernprogramme.

2.5.1.3 Überblick über das Autorenwerkzeug Flash MX 2004

Flash ist eines der preiswerteren Autorensysteme, dass nicht primär auf das Erstellen von Lernprogrammen ausgerichtet ist. Das macht Flash oft nicht zur ersten Wahl als Autorensystem, allerdings sollte man beachten, dass die Stärken von Flash bei der Erstellung multimedialer und interaktiver Inhalte, die im Internet angeboten werden, liegen. Mit Flash lassen sich sowohl einfache Animationen als auch „wirkliche Kunstwerke“ erstellen. Ein wichtiger Fakt in der Zeit, in der die Ansprüche an Multimediapräsentationen immer mehr wachsen. Flash lässt sich heutzutage bereits in allen Websites einbinden und bietet einen Vielzahl von Techniken an, die bereits fester Bestandteil des Werkzeugs sind:

- Tweeningvarianten, um Bewegungsabläufe umzusetzen
- Import/Export von diversen Dateiformaten wie Bitmap-Grafiken, Sound und Video
- Gestalten eigener grafischer Ideen mit Hilfe der integrierten Werkzeuge

[Kan05, S. 21]

Die Entwicklungsumgebung von Flash MX 2004

Die Entwicklungsumgebung von Flash MX 2004 kann den verschiedenen Bedürfnissen des Nutzers angepasst werden. Bedienelemente können hinzugefügt, entfernt oder neu angeordnet werden. Im Folgenden sollen die am häufigsten benutzten Bedienelemente kurz erläutert werden:

Bühne Die Bühne ist der „Auftrittsort“ für den Darsteller auf ihr lassen sich Inhalte platzieren und modifizieren.

Zeitleiste Über die Zeitleiste wird der Ablauf eines Projektes, also das Nacheinander von Einzelbildern, gesteuert. Sie besteht aus Ebenen, Bildern und Abspielknopf. Über die Ebenen werden die Objekte hierarchisch angeordnet. Es wird also festgelegt welche Objekte im Vorder- und Hintergrund erscheinen. Die zeitliche Organisation eine Szene geschieht über die Abfolge von Bildern. Je nach dem wo sich der Abspielknopf befindet, wird der zeitliche Zustand auf der Bühne abgebildet. Bilder dienen neben der Darstellung der einzelnen Zustände auch als eine Art Container für ActionScript.

Werkzeugleiste Über die Werkzeugleiste lassen sich Objekte erstellen, ausrichten und modifizieren. Sie enthält Werkzeuge wie zum Beispiel: Lassowerkzeug, Auswahlwerkzeug, Pipette, Radiergummi, Linienwerkzeug, Stiftwerkzeug usw.

Bibliothek Importierte Flashinhalte und Inhalte in Form eines Symbols, einer Schaltfläche oder Sound werden in der Bibliothek gespeichert.

Eigenschaftsinspektor Ist ein Element oder Objekt aktiviert (angeklickt) steht der Eigenschaftsinspektor für dieses Element zur Verfügung. Der Eigenschaftsinspektor ist ein Dialogfeld über das sich verschiedene Eigenschaftsparameter einstellen lassen.

Aktionenfenster In diesem Fenster lässt sich ActionScript-Code einfügen und editieren. Für Professionelle Entwickler ist dieses Fenster völlig frei benutzbar. Für Anfänger stellt Flash vorgefertigten Quellcode zur Verfügung.

2.5.2 Entwicklung des Lernmoduls mit Flash MX 2004

2.5.2.1 Grundlagen der Entwicklung

Die Grundlage für die Entwicklung des Lernmoduls „Digitale Bildbearbeitung mit Adobe Photoshop CS2“, bildete die von Rülke [Rül06] entwickelte Klassenstruktur. In dieser Klassenstruktur bestand das wesentliche Ziel darin, die grafischen Elemente der Benutzeroberfläche über ActionScript zu zeichnen und dadurch flexible Eingriffsmöglichkeiten in die visuelle Darstellung zu bieten. Rülke hat die Klassen in Paketordnern sortiert um einen besseren Überblick zu erhalten. Bei der Entwicklung wurde von der Möglichkeit der Vererbung Gebrauch gemacht. Die Vererbung wurde sowohl bei selbst erstellten, als auch bei bereits in Flash integrierten Klassen angewendet. Fast alle Klassen werden über eine so genannte Klassenbibliothek eingebunden. Die Klassenbibliothek (swf-Film) wird zur Laufzeit geladen. Somit stehen alle Klassen nach dem Einlesen der Bibliothek im Initialisierungs- Film auch allen nachgeladenen Filmen zur Verfügung. Die benötigten Daten für den Inhalt werden vom jeweiligen Flashfilm aus den dazugehörigen XML- und HTML-Dateien geladen. Im folgenden Abschnitt sollen die wichtigsten Klassen von Herrn Rülke kurz vorgestellt werden. Eine vollständige und detaillierte Erläuterung der Klassen ist in der Arbeit von Rülke zu finden [Rül06]. Die Anmerkungen zur Anpassung der Klassen auf das Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ werden in einem separaten Abschnitt behandelt.

2.5.2.2 Klassenstruktur nach Rülke

Klassen zur Vorausladen von Dateien

Die flexible Architektur des Lernmoduls verlangt, dass viele externe Daten zur Laufzeit geladen werden müssen. Damit der Nutzer immer genau weiß, was gerade geschieht wurde eine Ladefortschrittanzeige entwickelt. Über diese kann der Nutzer erkennen, welche Informationen gerade eingelesen werden und wie der aktuelle Ladezustand aussieht. Für Realisierung dieser Anzeige wurden folgende Klassen entwickelt.

class loading.LoadDefinition

Diese Klasse wurde zur Übergabe an den Preloader entwickelt. Sie enthält die Informationen, welche Dateien über welche Objekte geladen werden sollen.

interface loading.PreloaderViewInterface

Diese Klasse registriert in der Preloader-Klasse die Ladefortschrittsanzeige und richtet sie für den Empfang von Statusinformationen ein.

Class loading.PreloaderView implements loading.PreloaderViewInterface

Diese Klasse generiert die Ladefortschrittsanzeige, welche sich bei Rülke als Kreis bestehend aus Linie und Füllung, sowie einem inneren Kreissegment präsentiert.

interface loading.PreloaderViewInterface

Diese Klasse registriert in der Preloader-Klasse die Ladefortschrittsanzeige und richtet sie für den Empfang von Statusinformationen ein.

class loading.Preloader

Diese Klasse überwacht den Status der nachgeladenen Daten. Sie sorgt dafür, dass nach dem Einlesen der Daten die Arbeit des Programms an gewünschter Stelle fortgesetzt werden kann.

Klassen zum Zeichnen der grafischen Elemente

Wie eingangs beschrieben, werden die grafischen Elemente mit ActionScript gezeichnet. Somit bilden die Klassen zum Zeichnen der grafischen Elemente eine wesentliche Grundlage des Lernmoduls. An dieser Stelle werden nur die Zeichenklassen von Rülke beschrieben, die auch für das Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop“ in der einen oder anderen Weise zum Einsatz kamen.

class graphics.drawing.LineStyle

Hier wird der Stil einer zu zeichnenden Linie festgelegt. Es kann die Linienstärke, -farbe sowie die -transparenz festgelegt werden.

class graphics.drawing.FillStyle

Mit Hilfe dieser Klasse kann die Füllung von Vektorformen definiert werden. Es ist möglich Transparenz und Farbe festzulegen.

class graphics.drawing.FormDefinition

Diese Klasse ist eine der wichtigsten Klassen überhaupt, denn sie bildet die Grundlage für fast alle Vektorgrafiken des Lernmoduls. Die einzige Ausnahme bildet hier die Ladefortschrittsanzeige. Mit dieser Klasse ist es möglich, Formen aus Linien und Kurven zusammenzusetzen.

Class *graphics.drawing.CircleDefinition* extends *graphics.drawing.FormDefinition*

Mit Hilfe dieser Klasse ist es möglich, Kreise oder Ringe zu definieren. Sie ist von Formdefinition abgeleitet.

class *graphics.drawing.RectangleDefinition* extends *graphics.drawing.FormDefinition*

Diese Klasse ermöglicht es, auf einfache Art und Weise abgerundete Ecken eines Rechtecks zu definieren.

Class *graphics.drawing.SplineDefinition* extends *graphics.drawing.FormDefinition*

Geschlossene Vektorformen lassen sich mit dieser Funktion nur über die Angabe der Koordinaten problemlos zeichnen.

class *graphics.drawing.FormDraw*

Diese Klasse dient dazu, erstellte FormDefinition-Objekte in MovieClips zu zeichnen.

Klassen zum Erstellen spezieller MovieClips

Die Klassen zum Erstellen spezieller MovieClips sind von MovieClip abgeleitet. Sie dienen als Grundlage für weitere ActionScript2-Klassendefinitionen.

dynamic class *graphics.objects.FramedClip* extends *MovieClip*

FrameClip ist eine Basisklasse, die von MovieClip abgeleitet wurde. Sie besitzt keinen eigenen Konstruktor und vererbt Ihre Eigenschaften und Methoden komplexeren Klassen. MovieClip verhindert, dass Unterklassen dynamisch sind, daher ist es an dieser Stelle notwendig im Klassenkopf das Schlüsselwort `dynamic` zu verwenden. In der `initClip()`-Funktion der Klasse werden drei leere MovieClips erstellt. Die oberste Ebene ist für den Rahmencustümig. Die unterste für die Farbfüllung und mittlere für die Inhalte aus der Bibliothek oder für externe SWF-Dateien.

Class *graphics.objects.MaskedClip* extends *graphics.objects.FramedClip*

Diese Klasse sorgt dafür, dass der Inhalt nur innerhalb des in FramedClip gezeichneten Rahmens dargestellt wird.

Class *graphics.objects.TextClip* extends *graphics.objects.MaskedClip*

Die Klasse MaskedClip wurde um die Möglichkeit der Textdarstellung erweitert. Der Text lässt sich ebenfalls maskieren.

class graphics.objects.TextClip extends graphics.objects.MaskedClip

Die Klasse MaskedClip wurde um die Möglichkeit der Textdarstellung erweitert. Der Text lässt sich ebenfalls maskieren.

Klassen zur Steuerung von Animationen

Diese Klassen dienen dazu, die Animationen des Lernmoduls anzuzeigen und zu steuern.

class animation.SimplePlayer

Dies Klasse realisiert einen einfach Player, der lediglich aus einer Wiedergabe und Pausieren-Schaltfläche besteht.

class animation.ComplexPlayer

Diese Klasse ist ähnlich der SimplePlayer aufgebaut, allerdings ist sie wesentlich komplexer. Im ComplexPlayer gibt es flexiblere Eingriffsmöglichkeiten in das Wiedergabegeschehen. Man kann an eine beliebige Position im Film springen, es ist möglich die Lautstärke des Sounds zu steuern außerdem ist es möglich die Steuerkonsole des Players ein- und auszublenden. Die Beschaffenheit der Klasse verlangt allerdings einen bestimmten Aufbau des Films, sonst gibt es Probleme in der Wiedergabe und Steuerung. Die Animation und der Sound muss direkt und komplett in der Hauptzeitleiste des Dokuments untergebracht sein.

Klassen zum Erstellen der Benutzungsoberflächenelemente

Diese Klassen dienen dazu, die verschiedenen Objekte der Benutzeroberfläche darzustellen. Als Beispiel wären da der Hintergrund, Steuerelemente und Animationen zu nennen.

class moduleObjects.ModuleBackground

Mit Hilfe dieser Klasse wird der Hintergrund des Lernmoduls generiert. Wobei sich die grafische Darstellung flexibel an die Bedürfnisse anpassen lässt.

class moduleObjects.ModulePage

Diese Klasse generiert die Textfelder für Haupt- und Zusatzbereich. Es lässt sich auch die Position bzw. der Abstand der Textfelder zu den Rändern festlegen. In diese Textfelder werden die HTML-formatierten Texte geladen.

class moduleObjects.ModuleAnimationPage

Mit dieser Klasse wird der Animationsbildschirm erzeugt, wobei der Animationsbildschirm bei Bedarf noch mit einem Hintergrund versehen den kann.

class moduleObjects.ModuleAnimationPage

Mit dieser Klasse wird der Animationsbildschirm erzeugt, wobei der Animationsbildschirm bei Bedarf noch mit einem Hintergrund versehen werden kann.

Dynamic class moduleObjects.NavigationButton extends MovieClip

Diese Klasse wurde entwickelt um die Navigationsschaltflächen zu generieren.

class moduleObjects.NavigationButtonBar

Diese Klasse generiert die gewünschte Anzahl NavigationButton-Instanzen. Außerdem ordnet sie diese in der gewünschten Position an.

dynamic class moduleObjects.ScreenshotButton extends MovieClip

Diese Klasse generiert Bildschaltflächen.

class moduleObjects.ScreenshotButtonContainer

Mit Hilfe dieser Klasse wird ein Container-Movieclip erstellt, in dem die Bildschaltflächen dargestellt werden.

class moduleObjects.SimplePageObjects

Mit Hilfe der über diese Klasse erzeugten Objekte lassen sich grafische Ressourcen des Lernmoduls darstellen. Dazu gehören zum Beispiel - Bilder, Download-Schaltflächen, Interaktionen sowie Beispielanimationen.

Klassen zum Laden der XML-Konfigurationsdateien

Die Konfiguration des Lernmoduls passiert über XML-Konfigurationsdateien, welche über die folgenden Klassen geladen werden.

Dynamic class util.DBoolean extends Boolean

dynamic class util.DNumber extends Number

dynamic class util.DString extends String

Diese Erweiterungen der jeweiligen Klassen dienen dazu, Objekte der Klassen untereinander als Eigenschaft zuzuweisen. Was die Abbildung der Element- und Attribut-Beziehungen der XML-Struktur erleichterte.

class loading.XMLLoader extends XML

Über den Konstruktor der Klasse, werden die verschiedenen Mitgliedvariablen mit Standardwerten initialisiert. Weiterhin dient diese Klasse dazu die XML-Elemente und -Attribute, die standardmäßig vom Typ *String* sind umzuwandeln in numerische und boolesche Variablen.

Klassen zum Anzeigen von Tooltips

Zur Unterstützung der Aussage eines interaktiven Elementes wurden Tooltips erstellt. Die folgende Klasse wurde entwickelt um diese Tooltips anzuzeigen.

class util.Tooltip

Für die Erstellung, sowie das Aussehen und die Position des Tooltips ist diese Klasse verantwortlich.

2.5.2.3 Die Generelle Struktur des Lernmoduls

Im Groben wurde die Struktur des Lernmoduls übernommen, wie sie von Rülke in seinem Lernmodul angelegt wurde. Der Aufruf der Dateien passiert in ähnlicher Art und Weise. Da wir in unserem Modul allerdings eine Navigationsebene mehr verwenden, musste die Struktur dahingehend leicht angepasst werden. Die folgende Grafik soll die Navigation unseres Lernmoduls verdeutlichen.

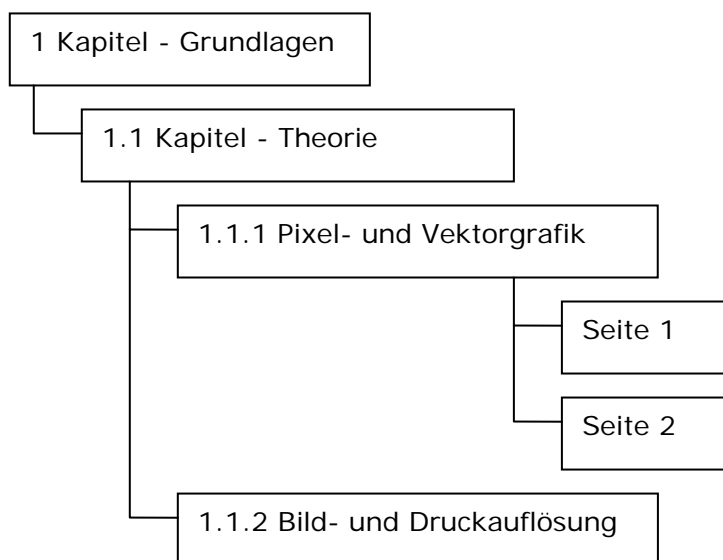


Abbildung 5 - Struktur des Lernmoduls

Das Lernmodul gliedert sich in einzelne Kapitel, diese wiederum in Unterkapitel. Die Unterkapitel sind wiederum in Abschnitte unterteilt, die aus einzelnen Seiten bestehen. Die vollständige Gliederung des Lernmoduls befindet sich im Anhang D.

Um diese Gliederung des Lernmoduls zu realisieren, musste der Prozess beim Laden eines Lernobjektes angepasst werden.

Wie schon in [Rül06] nachzulesen ist, besteht ein komplettes SCORM-Lernobjekt nicht aus einem einzelnen Film, sondern es setzt sich aus den verschiedenen Teilfilmen, XML-, HTML- sowie JavaScript-Dateien zusammen. Diese Daten werden zur Laufzeit nacheinander geladen. Unterstützt wird dieser Prozess durch die Preloader-Klassen, die auch dafür sorgen, dass der Anwender immer genau weiß, was gerade geladen wird und wie der Fortschritt ist.

Der Prozess des Ladens und die daran vorgenommenen Änderungen sollen im Folgenden kurz erläutert werden, wobei auf den Sinn und die Bedeutung der einzelnen Dateien nur sehr kurz eingegangen wird, da diese in [Rül06] beschrieben sind.

Prozess des Ladens

Beim Aufruf eines Lernobjektes wird die *Load.html* gestartet. Über die Datei *load.html* wird ein einzelnes Lernobjekt aufgerufen. Sie verfügt über keinerlei relevanten Inhalt und dient lediglich als eine Art Container für JavaScript-Programmcode. Die Datei ist für jedes Lernobjekt genau einmal vorhanden. Sie lädt die JavaScript-Dateien *ModulFunktion.js*, *SCOFunktion.js* und *APIWrapper.js*, wobei die *ModulFunktion* wiederum eine Funktion zum Laden des Initialisierungsfilms *init.swf* beinhaltet. Der Initialisierungsfilm lädt die Textstil-Definitionen für Lernmodul und Ladefortschrittsanzeige. Er lädt also die CSS-Dateien – *modul_textstyles.css* und *preloader_view_textstyle.css*. Außerdem ist er verantwortlich für das Laden der XML-Konfigurationsdateien – *modul_init.xml* und *preloader_view_init.xml*, sowie für den Hauptfilm sowie die Klassenbibliotheken. Bis zu dieser Stelle ist der Ladeprozess des Moduls Digitale Bildbearbeitung identisch mit dem Ladeprozess des Moduls „Die Interaktive Arbeit mit Flash“. Ab dem Hauptfilm gibt es Änderungen. Während bei Rülke direkt die *page.swf* geladen wird. Unterscheidet das Lernmodul „Digitale Bildbearbeitung“, ob es im Lernmodul noch Unterseiten gibt. Existiert nur ein Abschnitt in diesem Kapitel werden direkt die *page.swf* sowie die *chapter_init.xml* geladen, welche die XML-Konfigurationsdateien des Hauptfilms beinhaltet. Gibt es allerdings noch verschiedene Abschnitte wird vor der *page.swf* noch die *subchapter.swf* geladen, und erst aus der *subchapter.swf* heraus die *page.swf* aufgerufen. Zusätzlich wird bei der *subchapter.swf* noch die *subchapter_init.xml* aufgerufen, da an dieser Stelle

noch XML-Konfigurationsdateien für den Abschnitt benötigt werden. Die Datei *page.swf* lädt die XML-Konfigurationsdateien der einzelnen Seiten – *page_init.xml*, sowie die Ressourcen der Seite – Texte, Bilder und Animationen. Auch die *subchapter.swf* ist in der Lage Ressourcen in Form von Texten zu laden und darzustellen. Von dieser Möglichkeit wird beim Starten eines Kapitels gebrauch gemacht, wenn noch kein bestimmter Abschnitt ausgewählt ist.

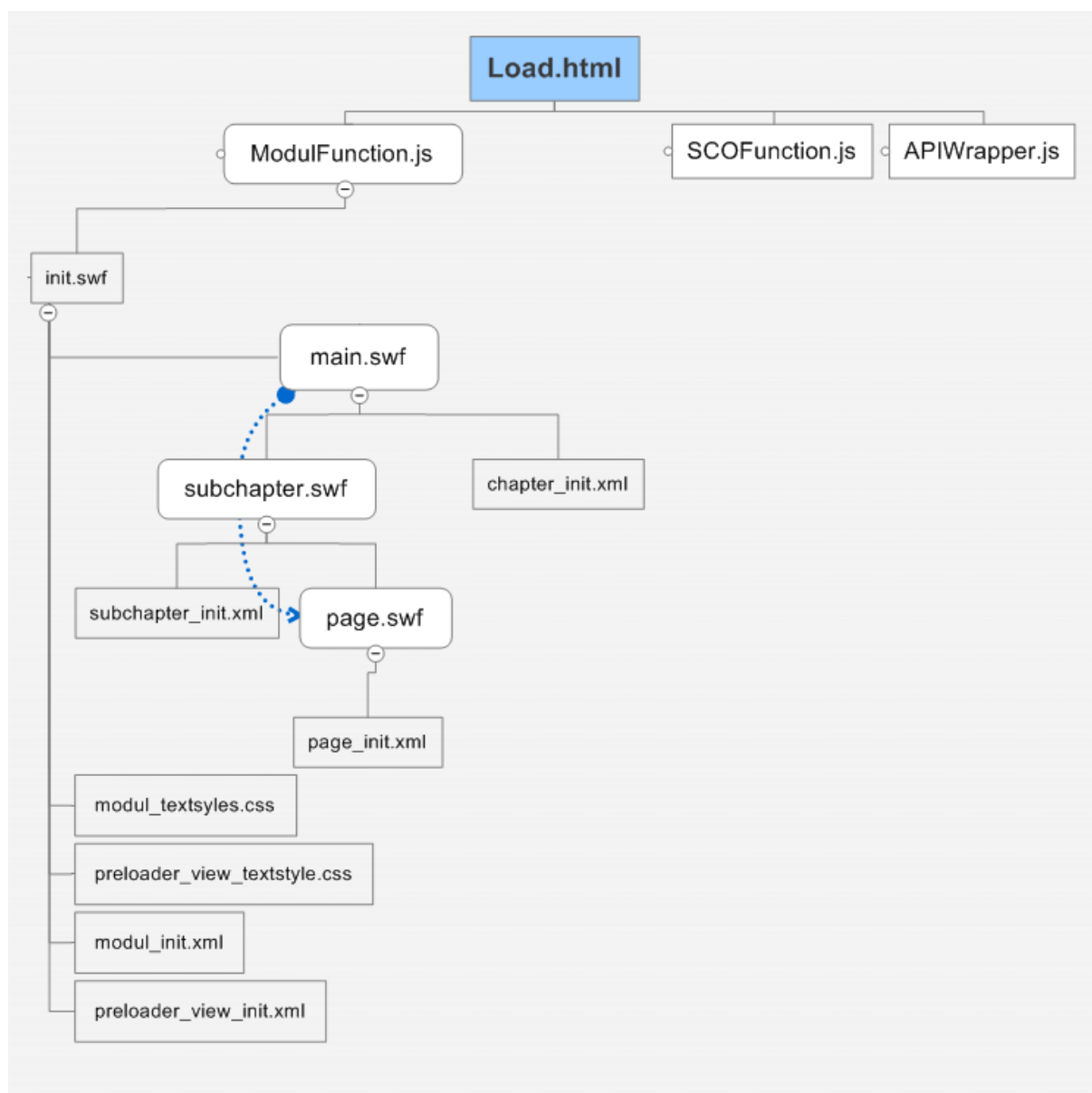


Abbildung 6 - Ablaufplan Lernmodul „Digitale Bildbearbeitung“

2.2.2.4 Anpassung des Lernmoduls

Um das Lernmodul in der Form zu realisieren, wie es beim Design und im Drehbuch vorgesehen ist, mussten einzelne Dateien des Lernmoduls „Einführung in die interaktive Arbeit mit Flash“ modifiziert werden. Unsere erste Aufgabe bestand darin, die Grafikdateien auf unser Design anzupassen. Die Klassen zum Zeichnen der grafischen

Elemente konnten ohne größere Probleme von uns übernommen werden, da das Design des Lernmoduls aus den grundlegenden Formen Rechteck und Linie besteht. Diese Formen sind auch die tragenden Elemente vom Lernmodul „Flash“.

Die Klassen zum Erstellen der Benutzungsoberflächenelemente mussten dagegen sehr stark modifiziert werden, da sich das Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ doch sehr in der Bedienung Lernmodul „Flash“ unterscheidet.

Navigation

Unser Lernmodul besteht aus drei verschiedenen Navigationen. Zum einen die Navigation, die mittels LMS bzw. mittels HTML-Seite zur Verfügung gestellt wird. Mit Hilfe dieser Navigation kann der Lerner zwischen den einzelnen Kapitel bzw. Unterkapitel navigieren.

Die zweite Navigation befindet sich im Flashfilm des jeweiligen Kapitels. Mit Hilfe dieser Navigation kann der Lerner zwischen den einzelnen Abschnitten navigieren. Realisiert wird diese Navigation mit den Dateien *NavigationButton.as* und *NavigationButtonBar.as*. Im Ausgangsmaterial waren beide Dateien schon vorhanden. Allerdings unterscheidet sich das Prinzip der Navigation, wie auf Abbildung 7 und Abbildung 8 zu erkennen, etwas von der ursprünglichen. Daher mussten diese Dateien auf unsere Bedürfnisse angepasst werden.



Abbildung 7 - Navigation "Einführung in die interaktive Arbeit mit Flash"



Abbildung 8 - Navigation "Digitale Bildbearbeitung mit Adobe Photoshop CS2"

Die Navigation zwischen den einzelnen Abschnitten, sollte so realisiert sein, dass das Element beim RollOver raus fährt, so dass der komplette Titel des Abschnitts zu lesen (Abbildung 10) ist. Bei RollOut soll das Navigationselement wieder hereinfahren (Abbildung 9). Das Navigationselement, des aktuell besuchten Abschnitts, sollte

gekennzeichnet und deaktiviert werden (Abbildung 11), so dass ein raus und reinfahren des jeweiligen Elements nicht möglich ist.

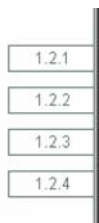


Abbildung 9 -
Ausgangszustand/RollOut

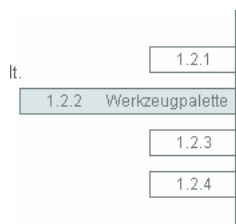


Abbildung 10 - RollOver

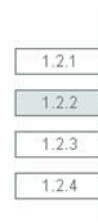


Abbildung 11 - Besucher Abschnitt

```

public function onReleaseAction():Void
{
    this.setActive();
}

public function onRollOverAction():Void
{
    this._x=0-((this.m_mcTextClip.setTextPosX())-100);
}

public function onRollOutAction():Void
{
    this._x=0;
}

```

Quellcode 1 - NavigationButton.as

Die dritte Navigation in unserem Lernmodul wird als Orientierungsleiste im unteren Bereich des UserInterface dargestellt. Mit Hilfe der dritten Navigationsebene ist es möglich in den einzelnen Seiten eines Abschnitts zu navigieren. Hier galt es, vier Zustände durch eine unterschiedliche Farbgebung der Navigationselemente zu unterscheiden. Die aktuell besuchte Seite wurde mit dem Farbwert 0xDBE5E5 gekennzeichnet, die schon besichtigte Seite trägt den Farbwert 0x6D7C7C und die noch nicht besuchte Seite hat den Farbwert 0xFFFFFFFF. Beim RollOver bekommt das Kästchen den Farbwert 0xB6CBCB. Mit Hilfe dieser Zustände behält der Nutzer den Überblick, welche Seiten er bereits besucht hat, wo er gerade ist und welche er noch nicht besucht hat. Realisiert wurde diese Navigation mit Hilfe der Klassen *OrientatationButton.as* und *OrientatationButtonBar.as*. Das sind Dateien, die im Ausgangsmaterial noch nicht vorhanden waren und zusätzlich von uns integriert wurden.

```
public function onReleaseAction():Void
{
    this.setActive();
    this.m_mcTextClipButton.drawFill(m_oFillStyle2);
}

public function onRollOverAction():Void
{
    this.m_oFillStyle5=this.m_mcTextClipButton.getFillStyle();
    this.m_mcTextClipButton.drawFill(m_oFillStyle3);
}

public function onRollOutAction():Void
{
    if(this.m_oFillStyle5 == this.m_oFillStyle2)
    {
        this.m_mcTextClipButton.drawFill(m_oFillStyle4);
    }
    else if (this.m_oFillStyle5 == this.m_oFillStyle1)
    {
        this.m_mcTextClipButton.drawFill(m_oFillStyle1);
    }
    else if (this.m_oFillStyle5 == this.m_oFillStyle4)
    {
        this.m_mcTextClipButton.drawFill(m_oFillStyle4);
    }
    else;
}

public function onSelectableButton(_selectable:Boolean):Void
{
    if(_selectable)
        this.m_mcTextClipButton.drawFill(m_oFillStyle2);
    else
        this.m_mcTextClipButton.drawFill(m_oFillStyle4);
}
```

Quellcode 2 - OrientationButton.as

Hintergrund

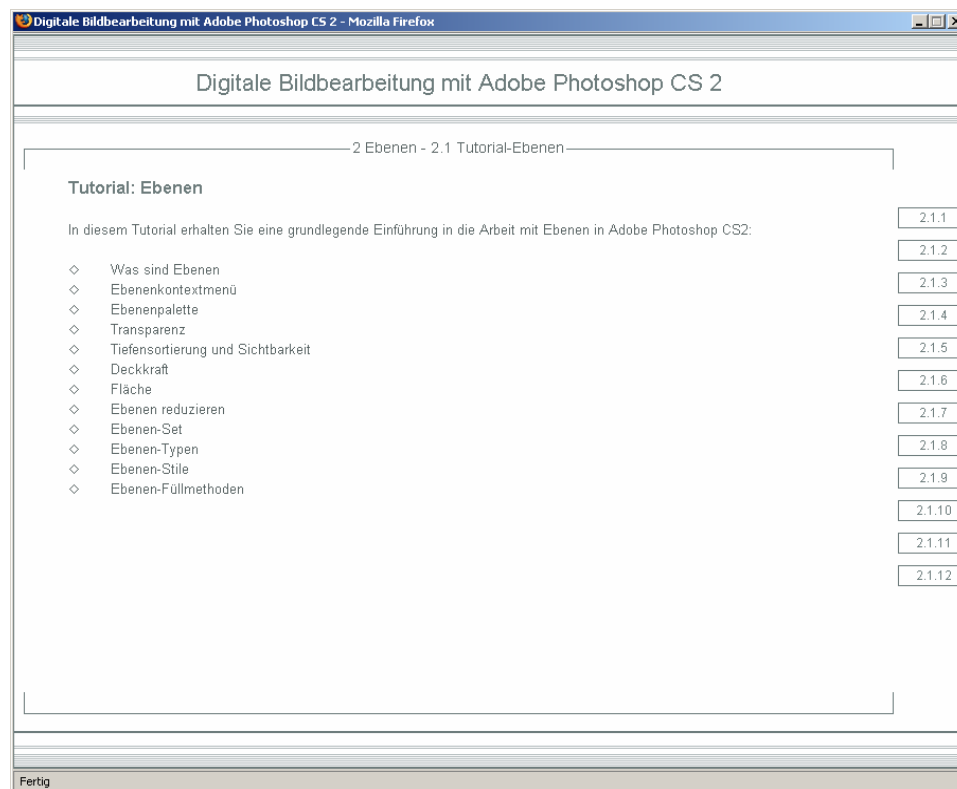


Abbildung 12 – Layout Lernmodul „Digitale Bildbearbeitung“

Um den Hintergrund des Lernmoduls, wie im Layout gezeigt, zu realisieren, musste die Datei *ModulBackground.as* komplett überarbeitet werden. In der *ModulBackground.as* ist geregelt, wie die einzelnen Zeichenelemente aussehen und wo sie dargestellt werden. Wegen der vielen Linien war es möglich an einigen Stellen mit Schleifen zu arbeiten, um nicht jede Linie separat zu zeichnen.

```

var lp:LinePatternDefinition = new LinePatternDefinition();
//Array mit y-Positionen und Linienanzahl der Muster
var pos:Array = new Array([3,7], [22,2], [78,4], [678,2], [686,6]);

for (var y = 0; y<pos.length; y++) {
    lp.definePattern( 0,
                    pos[y][0],
                    modul_w-1,
                    pos[y][0],
                    new LineStyle(1, color_line_light, 100),
                    pos[y][1],
                    0,
                    2);
    formd.addFormDefinition(lp);
}

```

Quellcode 3 - ModulBackground.as

Die Farbe der Linien wird über die `modul_init.xml` geregelt.

```
<bg>
  <colors>
    <color_line>0x6d7c7c</color_line>
    <color_line_light>0xb5bdbf</color_line_light>
  </colors>
  ....
</bg>
```

Quellcode 4 - `modul_init.xml`

XML

Die `modul_init.xml` ist eine der wichtigsten Dateien des Lernmoduls. Über diese Datei werden viele Einstellungsmöglichkeiten realisiert. So zum Beispiel die Farbe der Linien, die Position der Textfelder zum Rand und vieles mehr. Da unsere Datei auf einem völlig anderem Layout beruht als „Flash interaktiv“, musste die `modul_init.xml` in ihren Werten modifiziert werden. Die anderen xml-Dateien des Lernprogramms konnten weitestgehend unberührt bleiben, da dieselben Grundlagen verwendet wurde.

Im Einzelnen wurden folgende Rubriken angepasst:

<bg> - Background

Im Element `<bg>` wurden die Farben (`<color>`) der Linien verändert sowie die Größe (`<size>`) und Position (`<coordinates>`) der Textfelder.

<buttons> - Schaltflächen

Das Element Schaltflächen verschachtelt sich in die Elemente `<symbol>`, `<screenshot>`, `<labeled>`, `<navigation>` und `<orientationbar>`.

- `<symbol>` - Veränderung der Linienfarbe (`<color>`), der Größe der Symbole (`<sizes>`), sowie die Tooltips (`<tooltips>`), die beim RollOver der Symbolschaltflächen erscheinen.
- `<screenshots>` - Damit die Screenshotbuttons passend zu unserem Layout dargestellt werden, mussten Farben (`<colors>`) und Größen dieser Buttons (`<sizes>`) angepasst werden.
- `<labeled>` - Dieses Element und die Funktionalität dieser Buttons wurde für das Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ nicht verwendet und damit auch nicht modifiziert. Die Angaben für diese Funktionalität wurden jedoch in der `modul_init.xml` gelassen, damit bei eventuellen Erweiterungen darauf zurückgegriffen werden kann.

- `<navigation>` - Auch bei den Buttons der Navigation mussten wieder Farbe (`<color>`), Größe (`<size>`) und Position (`<coordinates>`) passend zum Design verändert werden.
- `<orientationbar>` - Dieses Element gab es in der `modul_init.xml` des Lernmodules „Flash interaktiv“ noch nicht. Es wurde integriert, um Farbe, Position und Größe der Orientierungsleiste festlegen zu können.

`<interactions>` - Interaktionen, `<small_player>` - Kleiner Player, `<big_player>` Großer Player

Wie auch schon in den vorhergehenden Elementen wurden bei diesen Elementen Farben (`<color>`), Größen (`<size>`) und Positionen (`<coordinates>`) verändert. Der Wortlaut der Tooltips in den Elementen (`<small_player>`) und (`<big_player>`) wurden wie in der Ursprungsdatei belassen.

`<tooltips>` - Tooltips

Der Wortlaut der Tooltips wurde in den einzelnen Elementen modifiziert bzw. wie in der Ursprungsdatei belassen. Das Element `<tooltip>` ist für das Aussehen der Tooltips verantwortlich. Damit sie in das Layout des Lernmoduls passen, wurden auch hier Farbe (`<color>`) und Größe (`<size>`) entsprechend angepasst.

`<directories>` - Verzeichnisse

Das wichtigste Element der `modul_init.xml` ist das Element `<directories>`, hier sind die Pfade zu den verschiedenen Medien-Ressourcen des Lernmoduls hinterlegt. Da die Verzeichnisstruktur des Lernmoduls anders ist, als bei „Flash interaktiv“ mussten hier die Pfade entsprechend der Struktur angepasst werden.

2.5.2.5 Die wichtigsten Funktionalitäten

Zusammenfassend werden im Folgenden die wichtigsten Funktionalitäten des Lernmoduls noch einmal dargestellt. Außerdem wird eine Zuordnung der wichtigsten Klassen für diese Funktionalitäten erfolgen.

Navigation

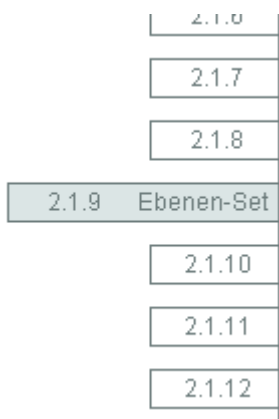


Abbildung 13 – Navigation des Lernmoduls „Digitale Bildbearbeitung“

Die linke Navigation dient dem navigieren zwischen den einzelnen Abschnitten. Beim RollOver über einen Navigationsbutton, fährt dieser heraus und zeigt den vollen Titel des gewählten Abschnitts. Für die Funktionalität dieser Navigation sind folgende Klassen verantwortlich:

NavigationButton.as

NavigationButtonBar.as

Orientierungsleiste



Abbildung 14 – Orientierungsleiste des Lernmoduls „Digitale Bildbearbeitung“

Die untere Navigation, die Orientierungsleiste dient dem navigieren zwischen den einzelnen Seiten eines Abschnitts. Anhand der Farbgebung lässt sich erkennen, welche Seite bereits besucht wurde, welche aktuell angeklickt ist und welche noch nicht besucht wurde. Für die Funktionalität dieser Orientierungsleiste sind folgende Klassen verantwortlich:

OrientationButton.as

OrientationButtonBar.as

Screenshotbutton



Text-Werkzeuge

- ◇ Dienen der Erstellung von Text im Bild
- ◇ Tastenkürzel: T

Abbildung 15 – Screenshot des Lernmoduls „Digitale Bildbearbeitung“

Screenshots dienen dazu, Symbole und Schaltflächen im Programm näher zu erläutern. Die Screenshots wurden als Buttons realisiert, so dass man beim Anklicken eines Symbols oder einer Schaltfläche eine Erläuterung erhält. Die Erläuterung erscheint je nach Screenshot links neben dem Bild oder darunter. Das angeklickte Symbol wird weiß hinterlegt, um es als aktiviert kenntlich zu machen.

Die Klasse, über die diese Funktionalität realisiert ist, ist die:
ScreenshotButton.as

Animationsbildschirm

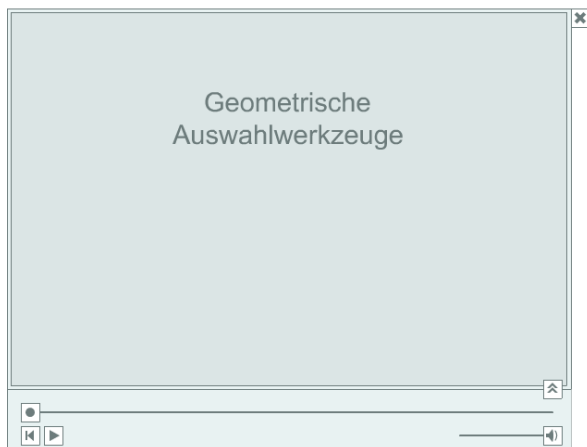


Abbildung 16 – Player des Lernmoduls „Digitale Bildbearbeitung“

Der Animationsbildschirm dient dazu, Videosequenzen und Animationen darzustellen. Die Filme lassen sich über die Steuerungsleiste individuell steuern. So ist es möglich, im Film an eine beliebige Stelle zu springen. Ebenso kann der Film jederzeit angehalten sowie beendet werden. Der Sound lässt sich in der Lautstärke variieren.

Verantwortlich für die Funktionalität des Players ist die Klasse:

ComplexPlayer.as

Kleine Animation

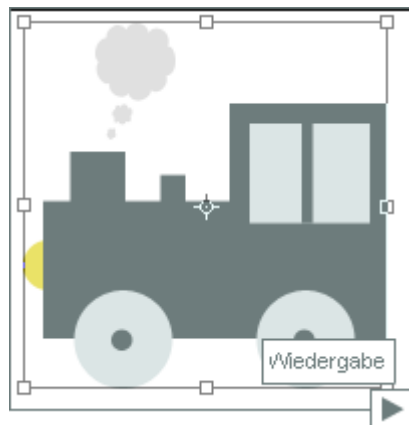


Abbildung 17 - kleiner Player

Der kleine Player dient dazu kleine Animationen abzuspielen, bei denen kein Springen innerhalb des Films notwendig ist. Er muss einen Film starten und an beliebiger Stelle wieder beenden können.

Verantwortlich für die Funktionalität des Players ist die Klasse:

SmallPlayer.as

Grafiken



Abbildung 18 – Grafik des Lernmoduls „Digitale Bildbearbeitung“

Grafiken dienen dazu beschriebene Sachverhalte bildlich darzustellen. Die Grafiken sind mit keinerlei Funktionalität versehen.

Verantwortlich für die Darstellung der Grafiken ist die Klasse:

SimplePageObjects.as

Download



Abbildung 19 – Download des Lernmoduls „Digitale Bildbearbeitung“

Die Downloadfunktion ermöglicht es, die für die Praxisaufgaben notwendigen Dateien herunter zu laden. Die Downloadfunktion wird über ein entsprechendes Symbol angezeigt.

Verantwortlich für die Darstellung dieses Symbols ist die Klasse:

SimplePageObjects.as

2.5.2.6 Verwendung von SCORM 1.2

Um unser Lernmodul sinnvoll in ein LMS einbinden zu können, war es wichtig einen SCORM 1.2 konformen Kurs zu entwickeln. Was genau SCORM konform bedeutet, soll im folgenden Kapitel 3. SCORM detaillierter erläutert werden. Wichtig zu wissen ist, dass ein

SCORM konformes Lernmodul aus einzelnen Lerneinheiten, so genannte SCOs⁵ besteht. In unserem Lernmodul ist ein Unterkapitel gleich einem SCO. In SCORM 1.2 besteht zwischen den einzelnen SCOs keine Verbindung, da die Navigation getrennt vom Inhalt erfolgt. SCORM 1.2 stellt keine Möglichkeit zur Verfügung um zum Beispiel direkt von Kapitel 1.1 zu Kapitel 1.3 zu gelangen, ohne die Navigation des LMS zu benutzen. Ohne die Verwendung von SCORM wäre das kein Problem. Inwieweit SCORM 1.3 diese Einschränkung behebt, wird in den Kapiteln 3 und 4 geklärt.

2.5.3 Technischer Test

Der technische Test des Lernmodules „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ wurde von Frau Jana Halgasch durchgeführt. Im Folgenden soll es eine kurze Erläuterung der Fehler und ihrer möglichen Behebung geben. Dabei wird die Fehlerbeschreibung wörtlich aus dem Fehlerprotokoll übernommen.

Im Fehlerprotokoll von Frau Halgasch (siehe Anhang E) wurde mit der Fehlernummerierung bei Fehler Nr. 3 begonnen. Aus diesem Grund wird auch an dieser Stelle beim Fehler Nr. 3 begonnen.

Fehler Nr. 3

Bei Klick auf die zip-Symbole öffnet sich die Fehler Fehlermeldung: Incorrect Data Typ. Danach wird der File-Dialog korrekt angezeigt. Dieser Fehler trat bei der Verwendung von Firefox 1.5 unter WinXP auf. Da die aktuelle Version des Firefox derzeit bei 2.0.0.4 liegt und dieser Fehler bei anderen Firefox Versionen nicht auftrat, wurde sich hier geeinigt, dass der Fehler nicht korrigiert werden muss.

Fehler Nr. 4

Das letzte Kapitel wird nicht als abgeschlossen markiert, wenn man auf der letzten Seite der Anwendung angekommen ist. Erst nach Bestätigen des Zurückblättern-Buttons der 1. Navigation wird dieser Abschlussstatus erreicht. Dieser Fehler trat unter allen Betriebssystemen mit dem Browser Firefox auf. Auch beim Internet Explorer 7.0 war dieses Problem erkennbar. Die Ursache dafür noch bisher jedoch nicht gefunden werden.

Fehler Nr. 5.1 und 5.2

Fehlermeldung: Not initialized und LMSGetValue(cmi.core.lesson_status)failed. Diese Fehler traten bei allen Betriebssystemen und in allen Browsern auf und betraf ausschließlich die Testlektionen. Die Testlektionen haben seit je her sehr viele Probleme gemacht, da sie nicht wie der Rest des Lernmoduls mit Flash erstellt wurden. Stattdessen

⁵ Die genaue Erklärung dieses Begriffs erfolgt im Kapitel 3.3.1 Content Model.

wurden über das Programm Captivate Flash-Dateien generiert. Vor diesem Hintergrund klappte das Zusammenspiel zwischen den Test-Flashfilmen, dem Lernmodul und SCORM so gut wie gar nicht. Aus Zeitgründen wurde der Bereich Test nach dem technischen Test herausgenommen. Damit wird die Anzahl der Fehler verringert. Die Testlektionen sollen zu einem späteren Zeitpunkt erneut in Flash erstellt werden, damit die Integration reibungsloser verläuft.

Fehler Nr. 6

Für die einzelnen Kapitel und Unterkapitel wird generell kein Abschlussstatus erreicht. Da dieser Fehler ausschließlich bei Flash Player Version 7.0 auftrat und die aktuelle Version derzeit 9.0 ist musste auch diesem Fehler nicht nachgegangen werden, da davon ausgegangen werden kann, dass die Studenten zumindest mit Flash-Player Version 8.0 arbeiten.

Fehler Nr. 7

Fehlermeldung: General Exception wird teilweise mehrmals hintereinander angezeigt (in Kombination mit der Fehlermeldung, Incorrect Data Type), immer wenn die erste Navigation oder das Menü benutzt wird, bei Benutzung der 2. + 3. Navigation trat dieser Fehler nicht auf. Ebenso wie Fehler Nr. 6 ist das ein Fehler, der ausschließlich bei Flash Player Version 7.0 auftrat.

Fehler Nr. 8

Die Navigation funktioniert nicht korrekt: Seiten werden übersprungen, z. B. bei Aufruf von 1.2 über das Menü wird sofort 1.2.4 angezeigt. Auch das ist ein Fehler der nur in Flash-Player Version 7.0 auftrat und somit nicht unbedingt behoben werden muss.

Fehler Nr. 9

Flashdatei wird nicht geladen: Film bleibt bei 100% Lade-Status stehen. Dieser Fehler trat unter dem Betriebssystem Win2000 und dem Internet Explorer 6.0 auf. Allgemein läuft es unter diesen System Bedingungen sehr instabil. Die Ursache dieses Fehlers konnte bisher nicht entdeckt werden.

Fehler Nr. 10

Das Programm konnte nicht geöffnet werden. Folgende Fehlermeldung wurde angezeigt: Der SCORM-Player benötigt mindestens die Browser IE 6.0, Mozilla 1.3 oder Firefox. Dieser Fehler trat bei den Browsern Safari und Opera auf.

Weitere Auffälligkeiten

Die Symbole für zip, pdf, usw. sind nicht durchgängig mit einem aktiven Bereich versehen, d.h. der Mauszeiger ändert sich nur, wenn er sich genau über einer gefüllten Stelle befindet. In den Bereichen ohne Zeichnung findet dieser Zeigerwechsel nicht statt. Dieses Problem wurde behoben. Die Icons, die als Buttons genutzt werden, waren nicht vollständig gefüllt und hatten transparente Bereiche, die nicht als Fläche erkannt wurden. Ein Kapitel wird auch als abgeschlossen markiert, wenn Seiten über die 3. Navigation nicht aufgerufen wurden. Kapitel werden ebenso für beendet erklärt, wenn man den Test nur aufruft, aber nicht bearbeitet hat. Das Problem mit dem Test ist hinfällig, da es im Moment keine Tests im Lernmodul gibt. Das Lernmodul ist so eingerichtet, dass geprüft wird, ob alle Abschnitte angeschaut wurden, es ist nicht wichtig, ob auch jede einzelne Seite angeschaut wurde. Das Prüfen der einzelnen Seiten hat sich als sehr, sehr schwierig herausgestellt, wodurch im Lernmodul nur auf Abschnitte geprüft wurde.

3. SCORM

Im Folgenden Kapitel soll die aktuelle Version des Sharable Content Object Reference Model erläutert werden. Es werden die einzelnen SCORM-Bücher kurz beschrieben, um einen kompletten Überblick zu bekommen. Das Thema Navigation Model des Sequencing and Navigation Books ist wichtigster Bestandteil dieser Diplomarbeit, daher wird diesem Thema ein separates Kapitel gewidmet.

3.1 Überblick

Seit Jahren wird nach Standardisierungen im Bereich des E-Learnings und der Metadaten geforscht. Die Standardisierungen werden von verschiedenen Organisationen, Initiativen und Projekten bearbeitet. Daher entwickelte sich eine unüberschaubare Anzahl von Standards. Dem Wunsch nach Vereinheitlichung für den Austausch von elektronischen Lerneinheiten, kommt die ADL-Initiative (Advanced Distributed Learning – Initiative des amerikanischen Verteidigungsministeriums) mit dem Sharable Content Object Reference Model nach. Dabei versucht die ADL mit SCORM⁶ die Arbeiten von vier Standardisierungsgruppen zusammenfließen zu lassen.

- **AICC** (Aviation Industry Computer-Based Training Committee – <http://www.aicc.org>)
- **ARIADNE Foundation** (Alliance of Remote Instructional Authoring and Distribution)
- **IEEE LTSC** (Learning Technology Standards Committee, eine Einrichtung des Institute of Electrical and Electronic Engineers – <http://www.ieee.org>)
- **IMS Global Learning Consortium** (<http://www.imsglobal.org>)

Das Sharable Content Object Reference Model soll folgende Anforderungen erfüllen:

- **Wiederverwendbarkeit:**
Unabhängig von den Entwicklungswerkzeugen können Lerninhalte in unterschiedlichen Applikationen und auf verschiedenen Plattformen weiter bearbeitet und verwendet werden.
- **Zugänglichkeit:**
SCORM-konforme Kurse sind für verschiedene Personen an unterschiedlichen Orten jederzeit erreichbar
- **Interoperabilität:**
Erstellte Lerninhalte sind plattform-, software- und betriebssystemunabhängig.

⁶ Die aktuelle Version der SCORM-Spezifikation ist 1.3 (Stand 14. September 2004)

- **Beständigkeit:**

Auch nach Weiterentwicklung der Basistechnologie sollen die Kurse möglichst uneingeschränkt verwendbar bleiben.

- **Anpassungsfähigkeit:**

Anweisungen können an individuelle und situationsabhängige Bedürfnisse angepasst werden

- **Erschwinglichkeit:**

Kosten- und Arbeitsaufwand sollen gesenkt, aber die Lerneffizienz gesteigert werden.

[Pie05, S. 16]

Die verschiedenen SCORM-Bücher

SCORM besteht aus vier Dokumenten, die zusammengehörende und aneinander angepasste Sammlungen von Spezifikationen und Standards darstellen und in der SCORM-Terminologie Bücher genannt werden.

1. Overview
2. Content Aggregation Model (CAM)
3. Run-Time Environment (RTE)
4. Sequencing and Navigation (SN)

Diese Einteilung der Bücher existiert erst seit Version 1.2, die im Oktober 2001 veröffentlicht wurde. Damals waren es noch drei Bücher. Seit Version SCORM 2004 (Januar 2004) gibt es zusätzlich das Buch Sequencing und Navigation. Seit dieser Version wird auch jedes Buch eigenständig versioniert, um Änderungen und Korrekturen übersichtlicher zu gestalten.

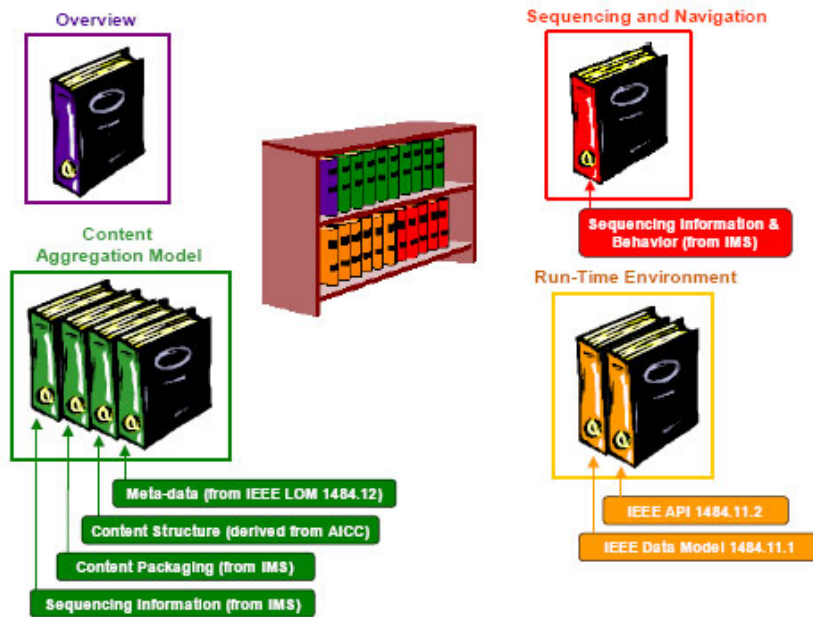


Abbildung 20 - SCORM-Bücher [SCO04a]

3.2 Overview

Das Buch Overview ist das Überblicksdokument. In diesem Dokument wird die Historie und Vision von SCORM beschrieben. Es gibt eine Einführung in die Bücher „Content Aggregation Model (CAM)“, „Run-Time Environment (RTE)“ sowie in das Buch „Sequencing and Navigation (SN)“. Außerdem werden in „Overview“ die Zusammenhänge zwischen den Büchern deutlich.

3.3 Content Aggregation Model (CAM)

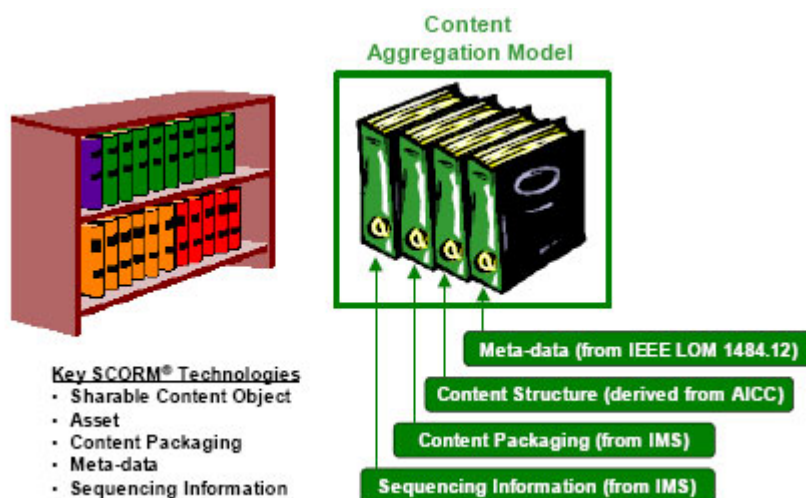


Abbildung 21 - Content Aggregation Model (CAM) [SCO04b]

Das "Content Aggregation Model" beschreibt die Ressourcen (Assets und SCOs) die in Lernpaketen verwendet werden können. Es zeigt Möglichkeiten zur Zusammenfassung und Strukturierung von Ressourcen zu wieder verwendbaren Lernobjekten oder „Sharable Content Objects“ (SCO). Es können Dateien in den unterschiedlichsten Formaten sowie URLs als Ressourcen verwendet werden. Die Organisation der Ressourcen erfolgt in Hierarchien, die „Organizations“ genannt werden. Für die strukturierten Lernpakete können Metadaten spezifiziert werden. Diese ermöglichen anhand von Beschreibungen und Schlüsselwörtern eine gezielte Suche nach Lerninhalten. Das Gesamtpaket wird durch eine so genannte Manifestdatei im XML-Format definiert. Außerdem definiert das Content Aggregation Model die Ablaufsteuerung und die Navigation von wieder verwendbaren Lerneinheiten.

Das Content Aggregation Model ist in folgende Teile untergliedert:

- Content Model
- Content Packaging
- Metadaten
- Sequencing and Navigation

3.3.1 Content Model

Das Content Model definiert die inhaltlichen Komponenten Asset und SCO, sowie deren Aggregation zu komplexeren Einheiten.

Asset

Ein Asset ist die kleinste, inhaltlich abgeschlossene Form einer Lernressource. Es ist wiederverwendbar. Mehrere Assets können miteinander zu einem weiteren Asset oder einem SCO kombiniert werden. Ein Asset kann jede Art von elektronischen Medien repräsentieren, die auch ein Webbrowser verarbeiten und darstellen kann. Als Beispiele sind hier Texte, Bilder, Klänge, Musik und JavaScript-Funktionen zu nennen.

SCO

Ein SCO (Sharable Content Object) kann eine einzelne Einheit, eine Lektion oder gar ein ganzes Modul in einem Kurs sein. Es besteht aus einem oder mehreren Assets oder Dateien. Nicht jede Datei muss extra als Asset definiert sein.

Der Unterschied zwischen SCO und Asset ist, dass ein SCO in der Lage ist mit dem LMS über eine Laufzeitumgebung zu kommunizieren. Das heißt, ein SCO muss in der Lage sein, die vom LMS bereitgestellte API-Instanz ausfindig zu machen und mindestens eine API-Methode – die Methode zum Initialisieren und Beenden der Kommunikation - aufzurufen. Je nach Inhalt können auch weitere API-Methoden vom SCO aufgerufen werden. API – Application Programming Interface - wird in 3.4.2 API näher erläutert.

Reine Informationsseiten können als Assets abgespeichert werden. Sobald sie allerdings der Auswertung dienen müssen sie als SCO angelegt werden. Damit zum Beispiel die Ergebnisse einer Testseite ausgewertet werden können, muss die Seite um die entsprechende JavaScript API-Methode erweitert werden.

SCOs sollten unabhängig vom Gesamtkontext sein, sowie inhaltlich abgeschlossen und sinnvoll, damit die Wiederverwendbarkeit gewährleistet ist.

Content Organization

Die Content Organization dient dazu, die Ablaufsteuerung und das Navigationsverhalten zwischen den Lerneinheiten zu beschreiben. Dazu bildet die Content Organization die Struktur der Inhalte eines Lernmoduls ab. Die Strukturierung passiert baumartig in einem SCORM standardisierten XML-Schema, in der sogenannten Manifest-Datei. Zur Laufzeit wird diese Struktur vom LMS interpretiert. Die Wurzel dieser Struktur bildet das `<organization>`-Element. Unter diesem Element folgen `<item>`-, die verschachtelt sein können. Jedes Element, ob `<organization>` oder `<item>` ist eine Activity, an die Sequencing-Informationen gebunden sein können.

Manifest

Die Manifest-Datei, ein XML Dokument, beschreibt die Struktur des Inhalts sowie Ressourcen eines Content Packages. Sie wird unter dem Namen `imsmanifest.xml` gespeichert und befindet sich im Wurzelverzeichnis des Content Packages. Die Gültigkeitsbereiche dieser Datei können verschieden sein und richten sich nach dem Umfang des Content Packages. Das heißt, die Manifestdatei kann:

- einen kompletten Kurs,
- einen Teil eines Kurses,
- eine Sammlung von Kursen oder gar
- nur eine Sammlung von Inhalten ohne Struktur

beschreiben.

Beschreibt eine Manifestdatei nur einen Teil eines Kurses, beinhaltet ein Package mehrere Manifeste. Dabei enthält ein Package ein Top-Level-Manifest, welches für das gesamte Package steht, sowie ein oder mehrere Sub-Manifeste.

Eine Manifestdatei ist in drei bis vier Abschnitte gegliedert.

- `<metadata>` - Metadaten. Sie dienen der besseren Identifikation und Wiederverwendbarkeit der Lerninhalte. Sie können direkt in der Manifestdatei festgelegt werden oder in einer externen XML-Datei, die über `<location>` referenziert wird.
- `<organization>` - In diesem Abschnitt wird die Struktur des Packages festgelegt.

- `<resources>` - Dieser Bereich enthält eine Auflistung der Ressourcen, auf welche innerhalb der Content Organization verwiesen wurde. Innerhalb dieses Bereiches wird auf die physischen Dateien verwiesen.
- `<imsss:sequencingCollection>` - Der vierte Abschnitt ist abhängig davon, ob man mehrfachverwendbare Sequencing Objekte benötigt. Er existiert erst seit der Einführung des Simple Sequencing.

3.3.2 Content Packaging

Das Content Packaging beschreibt, wie die Lerneinheiten in einem standardisierten Format strukturiert und gespeichert werden. Diese Standardisierung dient der Wiederverwendbarkeit der Lernmodule in verschiedenen LMS.

In einem Content Package sind neben der Manifest-Datei noch die physischen Ressourcen eines oder mehrerer Lernprogramme enthalten. Auch externe Ressourcen sind per URL-Angabe im Content Package möglich. Das Content Package wird als eine Archivdatei im Zip-Format verpackt und kann nur als solche in ein LMS importiert werden.

Nr. Name

1	Manifest				
	1.1	Identifier			
	1.2	Version			
2	Metadata				
	2.1	Schema			
	2.2	Schema Version			
	2.3	Location			
	2.4	(Meta-data)			
3	Organizations				
	3.1	Default			
	3.2	Organization			
		3.2.1	Identifier		
		3.2.2	Structure		
		3.2.3	Title		
		3.2.4	Item		
			3.2.4.1	Identifier	
			3.2.4.2	Identifier Ref	
			3.2.4.3	Is Visible	
			3.2.4.4	Parameters	
			3.2.4.5	Title	
			3.2.4.6	Item	
			3.2.4.7	Metadata	
				3.2.4.7.1	Schema
				3.2.4.7.2	Schema Version

			3.2.4.7.3	Location
			3.2.4.7.4	{Meta-data}
		3.2.4.8	Prerequisites	
			3.2.4.8.1	Type
		3.2.4.9	Max Time Allowed	
		3.2.4.10	Time Limit Action	
		3.2.4.11	Data From LMS	
		3.2.4.12	Mastery Score	
	3.2.5	Metadata		
		3.2.5.1	Schema	
		3.2.5.2	Schema Version	
		3.2.5.3	Location	
		3.2.5.4	{Meta-data}	
4	Resources			
	4.1	Resource		
		4.1.1	Identifier	
		4.1.2	Type	
		4.1.3	HREF	
		4.1.4	SCORM Type	
		4.1.5	Metadata	
			4.1.5.1	Schema
			4.1.5.2	Schema Version
			4.1.5.3	Location
			4.1.5.4	{Meta-data}
		4.1.6	Dependency	
			4.1.6.1	Identifier Ref
5	(Sub-)Manifest			

Tabelle 2 - Datenelemente in einem Content Package [ELA07]

3.3.3 Metadaten

Die Metadaten dienen der Beschreibung der Content Model Komponenten sowie des Content Package. Durch sie ist eine eindeutige Identifizierung und Katalogisierung möglich. Außerdem ermöglichen die Metadaten eine Such nach Lerneinheiten in einer Datenbank. Die Metadaten sind in neun Kategorien mit mehreren Unterelementen eingeteilt, die in der folgenden Tabelle kurz vorgestellt werden:

Kategorie	Beschreibung
<general>	Allgemeine Informationen, welche die Ressource als Ganzes beschreiben (z. B. Titel, Stichworte, Beschreibung)
<lifeCycle>	Geschichte und aktueller Stand der Ressource, sowie an der Entwicklung beteiligte Personen
<metaMetadata>	Informationen über den Metadatensatz selbst
<technical>	Beschreibung der technischen Anforderungen und Charakteristika der Ressource
<educational>	Beschreibung der pädagogischen Charakteristika der Ressource
<rights>	Informationen über Nutzungsrechte, Kosten, Copyright
<relation>	Definition von Beziehungen der Ressource zu anderen Ressourcen
<annotation>	Kommentare zum pädagogischen Nutzen der Lernressource
<classification>	Einordnung der Ressource in ein Klassifizierungsschema

Tabelle 3 - Metadatenkategorien [Ger04]

3.3.4 Sequencing and Navigation

Dieser Abschnitt beschreibt, wie die verschiedenen Sequencing and Navigation Strategien in die Manifest-Datei eingefügt werden. Dieser Teil ist erst seit Version 1.3 im CAM zu finden. Da das Thema Navigation ein wichtiger Bestandteil dieser Arbeit ist, wird an späterer Stelle noch einmal genauer darauf eingegangen.

3.4 Run-Time Environment (RTE)

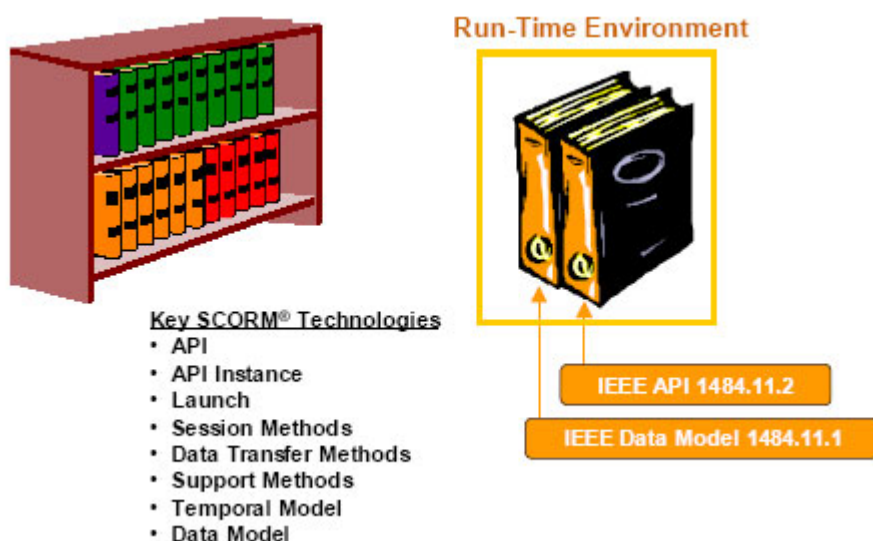


Abbildung 22 - Run-Time Environment [SCO04c]

Das „Run-Time Environment“ beschreibt das laufzeitbezogene Verhalten eines Learning Management Systems (LMS). Es bildet die Schnittstelle zwischen dem LMS und den CAM spezifizierten Lernpaketen. Außerdem ermöglicht es die Verwendung von Benutzerdaten zur Laufzeit, um Benutzer-Präferenzen, Lernfortschritte und Lernziele abzubilden. Mit Hilfe der Run-Time-Environment ist es möglich, dass einmal erstellte Lerninhalte ohne Veränderungen in verschiedenen LMS ausgeführt werden können.

Das RTE Book ist in drei Teile gegliedert:

- Launch
- API
- Datenmodell

3.4.1 Launch

Im RTE Book ist das Verhalten von LMS beim Starten von Lernressourcen definiert. Die verschiedenen Content Objekte stellen beim Starten unterschiedliche Anforderungen an das LMS, welches diese zu beachten hat. Als Content Objects werden SCOs und Assets bezeichnet.

Ein Asset muss über das http-Protokoll gestartet werden, dass ist die einzige Anforderung, die dieses Content Object an das LMS stellt.

Ein SCO muss im Child-Fenster bzw. in einem Child-Frame des Fensters gestartet werden, welches die API-Instanz enthält. Das heißt, ein SCO muss aus einem Fenster heraus gestartet werden, dass die API-Instanz enthält. Dabei kann es in einem Frame in diesem Fenster gestartet werden, oder es öffnet ein neues Fenster. Dies muss geschehen, um zu gewährleisten, dass das SCO eine Verbindung zum LMS aufbauen kann. Es darf nie mehr als ein SCO gleichzeitig gestartet werden. Über Sequencing Mechanismen bzw. das LMS wird das SCO identifiziert, was nach der Abarbeitung des aktuellen SCOs gestartet werden soll.

3.4.2 API – Application Programming Interface

Das API ermöglicht einen standardisierten Datenaustausch zwischen SCO und LMS. Dafür stellt das API Schnittstellen in Form von Funktionen zur Verfügung. Somit muss der Entwickler sich nicht mit Schnittstellen beschäftigen, sondern kann diese Funktionen nutzen. Der Zugriff durch ein SCO auf die API-Instanz erfolgt über JavaScript. Wie zuvor beschrieben muss ein SCO mindestens in der Lage sein, API-Methoden zum Initialisieren und Beenden der Kommunikation aufzurufen, dass heißt es muss die Methoden `Initialize()` und `Terminate()` aufrufen. Ein SCO ist die einzige Komponente im SCORM Content Model, die in der Lage ist mit dem LMS über eine bereitgestellt API-Instanz zu kommunizieren. Die API-Instanz muss vom SCO beim Starten lokalisiert werden.

3.4.3 Datenmodell

Zur Laufzeit werden über die API-Instanz Informationen zwischen SCO und LMS ausgetauscht. Dazu stellt das RTE-Datenmodell einen festgelegten Satz von Elementen

zur Verfügung. Durch die Benutzung eines einheitlichen Datenmodells ist die übereinstimmende Verarbeitung von Informationen in verschiedenen LMS gewährleistet. Im RTE-Datenmodell können verschiedene Informationen gespeichert und abgerufen werden: U. a. Daten über den Lernenden, über seine Interaktionen mit dem Lerninhalt, über den Lernerfolg oder den Abschluss von SCOs. Diese Daten sind auch Grundlage für LMS-interne Sequencing-Entscheidungen, da die Werte einiger Elemente in die Ablaufsteuerung einbezogen werden.

Alle Elemente des RTE-Datenmodells müssen von einem SCORM-konformen LMS unterstützt werden, in der Version 1.2 waren nur bestimmte Elemente obligatorisch. Für SCOs sind weiterhin alle Elemente optional. Zur Kennzeichnung des zugrunde liegenden Datenmodells beginnen alle Elemente mit „cmi“, danach folgt der Elementenname entsprechend des hierarchischen Aufbaus des Modells. Jedem Datenelement ist ein Datentyp zugeordnet [Ger04]. Die einzelnen Datenelemente sind als Anhang B in der Diplomarbeit von Anne-Kathrin Gericke „Untersuchung und Bewertung von SCORM 2004 sowie prototypische Realisierung von Online-Kursmaterial nach dieser Spezifikation“ aufgelistet. Die in SCORM 2004 neu hinzugekommenen Elemente werden ab Seite 29 der Diplomarbeit von Anne-Kathrin Gericke näher erläutert.

3.5 Sequencing and Navigation (SN)

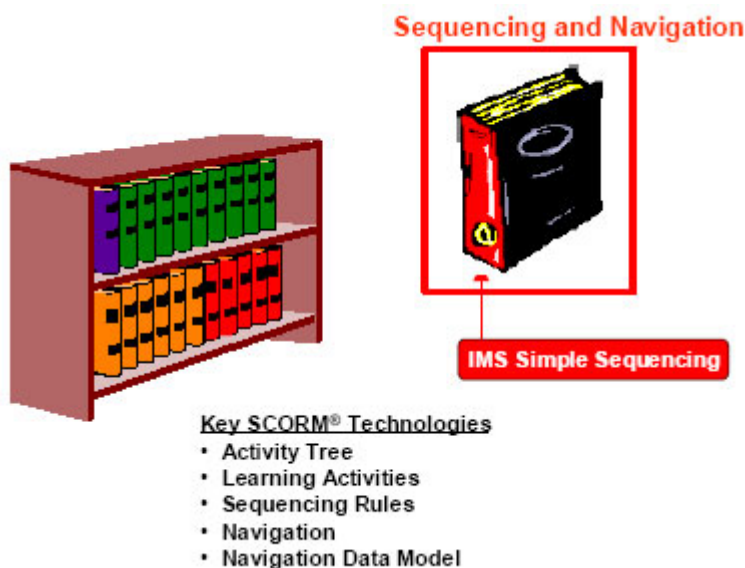


Abbildung 23 - Sequencing and Navigation (SN) [SCO04d]

Das Buch Sequencing and Navigation beinhaltet folgende Punkte:

- Sequencing Konzepte und Terminologie
- Sequencing Definition Model
- Sequencing Behavior Model
- Navigation Controls and Requirements
- Navigation Data Model

In diesem Teil des dritten Kapitels soll nur auf die Punkte

- Sequencing Konzepte und Terminologie
- Sequencing Definition Model
- und Sequencing Behavior Model

eingegangen werden, da das Navigation Model gesondert in Kapitel 4 bearbeitet wird.

Durch „Sequencing and Navigation“ kann eine individuelle Navigation Programm- bzw. Lerner gesteuert erzeugt werden. Das heißt, das LMS wertet die Nutzung des Lernmoduls und gestellte Aufgaben zur Laufzeit aus, um dem Nutzer den optimalen Weg durch das Lernmodul, angepasst an seine Fähigkeiten, zu zeigen. Bevor genauer auf das Thema Sequencing des SN-Buches eingegangen wird, soll es zunächst einen kurzen Überblick über Sequencing und Navigation geben, um den Einstieg in dieses Gebiet zu erleichtern.

SCORM Sequencing Überblick

Das SCORM Sequencing basiert auf der IMS Simple Sequencing Spezifikation. Diese Navigation definiert, wie das gewünschte Ablaufverhalten innerhalb eines Lernangebots festgelegt wird, so dass es in jedem SCORM konformen LMS gleichermaßen umgesetzt wird. Im SN Book werden die Anwendung der IMSSpezifikation und die SCORM-spezifischen Erweiterungen beschrieben. Durch die Einführung des Sequencing ist es möglich, die Ablaufsteuerung vom Inhalt zu trennen, so dass Lernmaterialien wiederverwendbar sind und der Ablauf individuell gestaltet werden kann.

SCORM Navigation Overview

Das SCORM Navigation Model ist nicht Bestandteil der IMSSpezifikation. Es wurde von ADL zusätzlich erarbeitet. Ebenso wie das Sequencing Model bezieht sich die Navigation nur auf die Navigation zwischen verschiedenen Activities und nicht auf die Navigation innerhalb eines Content Objects. Das Navigation Model besteht aus zwei Teilmodellen. Das Navigation Data-Model ist dafür verantwortlich, dass Navigation Requests vom SCO zum LMS kommuniziert werden können. Mit Hilfe des Presentation Information Model wird gesteuert, welche UI-Devices nicht angezeigt werden sollen, während das Content Object angezeigt wird. Mit dieser Funktion vermeidet man redundante UI-Devices. Im

Kapitel 4 soll genauer auf das Thema Navigation Model und die beiden Teilmodelle eingegangen werden. Dort sollen dann auch die Möglichkeiten dieses Models getestet und prototypisch umgesetzt werden.

3.5.1 Sequencing Konzept und Terminologie

Die wichtigsten Konzepte des Sequencing sind die Activity, der Activity Tree und die Objectives und sollen im Folgenden erklärt werden.

3.5.1.1 Activity

Eine Activity ist das, was der Lernende während der Abarbeitung eines Lernangebotes macht, das heißt eine Lernhandlung. Auf Activities wird das Sequencing angewendet. Über Sequencing Informationen und Interaktionen des Lernenden ermittelt das LMS die Abfolge der einzelnen Activities. Man unterscheidet Blatt-Activity und Sub-Activities. Eine Blatt Activity bezieht sich auf eine Lernressource oder besteht aus weiteren Sub-Activities. Die Sub-Activities können wiederum Blatt-Activities sein oder aus Activities aufgebaut sein. Das heißt die Activities sind verschachtelt. Es gibt keine Begrenzung für die Verschachtelungstiefe.

Die folgende Abbildung zeigt eine Activity mit Sub-Activities.

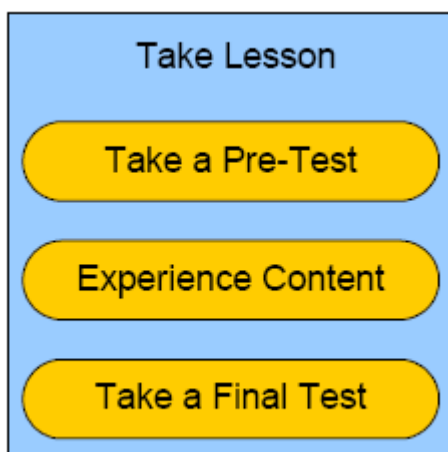


Abbildung 24 - Activity mit Sub-Activities [SCO04d, S. SN-2-9]

Die Activity „Take Lesson“ ist aus den Sub-Activities „Take a Pre-Test“, „Experience Content“ und „Take a Final Test“ aufgebaut.

Verschachtelte Activities werden als Cluster bezeichnet. Ein Cluster beinhaltet ein „Eltern“-Activity und die dazu gehörigen „Kinder“-Activities. Hat eine „Kind“-Activity wiederum eigene Kinder, bilden diese ein neues Cluster. Die folgende Grafik zeigt fünf einfache Cluster.

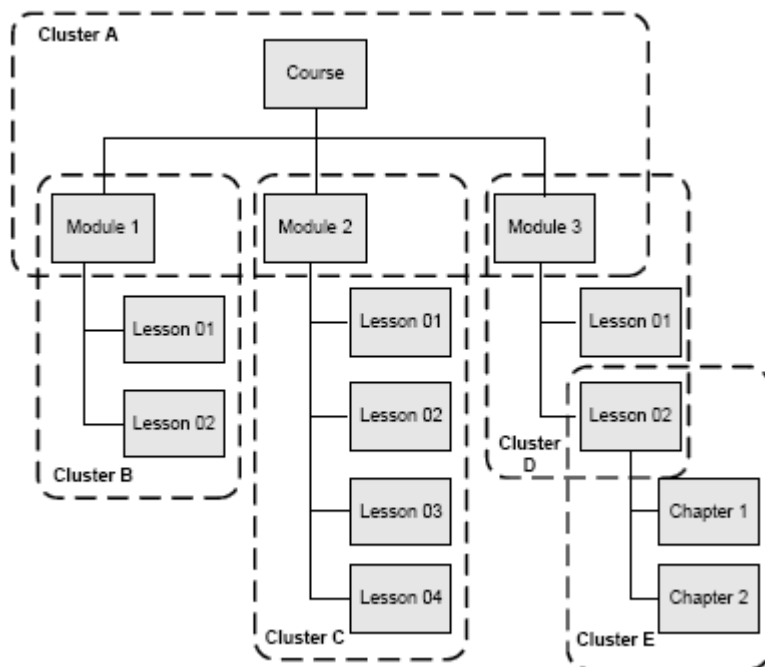


Abbildung 25 - Cluster Beispiel [SCO04d, S. SN-2-7]

Cluster A, der Course Cluster, beinhaltet die vier Aktivitäten. Die Course Aktivität und die Eltern Aktivitäten der Cluster B, C und D. Cluster B besteht aus drei Aktivitäten: Die Modul 1 – Aktivität sowie die Kinder Lesson 01 und Lesson 2. Die Aktivität Modul 2 sowie die Kinder-Aktivitäten Lesson 01-04 bilden zusammen Cluster C. Cluster D setzt sich wieder aus drei Aktivitäten zusammen. Die Eltern Aktivität bildet dabei Modul 3. Die Kinder-Aktivitäten sind Lesson 01 und Lesson 02. Lesson 02 wiederum bildet die Eltern-Aktivität für Cluster E. Dieses Cluster besteht aus 3 Aktivitäten – der eben genannten Aktivität Lesson 02 sowie aus den Kindern Chapter 1 und 2.

Activities stehen in direktem Zusammenhang mit der Content Organization und ihre Struktur wird in einem Activity Tree beschrieben. [Ger04]

3.5.1.2 Activity Tree

Abbildung 26 zeigt ein Beispiel für einen einfachen Activity Tree.

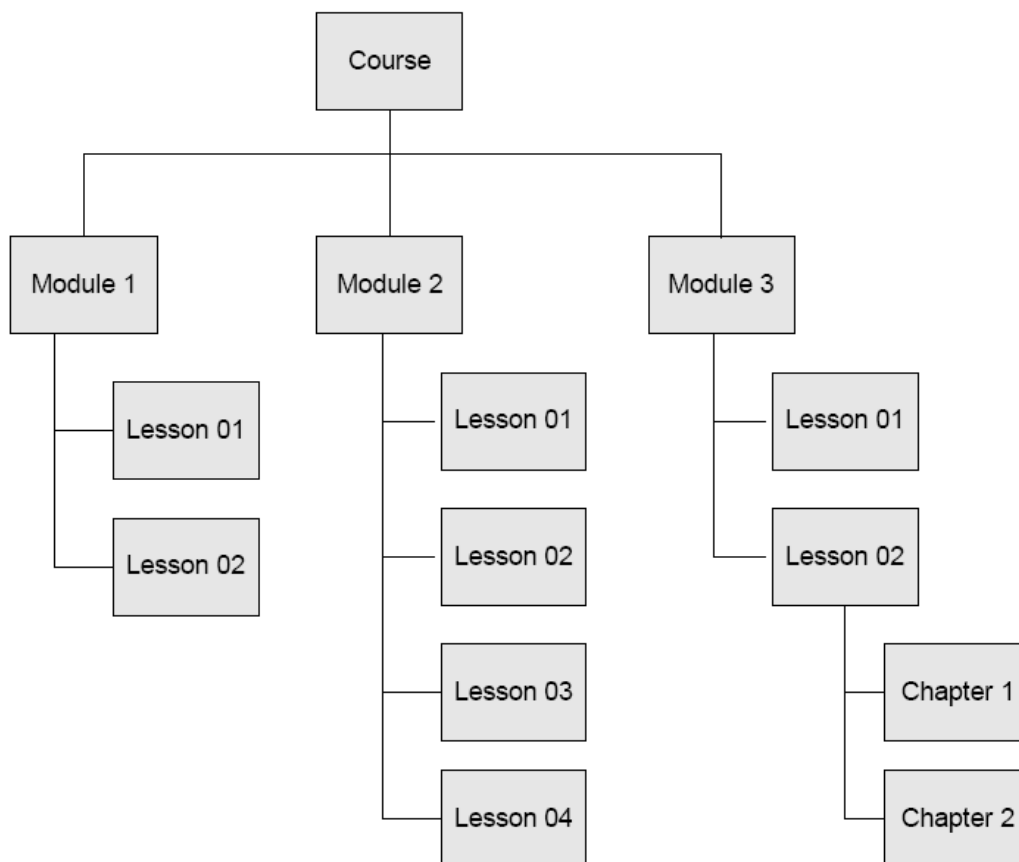


Abbildung 26 –Beispiel eines einfachen Activity Tree [SCO04d, S. SN-2-3]

Der Activity Tree wird aus der Content Organization abgeleitet. Dabei bildet das `<organization>` - Element die Wurzel-Activity des Baumes. Ein `<item>`-Element entspricht einer Activity.

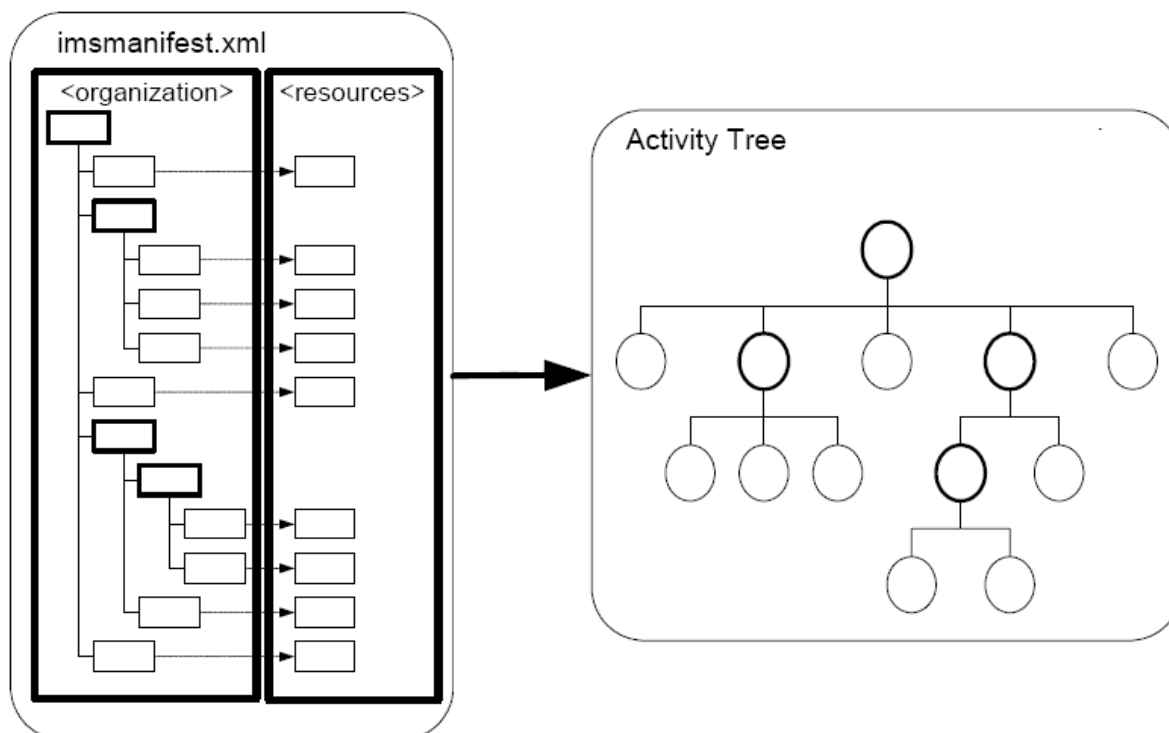


Abbildung 27 - Ableitung des Activity Tree aus der Content Organization, nach [SCO04d, S. SN-2-5]

Mit Hilfe des Activity Tree sind die Sequencing-Informationen unabhängig vom Lerninhalt. Außerdem kann das LMS aus dem Activity Tree das Navigationsmenü für die Lernumgebung generieren.

3.5.1.3 Objectives

Objectives sind „Variablen“ mit denen Statuswerte gespeichert werden um von Activities genutzt zu werden. Die Activities können Objectives gemeinsam nutzen und mit einem oder mehreren Objectives verbunden werden. Mit Hilfe der Objectives ist es möglich Lernerfolge festzuhalten.

Man unterscheidet lokale und globale Objectives:

- Lokale Objectives – Können nur von Activities genutzt werden, an die sie auch gebunden sind.
- Globale Objectives – Sind allen Activities zugänglich und nicht an eine gebunden.

3.5.2 Sequencing Definition Model

Das SCORM Sequencing Definition Model ist ein Informationsmodell, welches von der IMS Simple Sequencing (SS) Specification abgeleitet wurde. Das SCORM Sequencing Definition Model besteht aus einer Reihe von Elementen, die von einem Entwickler genutzt werden können, um Sequencing Strategien umzusetzen. Der Entwickler definiert

mit den Elementen des Sequencing Models, welche Inhalte präsentiert werden und wann diese präsentiert werden. Er übermittelt somit die nötigen Informationen für die Steuerung des Ablaufs und der Bereitstellung der Activities über das LMS.

Jedem `<item>`- Element und dem `<organization>`-Element können Sequencing Informationen zugeordnet werden, denn diese Elemente entsprechen Activities. Die Zuordnung geschieht über das Element `<sequencing>`, wobei jedes Element einen default-Wert hat, wenn kein anderer Wert gesetzt wird.

Um die Herkunft des Elements aus der IMS-Spezifikation deutlich zu machen bekommt jedes Sequencing Element das Namensraum-Präfix `imsss`. Im Folgenden sollen die drei wichtigsten Steuerungsmöglichkeiten kurz erläutert werden. Eine ausführliche und vollständige Erläuterung der Steuerungsmöglichkeiten und ihrer Anwendung finden sie in der Diplomarbeit von Anne-Kathrin Gerike [Ger04, S. 44].

3.5.2.1 Sequencing Control Modes

Über diese Steuerungsmöglichkeit werden die Inhalte des Navigationsmenüs beeinflusst. Es kann darüber festgelegt werden, wie der Weg durch den Activity Tree zurück gelegt werden kann und welche Activities direkt im Menü angewählt werden können. Das Element dafür in der Manifest-Datei ist `<controlMode>`. Dieses Element enthält folgende Attribute die mit `true` oder `false` belegt werden können.

- `Choice` – Wenn `choice = true`, kann jede Activity innerhalb des Clusters in beliebiger Reihenfolge ausgewählt werden.
- `ChoiceExit` – Wenn `choiceExit = false`, ist es nicht möglich per Choice Navigation Request eine andere Activity auszuwählen und somit die aktuelle zu beenden. An dieser Stelle ist es nur über vor- bzw. zurück-blättern möglich die aktuelle Activity zu verlassen.
- `Flow` - Wenn `flow = true` ist das Vor- und Zurück Blättern mit den entsprechenden Button möglich.
- `forwardOnly` – Die Activities des Clusters können nur vorwärts durchlaufen werden, wenn `forwardOnly = true`.
- `useCurrentAttemptObjectiveInformation` – Wenn das Attribut gleich `true`, soll das LMS die Lernerdaten des aktuellen Versuchs verwenden.
- `useCurrentAttemptProgressInformation` – Wenn das Attribut gleich `true`, soll das LMS die Lernerdaten des aktuellen Versuches speichern.

3.5.2.2 Sequencing Rules

Mit den Sequencing Rules kann festgelegt werden, wann Activities übersprungen werden, ob und wann sie angezeigt werden im Menü und dann sie wieder verlassen werden. Das Element dafür lautet: `<sequencingRules>`. Für eine Activity können eine oder mehrere solcher Sequencing Rules festgelegt werden.

Eine Sequencing Rule besteht aus ein oder mehreren Bedingungen und einer Aktion. Der Bedingungsteil dieser Regel gibt `true` oder `false` zurück. Ist die Bedingung erfüllt, wird die Aktion ausgeführt.

Da das Element Sequencing Rules sehr umfangreich ist sollen folgend nur die drei Typen kurz genannt werden. Ausführlichere Informationen sind bei [Ger04, S. 49ff] zu finden.

Entsprechend dem Zeitpunkt Ihrer Prüfung, gibt es drei Typen für Sequencing Rules:

- Pre-Condition-Rule: wird ausgewertet, wenn die nächste bereitzustellende Activity ermittelt wird (Element `<preConditionRule>`)
- Post-Condition-Rule: wird ausgewertet, wenn der Zugriff auf die Activity beendet wird. (Element `<postConditionRule>`)
- Exit-Condition-Rule: wird ausgewertet, wenn der Zugriff auf eine untergeordnete Activity beendet wird (Element `<exitConditionRule>`)
-

3.5.2.3 Rollup Rules

Cluster Activities sind nicht direkt mit SCOs verbunden, daher gibt es für sie keinen direkten Lernerdatenaustausch (Tracking Informationen). Je nach Tracking Informationen der Blatt-Activities innerhalb eines Clusters, können mit Hilfe von Rollup Rules Informationen für die Parent-Activities ermittelt werden. Es werden den die Informationen von der Kind-Activity zur Cluster-Activity „hochgerollt“. Je Cluster-Activity können ein oder mehrere Rollup Rules zugeordnet werden. Element in der Manifestdatei ist: `<rollupRules>`.

3.5.3 Sequencing Behavior Model

Das Sequencing Behavior beschreibt, wie das LMS auf die im Sequencing Definition Model definierten Ereignisse reagieren muss. Dieses Verhalten wird aus dem Tracking Model, dem Activity State Model und dem Sequencing Definition Model hergeleitet.

3.5.3.1 Tracking Model

Früher gab es nur das Datenmodell der Laufzeitumgebung, da wurden Interaktionen des Lernenden mit dem SCO registriert. Damit Sequencing-Prozesse verarbeitet werden

können, ist es seit der Einführung des Sequencing notwendig, dass Interaktionen des Lernenden in Bezug auf eine Activities aufgezeichnet werden. Das LMS speichert für jede Activity Tracking-Informationen. Diese Informationen bilden das Tracking Model. Das heißt es wird zum Beispiel gespeichert, welche Activity gerade aktiv ist, wie oft auf eine Activity zugegriffen wurde oder ob der Lernende erfolgreich beim Bearbeiten einer Activity ist. Zu Beginn der Benutzung durch den Lernenden ist für jedes Element im Tracking Model ein default-Wert definiert. Zu Laufzeit, das heißt während der Bearbeitung des Lernmoduls durch den Lernenden werden diese Wert verändert. Das Tracking Model wird also dynamisch zur Laufzeit angepasst.

Es werden verschiedene Tracking Informationen erfasst, die folgende Übersicht gibt eine kurze Information darüber:

- Objective Progress Information – beschreibt den Erfolg des Lernenden bezüglich eines Objectives und wird für jedes Objective einer Activity registriert.
- Activity Progress Information – beschreibt den Fortschritt des Lernenden bezüglich der ganzen Activity.
- Attempt Progress Information – beschreibt den Fortschritt des Lernenden bezüglich des aktuellen Zugriffs.
- Activity State Information – es werden für jede Activity noch zusätzliche Informationen gespeichert, die während des Overall Sequencing Process benötigt werden.
- Global State Information – es werden Status-Informationen bezüglich des Activity Tree registriert.

3.5.3.2 Activity State Model

Auch das Activity State Model ist ein dynamisches Model. In diesem Modell werden der Zustand jeder Aktivität und der globale Zustand des Aktivitäten Baums verwaltet.

Eine Aktivität kann folgende Zustände haben.

Element	Beschreibung	Platzhalter	Default Wert
Activity is Active	Gibt an, ob auf eine Aktivität zugegriffen wird (<code>true</code>) oder nicht (<code>false</code>). Mit dem Beginn des Zugriffs wird der Wert auf „ <code>true</code> “ gesetzt, mit dem Ende ist der Wert „ <code>false</code> “	Boolean	<code>false</code>
Activity ist Suspended	Gibt an ob eine Aktivität unterbrochen ist. Ist sie unterbrochen ist der Wert „ <code>true</code> “ ist sie nicht unterbrochen „ <code>false</code> “	Boolean	<code>False</code>
Available Children	Enthält eine Liste aller verfügbaren „Kind-Aktivitäten“.	Geordnete Liste von Aktivitäten	Alle Kinder

Tabelle 4 - Activity State Information

Ein Baum kann folgende globale Informationen haben:

Element	Beschreibung	Platzhalter	Default Wert
Current Activity	Dieses Element kennzeichnet die Aktivität die gerade verwendet wird.	Activity	keine
Suspended Activity	Beinhaltet die Aktivität, von der aus die Unterbrechung ausgelöst wurde.	Activity	Keine

Tabelle 5 - Global State Information

3.5.3.3 Sequencing Definition Model

Das Sequencing Definition Model ist ein statisches Model und beschreibt, wie die verschiedenen Sequencing-Prozesse die Informationen des Tracking Modells interpretieren müssen, um die definiert Ablaufsteuerung umzusetzen. Siehe auch 3.5.2 Sequencing Definition Model (S. 53).

4. Verwendung des Navigation-Modells

SCORM gibt dem Lernenden die Möglichkeit, aus Lernaktivitäten, die aus einem gegebenen Aktivitäten-Baum generiert werden, Lernerfahrungen zu sammeln. Navigation ist dabei der Prozess, bei dem der Lernende und das LMS kooperieren, um Navigation Requests zu erkennen und daraus eine Lernerfahrung zu realisieren.

Typischerweise bietet das LMS dem Lerner ein Set von UI-Devices, die der Lerner nutzen kann, um seine Navigationswünsche zu äußern. In manchen Fällen möchte der Content-Entwickler, dass dem Lernenden keine Schnittstellen vom LMS angeboten werden, da das SCO selbst Schnittstellen bereit stellt. Möglich ist auch eine Kombination von Schnittstellen. In jedem Fall entsprechen die Navigation Requests dem Aktivitäten Baum. Dieser ist Lerner- und Inhaltsgesteuert.

SCORM kennt keine Pflicht-Anforderungen an Typen oder Stile von Navigations-schnittstellen, die dem Lerner zur Laufzeit zur Verfügung gestellt werden. Die Art der Nutzerschnittstelle und der Mechanismus zum Aufzeichnen von Interaktionen zwischen Lernendem und LMS sind absichtlich unspezifiziert. Ergebnisse, wie das Aussehen und Empfinden, Präsentationsstile, Platzierung oder Kontrollen liegen außerhalb der SCORM-Spezifizierungen.

Was genau zu SCORM gehört und was das Navigation-Modell kann, soll im Folgenden theoretisch betrachtet werden. Diese Theorie soll anschließend in Ideen zur Optimierung des Lernmoduls einfließen. Ziel ist es, diese Vorschläge später prototypisch umzusetzen. Am Ende soll erkennbar sein, inwieweit und ob das Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ durch das Navigation-Modell verbessert werden kann.

Im SCORM SN-Buch wird außer dem Navigation-Modell noch das Sequencing erläutert, da über dieses Thema jedoch schon eine ausführliche Diplomarbeit von Frau Gericke existiert, soll das Sequencing nicht genauer betrachtet werden.

4.1 Theoretische Betrachtung des Navigation-Modells

In diesem Teil der Diplomarbeit soll das Navigation-Modell theoretisch betrachtet werden. Im ersten Teil wird in das Thema Navigation Modell eingeführt. An späterer Stelle sollen die zwei Teilmodelle des Navigation Modells genauer betrachtet werden.

4.1.1 Einführung

Das LMS stellt UI-Devices bereit, mit dem der Lernende das Lernmodul steuern und sich durch die einzelnen Lektionen navigieren kann. Möchte ein Entwickler eigene UI-Devices direkt in die Lerninhalte integrieren und Navigation Requests direkt aus dem SCO erzeugen, steht ihm das Navigation Model zur Verfügung. Das Navigation Model ist kein Bestandteil der IMSSpezifikation, sondern wurde von ADL zusätzlich erarbeitet.

Es findet nur in der Navigation zwischen Lernaktivität und dem LMS Anwendung. In die Navigation innerhalb eines SCOs kann SCORM nicht eingreifen. Diese Navigation geschieht ausschließlich durch das SCO selbst.

Beispiel: Eine SCORM-Navigation ist nicht fähig, den Wechsel zwischen einzelnen Seiten eines mehrseitigen SCOs zu übernehmen. Das heißt, SCORM ist im Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ nicht in der Lage, die Aufgabe der Orientierungsleiste zu übernehmen.

Das SCORM Navigation-Model definiert eine Reihe von Navigationsereignissen. Diese Ereignisse können direkt durch ein SCO oder durch den Lerner ausgelöst werden. Der Lerner ist mit Hilfe von UI-Devices, die vom LMS oder direkt vom SCO zur Verfügung gestellt werden, in der Lage diese Navigationsereignisse auszulösen. Das heißt, Navigation Requests des Nutzers oder des SCOs führen zu Navigationsereignissen. Mit Hilfe von Navigation Requests lässt sich das Vorgehen im Aktivitäten-Baum steuern. Man ist zum Beispiel in der Lage einzelne Lernaktivitäten auszuwählen, in die nächste Aktivität zu wechseln oder zu einer früheren Aktivität zurück zu kehren.

Die folgende Tabelle beschreibt eine Reihe von Navigationsereignissen und erklärt, wie die einzelnen Navigation Requests unterteilt werden können. Zusätzlich definiert die Tabelle, welches der entsprechenden Navigation Requests jeweils ausgelöst werden kann.

Navigationsereignis	Beschreibung	Quellen
Start	Dieses Ereignis stellt eine Möglichkeit dar, die erste bzw. die Startaktivität im Aktivitäten-Baum zu identifizieren. Dieses Ereignis wird üblicherweise automatisch vom LMS generiert, wenn der Lerner einen neuen Versuch an der Wurzelaktivität des Aktivitäten-Baums unternimmt. Dieses Ereignis hat einen Start-Navigation Request zur Folge.	LMS Only
Resume All (alle fortsetzen)	Dieses Ereignis stellt eine Möglichkeit dar, einen früheren Versuch an der Wurzelaktivität des Aktivitäten-Baums fortzusetzen. Dieses Ereignis wird gewöhnlich automatisch vom LMS ausgelöst, wenn der Lerner einen früheren Versuch am Aktivitäten-Baum fortsetzt. „Resume All“ hat einen Resume all Navigation Request zur Folge.	LMS Only
Continue (weiter)	Dieses Ereignis stellt eine Möglichkeit dar, die nächste (in Relation zur aktuellen Aktivität) logisch erreichbare Lernaktivität im Aktivitäten-Baum zu erkennen. Dieses Ereignis hat einen Continue Navigation Request zur Folge.	LMS or SCO
Previous (zurück)	Dieses Ereignis stellt eine Möglichkeit dar, die letzte (in Relation zur aktuellen Aktivität) logische Lernaktivität im Aktivitäten-Baum zu erkennen. Dieses Ereignis hat einen Previous Navigation Request zur Folge.	LMS or SCO
Choose (wählen)	Dieses Ereignis stellt eine Möglichkeit dar direkt zu einer bestimmten Aktivität im Aktivitäten-Baum zu springen. Dieses Ereignis hat einen Choose Navigation Request für eine gezielte Aktivität zur Folge.	LMS or SCO
Abandon (verlassen)	Dieses Ereignis stellt eine Möglichkeit dar, zu einem ungewöhnlichen Zeitpunkt den aktuellen Versuch zu verlassen - ohne Absicht ihn fortzusetzen. Dieses Ereignis endet im aktuellen Versuch der aktuellen Aktivität. Wenn die aktuelle Aktivität eine Eltern Aktivität hat, ist der Versuch an der Elternaktivität nicht beendet. Abandon hat keinen unmittelbaren Einfluss auf irgendwelche Vorfahren der aktuellen Aktivität. Ein abgebrochener Versuch wird als Versuch gezählt. Auf keinen Fall bedeutet Abandon, dass die Informationen, die bereits aufgezeichnet wurden, verschwinden. Beispiel: Wenn die Aktivität bereits als bestanden oder komplett aufgezeichnet gilt, dann kann sie nicht mehr den Status „nicht komplett“ oder „nicht bestanden“ erhalten. Dieses Ereignis hat einen Abandon Navigation Request zur Folge.	LMS or SCO
Abandon all (alle verlassen)	Dieses Ereignis stellt eine Möglichkeit dar, zu einem ungewöhnlichen Zeitpunkt den aktuellen Versuch an der Wurzelaktivität des Aktivitäten-Baums zu verlassen. Eine Absicht zur späteren Fortsetzung besteht nicht. Dieses Ereignis beendet den aktuellen Versuch an der Aktivitäten-Baum-Wurzelaktivität und in allen aktiven Lernaktivitäten. Auf keinen Fall bedeutet es, dass bereits aufgezeichnete Informationen verloren gehen. Dieses Ereignis hat einen Abandon-All Navigation Request zur Folge.	LMS or SCO
Suspend All	Dieses Ereignis stellt eine Möglichkeit dar den aktuellen Versuch an der Wurzelaktivität zu unterbrechen. Es unterbricht den aktuellen Versuch an der Wurzelaktivität des Aktivitäten-Baums und in allen aktiven Lernaktivitäten. Keine der Versuche der unterbrochenen Aktivitäten endet. Wenn der nächste Versuch an der Wurzelaktivität des Aktivitäten-Baums durch Resume All initiiert wird, werden alle	LMS only

	Versuche der mit diesem Ereignis unterbrochenen Aktivitäten fortgesetzt. Die bereits aufgezeichneten Informationen gehen nicht verloren. Dieses Ereignis hat einen Suspend All Navigation Request zur Folge.	
Unqualified Exit	Dieses Ereignis zeigt an, dass der aktuelle Versuch normal beendet wurde und dass kein anderes Navigationsereignis ausgelöst wurde. Dieses Ereignis beendet den aktuellen Versuch bei der aktuellen Aktivität. Dieses Ereignis hat einen Exit Navigation Request zur Folge.	LMS or SCO
Exit All	Dieses Ereignis zeigt an, dass der aktuelle Versuch an der Wurzelaktivität des Aktivitäten-Baums normal beendet wurde. Es beendet den aktuellen Versuch an der Wurzelaktivität des Aktivitäten-Baums und in allen aktiven Lernaktivitäten. Dieses Ereignis hat einen Exit All Navigation Request zur Folge.	

Tabelle 6 - Navigations Ereignisse und Beschreibungen

4.1.1.1 Abarbeiten des Navigation Requests

Wenn der Lernende oder der Inhalt ein Navigationsereignis auslöst, setzt das LMS den entsprechenden Navigation Request mittels Sequencing-System in Gang. Das Resultat des Startens des Navigation Requests ist immer eines der Folgenden:

1. Wenn der Effekt des Navigation Requests die Beendigung des aktuellen Versuchs am Aktivitäten-Baum ist, startet das LMS einen Exit All Navigation Request, der den Versuch beendet und die Steuerung dem LMS zurück gibt.
2. Nach dem Ermitteln des aktuellen Tracking-Status und der anwendbaren Sequencing-Information am Aktivitäten-Baum, bestimmt das LMS, dass das Starten des beabsichtigten Navigation Requests NICHT beachtet werden soll. In diesem Fall ignoriert das LMS den Navigation Requests. Das LMS unternimmt keine Sequencing Aktion, bis ein anderer Navigation Request ausgelöst wurde.
3. Nach dem Ermitteln des aktuellen Tracking-Status und der anwendbaren Sequencing-Information am Aktivitäten-Baum, bestimmt das LMS, dass das Starten des beabsichtigten Navigation Requests BEACHTET werden sollte. Das LMS ruft den Overall Sequencing Process mit dem geplanten Navigation Request auf. Das Ergebnis des Overall Sequencing Process ist eines der Folgenden:
 - a. Eine Lernaktivität wurde zum Laden identifiziert – Das LMS soll die identifizierte Startaktivität vorbereiten und in Gang setzen.
 - b. Es wurde keine Lernaktivität zum Laden identifiziert – In diesem Fall hat SCORM keine Vorschriften an das Verhalten des LMS. Es wird empfohlen, dass das LMS die Verwirrung beim Lernenden minimal hält.
 - c. Eine Ausnahme ereignet sich während eines Sequencing Prozesses – In diesem Fall hat SCORM keine Vorschriften an das Verhalten des LMS. Das

LMS sollte die Ausnahme sauber handhaben und die Verwirrung beim Lerner klein halten.

4.1.1.2 Beenden eines Content Objects

Wenn ein LMS UI-Devices für das Auslösen von Navigation Requests anbietet, könnte ein Lernender seinen Wunsch zum Navigieren durch das Auslösen eines oder mehrerer dieser Schnittstellen anzeigen. Zeigt der Lernende seinen Wunsch zu navigieren, nimmt SCORM an, dass er mit der Bearbeitung des aktuell geladenen Objektes fertig ist – insofern dieses existiert. Beachtet das LMS das Auslösen des Navigationsereignisses, muss es zunächst das aktuelle Objekt wegräumen, bevor es den Navigation Request ausführen kann. Das aktuelle Objekt muss weggeräumt sein, damit sicher gestellt werden kann, dass die Lerninformationen vom Content Objekt übergeben wurden, da sich diese auf das Sequencing auswirken könnten.

SCOs können Navigationsabsichten mit Hilfe des Navigation-Data-Model direkt an das LMS weitergeben. Außerdem muss ein SCO dem LMS zeigen, dass es auf den Navigation Request reagieren soll. Dies geschieht mit Hilfe von `Terminate()`. Diese API-Methode signalisiert, dass die Kommunikation mit dem LMS beendet werden soll. Ist die Kommunikation beendet, kann und soll das LMS den Navigation Request ausführen.

Sobald die `Terminate()` Anforderung in Gang gesetzt wurde, wird das LMS zuerst schwebende, vom Lerner angestoßene, Navigations-Ereignisse starten. Wenn keine dieser Navigationsereignisse vorhanden sind, wird das LMS den letzten durch ein SCO kommunizierten Navigation Request ausführen - falls es einen gibt. Wenn weder der Lerner noch das SCO Navigationsabsichten anzeigen, muss das LMS warten bis der Lerner ein Navigationsereignis anzeigt.

Man stelle sich folgendes Szenario vor:

Ein SCO A ist geladen und ruft `Initialize()` auf. Während SCO A läuft, wählt der Lerner eine andere Aktivität und nutzt dazu eine Navigationsschnittstelle, die durch das LMS für das entsprechende SCO angeboten wird. Dieses SCO, SCO B, wird nun im selben Browserfenster geladen, dabei erzwingt es das Löschen des bisherigen SCOs, SCO A. In diesem Fall muss SCO A `Terminate()` aufrufen, wenn es bemerkt, dass es gelöscht werden soll. Die LMS Anwendung muss die Kommunikationssession abschließen, für die A geöffnet wurde, auch wenn SCO A „vergisst“ `Terminate()` aufzurufen.

SCO A muss eine Behandlungsroutine für das `onUnload()` Ereignis ausführen, welche die benötigte Kommunikation mit dem LMS erledigt und anschließend `Terminate()` aufruft. SCORM könnte einen Kommunikationsmechanismus einsetzen, der es dem LMS zukünftig erlaubt das SCO zu benachrichtigen, bevor es das SCO zum Beenden zwingt.

Generell gilt: Navigationsereignisse, die vom Lerner ausgelöst werden, haben Vorrang vor Navigation Requests, die durch das SCO ausgelöst werden.

4.1.1.3 Navigation und Hilfsmittel

Die IMS-SS-Spezifikation definiert das Konzept der Hilfsmittel als einen unterstützenden Service, der Lernern in Verbindung mit Lernaktivitäten bereitgestellt wird. Die Hilfsmittel beinhalten Themen wie Glossar, Referenz-Handbuch, Chat-Rooms und Diskussionsplattformen und so weiter. Die IMS-SS-Spezifikation stellt nur einen kleinen Aufhänger für Hilfsmittel zur Verfügung und SCORM hat bis jetzt nicht genügend Anforderungen, um zukünftig ihren Nutzen in einer wieder verwendbaren Art und Weise zu definieren. SCORM verbietet nicht die Nutzung von Hilfsmitteln, jedoch ist es zu empfehlen, dass Entwickler und LMS Händler Hilfsmittel sehr vorsichtig einsetzen, um zukünftige Wiederverwendbarkeit zu gewährleisten.

4.1.2 UI-Devices für die Navigation

Das Navigation Data Model besteht aus zwei Teilmodellen. Ein Teil ist das Run-Time Navigation Data Model. Über dieses Model können Navigation Requests vom SCO zum LMS kommuniziert werden. Der zweite Teil ist das Presentation Information Model. Über dieses Modell wird gesteuert, welche UI-Devices vom LMS angezeigt werden und welche nicht. Damit sollen redundante UI-Devices vermieden werden. Im Folgenden wird zunächst geklärt, was notwendig ist um UI-Devices für die Navigation vorzubereiten.

4.1.2.1 UI-Devices für die Navigation vorbereiten

Das LMS muss als Minimum die Fähigkeit haben, über UI-Devices Navigationsereignisse auszulösen. SCORM hat keine Regeln für den Typ oder das Aussehen von UI-Devices, die dem Lerner zur Laufzeit präsentiert werden. Ebenso wenig gibt es Vorschriften an die UI-Devices der Navigation. Die Art der UI-Devices und der Mechanismus, um Interaktionen zwischen Lerner und LMS zu erfassen, sind absichtlich nicht festgelegt.

Obwohl SCORM keine Anforderungen an den Typ oder die Stile von UI-Devices stellt, ist es zu empfehlen, dass das LMS nur auslösbare UI-Devices für Navigationsereignisse zur Verfügung stellen kann. Also nur Schnittstellen von denen das LMS einen Navigation

Request produzieren kann. Es ist außerdem zu empfehlen, eine Nutzerschnittstelle durch das komplette Content-Objekt beizubehalten. Weiterhin wird empfohlen, dass das Verhalten von Navigationsereignissen, die durch eine vom LMS unterstützte Schnittstelle ausgelöst werden, über die komplette „Lernerfahrung“ gleich bleiben.

SCOs können wahlweise UI-Devices zum Auslösen von Navigationsereignissen implementieren. SCORM spezifiziert, wie ein SCO auf einen Navigation Request hinweisen kann und diesen ausführt. Ebenso ist spezifiziert, wie ein SCO ein LMS zur Entscheidung darüber auffordert, ob ein spezieller Navigation Request gültig ist. Ein Entwickler kann für jedes Inhalts-Objekt anzeigen, dass selbiges bestimmte UI-Devices erlaubt. Zudem kann er erzwingen, dass das LMS keine doppelten UI-Devices für dieselben Navigationsereignisse erlauben soll.

Ziel der Möglichkeit UI-Devices ist verbieten ist es, den Lerner nicht mit doppelten UI-Devices zu verwirren. Ein solcher verwirrender Fall wäre es, wenn „previous“- und „continue“-Buttons in der SCO und in der UI des LMS erscheinen. Andere Verwendungen dieses Features sind außerhalb des Einflusses von SCORM.

Um den Nutzer nicht mit doppelt erscheinenden Navigationselementen in Form eines Strukturbaumes oder einer Linkliste zu verunsichern, steht eine weitere Möglichkeit zur Verfügung, das `isVisible` Attribut. Die Nutzung dieses Attributs soll im Folgenden erläutert werden.

4.1.2.2 Nutzung des `isVisible` Attributes

Wenn man die Struktur eines Paketes bzw. eines Lernmoduls anzeigen bzw. übergeben will, ist es unter Umständen nötig zunächst zu klären, welche Elemente angezeigt werden sollen und welche nicht. Dazu dient das `isVisible` Attribut. Dieses Attribut gibt an, ob das verknüpfte Element mit angezeigt wird, wenn die Struktur des Paketes angezeigt oder übergeben wird. Der Wert des `isVisible`-Attributes beeinflusst nur das Element, für das es definiert ist. Es wirkt sich nicht auf die Kinder dieses Elementes oder die Ressourcen aus, die mit diesem Element verbunden sind. Der Wert des Attributes kann `true` oder `false` sein. Ist `isVisible = false`, ist die Lernaktivität unsichtbar. Der Effekt von `isVisible` ist dabei strikt auf die Präsentation von UI-Devices beschränkt. Es hat keine Auswirkungen auf das SCORM Sequencing Verhalten, welches den Choice Navigation Requests zugeordnet ist. Unsichtbare Lernaktivitäten (`isVisible = false`), welche von einem SCORM Content Package abgeleitet wurden, können trotzdem Ziele von Choice Navigation Requests sein. Lernaktivitäten, die Ziel von Choice Navigation Requests sind, könnten fürs Überbringen und Starten gekennzeichnet sein. Wenn ein

Entwickler sicherstellen möchte, dass eine spezielle Lernaktivität nicht zum Überbringen durch einen Navigation Request erkannt werden soll, sollte eine bedingte „If always then hide from“ Sequencing-Regel auf die Lernaktivität angewandt werden.

4.1.2.3 Presentation Information Model

Das Presentation Information Model ist ein kleines Modell, das Entwicklern ermöglicht bestimmte Präsentationseigenschaften für ein Content Object anzugeben. Ein Entwickler hat mit diesem Teilmodell die Möglichkeit festzulegen, ob UI-Devices angeboten werden sollen oder nicht.

Die Informationen über die Präsentationseigenschaften werden in der Manifestdatei integriert und haben, die in der Tabelle 7 beschriebene Struktur.

Nr.	Name	Erklärung	Platzhalter	Datentyp	Default
1	Presentation	Informationen hinsichtlich Präsentation			
1.1	Navigation Interface	Eigenschaften über UI-Devices Kontrolle			
1.1.1	Hide LMS UI	Hinweise, dass kein speziellen UI-Devices durch das LMS zur Verfügung gestellt werden sollen, die es dem Lerner ermöglichen verbundene Navigationsereignisse auszulösen	Null oder mehr Vokabel Zeichen	Offener, erweiterbarer Wortschatz, mit speziell definierten Zeichen (siehe Tabelle 6)	(leer)

Tabelle 7 - Präsentation des Information Model

Tabelle 5 spezifiziert das Set von Nutzer-Eingabeschnittstellen, die der Inhalt möglicherweise unterstützen will. Es ist zu empfehlen, dass ein LMS die Anfragen zum Verstecken von Nutzer-Eingabeschnittstellen beachtet und redundante und potentiell verwirrende Schnittstellen nicht unterstützt.

Die `HideLMS UI` kann folgende in Tabelle 8 erläuterte Datentypen besitzen.

Zeichen	Definitionen	Erklärung
Previous	Vorher Navigations-eingabe	Wenn dieses Zeichen angegeben wurde, sollte das LMS keine Nutzereingabeschnittstelle zur Verfügung stellen, die es möglich macht ein vorheriges Navigationsereignis auszulösen, so lange das verbundene Content Object geladen ist.
Continue	Weiter Navigations-eingabe	Wenn dieses Zeichen angegeben wurde, sollte das LMS keine Nutzereingabeschnittstelle zur Verfügung stellen, die es möglich macht ein weiteres Navigationsereignis auszulösen, so lange das verbundene Content Object geladen ist.
Exit	Ausgang Navigations-eingabe	Wenn dieses Zeichen angegeben wurde, sollte das LMS keine Nutzereingabeschnittstelle zur Verfügung stellen, die es möglich macht ein Exit Navigationsereignis auszulösen, so lange das verbundene Content Object geladen ist.
Abandon	Verlassen Navigations-eingabe	Wenn dieses Zeichen angegeben wurde, sollt das LMS keine Nutzereingabeschnittstelle zur Verfügung stellen, die es möglich macht, ein Verlassen Navigationsereignis auszulösen, so lange das verbundene Content Object geladen ist.

Tabelle 8 - Laufzeit Nutzereingabeschnittstellen Wortschatz

Das Presentation Information Model, kann nur auf Content Objekte angewandt werden. Das heißt es darf nur innerhalb von `<item>`-Elementen benutzt werden, die ein SCO referenzieren, also nur an Blatt-Activities. Der Effekt des Presentation Information Model kann nur in der Zeit vom Laden bis zum Wegräumen eines Content Objekts vorkommen. Sie gelten nur für das entsprechende SCO. Sobald ein anderes SCO angezeigt wird, sind die Angaben nicht mehr gültig und alle Navigationselemente werden angezeigt. Das Präsentation Model hat keine Auswirkungen auf ein LMS ohne geladenes Content Objekt. In diesem Sinne kann das LMS jedes selbstgewählte UI-Device auch anbieten. Sobald das LMS frei zum Unterstützen von Nutzereingabeschnittstellen ist, ist es wählbar.

4.1.2.4 Kommunikation zwischen SCO und LMS bei einem Navigation Request

Ein SCO fragt an, ob für einen Navigation Request eine passende Lernaktivität gefunden wurde. Das heißt ein SCO möchte wissen, ob der gegebene Navigation Request gültig ist. Das SCO ist in der Lage das LMS nach der Gültigkeit der verschiedenen Navigation Requests zu befragen. Diese Informationen können genutzt werden, um eine Reihe optimierter UI-Devices zu erstellen. Ungeachtet dessen ob ein SCO Nutzereingabeschnittstellen anbietet, kann ein SCO einem LMS direkt die Navigationsabsicht übermitteln. Ein SCO kann bis zu dessen Beendigung auf einen - und nur auf einen - Navigation Request zur Abarbeitung durch das LMS aufmerksam machen. Als Beispiel: Ein SCO kann einen Navigation Request wie Previous, Exit und Choose an das LMS übermitteln. Nachdem das SCO abgelegt wurde, führt das LMS die übermittelten Navigation Requests aus und überliefert diesen zur angestrebten Aktivität.

Alle eintretenden Übermittlungen zum SCORM Navigation Data Model nutzt die SCORM Run-Time API. Die Übermittlung zwischen SCOs und LMS betreffs Navigation Requests ist in der folgenden Tabelle definiert.

Nr	Name	Erklärung	Platzhalter	Datentyp
1	Navigation	Informationen bezüglich Navigation Requests		
1.1	Request	Informationen bezüglich Navigation Requests. SCO verlangt vom LMS die Abarbeitung bis zum Ende des SCOs.	"_none_" "continue" "previous" "choice" {target} "abandon" "abandonAll" "exit" "exitAll"	Vocabulary (Restricted)
1.2	Valid Request	Informationen, die darauf hinweisen, ob ein Navigation Request gültig ist oder nicht.		
1.2.1	Continue	Mit diesem Elementen wird bestimmt, ob es zum Continue Navigation Request eine Aktivität gibt, die übergeben werden kann. ADL-Note: Dieses Element kann von einem SCO dazu benutzt werden, zu entscheiden, ob das SCO eine Nutzereingabeschnittstelle, als Navigationssteuerung übergibt, die es Lernern erlaubt ein Continue Navigationsereignis auszulösen.	"true" "false" "unknown"	Vocabulary (Restricted)
1.2.2	Previous	Mit diesem Element wird bestimmt, ob es zum Previous Navigation Request eine Aktivität gibt, die übergeben werden kann. ADL Note: Dieses Element kann von einem SCO dazu benutzt werden, zu entscheiden, ob das SCO eine Nutzereingabeschnittstelle als Navigationssteuerung übergibt, die es Lernern erlaubt ein Previous Navigationsereignis auszulösen	"true" "false" "unknown"	Vocabulary (Restricted)
1.2.3	Choice (target)	Mit diesem Element wird bestimmt, ob es zum Choice Navigation Request für die aktuelle Aktivität, eine Aktivität gibt, die übergeben werden kann. ADL Note: Dieses Element kann von einem SCO dazu benutzt werden, zu entscheiden, ob das SCO eine Nutzereingabeschnittstelle als Navigationssteuerung übergibt, die es Lernern erlaubt ein Choice Navigationsereignis für die ausgewählte Aktivität auszulösen	"true" "false" "unknown"	Vocabulary (Restricted)

Tabelle 9 - SCORM Navigation Data Model

4.1.2.5 Run-Time Navigation Data Model

Mittels der Elemente des Run-Time Navigation Data Model können Navigation Requests vom SCO zum LMS übertragen werden. Es ist auch möglich die Gültigkeit dieser Requests mit Hilfe des Run-Time Navigation Data Model zu prüfen. Das Modell besteht aus zwei Elementen. Dem Request (`adl.nav.request`) und dem Request Valid (`adl.nav.request_valid`).

Request

Über dieses Element kann das SCO den gewünschten Navigations Request zum LMS übertragen. Ein SCO kann bis zu seiner Beendigung des SCO, nur einen Navigation Request für die Abarbeitung durch ein LMS abbilden. Es kann Navigation Requests wie Previous, Exit, und Choice an das LMS übermitteln. Nachdem das SCO beendet wurde, wird das LMS diesen übermittelten Navigation Request abarbeiten und dann die gewünschte Lernaktivität liefern.

Bevor die Kommunikation zwischen SCO und LMS nicht beendet wurde, hat die Übermittlung des Navigation Requests zum LMS keine Folgen. Die Kommunikation zwischen SCO und LMS wird über den erfolgreichen Aufruf von `Terminate()` beendet. Das Run-Time Navigation Data Model ist nur während einer Lerner Session auf einem SCO zulässig. Es wird durch das LMS gesteuert, bis das SCO endet und es besteht nicht unter Exit-Bedingungen.

Es gelten folgende Szenarien:

Szenario: A

Während des Lernerlebnisses, begegnet der Lerner der Lernaktivität A, die wird durch SCO A (Lerner Session 1 bei SCO A) präsentiert, welches mit Lernaktivität A verbunden ist.

- SCO A setzt das `adl.nav.request` Element auf „Continue“.
- `Terminate()` wurde durch den SCO aufgerufen, bevor der Lerner irgendein Navigationsereignis ausgelöst hat. Daraus folgt das Ende der Kommunikation zwischen SCO und API-Instanz

Das Resultat dieses Szenarios ist, dass der Continue Navigation Request durch das LMS abgearbeitet wird.

Szenario: B

Später, während desselben Lernererlebnisses, begegnet der Lerner der Lernaktivität A wieder, somit wird SCO A dem Lerner erneut präsentiert (Learner Session 2 bei SCO A).

- Wenn SCO A sofort `GetValue(adl.nav.request)` aufruft, wird „_none_“ zurück gegeben, da das SCO während der Lersersession keinen Navigation Request übermittelt hat.
- Während dieser Lersersession setzt das SCO kein `adl.nav.request` Element.
- `Terminate()` wurde durch das SCO aufgerufen, bevor der Lerner irgendein Navigationsereignis ausgelöst hat. Daraus folgt das Ende der Kommunikation zwischen SCO und API Instanz.

In diesem Szenario bestand das `adl.nav.request` Data Model Element aus Szenario A nicht mehr. Es beinhaltete nicht mehr den vorherigen „continue“-Wert. `Terminate()` würde in diesem Fall keinen Navigation Request auslösen. Das LMS würde auf ein Navigationsereignis des Lerners warten, bevor es irgendeinen Navigation Request auslöst.

An folgendem Quellcodebeispiel soll die Anwendung des Run-Time Navigation Data Models näher erläutert werden.

```
doContinue()
{
    doSetValue(„adl.nav.request“, „continue“)
    doTerminate()
}
```

Quellcode 5 - Anwendung `adl.nav.request`

Über ein Navigationselement, dass in einem SCO integriert wurde, wird die Funktion `doContinue()` aufgerufen. Es wird der Wert „continue“ für `adl.nav.request` gesetzt, was später einen Continue Navigation Request auslöst. Über die API-Instanz wird dieser Wert an das LMS übertragen. Die Funktion ruft `Terminate()` auf, um die Kommunikation zwischen SCO und LMS zu beenden. Erst danach kann der Continue Navigation Request ausgeführt werden. Möchte man einen Choice Navigation Request auslösen, ist zusätzlich zum Wert `choice` noch die Angabe des Ziels notwendig. Das Ziel ist die ID der zu wählenden Aktivität. In einem Beispiel sieht das folgendermaßen aus:

```
doChoice()
{
    doSetValue(„adl.nav.request“, „{target=SITE_1}choice“)
    doTerminate()
}
```

Quellcode 6 - Anwendung `adl.nav.request` mit `choice`

`Adl.nav.request` kann folgende Werte übernehmen:

- `continue`
- `previous`
- `choice`
- `exit`
- `exitAll`
- `abandon`
- `abandonAll`

Eine detaillierte Erläuterung des `adl.nav.request` sind im Anhang A zu finden.

Request Valid

An bestimmten Stellen kann es von Vorteil bzw. wünschenswert sein, zu prüfen, ob ein Navigation Request gültig ist oder nicht. Das heißt ob eine Benutzerschnittstelle in der Lage ist eine Aktivität zu identifizieren.

Als Beispiel: Ein Content-Entwickler möchte, dass „Continue“ und „Next“ Button in einem SCO nur sichtbar sind, wenn das SCO in einer logischen Reihenfolge ist. Das SCO selbst, kann nicht entscheiden, ob es in der logischen Reihenfolge ist, das heißt, ob der Navigation Request zulässig ist. Aber das LMS kann diese Entscheidung fällen, da es die Informationen durch das Sequencing bekommt. Das SCO kann nun Aufrufe zum Run-Time Navigation Data Model machen, um die Zulässigkeit verschiedener Navigation Requests zu prüfen.

Die Prüfung der Zulässigkeit erfolgt mit `adl.nav.request_valid`.

Ein Beispielquellcode für eine solche Prüfung könnte wie folgt aussehen:

```
If (doGetValue(„adl.nav.request_valide.continue“) != true)
{
    disableButton(“continue“)
}
```

Quellcode 7 - Anwendung `adl.nav.request_valid`

In diesem Beispiel wird abgefragt, ob ein Continue Navigations Request gültig ist. Bei Ungültigkeit, soll der Weiter-Button nicht angezeigt werden.

Dass ein Continue Navigation Request ungültig ist, kann verschiedene Ursachen haben. Eine wäre, dass der Control Mode `flow = false` ist. Also kein Continue Navigation Request möglich ist. Es kann auch sein, dass im Aktivitäten Baum keine nachfolgenden Aktivitäten mehr vorhanden sind. Auch dann ist ein Continue Navigation Request ungültig.

Die folgenden Navigation Request können auf Ihre Gültigkeit geprüft werden:

- Continue
- Previous
- Choice

Eine detailliertere Erläuterung des `adl.nav.request_valid` ist im Anhang A zu finden.

4.2 Ideen zur Optimierung des Lernmoduls

Interessant für das vorliegende Lernmodell ist die Anwendung des Navigation-Modell in Form von in SCOs eingebetteten Navigationselementen.

4.2.1 Weiter- bzw. Zurück Button

So könnte man zum Beispiel versuchen einen Weiter- bzw. einen Zurück Button im SCO zu implementieren.

Derzeit ist das Blättern innerhalb einzelner Seiten eines Abschnittes über die Orientierungsleiste möglich. Statt dieser Orientierungsleiste könnte man nun einen Weiter-Button implementieren um die einzelnen Seiten in der richtigen Reihenfolge durcharbeiten zu können. Der Zurück-Button ersetzt auch die Orientierungsleiste. Somit ist es möglich innerhalb der Seite auch zurück zu blättern.

4.2.2 Link innerhalb eines Textes

Eine weitere Verwendung von eingebetteten Navigationselementen könnte die Integration von Links innerhalb eines Textes sein. Über diese Links sollte es möglich sein ein neues SCO aufzurufen. Dieses SCO könnte ein neues Kapitel sein, in dem verwendetes Stichwort näher erläutert wird. Es könnte aber auch eine Art Glossar sein, in dem dieses Stichwort erklärt wird. SCORM stellt keine Möglichkeiten für Hilfsmittel zur Verfügung. Allerdings bietet es kleine Aufhänger für Hilfsmittel. Als solch ein Aufhänger könnte ein eingebettetes Navigationselement gesehen werden.

4.2.3 Veränderung der rechten Navigation

Derzeit gehören zu einem SCO mehrere Abschnitte, die wiederum mehrere Seiten beinhalten können. Durch diese Verschachtelung ist es nur schwer zu überprüfen ob alle Seiten eines Abschnittes abgearbeitet wurden. Verändert man die rechte Navigation dahingehend, dass ein Anklicken eines Navigationsbuttons ein neues SCO aufruft, können

die einzelnen Abschnitte in SCOs gepackt werden. Durch diese Möglichkeit der Verwendung bleibt die Navigation scheinbar innerhalb eines SCO, jedoch stellen die Abschnitte einzelne SCOs dar, was das Nachvollziehen der Abarbeitung erleichtert.

4.3 Prototypische Umsetzung der Ideen

Die oben genannten Ideen zur Optimierung sollen unter Verwendung des Navigation Models auf ihre Möglichkeiten getestet werden. Dazu wird das Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ in Auszügen prototypisch umgesetzt und mit eingebetteten Navigationselementen versehen. Als Grundlage für die Umsetzung soll die erarbeitete Theorie dienen.

4.3.1 Vorbetrachtungen

Die in Kapitel „4.2 Ideen zur Optimierung des Lernmoduls)“ genannten Ideen sollen am Beispiel des Lernmoduls „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ umgesetzt werden. Auf Grund des Umfangs dieses Lernmoduls wurden ausgewählte Seiten in HTML nachgebaut. Dadurch ließen sich die einzelnen Navigationsideen einfacher testen. Zum Entwickeln der Seiten wurde ein einfacher Text-Editor verwendet.

Als Werkzeuge wurden außerdem eine Beispiel-Implementierung der Laufzeitumgebung von ADL und ein Editor verwendet, der es ermöglicht schnell SCORM Content Packages zu erstellen.

4.3.1.1 Reload Editor

Der Reload Editor ist kostenlos im Internet erhältlich und wurde in der Version 2.5.4 vom 02. März 2007 verwendet. Der Reload Editor unterstützt Entwickler bei der Erstellung eines SCORM Content Packages. Mit Unterstützung dieses Editors ist es nicht mehr nötig mühevoll XML-Code einzugeben. Man stellt sich sein Package zusammen und fügt über Eingabemasken die Metadaten hinzu, daraus erstellt der Editor den benötigten XML-Code. Am Ende lässt die ZIP-Datei – das Package generieren. Der Editor ist ein java-basiertes Open-Source-Projekt. Nachteilig ist, dass es beim Editor zwar eine Vorschau des Lernmoduls gibt, allerdings lassen sich damit nicht alle Funktionen testen. Die richtige Funktionalität von Sequencing und Navigation konnte mit dem Reload-Editor nicht getestet werden. Aus diesem Grund war zum Umsetzen des Navigation Models noch eine Beispiel-Implementierung der Laufzeitumgebung von ADL notwendig.

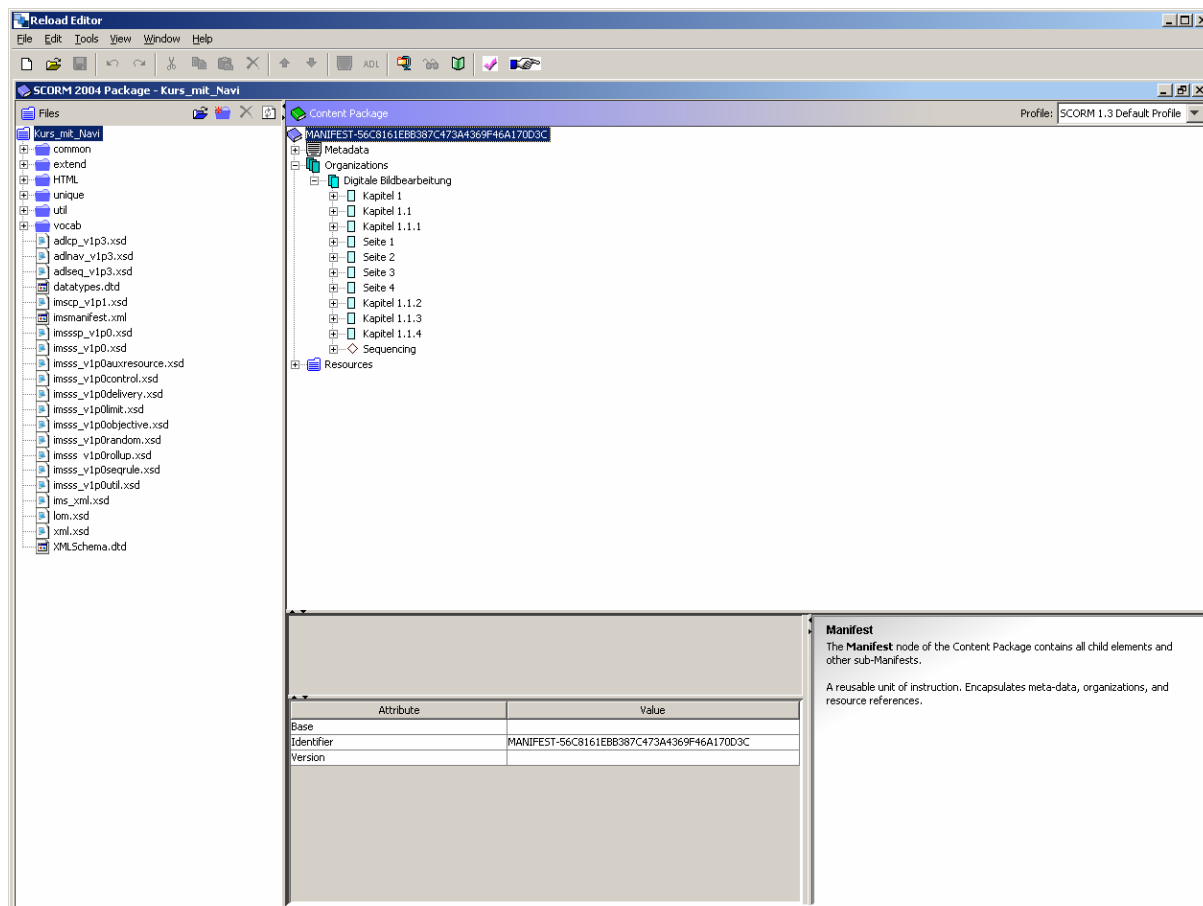


Abbildung 28 - Screenshot Reload Editor Version 2.5.4

4.3.1.2 ADL Sample Runtime Environment

Das ADL Sample Run-Time Environment kurz SampleRTE genannt, setzt die in SCORM beschriebenen Konzepte in Form einer Beispiel Implementierung um. Beispielimplementierung heißt, dass man mit diesem Programm kein vollständiges LMS zu Verfügung hat. Mit dem Programm lässt sich aber testen, wie SCORM Sequencing und Navigation in einem LMS umgesetzt werden können. In dieser Beispielimplementierung lassen sich auch die Funktionen testen, welche die Vorschau des Reload-Editors nicht darstellen konnte. Somit ist das Sample RTE für Content-Entwickler zum Testen von Kursen sehr gut geeignet.

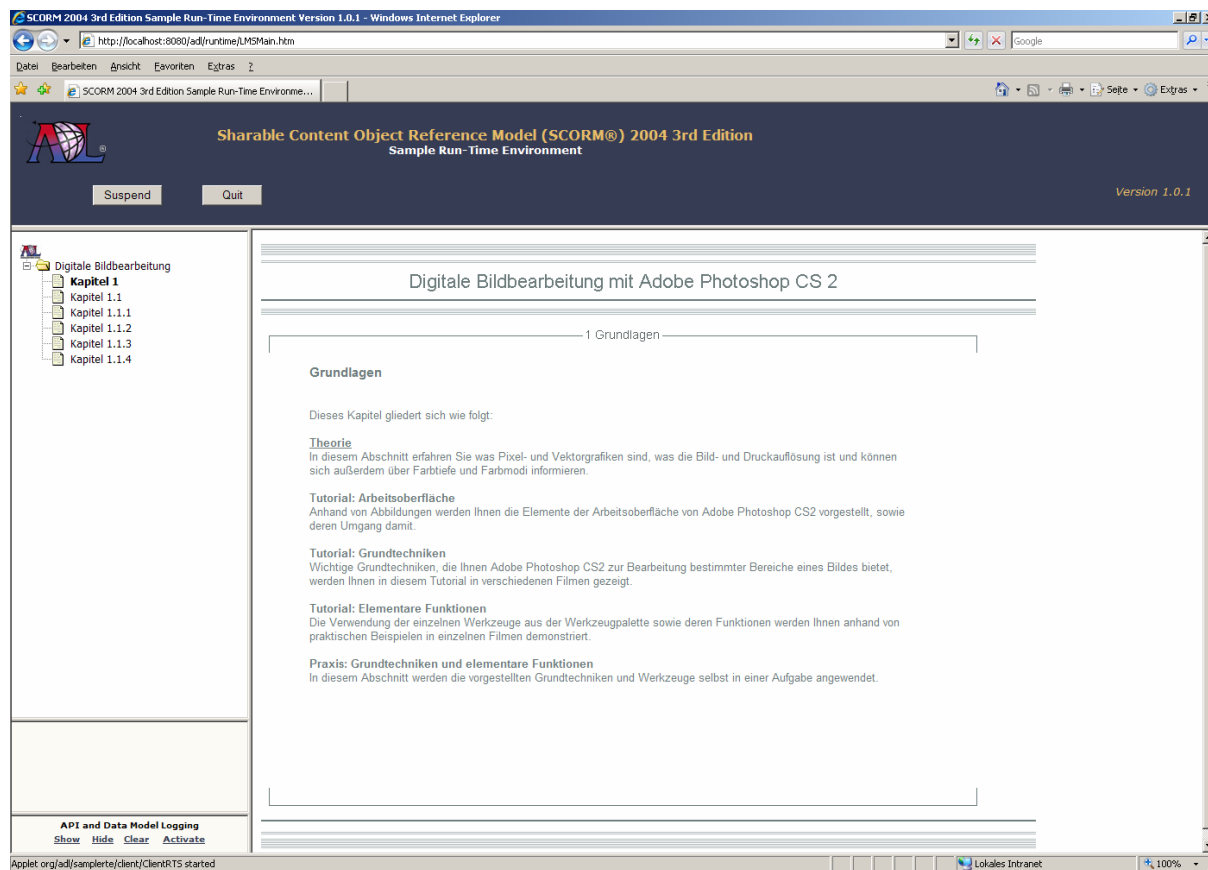


Abbildung 29 - Screenshot Sample Run-Time Environment

In der Sample RTE ist es wie in einem LMS möglich Kurse zu importieren. Nutzer können sich für Kurse anmelden und es ist möglich diese Kurse anzuschauen. Außerdem kann ein Administrator diese Kurse verwalten. Natürlich haben diese Funktionalitäten nur einen begrenzten Umfang.

Sehr nützlich bei der Verwendung dieses Tools ist die Möglichkeit in Form eines Loggs die Kommunikation zwischen LMS und SCO zu verfolgen. Mit Hilfe dieser Funktionalität lassen sich Fehler sehr gut erkennen. Die Darstellung des Loggs kann angezeigt und ausgeblendet, sowie aktiviert und deaktiviert werden.

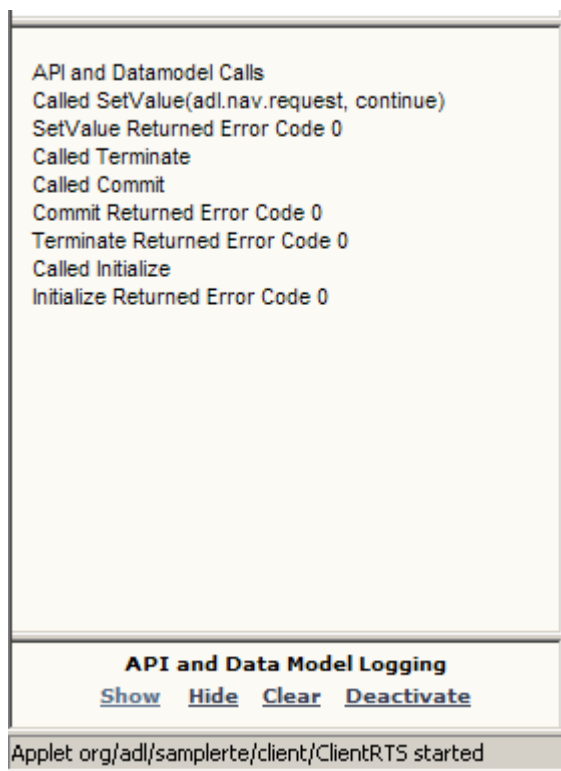


Abbildung 30 - Logganzeige der Sample RTE

4.3.2 Beschreibung des Lernmodulauszugs

Zur prototypischen Umsetzung wurde das Lernmodul auszugsweise in HTML in der folgenden Struktur umgesetzt:

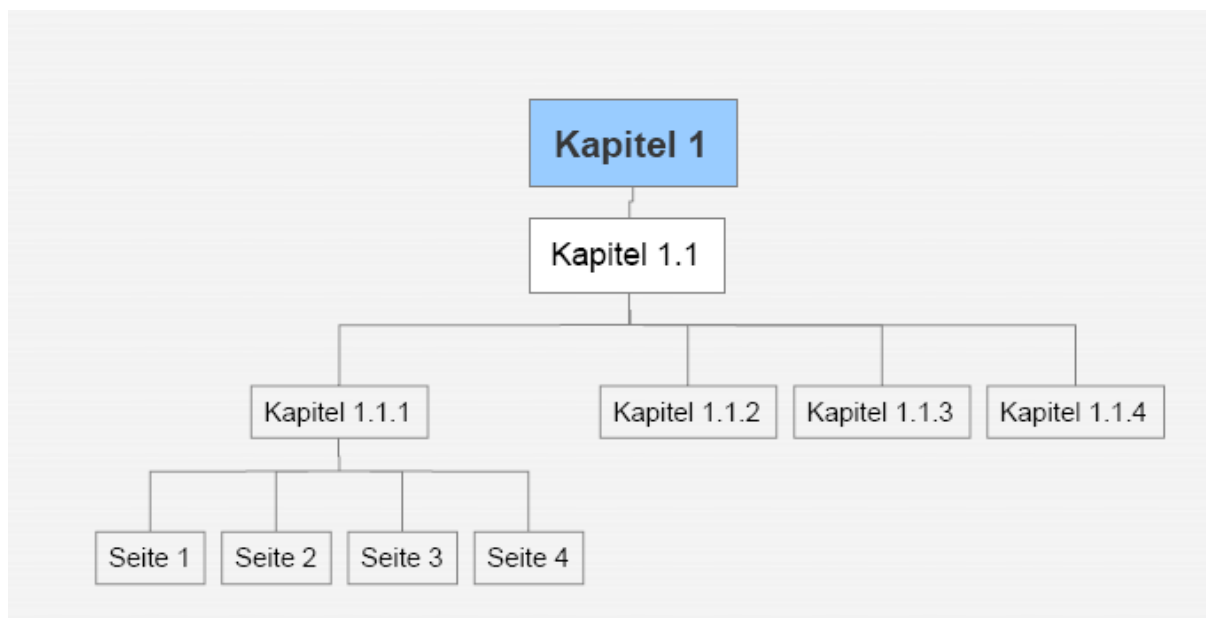


Abbildung 31 - Struktur Prototyp Digitale Bildbearbeitung mit Adobe Photoshop CS2

Kapitel 1 bildet das Wurzel-Element vom Wurzel-Element kommt man über einen Link im Text oder über die vom LMS angebotene Navigation auf das Kapitel 1.1.

Theorie

In diesem Abschnitt erfahren Sie was Pixel- und Vektorgrafiken sind, was die Bild- und Druckauflösung ist und können sich außerdem über Farbtiefe und Farbmodi informieren.

Abbildung 32 - Textlink im Wurzel-Element Kapitel 1

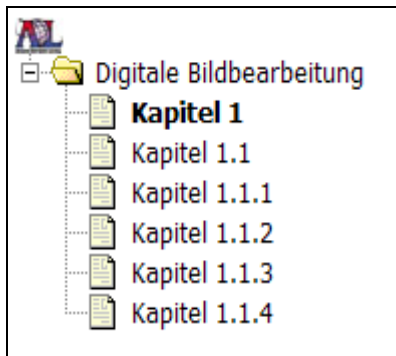


Abbildung 33 - Navigation wie sie vom LMS angeboten wird

Von Kapitel 1.1 kann der Lerner über die Navigation des LMS oder über die rechte Navigation des SCOs in die Kapitel 1.1.1, 1.1.2, 1.1.3 und 1.1.4 wechseln.



Abbildung 34 - Rechte Navigation des Lernmoduls

Für die Kapitel 1.1.2 – 1.1.4 wurde an dieser Stelle nur das Navigieren zwischen den Kapiteln realisiert. Im Kapitel 1.1.1 ist es möglich zwischen den einzelnen Seiten über einen Weiter- bzw. Zurück-Button zu blättern.

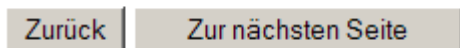


Abbildung 35 - Weiter- bzw. Zurück-Button zum Navigieren zwischen den Seiten

In der Navigation die vom LMS dargestellt wird, sind nur die Kapitel sichtbar. Die Seiten wurden auf `isVisible=false` gesetzt.

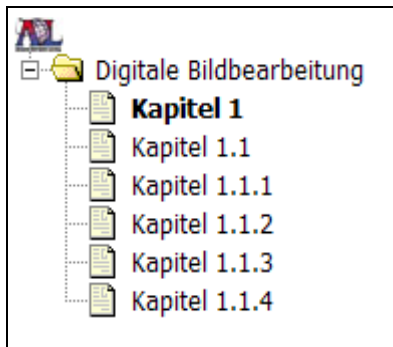


Abbildung 36 - LMS Navigation ohne Darstellung der einzelnen Seiten

4.3.3 Beispielumsetzung: Allgemeines

Um das Navigation Modell zu implementieren, ist es zunächst notwendig eine Abarbeitungsstrategie festzulegen.

Beim Starten des Kurses werden die Grundlagen angezeigt. Von dort aus kommt der Lerner über den Link Theorie bzw. das Menü des LMS in die nächsten Unterkapitel. Im Menü des LMS ist das Kapitel 1 bis in die dritte Ebene sichtbar. Die einzelnen Seiten im Kapitel 1.1.1 werden nicht im Menü des LMS angezeigt. Die Vor- und Zurück-Button des LMS sind ausgeblendet. Durch Anklicken der einzelnen Kapitel in der Navigation des LMS bzw. in der rechten Navigation des SCOs kann man sich in beliebiger Reihenfolge durch die Lektionen bewegen. Auf die Seiten des Kapitels 1.1.1 kommt der Lerner über einen Weiter- bzw. Zurück-Button, der in den SCOs implementiert wurde. Die Kursstruktur und das Sequencing wurden im Vorfeld der Beispielumsetzung angelegt und gelten für alle drei Beispiele.

Alle drei Beispiele wurden in einem Kurs implementiert und greifen dort ineinander über. Dieser Kurs ist auf der beiliegenden CD dieser Arbeit zu finden.

4.3.3.1 Kursstruktur

Um die oben genannte Abarbeitung zu ermöglichen, muss zunächst die Struktur des Kurses festgelegt werden. Dies geschieht in der Content *Organization*. Die Content Organization enthält sämtliche Activities. Jede einzelne Seite des Lernmoduls ist, anders als in der Flash-Version, ein SCO.

Activities	Bezeichnung	ID	Sichtbarkeit
Wurzel-Activity	ORGANIZATION		Sichtbar
Activity	Kapitel 1	KAPITEL_1	Sichtbar
Activity	Kapitel 1.1	KAPITEL_1_1	Sichtbar
Activity	Kapitel 1.1.1	KAPITEL_1_1_1	Sichtbar
Activity	Seite 1	SITE_1	Nicht sichtbar
Activity	Seite 2	SITE_2	Nicht sichtbar
Activity	Seite 3	SITE_3	Nicht sichtbar
Activity	Seite 4	SITE_4	Nicht sichtbar
Activity	Kapitel 1.1.2	KAPITEL_1_1_2	Sichtbar
Activity	Kapitel 1.1.3	KAPITEL_1_1_3	Sichtbar
Activity	Kapitel 1.1.4	KAPITEL_1_1_4	Sichtbar

Aus dieser Struktur resultierend, sehen die `<item>`-Elemente der `<organization>` wie folgt aus.

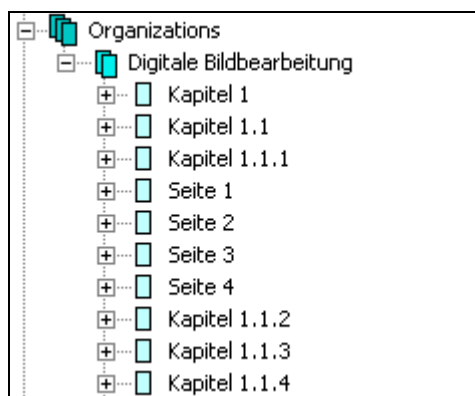


Abbildung 37 - Darstellung `<item>`-Elemente im Reload Editor

```
- <organizations default="Bildbearbeitung">
- <organization identifier="Bildbearbeitung">
  <title>Digitale Bildbearbeitung</title>
  + <item identifier="KAPITEL_1" identifierref="RES_KAPITEL_1" isvisible="true">
  + <item identifier="KAPITEL_1_1" identifierref="RES_KAPITEL_1_1" isvisible="true">
  + <item identifier="KAPITEL_1_1_1" identifierref="RES_KAPITEL_1_1_1" isvisible="true">
  + <item identifier="SITE_1" identifierref="RES_SITE_1" isvisible="false">
  + <item identifier="SITE_2" identifierref="RES_SITE_2" isvisible="false">
  + <item identifier="SITE_3" identifierref="RES_SITE_3" isvisible="false">
  + <item identifier="SITE_4" identifierref="RES_SITE_4" isvisible="false">
  + <item identifier="KAPITEL_1_1_2" identifierref="RES_KAPITEL_1_1_2" isvisible="true">
  + <item identifier="KAPITEL_1_1_3" identifierref="RES_KAPITEL_1_1_3" isvisible="true">
  + <item identifier="KAPITEL_1_1_4" identifierref="RES_KAPITEL_1_1_4" isvisible="true">
  + <imsss:sequencing>
  </organization>
</organizations>
```

Quellcode 8 - Darstellung `<item>`-Elemente in XML

4.3.3.2 Sequencing

Da es in dieser Umsetzung in erster Linie um das Navigation Modell geht, wurde das Sequencing nur soweit wie nötig umgesetzt. Notwendig für die Anwendung von „continue“ und „previous“ ist die Steuerung des Inhalts und der Navigation. Dies

geschieht über die Control Modes. Da die Control Modes für alle Kapitel gleichermaßen gelten, wurden sie direkt auf die Wurzel Activity angewendet. Die Control Modes `choice` und `flow` wurden auf `true` gesetzt. Control Mode `choice` sorgt dafür, dass der Anwender alle sichtbaren Kapitel in beliebiger Reihenfolge anklicken kann. `Flow` gibt an, ob Vor- und Zurück-Blättern mit Hilfe von Buttons möglich ist. In unserem Beispiel ist `flow` auf `true` gesetzt, daher ist das Blättern möglich und das LMS stellt den Button dafür zur Verfügung. Ohne `flow = true` wäre die Anwendung von `continue` und `previous` nicht möglich. In der Manifestdatei sieht dies wie folgt aus:

```
<imsss:sequencing>
  <imsss:controlMode choice="true" flow="true" />
</imsss:sequencing>
```

Quellcode 9 - Sequencing Manifestdatei

Für die Anwendung von `choice` ist die Implementierung der Sequencing Control Modes nicht unbedingt notwendig. Dennoch ist es ratsam, da die Control Modes für die Steuerung von Inhalt und Navigation notwendig sind.

4.3.4 Beispielumsetzung: Weiter-Button

Ziel ist, das Weiter - bzw. Zurück-Blättern über einen Link oder Button innerhalb eines SCOs zu ermöglichen. Dazu musste zunächst in der Seite ein Button implementiert werden, der einen Continue Navigation Request für Weiterblättern und einen Previous Navigation Request für Zurückblättern auslöst.

Nachteile

- ◇ Der wohl größte Nachteil einer Pixelgrafik besteht darin, dass zu jedem Pixel dessen exakte *x/y*-Koordinaten und Farbwert gespeichert werden müssen. Dies führt bei hohen Auflösungen zu einem **hohen Speicherbedarf** und im Internet zu **hohen Ladezeiten**.
- ◇ Einen weiteren Nachteil stellt die **Skalierbarkeit** dar. Hier wird bei hohen Vergrößerungen die Zusammensetzung aus einzelnen Pixeln ersichtlich und das Bild verliert an Qualität.



Zurück

Zur nächsten Seite

Abbildung 38 - Seiten Ausschnitt Kapitel 1.1.1 Seite 2

Hier findet das Run-Time Navigation Data Model Anwendung. Über dieses Model können Navigation Requests vom SCO zum LMS übertragen werden. Realisiert wird das mittels JavaScript Funktionen die den Wert entweder auf `continue` oder auf `previous` setzen

und anschließend die Kommunikation mit dem LMS beenden. Für die jeweilige Funktionalität des Buttons wurde jeweils eine Funktion geschrieben.

```
function doContinue()
{
    doSetValue("adl.nav.request","continue");
    doTerminate();
}
```

Quellcode 10 - Funktion für Weiterbutton

```
function doPrevious()
{
    doSetValue("adl.nav.request","previous");
    doTerminate();
}
```

Quellcode 11 - Funktion für Zurückbutton

DoContinue() und doPrevious() übermitteln jeweils einen Navigation Request aus dem SCO an das LMS. Sie beenden danach die Kommunikation mit dem LMS. Die Datentypen continue und previous machen dem LMS kenntlich um welchen Navigation Request es sich handelt (Weiter oder Zurück blättern). DoTerminate() sorgt dafür, dass die Kommunikation mit dem LMS beendet wird. DoTerminate() ruft eine Funktion in der *APIWrapper.js* auf, die von ADL zur Verfügung gestellt wird, um dem Content Entwickler die Arbeit mit SCORM zu erleichtern.

Wenn nun der Button und die Funktionalität im SCO implementiert werden, ist es wichtig die Navigations Elemente des LMS auszublenden, damit keine doppelten Navigationsmöglichkeiten bestehen. Das Ausblenden der vom LMS angebotenen Navigationselemente wird direkt in der Manifestdatei mit Hilfe des Presentation Navigation Model gesteuert. Mit diesem kann, wie zuvor beschrieben, festgelegt werden, welche UI-Devices nicht bereit gestellt werden sollen. In diesem Fall sollen Weiter- und Zurück-Button nicht bereit gestellt werden. In der Manifestdatei sieht das wie folgt aus:

```
- <adlnav:presentation>
  - <adlnav:navigationInterface>
    <adlnav:hideLMSUI>previous</adlnav:hideLMSUI>
    <adlnav:hideLMSUI>continue</adlnav:hideLMSUI>
  </adlnav:navigationInterface>
</adlnav:presentation>
```

Die Navigationselemente Weiter (continue) und Zurück (previous) wurden mit hideLMSUI ausgeblendet. Sie stehen dem Lerner somit nicht mehr zur Verfügung. Da sie dem Lerner auf keiner Seite und in keinem der drei Beispielumsetzungen zur Verfügung stehen sollen, wurde das Presentation Navigation Model auf jedes SCO angewandt. Daher sind die Element zwar durch die Anwendung der Sequencing Control Modes zwar vorhanden,

aber nicht sichtbar. Ihre Funktionalitäten wurden auf die Buttons innerhalb der SCOs übertragen.

4.3.5 Beispielumsetzung: Verlinkung

Damit man von der Seite Grundlagen auf die folgende Seite des Kapitels kommt, wurde ein Textlink auf der Seite implementiert, der direkt zum Kapitel 1.1.1 Theorie führt. Realisiert wurde dieses mit dem Run-Time Navigation Data Model. dieses sorgt dafür, dass ein Choice Navigation Request beim Anklicken des Links ausgelöst wird und anschließend die Kommunikation mit dem LMS beendet wird. Analog zum Weiter-Button geschieht auch dies über eine JavaScript Funktion.

```
function doChoice()  
{  
    doSetValue("adl.nav.request", "{target=KAPITEL_1_1}choice");  
    doTerminate();  
}
```

Quellcode 12 - Funktion für Textlink

DoChoice() unterscheidet sich nicht wesentlich zur doContinue() bzw. doPrevious(). Anstelle von continue oder previous wird mit dieser Funktion der Wert auf choice gesetzt. Außerdem wird gesagt welche SCO mit diesem Navigation Request aufgerufen werden soll. Dazu gibt man die ID des jeweiligen SCOs an. In unserem Fall ist das KAPITEL_1_1. Ist das Run-Time Navigation Data Model umgesetzt, findet wieder das Presentation Navigation Model Anwendung. Auch in diesem Beispiel wurden Weiter- und Zurück-Button ausgeblendet, daher ist die Anwendung des Models analog zum ersten Beispiel und wird nicht gesondert aufgeführt.

4.3.6 Beispielumsetzung: Rechte Navigation

Für die Beispielumsetzung der rechten Navigation wurde auf dasselbe Prinzip wie bei der Realisierung des Textlinks zurückgegriffen. Auch für dieses Beispiel musste ein choice Navigation Request ausgelöst und anschließend die Kommunikation mit dem LMS beendet werden. Um die verschiedenen Ziele der Navigation anzusteuern, wurde die Grafik mit einem ImageMap versehen. Beim Klicken auf die sensitiven Bereiche der ImageMap wurde ein Link angeklickt, der die entsprechend hinterlegte JavaScript-Funktion aufruft. Der Einfachheit halber wurde für jeden einzelnen Link eine Funktion geschrieben, die dann die jeweilige ID des SCO beinhaltet. Natürlich ließe sich das auch über eine einzige Funktion regeln, die über Variablen die jeweilige ID vergibt. Auch das Presentation Model findet hier wieder Anwendung, da auf Grund der Verlinkung der rechten Navigation ein Weiter- bzw. Zurück-Button benötigt wurden.

4.4 Fazit

Nach der Umsetzung der einzelnen Möglichkeiten konnten verschiedene positive und auch negative Möglichkeiten des Navigation Model festgestellt werden. Das Navigation Model kann gut bei Lernmodulen angewandt werden, die aus einzelnen HTML-Seiten oder sehr kleinen Flashfilmen bestehen. Das Navigation Model findet nur Anwendung zwischen SCO und LMS. Die Navigation innerhalb eines SCO wird dabei nicht unterstützt. Mit jedem choice, previous und continue Navigation Request eines SCOs an das LMS wird ein neues SCO geladen. Bei HTML-Seiten macht das bei der Arbeit nichts aus. Bei größeren Flashfilmen könnte das wiederum auf Grund der Ladezeiten zu Problemen führen.

In der Beispielumsetzung wurden daher alle HTML-Seiten als einzelne SCOs deklariert um die Navigation zu testen. Inwieweit das Sinn macht, muss jeder Content Entwickler individuell entscheiden. Beim ursprünglichen Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ sind die SCOs sehr grob gefasst. Ein SCO beinhaltet dabei die verschiedenen Abschnitte und dazu die jeweiligen Seiten. Mit Hilfe von SCORM ließ sich prüfen, ob alle Abschnitte abgearbeitet wurden. Bei der Prüfung der einzelnen Seiten, sind wir jedoch auf Grenzen gestoßen. Bei der Beispielumsetzung wurden die SCOs sehr fein definiert: 1 SCO gleich eine Seite. Daher war die Prüfung der Abschlussstati der einzelnen Seiten gar kein Problem mehr. Jedoch sollte man an dieser Stelle genau überprüfen was mehr Sinn macht.

Weiterhin fällt auf, dass das Navigation Model in der jetzigen Form den Grundsatz von SCORM verletzt. Löst man einen Continue oder Previous Navigation Request aus, bleibt der Grundsatz der Unabhängigkeit der SCO *fast* gewahrt. Denn es wird einfach zum nächsten oder letzten SCO im Aktivitäten-Baum gegangen egal welche SCO dort jetzt steht. Es ist nur fraglich, ob nicht der Grundsatz der Unabhängigkeit der SCOs schon dadurch verletzt wird, dass ein Continue oder Previous Navigation Request nur Anwendung finden kann, wenn der Sequencing Control Mode `flow` auf `true` gesetzt wird. Inwieweit die Verletzung hier zutrifft kann nicht mit voller Bestimmtheit gesagt werden. Sicher ist allerdings, dass bei der Verwendung des Choice Navigation Requests der Grundsatz der Unabhängigkeit in jedem Fall verletzt wird. An dieser Stelle muss im SCO genau die ID festgelegt werden auf die verlinkt werden soll. Kommt jetzt statt dem SCO mit der entsprechenden ID ein anderes SCO, funktioniert die Verlinkung nicht mehr. Die SCOs sind durch die Verlinkung nicht mehr unabhängig voneinander. Sie können nicht mehr beliebig kombiniert, ausgetauscht und weggelassen werden. Da ich die einzelnen HTML-Seiten als jeweils ein SCO definiert habe, wurde schon dadurch der Grundsatz der Unabhängigkeit von mir selbst verletzt, da die SCOs nun keine inhaltlich abgeschlossenen sinnvollen Lerneinheiten sind.

Das Navigation Presentation Model dient dazu, Dopplungen in der Navigation zu vermeiden. Aus Usability-Sicht ist das eine sehr sinnvolle Funktion, da das doppelte Vorhandensein einer Navigation den Nutzer eher verwirrt als unterstützt.

Die SCORM SN Spezifikation spezifiziert weder Aussehen oder Art der Anwenderschnittstelle, noch Mechanismen für die Interaktionen. Merkmale wie Aussehen, Verhalten, Präsentation und Anwenderschnittstelle bzw. deren Kontrollmechanismus sind noch nicht Gegenstand von SCORM. Es gibt lediglich Empfehlungen zur Interpretation des SCORM Navigation Model. An dieser Stelle macht sich bemerkbar, dass es noch keine formalen Standards für Navigation und Präsentation gibt. SCORM ist an dieser Stelle noch sehr unausgereift.

Zusammenfassend hat das Navigation Model gute Ansätze und lässt sich sehr einfach umsetzen. Allerdings ist nach meiner Meinung das SCORM Buch 4 nicht sehr dienlich, da es an dieser Stelle sehr knapp gehalten ist und nur Interpretationshinweise gibt. Bei der Umsetzung des Navigation Data Models sollte der Sinn der einzelnen Navigationselemente nicht außer Acht lassen und Prioritäten gesetzt werden. Will man den Grundsatz von SCORM nach der Unabhängigkeit der einzelnen Elemente beherzigen, sollte man auf das Navigation Model doch eher verzichten, da es diese Grundsätze verletzt. Im Lernmodul „Digitale Bildbearbeitung mit Adobe Photoshop CS2“ macht eine Anwendung des Navigation Models nur Sinn, wenn darüber eine Art Glossar gesteuert wird. Ansonsten ist die gröbere Einteilung der SCOs sinnvoller, auch wenn die Überprüfung der Beendigungsstati nicht ganz unproblematisch ist. Im Buch SCORM SN ist das Sequencing der ausgereifere und sinnvollere Teil.

Abbildungsverzeichnis

Abbildung 1 - Phasen der Produktion [Ris06, S. 98]	9
Abbildung 2 - Unterteilung Konzeptions- und Realisierungsphase	10
Abbildung 3 - Screenshot Basalttext Lernmodul "Digitale Bildbearbeitung mit Adobe Photoshop CS2"	11
Abbildung 4 - Finales Layout des Lernmoduls	14
Abbildung 5 - Struktur des Lernmoduls.....	23
Abbildung 6 - Ablaufplan Lernmodul „Digitale Bildbearbeitung“	25
Abbildung 7 - Navigation "Einführung in die interaktive Arbeit mit Flash".....	26
Abbildung 8 - Navigation "Digitale Bildbearbeitung mit Adobe Photoshop CS2"	26
Abbildung 9 - Ausgangszustand/RollOut	27
Abbildung 10 - RollOver	27
Abbildung 11 - Besucher Abschnitt	27
Abbildung 12 – Layout Lernmodul „Digitale Bildbearbeitung“	29
Abbildung 13 – Navigation des Lernmoduls „Digitale Bildbearbeitung“	32
Abbildung 14 – Orientierungsleiste des Lernmoduls „Digitale Bildbearbeitung“.....	32
Abbildung 15 – Screenshot des Lernmoduls „Digitale Bildbearbeitung“	33
Abbildung 16 – Player des Lernmoduls „Digitale Bildbearbeitung“	34
Abbildung 17 - kleiner Player	34
Abbildung 18 – Grafik des Lernmoduls „Digitale Bildbearbeitung“	35
Abbildung 19 – Download des Lernmoduls „Digitale Bildbearbeitung“	35
Abbildung 20 - SCORM-Bücher [SCO04a].....	41
Abbildung 21 - Content Aggregation Model (CAM) [SCO04b]	41
Abbildung 22 - Run-Time Environment [SCO04c]	46
Abbildung 23 - Sequencing and Navigation (SN) [SCO04d].....	48
Abbildung 24 - Activity mit Sub-Activities [SCO04d, S. SN-2-9]	50
Abbildung 25 - Cluster Beispiel [SCO04d, S. SN-2-7].....	51
Abbildung 26 –Beispiel eines einfachen Activity Tree [SCO04d, S. SN-2-3]	52
Abbildung 27 - Ableitung des Activity Tree aus der Content Organization, nach [SCO04d, S. SN-2-5]	53
Abbildung 28 - Screenshot Reload Editor Version 2.5.4	73
Abbildung 29 - Screenshot Sample Run-Time Environment.....	74
Abbildung 30 - Logganzeige der Sample RTE	75
Abbildung 31 - Struktur Prototyp Digitale Bildbearbeitung mit Adobe Photoshop CS2 ...	75
Abbildung 32 - Textlink im Wurzel-Element Kapitel 1	76
Abbildung 33 - Navigation wie sie vom LMS angeboten wird	76
Abbildung 34 - Rechte Navigation des Lernmoduls	76
Abbildung 35 - Weiter- bzw. Zurück-Button zum Navigieren zwischen den Seiten	76

Abbildung 36 - LMS Navigation ohne Darstellung der einzelnen Seiten	77
Abbildung 37 - Darstellung <item>-Elemente im Reload Editor	78
Abbildung 38 - Seiten Ausschnitt Kapitel 1.1.1 Seite 2	79

Tabellenverzeichnis

Tabelle 1 - Typografische Aspekte der Textgestaltung.....	13
Tabelle 2 - Datenelemente in einem Content Package [ELA07].....	45
Tabelle 3 - Metadatenkategorien [Ger04].....	46
Tabelle 4 - Activity State Information.....	56
Tabelle 5 - Global State Information.....	57
Tabelle 6 - Navigations Ereignisse und Beschreibungen.....	61
Tabelle 7 - Präsentation des Information Model.....	65
Tabelle 8 - Laufzeit Nutzereingabeschnittstellen Wortschatz.....	66
Tabelle 9 - SCORM Navigation Data Model.....	67
Tabelle 10 - Dot-notation Binding für das Request Data Model Element.....	92
Tabelle 11 - Dot-notation Binding für das Request Valid Data Model Element.....	95

Quellcodeverzeichnis

Quellcode 1 - NavigationButton.as	27
Quellcode 2 - OrientationButton.as	28
Quellcode 3 - ModulBackground.as	29
Quellcode 4 - modul_init.xml	30
Quellcode 5 - Anwendung adl.nav.request	69
Quellcode 6 - Anwendung adl.nav.request mit choice	69
Quellcode 7 - Anwendung adl.nav.request_valid	70
Quellcode 8 - Darstellung <item>-Elemente in XML	78
Quellcode 9 - Sequencing Manifestdatei.....	79
Quellcode 10 - Funktion für Weiterbutton.....	80
Quellcode 11 - Funktion für Zurückbutton	80
Quellcode 12 - Funktion für Textlink.....	81

Literaturverzeichnis

- [Ger04] GERICKE, ANNE-KATHRIN: *Untersuchung und Bewertung von SCORM 2004 sowie prototypische Realisierung von Online-Kursmaterial nach dieser Spezifikation*. Diplomarbeit der Hochschule für Technik und Wirtschaft Dresden, Dezember 2004.
- [Kan05] KANNENGIESSER, Caroline und Matthias: *ActionScript Praxisbuch*. Franzis Verlag GmbH Poing, 2005.
- [Pie05] PIETZSCH, MELANIE: *Entwicklung einer benutzerfreundlichen Anwendung zur Generierung von Navigationsstrukturen für E-Learning-Kurse nach der SCORM 1.3 Spezifikation*. Diplomarbeit der Hochschule für Technik und Wirtschaft Dresden, September 2005.
- [Ris02] RISER, U.: *Konzeption und Entwicklung interaktiver Lernprogramme*. Springer-Verlag Berlin Heidelberg, 2002.
- [RüI06] RÜLKE, TORSTEN: *Anwendung objektorientierter Konzepte mit ActionScript 2 zur Realisierung wieder verwendbarer Lerninhalte - Design und Entwicklung des Kurses „Die interaktive Arbeit mit Flash MX 2004“ für das Bildungsportal Sachsen*. Diplomarbeit der Hochschule für Technik und Wirtschaft Dresden, Juni 2006.
- [SCO04a] *Sharable Content Object Reference Model (SCORM) 2004 - book 1 - Overview*, Juli 2004.
- [SCO04b] *Sharable Content Object Reference Model (SCORM) 2004 - book 2 – Content Aggregation Model*, Juli 2004.
- [SCO04c] *Sharable Content Object Reference Model (SCORM) 2004 - book 3 - Run-Time-Environment*, Juli 2004.
- [SCO04d] *Sharable Content Object Reference Model (SCORM) 2004 - book 4 – Sequencing and Navigation*, Juli 2004.
- [ELA07] *ELAN Portal Niedersachsen* <http://www.elan-niedersachsen.de/index.php?id=707> Stand: 29.Juli 2007

Anhang A

Ergänzung zum Run-Time Navigation Data Model

Die folgenden Punkte definieren die Anforderungen für die Implementierung des SCORM Navigation Data Model. Jedes Data Model Element wird in einem neuen Punkt erläutert. Jeder Punkt beinhaltet eine Tabelle, welche die Anforderungen für das erläuterte Element enthält. Die Anforderungen können sowohl auf das LMS als auch auf das SCO angewandt werden. Einige Anforderungen beeinflussen LMS-Implementierungen, andere beeinflussen SCO-Implementierungen und einige beeinflussen beides. Die nachstehende Tabelle erläutert, wie die folgende Tabelle aussieht und zu verstehen ist.

Dot-Notation Binding	Details
<p><not-notation characterstring representation of the data model element></p>	<p><u>Data Element Implementierungsanforderungen:</u> Dieser Teil der Tabelle definiert die Implementierungsanforderung des Data Model Elements. Dieser Teil behandelt die Anforderungen, die für LMS und SCO gleichermaßen gelten. Er ist untergliedert in drei Teile: Daten Typ, Platzhalter und Format.</p> <ul style="list-style-type: none"> • Daten Typ: Beschreibt den speziellen Daten Typ für das Data Model Element. Diese Daten Typen sind definiert im Data Typ Teil des SCORM RTE Buches. • Platzhalter: Stellt die Menge der Werte dar, die im Datentyp verwendet werden können. • Format: Beschreibt die Formatbeschränkungen für die Werte dieses Datentyps. <p><u>LMS Verhaltensanforderungen:</u> Dieser Teil beschreibt die Gruppe der Anforderungen, die für das LMS gelten.</p> <p><u>SCO Verhaltensanforderungen:</u> Dieser Teil beschreibt die Gruppe der Anforderungen, die für ein SCO gelten.</p> <p><u>API Implementierungsanforderungen:</u></p> <ul style="list-style-type: none"> • GetValue(): Dieser Abschnitt behandelt das spezielle Verhalten, welches ein LMS befolgen sollte, wenn es den GetValue()-Aufruf für ein spezielles Data Model Element abarbeitet. Außerdem behandelt dieser Abschnitt die Fehler, die bei der Verwendung des Data Model Element mit einem GetValue()-Aufruf auftreten können. • SetValue(): Dieser Abschnitt behandelt das spezielle Verhalten, welches ein LMS befolgen sollte, wenn es den SetValue()-Aufruf für das spezielle Data Model Element abarbeitet. Außerdem behandelt dieser Abschnitt die Fehler, die bei der Verwendung des Data Model Elements mit einem SetValue()-Aufruf auftreten können. <p><u>Ergänzende Verhaltensanforderungen</u> Dieser Abschnitt behandelt einige ergänzende Verhaltensanforderungen, die speziell für diese Data Model Element gelten.</p> <p><u>Beispiel:</u> Dieser Abschnitt stellt ein Beispiel für gültige API Methoden Aufrufe dar, die das Data Model Element verwendet.</p>

Adl.nav.request

Dot-Notation Binding	Details
adl.nav.request	<p>Dieses Element wird vom SCO genutzt, um einen verlangten Navigation Request zum Abarbeiten abzubilden, sofort gefolgt vom erfolgreichen Aufruf <code>Terminate()</code>.</p> <p>Data Element Implementierungsanforderungen:</p> <ul style="list-style-type: none"> • Datentyp: (begrenzt) Buchstabenketten (continue, previous, choice, exit, exitAll, abandon, abandon All, and _non_), und eventuell ein Target Trennsymbol als Buchstabenkette • Format: Das Format des Characterstring sollte folgendes sein: <ul style="list-style-type: none"> ○ {target=<STRING>} <navigation request> <p>Das Trennsymbol {target=<STRING>} kennzeichnet das Ende eines Choice-Navigation Requests. Das Trennsymbol wird verlangt und sollte die erste Serie von Buchstaben sein, die im parameter_2 des SetValue Aufrufes stehen, wenn der Navigation Request „choice“ ist. Der Wert des <STRING> weist typischerweise auf die Attributbezeichnung eines <item> Elementes vom Content Package hin.</p> <p>Alle anderen Navigation Request, inklusive des Trennsymbols, führen zu einem Fehler.</p> • Value Space: SCORM verbindet die erlaubten Werte für die Buchstabenketten zu den folgenden begrenzten Vokabelmerkmalen. <ul style="list-style-type: none"> ○ „continue“: Weist das LMS darauf hin, dass der Inhalt behauptet, dass ein Continue Navigation Request sofort abgearbeitet werden soll, gefolgt von der Beendigung des SCOs. ○ „previous“: Weist das LMS darauf hin, dass der Inhalt behauptet, dass ein Previous Navigation Request sofort abgearbeitet werden soll, gefolgt von der Beendigung des SCOs. ○ „choice“: Weist das LMS darauf hin, dass der Inhalt behauptet, dass ein Choice Navigation Request sofort abgearbeitet werden soll, gefolgt von der Beendigung des SCOs. ○ „exit“: Weist das LMS darauf hin, dass der Inhalt behauptet, dass ein exit Navigation Request sofort abgearbeitet werden soll, gefolgt von der Beendigung des SCOs. ○ „exitAll“: Weist das LMS darauf hin, dass der Inhalt behauptet, dass ein exitAll Navigation Request sofort abgearbeitet werden soll, gefolgt von der Beendigung des SCOs. ○ „abandon“: Weist das LMS darauf hin, dass der Inhalt behauptet, dass ein Abandon Navigation Request sofort abgearbeitet werden soll, gefolgt von der Beendigung des SCOs. ○ „AbandonAll“: Weist das LMS darauf hin, dass der Inhalt behauptet, dass ein AbandonAll Navigation Request sofort abgearbeitet werden soll, gefolgt von der Beendigung des SCOs. ○ „_none_“: Weist das LMS darauf hin, dass der Inhalt behauptet, dass einige frühere Navigation Requests, angegeben durch das SCO nicht sofort abgearbeitet werden sollen, gefolgt von der Beendigung des SCOs. Das Setzen dieses Wertes ist sinnvoll, um schwebende Navigation Request zu beenden.

LMS Verhaltensanforderungen:

- Dieses Element ist zwingend und sollte vom LMS auf Lesen und Schreiben gesetzt werden
- Normalerweise möchten die Lerner ihren Navigationswunsch durch einige Navigations UI-Devices anzeigen, die vom LMS angeboten werden. In einigen Fällen könnte die Nutzerschnittstelle in ein SCO eingebettet sein oder das SCO könnte den Wunsch haben, einen Navigation Request im Interesse des Lernenden anzubieten. Beim Ausbleiben einer durch den Lerner aufgerufenen Navigationsanfrage über eine vom LMS angebotene UI-Devicessteuerung arbeitet das LMS den Navigation Request ab, der durch ein SCO identifiziert wurde.
- Die default Navigationsanfrage, wenn sie nicht durch das SCO gesetzt wurde, sollte „_none_“ sein.
- Nach dem normalen Ende eines SCO (das SCO setzt cmi.exit to „“ oder „normal“) und dem Fehlen einer Navigationsanfrage durch den Lerner, soll das LMS die Navigationsanfrage abarbeiten, die durch dieses Element mit Hilfe des Aktivitäten-Baums identifiziert wurde.
- Wenn das SCO in einem verschobenen Status endet (das SCO setzt cmi.exit zu „suspend“ oder „logout“), soll das LMS die Navigationsanfrage dieses Elements nicht abarbeiten. Es soll stattdessen eine Suspend oder SuspendAll Anfrage abarbeiten.

SCO Verhaltensanforderungen:

- Dieses Element ist durch das LMS als Lesen und Schreiben implementiert. Das SCO darf den Wert des adl.nav.request Data Model Elementes abfragen und ablegen.

API Implementierungsanforderungen:

- **GetValue():**
 - Das LMS soll den zugehörigen Navigation Request zurückgeben, der derzeit durch das LMS für das SCO betreut wird und einen Error Code von „0“ (No Error) angeben. Die zurück gegebene Zeichenkette soll den Data Element Implementierungsanforderungen zugeordnet werden.
 - Bis das SCO etwas anderes abbildet, ist der Default-Wert vom adl.nav.request „_none_“.
 - Wenn die Navigationsanfrage zwischenzeitlich vom LMS betreut wird, ist das Format des zurückgegebenen String für Choice {target=<STRING>}choice
 - Wobei <STRING> das Ziel der bevorstehenden Navigationsanfrage ist.

ADL Note: Die generelle Syntax des Separators ist definiert im SCORM RTE Buch.

- **SetValue():**
 - Wenn das SCO eine Anfrage aufruft, um den Navigation Request zu setzen, und der Wert kein Mitglied der beschränkten Vokabel-Zeichen ist, die oben beschrieben wurden, gibt das LMS false zurück und der Fehlercode wird auf 406 - Data Model Element Type unpassend - gesetzt. Das LMS soll den Status des Elements nicht ändern, welches auf dem Aufruf beruht.
 - Wenn der Navigation Request „choice“ ist und der Trenner {target=<STRING>} wird nicht angeboten

	<p>bzw. ist unpassend formatiert, soll das LMS false zurück geben und den Fehlercode auf 406 setzen, sowie den aktuellen Status des Elementes nicht ändern.</p> <ul style="list-style-type: none"> • Wenn der Navigation Request nicht „choice“ ist und der Trenner {target=<String>} angeboten wird, dann soll das LMS false zurückgeben und den Fehlercode auf 406 setzen, sowie den aktuellen Status des Elements nicht ändern. <p>Beispiele:</p> <ul style="list-style-type: none"> • GetValue(„adl.nav.request“) • SetValue(„adl.nav.request“,“{target=intro}choice“) • SetValue(„adl.nav.request“,„continue“)
--	---

Tabelle 10 - Dot-notation Binding für das Request Data Model Element

Adl.nav.request_valid

Dot-Notation Binding	Details
adl.nav.request_valide. continue	<p>Dieses Data Model Element wird von einem SCO genutzt, wenn ein „Continue“ Navigation Request am gegenwärtig bekannten Standort des Aktivitäten-Baums abgearbeitet werden soll.</p> <p><u>Data Element Implementierungsanforderungen:</u></p> <ul style="list-style-type: none"> • Daten Typ: Status (true, false, unknown) • Platzhalter: SCORM verbindet diese Status-Werte zu den folgenden Vokabelzeichen: <ul style="list-style-type: none"> ○ „true“: Das LMS fordert einen „Continue“ Navigation Request, der am aktuell bekannten Status am Aktivitäten-Baum abgearbeitet wird. ○ „false“: Das LMS verlangt, dass ein „Continue“ Navigation Request, der am aktuell bekannten Status des Aktivitäten Baums abgearbeitet wird, nicht zur Identifikation einer Aktivität verwendet wird. ○ „unkown“: Das LMS kann sich nicht für ein Ergebnis aus der Continue Navigation Requests entscheiden. Diese Anforderung steht am aktuell bekannten Status des Aktivitäten-Baums. Das SCO sollte keine Annahmen über das LMS oder das Resultat der Abarbeitung der Navigationsanfrage machen. • Format: Das Format des Data Model Wertes sollte eins der drei eingeschränkten Zeichen sein, die oben aufgelistet wurden („true“, „false“, „unkown“). <p><u>LMS Verhaltensanforderungen:</u></p> <ul style="list-style-type: none"> • Dieses Element ist zwingend und soll vom LMS nur auf Lesen gesetzt werden. • Es ist zu empfehlen, dass das LMS Gültigkeitsprüfungen an einer Continue-Navigationsabfrage vor dem Starten des aktuellen Content Objektes verrichtet. Dies befähigt das LMS richtige und sinnvolle Navigationssteuerungen in UI-Devices anzubieten und gültige Navigation Requests vom SCO zu beantworten. • Es ist zu empfehlen, dass das LMS Gültigkeitsprüfungen an einer Continue-Navigationsanfrage ausführt. Und zwar immer zu der Zeit, wenn ein Commit() Aufruf von einem SCO abgearbeitet wird und Sequencing verwandte Tracking-Informationen (Fortschritt, Objekt, Maße, Ziele) aktualisiert werden müssen. • Der Default-Status, bis er vom LMS evaluiert wurde, soll „unkown“ sein.

	<p><u>SCO Verhaltensanforderungen:</u></p> <ul style="list-style-type: none"> • Das Element ist als nur Lesen durch das LMS implementiert. • Dem SCO ist es gestattet den Wert des adl.nav.request_valid.continue Data Model Elements abzufragen. • Wenn „unknown“ von einem GetValue()-Aufruf zurückgegeben wird, ist es zu empfehlen, dass das SCO für eine Weile wartet und den Aufruf wiederholt. <p><u>API Implementierungssanforderungen:</u></p> <ul style="list-style-type: none"> • GetValue(): Das LMS beabsichtigt, das Ergebnis eines bestätigten Continue-Navigation Requests, entgegen des aktuellen Status des Activity-Tree vom LMS für den Lerner bearbeitet wird, zurück zu geben. Zudem beabsichtigt das LMS einen Error Code von „0“ – „Kein Fehler“ zu schreiben. Der Rückgabewert wird den Anforderungen folgen, die in den DATA ELEMENT-Implementierungen festgelegt sind. • SetValue(): Wenn das SCO SetValue() aufruft, um adl.nav.request_valid.continue zu setzen, dann soll das LMS den Error Code auf 404 – <i>Data Model Element ist nur zum Lesen</i> - setzen und <i>falsch</i> zurückgeben. Das LMS soll nicht den Status des Elements ändern, dass auf diesem Aufruf basiert. <p><u>Beispiel:</u></p> <ul style="list-style-type: none"> • GetValue(„adl.nav.request_valid.continue“)
Adl.nav.request_valid.previous	<p>Dieses Data Model Element wird von einem SCO zum Aufrufen genutzt, wenn ein „Previous“ Navigation Request identifiziert wurde. Der „Previous“ Navigation Request wird im gegenwärtigen Status des Aktivitäten-Baums bearbeitet.</p> <p><u>Data Element Implementierungsanforderungen:</u></p> <ul style="list-style-type: none"> • Daten Typ: Status (true, false, unknown) • Platzhalter: SCORM verbindet diese Statusanzeiger zu den folgenden Vokabelzeichen: <ul style="list-style-type: none"> ○ „true“: Das LMS fordert die Abarbeitung eines „Previous“ Navigation Requests am aktuellen Status des Aktivitäten Baums. Der Aufruf führt zum Auffinden eines entsprechenden Elementes. ○ „false“: Das LMS fordert die Abarbeitung eines „Previous“ Navigation Requests am aktuellen Status des Aktivitäten Baums. Im Ergebnis wird keine Aktivität zurückgegeben. ○ „unkown“: Das LMS entscheidet sich nicht für das Resultat der Previous Navigation Requests, die am aktuell bekannten Status des Aktivitäten Baums abgearbeitet wird. Das SCO sollte keine Annahmen über das LMS oder das Resultat der Abarbeitung der Navigationsanfrage aufstellen. • Format: Das Format des Data Model Wertes sollte eins der drei eingeschränkten Zeichen sein, die oben aufgelistet wurden („true“, „false“, „unkown“). <p><u>LMS Verhaltensanforderungen:</u></p> <ul style="list-style-type: none"> • Dieses Element ist zwingend und soll vom LMS nur auf Lesen gesetzt werden. • Es ist zu empfehlen, dass das LMS Gültigkeitsprüfungen an einer Previous Navigationsabfrage vor dem Starten des aktuellen Content Objektes verrichtet. Dies befähigt das LMS richtige und sinnvolle Navigationssteuerungen in UI-Devices anzubieten und gültige Navigation

	<p>Requests vom SCO zu beantworten.</p> <ul style="list-style-type: none"> • Es ist zu empfehlen, dass das LMS Gültigkeitsprüfungen an einer Previous Navigationsanfrage ausführt - immer zu der Zeit, wenn ein Commit() Aufruf von einem SCO abgearbeitet wird und Sequencing-nahe Tracking Informationen (Fortschrittsanzeige, Objektstatus, Messwerte, Ziele) aktualisiert werden müssen. • Der Default-Status, bis er vom LMS evaluiert wurde, soll „unkown“ sein. <p><u>SCO Verhaltensanforderungen:</u></p> <ul style="list-style-type: none"> • Das Element ist als nur Lesen durch das LMS implementiert. • Dem SCO ist es gestattet, den Wert des adl.nav.request_valid.Previous Data Model Elementes abzufragen. • Wenn „unknown“ von einem GetValue()-Aufruf zurück gegeben wird, ist es zu empfehlen, dass das SCO für eine Weile wartet und den Aufruf wiederholt. <p><u>API Implementierungsanforderungen:</u></p> <ul style="list-style-type: none"> • GetValue(): Das LMS fordert das Ergebnis eines bestätigten Continue-Navigation Requests, die entgegen des aktuellen Status des Activity-Tree vom LMS für den Lerner bearbeitet wird. Zudem wird das LMS einen Error Code von „0“ – „Kein Fehler“ schreiben. Der Rückgabewert wird den Anforderungen folgen, die in den DATA ELEMENT-Implementierungen festgelegt sind. • SetValue(): Wenn das SCO SetValue() aufruft, um adl.nav.request_valid.previous zu setzen, dann soll das LMS den Error Code auf 404 – Data Model Element ist nur zum Lesen - setzen und falsch zurück geben. Das LMS soll nicht den Status des Elements ändern, dass auf diesen Aufruf basiert. <p><u>Beispiel:</u> GetValue(„adl.nav.request_valid.Previous“)</p>
<p>Adl.nav.request_valid.choice. { target=STRING}</p>	<p>Dieses Data Model Element wird von einem SCO genutzt, wenn ein „Choice“ Navigation Request für eine abgebildet Aktivität, abgearbeitet am gegenwärtig bekannten Status des Aktivitäten-Baums, eine Aktivität identifiziert.</p> <p>Aktivitäten-Baums Adl.nav.request_valid.choice.{ target=<STRING>}</p> <p>Der Argumenttrenner { target=<STRING>} bildet das Ziel eines „Choice“ Gültigkeitsaufrufes ab. Der Argumenttrenner ist gefordert und muss sofort nach dem letzten Punkt im GetValue() Aufruf erscheinen.</p> <p><STRING> präsentiert einen Characterstring. Der Wert des <STRING> weist typischerweise auf die Attributbezeichnung eines <item> Elementes vom Content-Package hin, welches den Aktivitäten Baum ableitet.</p> <p><u>Data Element Implementierungsanforderungen:</u></p> <ul style="list-style-type: none"> • Daten Typ: Status (true, false, unknown) • Platzhalter: SCORM verbindet diese Status-Werte zu den folgenden Vokabelzeichen: <ul style="list-style-type: none"> ○ „true“: Das LMS fordert die Abarbeitung eines „Choice“ Navigation Requests am aktuellen Status des Aktivitäten-Baums. Der Aufruf führt zum Auffinden eines entsprechenden Elementes. ○ „false“: Das LMS fordert die Abarbeitung eines

	<p>„Choice“ Navigation Requests am aktuellen Status des Aktivitäten Baums. Im Ergebnis wird keine Aktivität zurückgegeben.</p> <ul style="list-style-type: none"> ○ „unkown“: Das LMS entscheidet sich nicht für das Resultat der Choice Navigation Requests, die am aktuell bekannten Status des Aktivitäten Baums abgearbeitet wird. Das SCO sollte keine Annahmen über das LMS oder das Resultat der Abarbeitung der Navigationsanfrage aufstellen. • Format: Das Format des Data Model Wertes sollte eins der drei eingeschränkten Zeichen sein, die oben aufgelistet wurden („true“, „false“, „unkown“). <p><u>LMS Verhaltensanforderungen:</u></p> <ul style="list-style-type: none"> • Dieses Element ist zwingend und soll vom LMS nur auf Lesen gesetzt werden. Dem SCO ist es gestattet den Wert des adl.nav.request_valid.choice Data Model Elements zurück zu setzen. • Das LMS ist nicht erforderlich, um für jede Aktivität im Aktivitäten-Baum das Element zu handhaben und zu unterstützen. Das LMS darf nur Anforderungen derart beantworten, wie sie die Data Element Implementierung vorschreibt. Empfehlung: Das LMS bewahrt abgeschlossene Gültigkeitsprüfungen solange wie möglich auf (bis ein Statuswechsel im Aktivitäten-Baum die aufbewahrten Werte beeinflusst), um die Beantwortungszeit für diese Anfragen zu verkürzen. • Der Default-Status, bis er vom LMS evaluiert wurde, soll „unkown“ sein. <p><u>SCO Verhaltensanforderungen:</u></p> <ul style="list-style-type: none"> • Das Element ist als nur Lesen durch das LMS implementiert. • Dem SCO ist es gestattet, den Wert des adl.nav.request_valid.choice Data Model Elementes abzufragen. • Wenn „unknown“ von einem GetValue()-Aufruf zurück gegeben wird, ist es zu empfehlen, dass das SCO für eine Weile wartet und den Aufruf wiederholt. <p><u>API Implementierungsanforderungen:</u></p> <ul style="list-style-type: none"> • GetValue(): <ul style="list-style-type: none"> ○ Bei Vorhandensein eines Zielbegrenzers gibt das LMS das Ergebnis einer geprüften Choice-Navigation Request zurück. Die entsprechende Zielaktivität, die entgegen des aktuellen Status des Aktivitäten-Baum. Zurückgegeben wird der Error-Code 0, kein Fehler. Der zurückgegebene Status entspricht den Anforderungen der Data Element Implementierung. ○ Bei Nichtvorhandensein oder Fehlerhaftigkeit eines Zielbegrenzers {target=<STRING>} wird das LMS „false“ zurückgeben. Dies zeigt der Error Code 301 an – General Get Failure. • SetValue(): Wenn das SCO SetValue() aufruft, um adl.nav.request_valid.previous zu setzen, dann soll das LMS den Error Code auf 404 – Data Model Element ist nur zum Lesen - setzen und falsch zurückgeben. Das LMS soll nicht den Status des Elements ändern, dass auf diesem Aufruf basiert. <p><u>Beispiel:</u> GetValue(„adl.nav.request_valid.choice.{target=intro}“)</p>
--	--

Tabelle 11 - Dot-notation Binding für das Request Valid Data Model Element

Anhang B

Drehbuch Beispielseite

Projekt	Autor	Datum	Bildschirmseite
Digitale Bildbearbeitung mit Photoshop	Wolfgang Schilling	2006-07-20	1
Kapitel	0 Beispieltext		
Unterkapitel	0.1 Beispieltext		
Unterunterkapitel	0.0.1 Beispieltext2		
Seite	1		
Bildschirmtext/Aufgaben bei Test		Grafiken und Foto	
<p>Hier steht der Text für die Seite. Wörter die auf der Seite fett oder vorgehoben erscheinen sollen, bitte auch hier hervorheben. Überschriften fett formatieren. Der Text ist vollständig ausformuliert.</p> <p>Bei Linktexten oder ähnlichem eine Bemerkung dazu, sowie wo der Linktext stehen soll.</p> <p>Beim Text sollten nicht mehr als 8-10 Wörter in einer Zeile stehen (60-100 Buchstaben).</p>		<p>Hier wird angegeben, welche Grafiken auf die Seite sollen und wo sie platziert werden. Die Grafiken müssen hier nicht eingefügt werden. Zu dem Drehbuch gibt es eine Ordnerstruktur, wo die Grafiken mit einem eindeutigen Namen eingeordnet werden. Im Drehbuch muss nur der Name angegeben werden. In welchem Ordner sich die Grafik (Beispielgrafik) befindet, ist aus den Kapiteln ersichtlich. Wichtig ist auch die Quelle der Grafiken, die sollte in Klammern hinter dem Namen erscheinen.</p> <p>Beispiel: <i>Werkzeugleiste (Schilling) = mittig zwischen Absatz eins und zwei.</i></p> <p>Wenn sich hinter einer Grafik ein Link verbirgt, zum Beispiel zum Starten des Videos, dann hier angeben mit einem Namen für die Video- oder Animationssequenz</p>	
Regie	Sprechertext		
Was passiert während des Sprechertextes, z. B. eine Grafik wird eingeblendet.	Sprechertexte sind Texte, die unabhängig von einer Animation sind. Die zum Beispiel ein Bild erklären. Reine Sprechertexte werden wir wahrscheinlich haben.		
Geräusche/Sound			
Art des Sounds, Wann erscheint er und wie lange			
Ablauf der Seite/Programmieranweisungen/Interaktionsbeschreibungen			
<p>Hier wird erklärt, was auf der Bildschirmseite abläuft. <i>Zum Beispiel beim Intro die Seite baut sich auf. Erst fliegen die Linien ein, dann das Bild und dann erscheint „Willkommen“ auf dem Bildschirm oder so.</i></p> <p>Weiterhin können hier Interaktionsbeschreibungen stehen, sowie Programmieranweisungen, falls sich die Seite anders als gewöhnlich verhalten soll.</p>			
Navigation			
Weiter		Zurück	

Hinweise

Die Felder Kapitel, Unterkapitel, Unterunterkapitel sowie Seite sind so formatiert, dass sich daraus automatisch eine Gliederung erstellen lässt, um sich später im Drehbuch einfacher zu Recht zu finden. Also am besten die Beispielseite kopieren, damit die Generierung der Gliederung funktioniert. Zum Generieren der Gliederung muss nur einfach die rechte Maustaste gedrückt und dann „Felder aktualisieren“, sowie „Gesamtverzeichnis aktualisieren“ ausgewählt werden. In die Historie einfach die Version des Drehbuchs eintragen und wann es bearbeitet wurde, damit wir den Überblick behalten, wann wer was gemacht hat.

Anhang C

STYLEGUIDE

1. Schriftgrößen und Schriftfarben

Titel des Lernmoduls: 22 px Arial – normal, Farbe: #6d7c7c

Digitale Bildbearbeitung mit Adobe Photoshop CS2

Orientierungszeile: 14 px Arial – normal, Farbe: #6d7c7c

1 Grundlagen – 1.1 Theorie

Fließtext: 13 px Arial – normal, Farbe: #6d7c7c

Als Pixel- und auch Rastergrafik....

Hauptüberschrift: 16 px Arial – Fett, Farbe: #6d7c7c

Pixelgrafiken

Zwischenüberschriften: 13 px Arial – Fett, Farbe: #6d7c7c

Vorteile

Hervorgehobener Text: 13 px Arial – Fett, Farbe: #6d7c7c

standardisiertes Bilddateiformat

Link: 13 px Arial – Kursiv, Farbe: #6d7c7c

standardisiertes Bilddateiformat

Link: besucht 13 px Arial – Kursiv, Farbe: #c0caca

standardisiertes Bilddateiformat

Link: rollOver 13 px Arial – Kursiv, Farbe: #6d7c7c

standardisiertes Bilddateiformat

Liste 1. Ebene: 13 px Arial – normal, Farbe: #6d7c7c

◇ Vektor- und Pixelgrafiken

◇

Beschriftung der Navigationsschaltflächen: 12 px Arial – normal, Farbe: #6d7c7c

1.1.1 Pixel- und Vektorgrafiken

Tooltipps: 11 px Arial – normal, Farbe: #6d7c7c

Seite 1

2. Elemente

Linien

Linie 1: 1px – Farbe: #6d7c7c



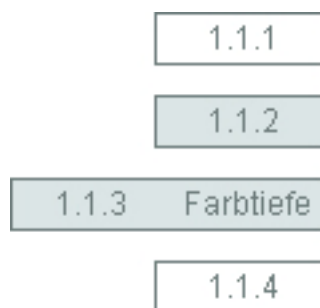
Linie 2: 2px – Farbe: #6d7c7c



Linie 3: 3px – Farbe: #6d7c7c



Navigationsschaltfläche



Inaktive Navigationsschaltfläche:

Background: #ffffff

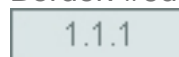
Border: #6d7c7c



Aktive Navigationsschaltfläche:

Background: #dbe5e5

Border: #6d7c7c



RollOver Navigationsschaltfläche:

Background: #dbe5e5

Border: #6d7c7c



Orientierungsschaltfläche



Inaktive Orientierungsschaltfläche

Background: #ffffff

Border: #6d7c7c



Aktive Orientierungsschaltfläche

Background: #6d7c7c

Border: #6d7c7c

Besuchte Orientierungsschaltfläche

Background: #dbe5e5

Border: #6d7c7c



Anhang D

Kapitel 0 - Einführung

Kapitel 0.1 - Lernziele

Kapitel 0.2 - Modulaufbau

Abschnitt 1 - Aufbau

Abschnitt 2 - Symbole

Kapitel 1 - Grundlagen

Kapitel 1.1 - Theorie

Abschnitt 1 - Pixel- und Vektorgrafik

Abschnitt 2 - Bild- und Druckauflösung

Abschnitt 3 - Farbtiefe

Abschnitt 4 - Farbmodus

Kapitel 1.2 - Tutorial: Arbeitsoberfläche

Abschnitt 1 - Menüleiste

Abschnitt 2 - Werkzeugpalette

Abschnitt 3 - Optionsleiste für Werkzeugeinstellungen

Abschnitt 4 - Paletten

Kapitel 1.3 Tutorial: Grundtechniken

Abschnitt 1 – Auswahl erstellen

Abschnitt 2 - Zoomen

Abschnitt 3 - Bildausschnitt verschieben

Kapitel 1.4 Tutorial: Elementare Funktionen

Abschnitt 1 - Geometrische Auswahlwerkzeuge

Abschnitt 2 - Hinzufügen, Entfernen von bestehender Auswahl

Abschnitt 3 - Punktorientierte Auswahlwerkzeuge

Abschnitt 4 - Freistellen

Abschnitt 5 - Zauberstab

Abschnitt 6 - Maskierungsmodus

Abschnitt 7 - Auswahl speichern

Abschnitt 8 - Auswahl transformieren

Kapitel 1.5 – Praxis

Kapitel 2 - Ebenen

Kapitel 2.1 - Tutorial Ebenen

Abschnitt 1 - Was sind Ebenen

Abschnitt 2 - Ebenenkontextmenü

Abschnitt 3 - Ebenen-Palette

Abschnitt 4 - Transparenz

Abschnitt 5 - Tiefensortierung und Sichtbarkeit

Abschnitt 6 - Deckkraft

Abschnitt 7 - Fläche

Abschnitt 8 - Ebenen reduzieren

Abschnitt 9 - Ebenen-Set

Abschnitt 10 - Ebenen-Typen

Abschnitt 11 - Ebenen-Stile

Abschnitt 12 - Ebenen-Füllmethoden

Kapitel 2.2 - Praxis: Ebenen

Abschnitt 1 - Ebenencollage

Abschnitt 2 - Ebenenbasierter Aquabutton

Kapitel 3 - Bildoptimierung

Kapitel 3.1 - Tutorial: Bildoptimierung

Abschnitt 1 - Bildoptimierung, Retusche

Abschnitt 2 - Histogramm

Abschnitt 3 - Gradationskurven

Abschnitt 4 - Tonwertkorrektur

Abschnitt 5 - Ausgewählte Filter

Kapitel 3.2 - Praxis: Bildoptimierung

Abschnitt 1 - Alte Fotografie

Abschnitt 2 - Porträtretusche

Abschnitt 3 - Urlaubsfotos optimieren

Kapitel 4 - Webdesign

Kapitel 4.1 - Theorie

Abschnitt 1 - Bilddateiformate im Web

Abschnitt 2 - Verlustbehaftete und verlustfreie Komprimierung

Kapitel 4.2 Tutorial Slicen

Abschnitt 1 - Slicen

Kapitel 4.2 Theorie: ImageReady

Abschnitt 1 - Animated GIF

Abschnitt 2 - ImageMaps

Abschnitt 3 - Hover-Schaltfläche

Kapitel 4.2 Praxis: Webdesign

Abschnitt 1 - Slicen und RollOver

Abschnitt 2 - Animated GIF

Abschnitt 3 - ImageMaps

Anhang E

Testbericht

für das Lernmodul "Digitale Bildbearbeitung mit Adobe Photoshop CS2"

<i>lfd.Nr.</i>	<i>Plattform</i>	<i>Browser</i>	<i>Flash Player</i>	<i>Internet- verbindung</i>	<i>Betroffene Kapitel/Seiten</i>	<i>Fehler-Nr.</i>
1	PC / WinXP	Internet Explorer 6.0	Version 9	DSL	3.2.2 Seite 1 3.2.3 Seite 1 3.2.3 Seite 2 4.5.2 Seite 1 4.5.3 Seite 1 4.5.3 Seite 2	Fehler Nr. 5.1 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1 Fehler Nr. 5.1 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1
2	PC / WinXP	Internet Explorer 6.0	Version 8	DSL	Steht nicht mehr zur Verfügung	
3	PC / WinXP	Internet Explorer 6.0	Version 7	DSL	Steht nicht mehr zur Verfügung	
4	PC / WinXP	Mozilla Firefox 2.0	Version 9	DSL	2.2.2 Seite 1 2.2.2 Seite 2 3.2.2 Seite 1 3.2.3 Seite 1 3.2.3 Seite 2 4.5.2 Seite 1 4.5.3 Seite 1 4.5.3 Seite 2	Fehler Nr. 4 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1 Fehler Nr. 5.1 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1 Fehler Nr. 5.1 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1
5	PC / WinXP	Mozilla Firefox 2.0	Version 8	DSL	2.2.2 Seite 1 2.2.2 Seite 2 3.2.2 Seite 1 3.2.3 Seite 1 3.2.3 Seite 2 4.5.2 Seite 1 4.5.3 Seite 1 4.5.3 Seite 2	Fehler Nr. 4 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1 Fehler Nr. 5.1 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1 Fehler Nr. 5.1 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1
6	PC / WinXP	Mozilla Firefox 2.0	Version 7	DSL	Generell bei Benutzen der ersten Navigation 1.2/1.3/1.5/4.5 1.2.2/1.2.3 1.6.1/2.2.1/3.2.1 von 1.6.1 nach 2	Fehler Nr. 4 Fehler Nr. 6 Fehler Nr. 7 Fehler Nr. 8 Fehler Nr. 3 6x Fehler Nr. 7
7	PC / WinXP	Internet Explorer 7.0	Version 9	DSL		Fehler Nr. 4
8	PC / WinXP	Internet Explorer 7.0	Version 8	DSL	Steht nicht mehr zur Verfügung	
9	PC / WinXP	Internet Explorer 7.0	Version 7	DSL	Steht nicht mehr zur Verfügung	
10	PC / WinXP	Mozilla Firefox 1.5	Version 9	DSL	1.2.1	Fehler Nr. 4
11	PC / WinXP	Mozilla Firefox	Version 8	DSL		Fehler Nr. 4

		1.5				
12	PC / WinXP	Mozilla Firefox 1.5	Version 7	DSL	1.2, 1.4, 2.2 1.6.1	Fehler Nr. 4 Fehler Nr. 6 Fehler Nr. 7 Fehler Nr. 8 Fehler Nr. 3
13	PC/Win2000	Firefox 1.0	Version 9	10,0 MBit/s (Standleitung)	Steht nicht mehr zur Verfügung	
14	PC/Win2000	Firefox 1.0	Version 7	10,0 MBit/s (Standleitung)	Steht nicht mehr zur Verfügung	
15	PC/Win2000	Firefox 2.0	Version 9	10,0 MBit/s (Standleitung)	2.2.2 Seite 1 2.2.2 Seite 2	Fehler Nr. 4 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1
16	PC/Win2000	Firefox 2.0	Version 8	10,0 MBit/s (Standleitung)	3.2.2 Seite 1 3.2.2 Seite 2 3.2.3 Seite 2 4.5.2 Seite 1 4.5.3 Seite 1 4.5.3 Seite 2	Fehler Nr. 4 Fehler Nr. 5.1 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1 Fehler Nr. 5.1 Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1
17	PC/Win2000	Firefox 2.0	Version 7	10,0 MBit/s (Standleitung)	Zu 1.2.2) 1.2 (sofort Zu 1.4 (sofort 1.4.2) Zu 2.2 (sofort 2.2.2) Zu 3.2 (sofort 3.2.2) Zu 4.2 (sofort 4.2.2) Schließen d. Anwendung 1.2.1 alle anderen Tests	Fehler Nr. 4 Fehler Nr. 6 Fehler Nr. 7 Fehler Nr. 8 Fehler Nr. 8 Fehler Nr. 8 Fehler Nr. 8 Fehler Nr. 7
18	PC/Win2000	IE 6.0	Version 9	10,0 MBit/s (Standleitung)	1.6.1 Seite 2 1.6.1 Seite 2 2.2.2 Seite 1 2.2.2 Seite 2 3.2.2 Seite 1 3.2.3 Seite 1 Insgesamt sehr instabil!	Fehler Nr. 9 Fehler Nr. 9+Absturz IE Absturz IE b. Klick auf PDF-Symbol Fehler Nr. 5.1/5.2/5.1 Fehler Nr. 5.2/5.1 Fehler Nr. 5.1 Fehler Nr. 9 + Absturz IE
19	PC/Win2000	IE 6.0	Version 8	10,0 MBit/s (Standleitung)	Steht nicht mehr zur Verfügung	
20	PC/Win2000	IE 6.0	Version 7	10,0 MBit/s (Standleitung)	Steht nicht mehr zur Verfügung	
21	MAC / OS X 10.4	Safari 2.0	Version 9			Fehler Nr. 10
22	MAC / OS X 10.4	Firefox 2.0	Version 9			Fehler Nr. 4
23	PC/Win2000	Opera		10,0 MBit/s (Standleitung)		Fehler Nr. 10

Fehler Nr. 3:

Bei Klick auf die zip-Symbole öffnet sich die Fehlermeldung: Incorrect Data Type. Danach wird der File-Dialog korrekt angezeigt.

Fehler Nr. 4:

Das letzte Kapitel wird nicht als abgeschlossen markiert, wenn man auf der letzten Seite der Anwendung angekommen ist. Erst nach Betätigen des Zurückblättern-Buttons der 1. Navigation wird dieser Abschlussstatus erreicht.

Fehler Nr. 5.1:

Fehlermeldung: Not initialized

Fehler Nr. 5.2:

Fehlermeldung: LMSGetValue(cmi.core.lesson_status)failed.

Fehler Nr. 6:

Für die einzelnen Kapitel und Unterkapitel wird generell kein Abschlussstatus erreicht.

Fehler Nr. 7:

Fehlermeldung: General Exception wird teilweise mehrmals hintereinander angezeigt (in Kombination mit der Fehlermeldung: Incorrect Data Type), immer wenn die erste Navigation oder das Menü benutzt wird, bei Benutzung der 2.+3. Navigation trat dieser Fehler nicht auf

Fehler Nr. 8:

Die Navigation funktioniert nicht korrekt: Seiten werden übersprungen, z.B. bei Aufruf von 1.2 über das Menü wird sofort 1.2.4 angezeigt!

Fehler Nr. 9:

Die Flashdatei wird nicht geladen: Film bleibt bei *100% Lade Seite* stecken.

Fehler Nr. 10:

Das Programm konnte nicht geöffnet werden. Folgende Fehlermeldung wurde angezeigt: Der SCORM-Player benötigt mindestens die Browser IE 6.0, Mozilla 1.3 oder Firefox.

Was noch aufgefallen ist:

Ein Kapitel wird auch als abgeschlossen markiert, wenn Seiten über die 3. Navigation nicht aufgerufen wurden. Kapitel werden ebenso für beendet erklärt, wenn man den Test nur aufgerufen, aber nicht bearbeitet hat.

Die Symbole für zip, pdf usw. sind nicht durchgängig mit einem aktiven Bereich versehen, d.h. der Mauszeiger ändert sich nur, wenn er sich genau über einer gefüllten Stelle befindet. In den Bereichen ohne Zeichnung findet dieser Zeigerwechsel nicht statt.

Selbständigkeitserklärung

Ich versichere, dass ich die Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ferch, den 4. Juli 2007

Jana Krüger