

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

Diplomarbeit
im Studiengang Medieninformatik

Thema: **Konzeption und Entwicklung eines Lernmoduls für das Bildungsportal Sachsen gemäß des E-Learning Standards SCORM 1.2**

Eingereicht von: Kerstin Wagner

Eingereicht am: 07.10.2005

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

Inhaltsverzeichnis

Einleitung	5
1 Lernplattformen und E-Learning Standards	7
1.1 Begriffserklärung E-Learning	7
1.2 Webbasierte Lernplattformen	9
1.2.1 Funktionen und Funktionsweise webbasierter Lernplattformen	9
1.2.2 Learning Management Systeme in Abgrenzung zu Autorenwerkzeugen, Content Management Systemen und Learning Content Management Systemen	13
1.3 E-Learning-Standards	19
1.3.1 Ziele und Nutzen von Standards	19
1.3.2 Standardisierungsprojekte	21
1.4 Das SCORM Referenzmodell	27
1.4.1 Das SCORM Content Aggregation Model	29
1.4.2 Die SCORM Run Time Environment	38
2 Die Lernplattform des Bildungsportals Sachsen	44
2.1 Das Bildungsportal Sachsen	44
2.1.1 Entstehungsgeschichte und Förderung	44
2.1.2 Ziele und Zielgruppen	45
2.1.3 Studienangebot	46
2.2 Die Lernplattform der Firma Saba	48
2.2.1 Bestandteile der Saba Lernplattform	48
2.2.2 Veröffentlichung von Lerninhalten	53
2.2.3 Funktionen im Bereich der Kommunikation	56
2.2.4 Autorenwerkzeuge	58
2.2.5 Unterstützte E-Learning Standards	63

2.3	Die Nutzung der Lernplattform Saba3 Release4 für das Bildungsportal Sachsen	67
2.3.1	Das Internetportal	67
2.3.2	Die Implementierung der Lernplattform Saba3 Release4 für das Bildungsportal	69
2.3.3	Kommunikations- und Kollaborationsfunktionen des Systems ACollab	72
3	Konzeption von Lerninhalten für das Bildungsportal Sachsen am Beispiel des Lernmoduls ‚Typografie praktisch!‘	76
3.1	Vorgehensweise bei der Entwicklung von Lernmodulen	76
3.2	Vorgaben	78
3.2.1	Zielgruppe und Einsatzbereich	78
3.2.2	Lernziele	79
3.2.3	Technische Vorgaben	80
3.2.4	Vorgaben zum visuellen Design	82
3.2.5	Rechtliche Aspekte	83
3.3	Didaktische Konzeption	84
3.3.1	Lehrstrategie	85
3.3.2	Einteilung der Lerninhalte in Lernobjekte	87
3.3.3	Navigation zwischen den Lernobjekten	88
3.3.4	Interaktionsmöglichkeiten	89
3.3.5	Medieneinsatz	90
3.3.6	Drehbuch	94
3.4	Gestaltung und Evaluation der Benutzeroberfläche	95
3.4.1	Funktionelle Gestaltung	96
3.4.2	Screendesign	101
3.4.3	Nutzertest	104
3.4.4	Usability Evaluation	106

4	Technische Entwicklung des Lernmoduls ‚Typografie praktisch!‘	111
4.1	Erstellung der Lernobjekte	111
4.1.1	Erstellung der Lernobjekte mit dem Programm Flash MX	111
4.1.2	Erstellung von integrierten Medien	132
4.2	Umsetzung von SCORM 1.2 am Lernmodul	137
4.2.1	Kommunikation zwischen Lernobjekten und Learning Management System	137
4.2.2	Metadatenerstellung	144
4.2.3	Content Packaging	147
4.3	Einstellen und Veröffentlichen des SCORM Inhalts im System des Bildungsportals Sachsen	150
4.3.1	Trainingssystem	150
4.3.2	Technische Tests	155
	Anhang A – Inhaltliche Gliederung des Lernmoduls	158
	Anhang B – Nutzertest	161
	Anhang C – Kriterienkatalog zur Usability Evaluation	168
	Anhang D – Ergebnisse der Usability Evaluation	171
	Abkürzungsverzeichnis	175
	Glossar	176
	Abbildungsverzeichnis	182
	Tabellenverzeichnis	185
	Verzeichnis der Codebeispiele	186
	Literaturverzeichnis	187
	Selbständigkeitserklärung	192

Einleitung

Der rapide Anstieg der Internetnutzung seit Mitte der neunziger Jahre geht mit einer enormen Entwicklung im Bereich der neuen Informations- und Kommunikationstechnologien einher. Dies führte vor allem im Bereich E-Business dazu, dass bis zur Krise der New Economy im Jahr 2001 häufig überzogene Entwicklungserwartungen formuliert wurden.

Aber auch im Bereich der Bildung verband man oft sehr hohe Erwartungen mit den neuen Technologien. E-Learning, so glaubte man, könne schon bald Massenveranstaltungen an Hochschulen durch selbstbestimmtes und selbst-motiviertes virtuelles Lernen ersetzen. Dadurch erhoffte man sich auch finanzielle Einsparungen und Gewinne. In Deutschland werden daher seit den neunziger Jahren von Bund und Ländern zahlreiche Projekte gefördert, deren Aufgabe darin besteht, den Einsatz von neuen Medien im Bereich der Hochschullehre auszubauen.

Eines dieser Projekte ist das Bildungsportal Sachsen, welches als Internetportal der sächsischen Hochschulen zur mediengestützten Aus- und Weiterbildung dient. Die Erwartungen und die Zielsetzungen bezüglich E-Learning an Hochschulen haben sich mittlerweile jedoch relativiert. Man geht zwar davon aus, dass virtuelle Lehr- und Lernangebote an Hochschulen eine zentrale Position einnehmen werden, jedoch kein Ersatz für die traditionelle Präsenzlehre, sondern eine sinnvolle Ergänzung sein werden [Bildungsportal 2003, S. 4]. Die Lernmodule, die im Rahmen des Verbundprojektes Bildungsportal Sachsen gefördert und erstellt werden, stehen den Studierenden der Hochschulen des Freistaates Sachsen über das Internetportal des Bildungsportals zur Verfügung und werden in der Hochschullehre eingesetzt.

Diese Arbeit beschreibt die Konzeption und die Erstellung des Lernmoduls ‚Typografie praktisch!‘, welches an der Hochschule für Technik und Wirtschaft Dresden (HTW Dresden) im Rahmen Projektes Bildungsportal Sachsen entwickelt wurde. Das Modul soll in der Lehrveranstaltung ‚Typographie‘ im Studiengang Medieninformatik begleitend zur Präsenzveranstaltung zum Einsatz kommen.

Die ersten beiden Kapitel dieser Arbeit beschreiben Wissensgrundlagen, welche für die Entwicklung des Lernmoduls nach dem E-Learning Standard SCORM und dessen Integration in die Lernplattform des Bildungsportal Sachsens von Bedeutung sind. Das erste Kapitel beschäftigt sich dabei mit den Funktionen von Lernplattformen und E-Learning Standards sowie mit dem

SCORM Referenzmodell. Das zweite Kapitel beschreibt zunächst das Bildungsportal Sachsen und zeigt dann, wie das Bildungsportal als Lernplattform funktioniert.

Die folgenden Kapitel beziehen sich auf die eigentliche Entwicklung des Lernmoduls. Das dritte Kapitel beschreibt hierbei zunächst die konzeptionelle Arbeit. Diese reicht von der Einteilung des Lerninhalts in geeignete Lerneinheiten über die Festlegung der Navigation und des Screendesigns bis hin zur Überprüfung der Tauglichkeit des Designs durch Nutzertest und Usability Evaluation.

Das vierte Kapitel beschäftigt sich mit der technische Umsetzung des Lernmoduls. Dabei wird neben der Entwicklung der Lerninhalte auch die Anpassung des Lernmoduls an den E-Learning Standard SCORM 1.2 und die Integration in das Bildungsportal Sachsen beschrieben.

Am Ende dieser Arbeit befindet sich neben mehreren Anhängen und Verzeichnissen auch ein Glossar, worin Fachbegriffe und Fremdwörter, die in dieser Arbeit in Tabellen, Grafiken oder auch im Text vorkommen, erläutert werden.

Zudem beinhaltet diese Arbeit einen separaten Anhang, der das Drehbuch des Lernmoduls ‚Typografie praktisch!‘ enthält, sowie eine beigelegte CD mit dem Lernmodul.

Nun noch eine Bemerkung zu der verwendeten Schreibweise. In dieser Arbeit wird durchgehend die männliche Form wie z.B. Lehrer oder Lernender verwendet. Dies dient alleine dazu, das Lesen zu vereinfachen. Die weibliche Form ist in dieser Arbeit grundsätzlich in die männliche einbezogen.

1 Lernplattformen und E-Learning Standards

1.1 Begriffserklärung E-Learning

Electronic Learning, kurz E-Learning¹, steht genau genommen für die gesamte Bandbreite elektronisch unterstützten Lernens. Dazu wäre die computergestützte Offline-Lehre und die internetgestützte Lehre genauso zu rechnen wären wie Telelearning oder das Lernen mit Hilfe von Videobändern. Heutzutage versteht man unter dem Begriff E-Learning jedoch meist das Lernen mit digital gespeicherten Lerninhalten und softwareunterstützten Lernumgebungen. Dabei gehen die genauen Definitionen in der Fachliteratur auseinander. Manche Autoren bezeichnen als E-Learning nur das internetgestützte Lernen, andere schließen das Lernen am Computer mit lokal installierter Software mit ein. Die Autoren des Buches *Auswahl von Lernplattformen* bezeichnen E-Learning „als einen übergeordneten Begriff für softwareunterstütztes Lernen“ und beziehen somit auch mobile Endgeräte, wie z.B. Handys, mit ein [Baumgartner u.a. 2002, S. 15].

Da das Bildungsportal Sachsen ein Internetportal ist, also seine Lerninhalte über das Internet genutzt werden, wird in dieser Arbeit auf die Definition von E-Learning als internetgestütztes Lernen zurückgegriffen.

Bei dieser Definition von E-Learning lassen sich Eigenschaften, die man dem Internet zuschreibt, prinzipiell auch dem Lernen übers Internet zuordnen. Die *Freiheit*, die der Nutzung des Internets zugeschrieben wird, kann man in diesem Zusammenhang mit dem Begriff *Flexible Learning* gleichsetzen, was soviel wie die Unabhängigkeit von Zeit, Ort und auch der Person bedeutet. Dabei ist jedoch offensichtlich, dass E-Learning immer noch in gewissem Maße von diesen Faktoren abhängig ist. So macht z.B. eine völlige Unabhängigkeit von *Zeit* keinen Sinn, wenn eine E-Learning Anwendung Kommunikationsformen wie Chats, Audio- oder Videokonferenzen nutzt. Der *Ort* des Lernens ist dann von Bedeutung, wenn dieser Auswirkungen auf den Zugang zum Internet und die Datenübertragungsrate hat. Auch die Hardware- und Software-Ausstattung des Endgerätes ist von Bedeutung. Entsprechen die Zugangsvoraussetzungen des Gerätes nicht den Anforderungen der E-Learning-Anwendung, führt dies beim Lernenden wohl eher zur Frustration als zur Motivation. Die Unabhängigkeit der *Person* wird insofern einge-

1. Es sind verschiedene Schreibweisen verbreitet, z.B. auch e-Learning, eLearning, elearning, eLearn.

schränkt, als dass nicht jeder Lernstoff für jede Person geeignet ist. Hier spielen sowohl personenbezogene Faktoren, wie z.B. Alter oder Wissensstand, eine Rolle als auch Einschränkungen durch den Anbieter, z.B. durch kostenpflichtige Angebote oder Angebote für Betriebs- oder Hochschulangehörige.

Von diesen Einschränkungen abgesehen, handelt es sich bei E-Learning dennoch um eine flexible Lernform, welche sich durch besondere Interaktivitätsmöglichkeiten auszeichnet. *Interaktivität* lässt sich in diesem Zusammenhang in zwei Kategorien einteilen. Eine *steuernde* Interaktivität ermöglicht dem Nutzer den Programmablauf selbst zu steuern, z.B. durch Aufrufen einer neuen Seite. *Didaktische* Interaktivität nimmt hingegen direkten Einfluss auf das Lernerlebnis. Dies geschieht durch Feedback an den Lernenden, was sowohl programmgesteuert als auch durch Kommunikation mit anderen Lernenden oder dem Lehrer geschehen kann. [Baumgartner u.a. 2002, S. 15ff]

Weitere Eigenschaften, die mit E-Learning häufig in Zusammenhang gebracht werden sind Multimedialität und Virtualität. Von *Multimedialität* wird meist dann gesprochen, wenn eine Anwendung aus mehreren digitalen Medien besteht, wovon mindestens ein Medium zeitabhängig (Sound, Animation, Video) ist. Mit dem Einsatz verschiedener Medien wird häufig versucht, *Virtualität* zu erzeugen. Dies stellt den Versuch dar, künstlich erzeugte Lernumgebungen z.B. durch Simulationen, Animationen, Interaktionen und Sound so realistisch wie möglich zu gestalten.

Im Vergleich zur computergestützten Offline-Lehre wird dem E-Learning aufgrund der oben genannten Eigenschaften ein didaktischer Mehrwert zugeschrieben. Dabei grenzen sich internetgestützte Lernumgebungen von computergestützten Offline-Lernumgebungen vor allem durch ihre Kommunikationsmöglichkeiten ab, die tutorielle Betreuung oder auch ein teamorientiertes Lernen ermöglichen [Kleinmann/Wannemacher 2004 S. 3]. In diesem Zusammenhang ist auch das so genannte *hybride Lernen* oder auch *blended Learning* zu erwähnen. Dabei handelt es sich um eine Lern- bzw. Lehrform, die eine Kombination aus der traditionellen Präsenzlehre und dem internet- bzw. computergestützten Lernen darstellt. Der soziale Aspekt des Lernens in der Gruppe und gesprächsbasierende Stoffvermittlungsformen bleiben somit erhalten.

1.2 Webbasierte Lernplattformen

Um E-Learning in seiner gesamten Bandbreite – von der Steuerung des Lernprozesses bis hin zu der Bereitstellung verschiedener Kommunikationsformen – ermöglichen zu können, bedarf es einer Reihe an Instrumenten (Software). Neben *administrative Aufgaben* zur Verwaltung der Lernenden und der Lehrenden müssen *Lernangebote erstellt, veröffentlicht und koordiniert* werden. Zudem sind *Kommunikationsangebote* zur Verfügung zu stellen.

Zur Erfüllung dieser Aufgaben haben sich verschiedene Softwarebereiche entwickelt, wobei die Grenzen fließend sind. In dieser Arbeit wird der Bereich der *webbasierten Lernplattformen* behandelt. Diese stellen das Grundgerüst für die Präsentation der Lerninhalte und die Organisation von Lernprozessen über das Internet dar. Zusätzlich übernehmen sie auch häufig viele der oben genannten Aufgabenbereiche.

„Unter einer webbasierten Lernplattform ist eine serverseitig installierte Software zu verstehen, die beliebige Lerninhalte über das Internet zu vermitteln hilft und die Organisation der dabei notwendigen Lernprozesse unterstützt.“ [Baumgartner u.a. 2002, S. 24]

Werkzeuge, mit denen man lediglich Lerninhalte präsentieren, den Lernprozess aber nicht organisieren kann, sind demnach keine Lernplattformen. Zum besseren Verständnis werden in dem folgenden Abschnitt einige Funktionsbereiche sowie die Arbeitsweise webbasierter Lernplattformen vorgestellt. Dabei wird der Begriff *Lernmanagementsystem* bzw. *Learning Management System* (LMS) als ein Synonym für Lernplattform verwendet.

1.2.1 Funktionen und Funktionsweise webbasierter Lernplattformen

Bei der Umsetzung ihrer Funktionen greifen Lernplattformen auf verschiedenste *Basistechnologien* zurück. Diese stellen eigenständig nutzbare Applikationen dar, welche zwar im Bereich E-Learning eingesetzt werden können, deren Funktionalitäten aber nicht speziell auf diesen Bereich ausgerichtet sind. Beispiele dafür sind E-Mail, Chat oder auch die Verwaltung von Datenbanken.

Typisch für Lernplattformen ist die Nutzung von Basistechnologien aus den Bereichen *Kommunikation, Informationsbeschaffung, Administration, Produktion, Evaluation* und *Hardware/Systemsoftware* [Niegemann u.a. 2004, S. 248].

Tabelle 1.1 gibt einen Überblick über diese Einteilung und ordnet den jeweiligen Bereichen Technologien zu, die häufig im Bereich E-Learning zum Einsatz kommen.

Tab. 1.1: Beispiele für Basistechnologien, nach [Niegemann u.a. 2004, S. 249]

Kommunikation (synchron) <ul style="list-style-type: none"> ▪ Chat, Instant Messaging ▪ Audio-/Video-Konferenzen ▪ Whiteboarding ▪ Application-Sharing ▪ Web-Safari ▪ Online-Voting 	Kommunikation (asynchron) <ul style="list-style-type: none"> ▪ E-Mail ▪ Foren, Newsgroups ▪ Blackboard ▪ Newsletter ▪ Elektronische Agenda, Group Calendaring
Informationsbeschaffung <ul style="list-style-type: none"> ▪ Suchmaschinen ▪ Information Retrieval 	Administration <ul style="list-style-type: none"> ▪ Zugriffs- und Benutzerverwaltung
Produktion <ul style="list-style-type: none"> ▪ Multimedia- Authoring ▪ Entwicklungsumgebungen ▪ (Multimedia-)Standards 	Evaluation <ul style="list-style-type: none"> ▪ Logfile-Analyse ▪ Online-Befragung
Hardware/Systemsoftware <ul style="list-style-type: none"> ▪ (Produktions-)Server, Streaming-Server, Webserver ▪ Internetzugang ▪ Sicherheitstechnologien, Verschlüsselung, Logging ▪ Datenbanken 	

Im E-Learning Kontext dienen Basistechnologien als Grundlage für so genannte *Lerntechnologien*. Darunter versteht man Applikationen, deren Funktionalitäten sich am Lernprozess orientieren, und die als Bausteine für komplexe E-Learning Systeme eingesetzt werden können.

Ausschlaggebend bei der Definition von Lerntechnologien ist der Anwendungskontext. So gelten Basistechnologien mit einem konkreten Anwendungskontext im Bereich E-Learning als Lerntechnologien. Durch den speziellen Anwendungskontext ergeben sich häufig neue, spezifische Funktionalitäten. Als Beispiel können die Basistechnologien für synchrone Kommunikation im Kontext E-Learning als Lerntechnologien für einen ‚virtuellen Klassenraum‘ zum Einsatz kommen. [Niegemann u.a. 2004, S. 249f]

Die Abbildung 1.1 stellt einige Einsatzbereiche von Lerntechnologien vor. Diese beziehen sich auf den Lernkontext, der hier durch Daten über die Lernenden (Personendaten), die Lerninhalte (Wissens-/Inhalte-Pool) und Daten über den Lernprozess der einzelnen Lernenden (Prozessdaten) repräsentiert wird.

Zudem verdeutlicht die Abbildung den Zusammenhang zwischen Basistechnologien, Lerntechnologien und E-Learning Systemen (Lerntechnologien basieren auf Basistechnologien und sind die Bausteine für E-Learning Systeme).

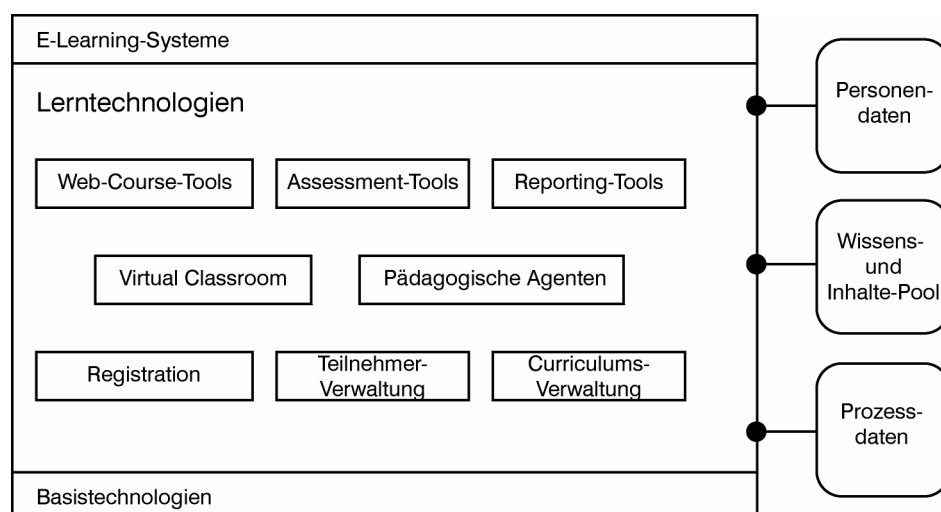


Abb. 1.1:
Beispiele für Lern-
technologien,
nach [Niegemann u.a.
2004, S. 251]

Lerntechnologien lassen sich hinsichtlich ihres Einsatzes in E-Learning Systemen bzw. Lernplattformen verschiedenen Funktionsbereichen zuordnen. Die fünf hauptsächlichsten Funktionsbereiche von Lernplattformen sind [Baumgartner u.a. 2002, S. 26]:

- Präsentation von Lerninhalten (Text, Bild, Grafik, Ton, Animation etc.)
- Kommunikationswerkzeuge (für asynchrone Kommunikation, wie z.B. Mail, Webforen, und synchrone Kommunikation, wie z.B. Chat)
- Werkzeuge zur Erstellung von Aufgaben und Übungen
- Evaluations- und Bewertungshilfen
- Administration (von Lernenden, Inhalten, Kursen, etc.)

Nach dieser Einteilung (vgl. Abb. 1.2) werden beispielsweise die Lerntechnologien Assessment- und Reporting-Tools den Evaluations- und Bewertungshilfen zugeordnet. Registration, Teilnehmer- und Curriculums-Verwaltung dienen hingegen der Administration.

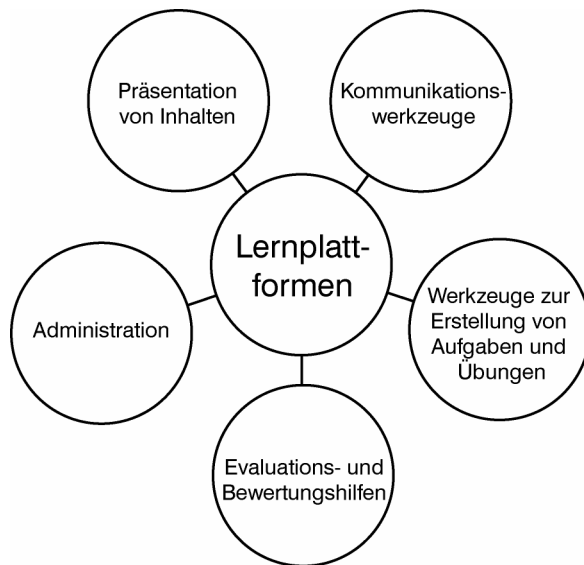


Abb. 1.2:
Die fünf hauptsächlichen Funktionsbereiche von Lernplattformen, nach [Baumgartner u.a. 2002, S. 27]

Abbildung 1.3 verdeutlicht das Zusammenspiel verschiedener Funktionsbereiche in einem Learning Management System. Selbst erstellter oder kommerziell zugekaufter Lerninhalt (Online-Kurs) wird in der Datenbank des LMS verwaltet und den Lernenden zur Verfügung gestellt. Dabei stellen Online-Kurse die kleinsten wiederverwendbaren Lerneinheiten dar.

In einem LMS kann der individuelle Lernprozess kann vom System mitverfolgt und gespeichert werden. In vielen Fällen stellt das System auch synchrone und asynchrone Kommunikationstools zum Kommunizieren bzw. Kollaborieren zur Verfügung. [Baumgartner u.a. 2002, S. 30]

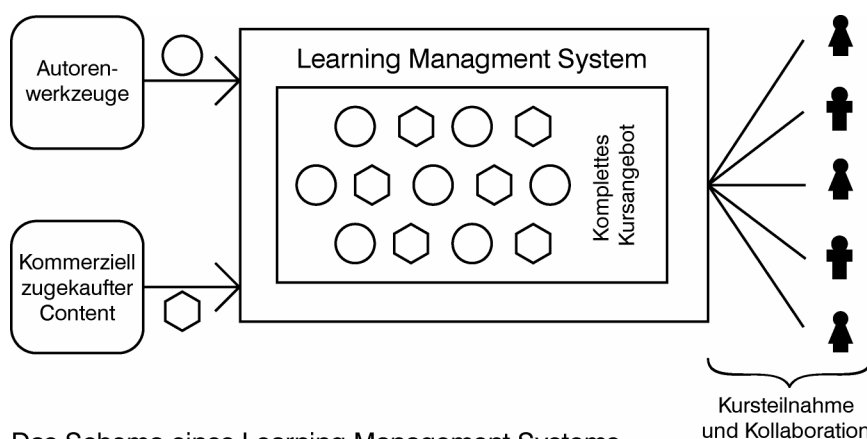


Abb. 1.3:
Learning Management System, nach [Baumgartner u.a. 2002, S. 30]

Das Schema eines Learning Management Systems (die Symbole ○ und ⬡ stehen für Online-Kurse [=Content])

1.2.2 Learning Management Systeme in Abgrenzung zu Autorenwerkzeugen, Content Management Systemen und Learning Content Management Systemen

Aus dem vorausgegangenen Abschnitt geht hervor, dass Learning Management Systeme zwar für die Präsentation von Lerninhalten zuständig sind, die Erstellung dieser Inhalte wird aber nicht zu den eigentlichen Funktionen eines Lernplattformen gerechnet. Zwar haben viele Anbieter von LMS so genannte *Autorenwerkzeuge* (Entwicklungswerkzeuge für interaktive Anwendungen) in ihr Produkt integriert, diese sind aber häufig extern zugekauft oder nur rudimentär implementiert [Baumgartner u.a. 2002, S. 32].

Als Entwickler von Lerninhalten kann es daher sinnvoll sein, auf eines der zahlreichen externen Autorenwerkzeuge zurückzugreifen. Dabei hat man die Möglichkeit zwischen Werkzeugen mit relativ hohem bis hin zu sehr geringem Einarbeitungsaufwand zu wählen.

Autorenwerkzeuge zur Erstellung von Online-Kursen lassen sich folgendermaßen unterteilen [Baumgartner u.a. 2002, S. 32f]:

- Professionelle Autorenwerkzeuge: Darunter sind Werkzeuge mit integrierter Programmiersprache und hohem Einarbeitungsaufwand gemeint. Sie besitzen den Vorteil, nicht auf wenige, vorgegebene Funktionen eingeschränkt zu sein, und sind daher sehr flexibel einsetzbar.
- WYSIWYG-HTML-Editoren: Die Bezeichnung ‚What You See Is What You Get‘ beruht darauf, dass die Autoren über eine grafischen Schnittstelle ähnlich wie in einem Textverarbeitungsprogramm arbeiten können. Der HTML-Code wird automatisch über das Programm erzeugt. Zusätzliche Plug-Ins bieten Funktionen speziell für die Erstellung von E-Lerninhalten.
- Rapid Content Development Tools: Dabei sind so genannte ‚Autorenwerkzeuge der neuen Generation‘ zu verstehen, die ohne großen Einarbeitungsaufwand schnell zu ansehnlichen Ergebnissen führen.

Die Abbildung 1.4 stellt Beispiele zu den jeweiligen Gruppen vor und zeigt den Einarbeitungsaufwand.

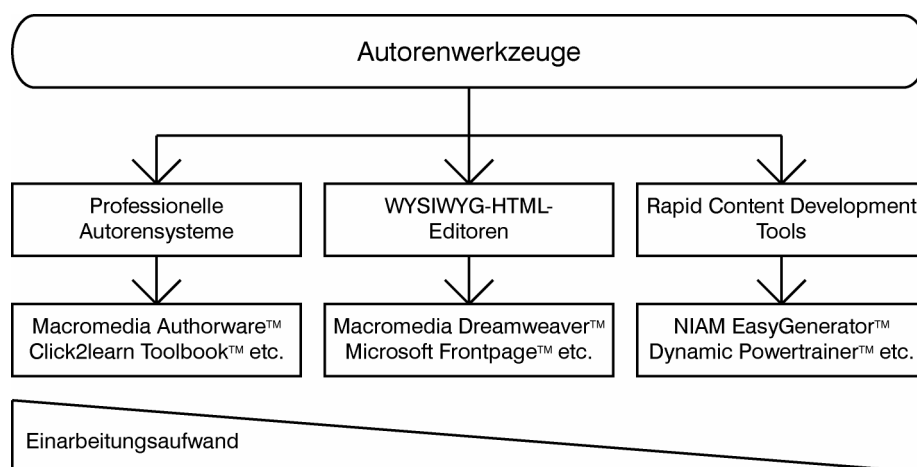


Abb. 1.4:
Die Familie der
E-Learning-Content
Autorenwerkzeuge,
nach [Baumgartner u.a.
2002, S. 34]

Eine weitere Produktgruppe, die im Zusammenhang mit E-Learning genannt werden sollte, ist die der *Content Management Systeme* (CMS). Unter dem englischen Begriff *Content Management* versteht man *Verwaltung von Inhalten*. Content Management Systeme stellen demnach genau genommen *Softwarewerkzeuge zur Verwaltung von Inhalten* dar. Je nach Einsatzbereich verfügen Content Management Systeme neben der Verwaltung von Inhalten in der Regel über weitere Funktionsbereiche. Typisch dafür sind [Strähle 2004, S. 32ff]:

- Erstellung von Inhalten: z.B. Erstellung verschiedener Medientypen
- Aufbereitung von Inhalten: Zusammenfassung verschiedener Inhaltsobjekte zu größeren Einheiten wie z.B. Seite, Kapitel oder Kurs
- Publikation und Präsentation von Inhalten: beinhaltet sowohl die Freigabe des Inhalt (z.B. durch Redakteure) als auch die Datenvorbereitung für die Druckvorstufe bzw. für die Übertragung von Webseiten auf einen Server; die eigentliche Präsentation findet in der Regel außerhalb des CMS statt, z.B. über einen Webbrowser bzw. über Papier bei gedruckten Inhalten
- Wiederverwendung von Inhalten: bedeutet einerseits die *dokumentenübergreifende* Wiederverwendung von Inhalten (z.B. bei der Nutzung einer Grafik für mehrere Dokumente) und andererseits die *zeitliche* Wiederverwendung von Inhalten (z.B. beim Einsatz einer ‚alten‘ Grafik in einem neuen Kontext)

- Aktualisierung von Inhalten: z.B. zur zeitlichen Wiederverwendung von Inhalten; beinhaltet häufig Versionierungsfunktionen zur Speicherung von älteren Fassungen

Etabliert hat sich der Begriff des Content Management Systems erst richtig durch den Einsatz solcher Systeme für Online-Inhalte [Strähle 2004, S. 32]. In diesem Zusammenhang werden so genannte *Web Content Management Systeme* (WCMS) in der Regel für Webseiten mit hohem Aktualitätsgrad verwendet, wie dies z.B. bei Online-Zeitungen der Fall ist. Das Management System dient dabei der prozessorientierten Publikation von Online-Inhalten und zeichnet sich durch folgende Merkmale aus (vgl. Abb. 1.5) [Baumgartner u.a. 2002, S. 34f]:

- Strikte Trennung von Inhalt und Layout: Inhalte (Texte, Bilder, Videoclips, etc.) werden von den Formatvorlagen (Templates) getrennt gespeichert. Beim Aufruf einer Webseite wird diese dynamisch erzeugt.
- Komponenten-Management: Die einzelnen Inhalte besitzen beschreibende Metadaten und sind in einer Komponenten-Datenbank (content component database) abgelegt. Aus dieser Datenbank kann auf die einzelnen Komponenten (Texte, Bilder, etc.) zugegriffen werden und diese zu gesamten Dokumenten oder Kursen zusammensetzen und publizieren.
- Workflow-Management: Darunter sind Mechanismen zu verstehen, die eine Definition und Kontrolle des Workflows (Ablauf der Arbeitsschritte) ermöglichen. So steht z.B. zu Beginn eines Workflows die Erstellung eines Dokuments. Im zweiten Schritt muss das Dokument vom Chefredakteur freigegeben werden. Danach wird es online gestellt und nach einer gewissen Zeit archiviert.

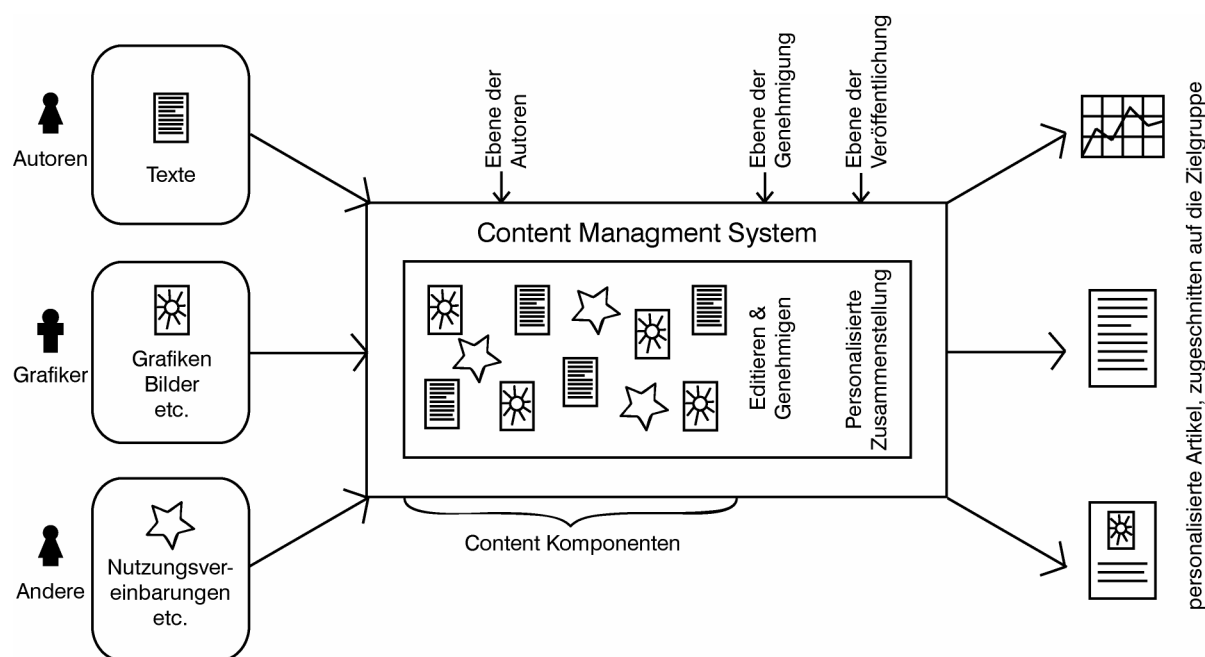


Abb. 1.5: Content Management System, nach [Baumgartner u.a. 2002, S. 35]

Auch im Bereich E-Learning werden Content Management Systeme eingesetzt. Man spricht dann auch von so genannten Learning Content Management System (LCMS). Diese kombinieren typischen Funktionen von Content Management Systemen und Learning Management Systemen [Baumgartner u.a. 2002, S. 43].

Die Funktionsweise von Content Management Systeme für virtuelles Lernen orientiert sich in der Praxis häufig an der von Web Content Management Systemen. Dabei erscheint besonders der Aspekt des Komponenten-Management für den Einsatz im E-Learning Bereich als sehr nützlich. Die Beschreibung durch Metadaten und die Zugriffsmöglichkeit auf einzelne Komponenten ermöglichen die Wiederverwendbarkeit von Inhalten auf der Ebene der einzelnen Content-Komponenten (Bilder, Animationen, etc.). Der Vorteil einer Wiederverwendbarkeit wird dann besonders deutlich, wenn es sich um aufwendig erstellte Animationen oder Videoclips handelt. Diese werden somit nicht mehr ausschließlich für einen einzelnen Online-Kurs produziert, sondern können unter Umständen in mehreren Kursen genutzt werden. Abbildung 1.5 veranschaulicht die Verwendung von Content-Komponenten in Zusammenhang mit Workflow-Management.

Bei der Verwendung von Content-Komponenten im E-Learning Kontext spricht man von Lernobjekten bzw. *wiederverwendbaren Lernobjekten*. Lernobjekte (Learning Objects bzw.

LOs) stellen die kleinste sinnvolle Lerneinheit dar, in die ein Online-Kurs zerlegt werden kann. Dabei kann es sich sowohl um einzelne Komponente als auch um eine Kombination verschiedener Komponenten handeln. So kann eine einzelne Grafik oder eine Animation ebenso wie ein Test zur Lernerfolgskontrolle ein LO darstellen.

Von wiederverwendbaren Lernobjekten (Reusable Learning Objects bzw. RLOs) spricht man, wenn ein Lernobjekt mit Metadaten versehen wird und zu größeren Kurseinheiten zusammengesetzt werden kann. [Baumgartner u.a. 2002, S. 42]

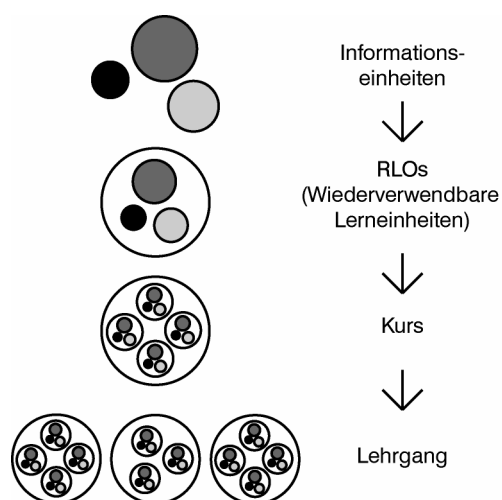


Abb. 1.6:
Hierarchie der RLOs,
nach [Baumgartner u.a. 2002, S. 42]

Learning Content Management System ermöglichen die Erstellung, Speicherung und Verwaltung von RLOs, zudem organisieren und betreuen sie webunterstütztes Lernen. Der Vorteil von Learning Content Management Systemen im Vergleich zu Learning Management Systemen ohne Funktionen für das Content Management liegt somit darin, dass die Entwickler von Online-Kursen Zugriff auf RLOs haben. Somit besteht die Möglichkeit, bereits entwickelte Lerneinheiten bei der Erstellung neuer Kurse wieder zu verwenden. Dies ermöglicht ein effizientes und kostengünstiges Arbeiten.

Die Abbildung 1.7 verdeutlicht die Funktionsweise eines Learning Content Management Systems.

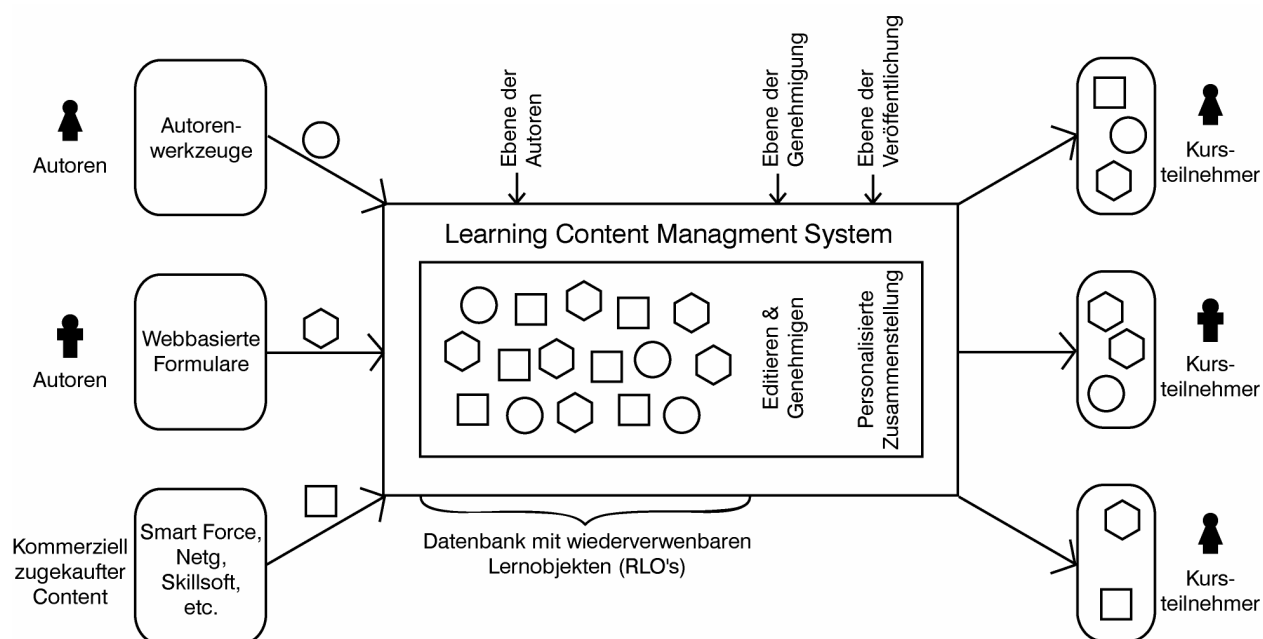


Abb. 1.7: Learning Content Management System, nach [Baumgartner u.a. 2002, S. 44]

Beispiele für LCMS sind *Mentorware* (von Mentorware), *Theorix* (von Theorix), *Top Class* (von WBT Systems) oder auch *Trainersoft Manager* (von Trainersoft). [Baumgartner u.a. 2002, S. 47ff]

Im Bereich der Content Management Systeme besteht eine große Angebotsvielfalt. Recherchen aufgrund einer Evaluation von deutschsprachigen Systemen, die im Auftrag des österreichischen Bundesministeriums für Bildung, Wissenschaft und Kultur (bm:bwk) durchgeführt wurde, ergaben, dass es sich um rund 220 Produkte weltweit handelt. Einen Überblick über deutschsprachige Content Management Systeme wie z.B. *Manila* (UserLand Software, Inc.) oder das Open Source Produkt *Postnuke* (Postnuke) erhält man über die Internetseite [virtual-learning 2004] zur Evaluation.

Verschiedene Webportale bieten einen Einblick in die Funktionsweise von Content Management Systemen und sollen bei der Wahl eines Systems behilflich sein. Nach dem Motto ‚try before you ... install‘ bietet *opensourceCMS* [opensourceCMS] die Möglichkeit, verschiedene Open Source Systeme online zu testen. Das Content Management Portal *contentmanager.de*

[contentmanager] stellt sowohl kostenlose als auch lizenzpflichtige Content Management Systeme zum Testen zur Verfügung. [Strähle 2004, S. 59]

1.3 E-Learning Standards

„Dem gemeinsamen Interesse von Lernenden, Entwicklern und Anbietern für Lehr- und Lernanwendungen nach vielfältiger und erweiterter Anwendbarkeit, einfacher Aktualisierung und geringeren Kosten bei der Produktion kann durch die Einhaltung von Standards begegnet werden.“ [Gries 2003, S. 1]

Dazu werden bei der Entwicklung von E-Learning Standards verschiedene Strategien verfolgt. Diese reichen von der Entwicklung von Metadatenschemas zur einheitlichen Beschreibung von Lernobjekten, um deren Archivierung, Auffindbarkeit und somit die Wiederverwendbarkeit zu erleichtern, bis hin zur Vereinheitlichung der Schnittstellen zwischen Online-Kurs und Lernmanagementsystemen. Wichtige Aspekte hierbei sind die Standardisierung der Kommunikation zwischen Lernplattform und –inhalten sowie der Möglichkeiten für Darstellung und Ablaufsteuerung eines Kurses.

Dabei spielt der wirtschaftliche Aspekt von E-Learning Standards sowohl für Entwickler- und Anbieterseite als auch für Nutzer von Online-Kursen eine bedeutende Rolle. Standards senken das Investitionsrisiko der Entwickler und Anbieter, sichern die Konkurrenzfähigkeit ihrer Produkte und können den Entwicklungsaufwand und somit die Entwicklungskosten senken. Nutzer profitieren von sinkenden Preisen, einer steigenden Angebotsvielfalt. [Wikipedia - Standard].

Verschiedene Ziele und der Nutzen von E-Learning Standards werden im folgenden Abschnitt genauer erläutert.

1.3.1 Ziele und Nutzen von E-Learning Standards

Interoperabilität ist eines der Schlagworte, welche häufig in Zusammenhang mit E-Learning Standards erwähnt werden. Gemeint ist damit die Fähigkeit zur Zusammenarbeit von verschiedenen Systemen oder Techniken [Wikipedia - Interoperabilität]. Standards im Bereich E-Learning ermöglichen die Zusammenarbeit bzw. den Datenaustausch zwischen Lerninhalten

und Lernplattformen durch die Definition einheitlicher Schnittstellen und Datenmodelle. Dadurch kann ein LMS beispielsweise den Lernenden Rückmeldung über ihren Lernfortschritt in einem Online-Kurs geben.

Als ein weiterer Vorteil von Standards gilt die *Portabilität*. Wenn bestimmte E-Learning Standards eingehalten werden, können Lerninhalte auf unterschiedlichen Plattformen laufen, ohne dass sie einer aufwändigen Systemanpassung unterzogen werden müssen.

In diesem Zusammenhang ist auch die bereits erwähnte *Wiederverwendbarkeit* von Lerninhalten (vgl. 1.2.1 u. 1.2.2) zu nennen. Lerninhalte können durch einheitliche Modelle zur Beschreibung leichter wieder verwendet und miteinander kombiniert werden. Somit lassen sich neue Kurse erstellen, ohne neuen Content zukaufen zu müssen.

Außerdem spricht *Beständigkeit* für die Verwendung von Standards. Bei gängigen E-Learning Standards wie z.B. SCORM ist nicht zu befürchten, dass diese in naher Zukunft komplett umgearbeitet werden. Dies gibt beim Kauf von Lernplattformen oder Lerninhalten, die Standards unterstützen, Sicherheit.

Zur funktionellen Kategorisierung (vgl. Abb. 1.8) lassen sich E-Learning Standards den drei Bereichen Lerninhalt, Lernende/Lernprozess und Interoperabilität zuordnen [Epolos o.J.]:

- *Lerninhalt*: Standards für Lerninhalte legen fest, wie Lerninhalte beschrieben (*Metadaten*) und in welcher Form gespeichert bzw. verpackt (*Packaging*) werden müssen, um problemlos vom LMS verarbeitet werden zu können. Mit Informationen über das *Sequenzing* kann das Ablaufverhalten innerhalb eines Lerninhalts definiert werden. Dadurch lässt sich festlegen, in welcher Reihenfolge und nach welchen Kriterien der Lernstoff abgearbeitet werden soll.
- *Lernende/Lernprozess*: Standards aus diesem Funktionsbereich beschreiben, wie allgemeine *Informationen über Lernende* (z.B. Name, Alter, Wissensstand) und Daten zum individuellen Lernprozess (*Leistungen der Lernenden*) gespeichert bzw. vom LMS verarbeitet werden können. Wichtig hierbei ist, dass Standards zum *Datenschutz* eingehalten werden.
- *Interoperabilität*: Standards zur Beschreibung der Interoperabilität legen die Integration der einzelnen Komponenten in einem LMS fest und sind dafür verantwortlich, dass deren Zusammenarbeit funktioniert. Sie betreffen die *Architekturen* von Learning Management Systemen und beschreiben *Protokolle*, die für die Kommunikation zwischen den Komponenten genutzt werden. Eine wichtige Aufgabe bei der Umsetzung von Interoperabilität übernehmen die *APIs* (Application Programming Interfaces), indem sie verschiedene Funktionen als standardisierte Schnittstellen zur Verfügung stellen.

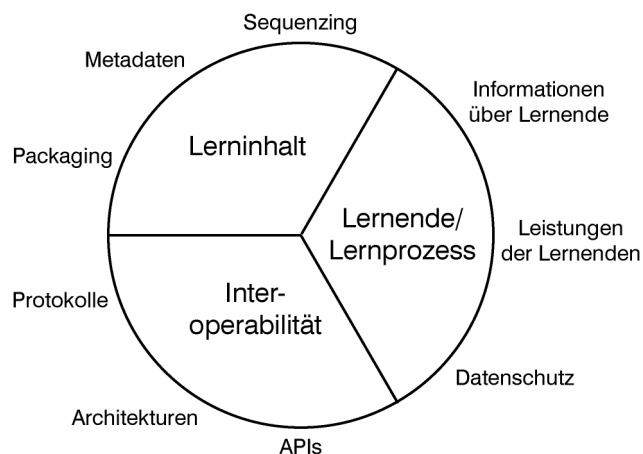


Abb. 1.8:
Funktionelle Kategorisierung
von E-Learning Standards,
nach [Epolos o.J.]

1.3.2 Standardisierungsprojekte

In den verschiedensten Bereichen arbeiten Vertreter aus Wirtschaft, Wissenschaft und Regierung zusammen um gemeinsame Standards zu entwickeln. Dies soll der Entwicklung proprietärer (herstellerspezifischer) Formate entgegenwirken und durch gemeinsame, offene Standards die Koexistenz verschiedener, aber dennoch kompatibler Produkte ermöglichen. Dies sorgt sowohl auf Verbraucher- als auch auf Herstellerseite für Investitionssicherheit.

Zur Beschreibung der Arbeit der Standardisierungsprojekte ist zunächst die Klärung der Begriffe Norm, Standard, Spezifikation, Empfehlung wichtig.

Während unter einer *technischen Norm* eine allseits rechtlich anerkannte und durch ein offizielles Normungsverfahren beschlossene, allgemeingültige sowie veröffentlichte Regel zur Lösung eines Sachverhaltes zu verstehen ist, kann der Begriff Standard verschiedene Bedeutungen besitzen. Zum einen wird er als Synonym für Norm eingesetzt. Man spricht dann auch von einem *akkreditierten Standard* oder einem *De-jure-Standard* („durch das Recht“). Zum anderen wird der Begriff Standard im Sinne von *Industriestandard* bzw. *De-facto-Standard* („durch die Tatsache“) verwendet. Industriestandards wurden zwar keinem offiziellen Normungsverfahren unterzogen, haben sich aber im Laufe einer gewissen Zeit in der Praxis als nützlich und richtig erwiesen.

Nur wenige Standardisierungsorganisationen sind dazu befugt, Normungsverfahren zur Akkreditierung von Standards durchzuführen. Dabei handelt es sich meist um nationale oder inter-

ationale Normungseinrichtungen, wie z.B. das *Deutsche Institut für Normung* (DIN), das *American National Standards Institute* (ANSI), das europäische *Comité Européen de Normalisation* (CEN) und der *International Organisation for Standardization* (ISO). Im Bereich E-Learning spielt das IEEE (*Institute of Electric and Electronic Engineers*) eine bedeutende Rolle bei der Durchführung von Normungsverfahren und besitzt das Recht, akkreditierte Standards zu definieren.

Viele Gremien und Initiativen, die sich mit Fragen der Standardisierung auseinandersetzen, sind jedoch nicht dazu befugt, offizielle Normungsverfahren durchzuführen. Die Ergebnisse solcher Gremien werden als *Spezifikation*, *Empfehlungen* oder *Richtlinien* bezeichnet und stellen häufig die Grundlage für eine anschließende Normungsarbeit dar. [Gries 2003, S. 2]

In den USA und in Europa haben sich in den letzten Jahren einige Gremien gebildet, die an der Standardisierung im Bereich E-Learning arbeiten. Einen wichtigen Beitrag hierbei leistet das SCORM Referenzmodell der *Advanced Distributed Learning Initiative* (ADL) mit der Zielsetzung, bestehende Spezifikationen bzw. Standards integrativ in einem Modell zu vereinen.

Im Folgenden werden wichtige Initiativen vorgestellt, deren Arbeit in Zusammenhang mit der Entwicklung des SCORM Referenzmodells steht.

AICC:

Das durch die amerikanische Luftfahrtindustrie im Jahre 1988 gegründete AICC (Aviation Industry Computer-Based Training Committee) ist ein internationaler Zusammenschluss von CBT-Herstellern und –Experten. Diese arbeiten gemeinsam an der Standardisierung von Lerntechnologien für den Einsatz in der Luftfahrtindustrie. Dennoch werden in Hinblick auf die Interoperabilität die durch das AICC entwickelten Richtlinien bewusst so allgemein formuliert, dass diese auch außerhalb der Luftfahrtindustrie eingesetzt und verbreitet werden können.

Die Richtlinien, die durch das AICC veröffentlicht werden, unterscheidet zwischen *AGRs* (AICC Guidelines and Recommendations) und *Technical Reports & Whitepapers*. AGRs beinhalten Empfehlungen zu verschiedenen Themen, wie z.B. ‚AGR 006‘ zur Interoperabilität von Systemen für CMI² (Computer Managed Instruction).

Zur Vertiefung der AGRs dienen Technical Reports & Whitepapers, die sich je nach Präfix einem Themenbereich zuordnen lassen. So steht zum Beispiel ‚COM‘ für den Bereich ‚Commu-

2. Unter CMI Systemen versteht AICC Learning Management Systeme.

nications‘ (Kommunikation) und ‚CMI‘ für den Themenbereich ‚Computer Managed Instruction‘. [AICC]

Im Bereich E-Learning hat AICC mit dem Dokument *AICC CMI Guidelines for Interoperability* (AICC CMI 001) einen wichtigen Beitrag zum Standardisierungsprozess beigetragen. Dieses liefert unter anderem ein Datenmodell zur Kommunikation (Communication Data Model) zwischen Lernobjekt und Lernmanagementsystem und gibt Richtlinien zur technischen Umsetzung dieser Kommunikation [AICC 2001]. Mittlerweile wurde auf Basis dieser Spezifikation der von IEEE akkreditierten Standard *IEEE 1.484.11.2-2003 – ECMAScript Application Program Interface for Content to Runtime Services Communication* und der Normentwurf *IEEE 1484.11.1 – Data Model for Content Object Communication (draft)* entwickelt.

Teile des SCORM Referenzmodells basieren bis einschließlich der Version 1.2 auf Richtlinien des Dokuments AICC CMI 001 (vgl. 1.4.2). Die aktuelle Version SCORM 2004 integriert bereits den von IEEE akkreditierten Standard IEEE 1.484.11.2-2003 und den IEEE Normentwurf IEEE 1484.11.1. [SCORM 2004b, S. 3-3 u. 4-3f]

Auf der Homepage von AICC befinden sich neben den Links zu den Veröffentlichungen unter anderem auch eine Auflistung der Produkte, die durch das AICC zertifiziert wurden, eine Downloadmöglichkeit für eine Testumgebung zur Überprüfung der Umsetzung bestimmter AICC-Richtlinien.

IEEE LTSC:

Das IEEE (Institute of Electric and Electronic Engineers) wurde im Jahr 1963 in den USA gegründet. Die IEEE stellt die einen Zusammenschluss von Menschen mit technischen Arbeitsfelder dar und besitzt heute mehr als 360 000 Mitglieder aus geschätzt 175 verschiedenen Ländern. [IEEE 2005]

Das LTSC (Learning Technology Standards Committee) wurde 1996 als eine Untereinheit des IEEE gegründet und besitzt, wie bereits erwähnt, als einzige der aufgeführten Initiativen das Recht, sowohl akkreditierte Standards als auch Richtlinien und Praxisempfehlungen zur Entwicklung und Implementierung von computer- und webbasierten Lernsystemen zu definieren. [Baumgartner u.a. 2002, S. 310]

ARIADNE Foundation:

Eine europäische Initiative ist dagegen die ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe) Foundation. Diese setzt die Arbeit der Vorgängerprojekte ARIADNE und ARIADNE II fort, die zum größten Teil durch die EU und die Schweizer Regierung finanziert wurden. Von 1996 bis 2000 wurden durch diese Projekte Werkzeuge und Methoden zur Produktion, Verwaltung und Wiederverwendung computerbasierter Lehr- und Lernelemente entwickelt. Darunter auch die ARIADNE Educational Metadata Specification, die einen wichtigen Beitrag zu der Entwicklung von E-Learning Standards darstellt. Diese Spezifikation, im englischen auch als *pedagogical header* bezeichnet, liefert eine der Grundlagen für den durch IEEE akkreditierten Standard *IEEE 1484.12.1-2002 – Learning Object Metadata (LOM)*. Dieser kommt auch unter SCORM 2004 zum Einsatz, wobei die SCORM Version 1.2 noch eine Entwurfsversion dieses Standards integriert (vgl. 1.4.1).

Die ARIADNE Foundation finanziert sich heute durch Mitgliedsbeiträge, eigene Einnahmen (z.B. die Austragung von Konferenzen) und Spenden, wird aber auch durch öffentliche und staatliche Einrichtungen, sowie durch die EU bezuschusst. Seine Aufgaben sieht die Foundation heute vorwiegend in der Pflege und der Weiterentwicklung bestehender, durch ARIADNE entwickelte Werkzeuge zur Erstellung, Verwaltung und Wiederverwendung von computerbasierten Lernressourcen. [ARIADNE]

IMS Global Learning Consortium:

Das IMS (Instructional Management Systems Project) Global Learning Consortium ist ein internationaler Zusammenschluss zur Entwicklung offener Standards für Lerntechnologien. Seine Mitglieder stammen aus den verschiedensten Bereichen, wie z.B. Bildungs- und Regierungseinrichtungen, Hersteller von Lernplattformen oder computer- und webbasierten Lernanwendungen.

1997 gegründet, stellte die IMS ursprünglich ein Projekt der National Learning Infrastructure Initiative innerhalb von EDUCAUSE dar, einer US-amerikanischen Organisation mit dem Ziel, die Hochschulbildung durch die intelligente Nutzung und Verbreitung von Informationstechnologien voranzutreiben. Auch das IMS Projekt setzte seinen Schwerpunkt zu Beginn auf den Bereich der Hochschulbildung. Aufgrund des breiten internationalen Interesses fand im Jahr 1999 eine Umwandlung des IMS Projects in das IMS Global Learning Consortium statt, und auch der Schwerpunkt der Arbeit hat sich mittlerweile auf die verschiedensten Bereiche der Aus- und Weiterbildung ausgedehnt.

Im Bereich E-Learning stellt die *IMS Learning Resource Meta-data Specification* eine weitere Grundlage des bereits erwähnten IEEE Standards LOM dar. [IMS 2005]

Auch haben sich die Spezifikationen *IMS Content Packaging* und *IMS Content Packaging XML Binding* in diesem Bereich durchgesetzt. Sie liefern Richtlinien zur Beschreibung und Erstellung von Lernressourcen als austauschbare und in unterschiedlichen Systemen einsetzbare Pakete. Beide Spezifikationen werden sowohl unter SCORM 1.2 als auch unter SCORM 2004 eingesetzt. Die Spezifikationen *IMS Simple Sequencing Information and Behavior Model* und *IMS Simple Sequencing XML Binding* sind dagegen erst seit SCORM 2004 im Referenzmodell integriert. Sie dienen der Festlegung des Ablaufs und der Steuerung einer Kurspräsentation (Sequencing) in Abhängigkeit vom Anwenderverhalten und dessen Lernergebnissen. [Gries 2003, S. 5]

ADL:

Die Advanced Distributed Learning Initiative wurde im Jahr 1997 vom Amerikanischen Verteidigungsministerium (Department of Defence) und vom Office of Science & Technology Policy (OSTP) des Weißen Hauses zum Leben erweckt. Die Zielsetzung dieser Initiative ist es, kostengünstigen Zugang zu Lern- und Übungsmaterialien zu verschaffen, die sowohl auf individuelle Bedürfnisse zugeschnitten und zeit- und ortsunabhängig genutzt werden können als auch hohe Qualitätsanforderungen erfüllen. Dazu bedarf es, laut ADL, der Akzeptanz und der Verbreitung eines Referenzmodells mit Richtlinien für webbasierte Lerninhalte. Dieses sollte folgende Eigenschaften erfüllen:

- Accessibility: Zugriffsmöglichkeit auf Lernobjekte von unterschiedlichen Orten aus
- Adaptability: Anpassbarkeit von Lernobjekten auf verschiedene individuelle oder organisatorische Bedürfnisse
- Affordability: Steigerung der Effizienz durch Zeit- und Kostenreduzierung bei der Erstellung von Lernanwendungen
- Durability: Beständigkeit gegenüber technischen Veränderungen
- Interoperability: die Möglichkeit, Lernkomponenten in verschiedenen Systemen oder unter anderen technischen Voraussetzungen nutzen zu können
- Reusability: Wiederverwendbarkeit von Lerninhalten in verschiedenen Anwendungen und Kontexten

Diese Anforderungen dienen als Grundlage bei der Entwicklung des Referenzmodells SCORM (Sharable Content Object Reference Model) und sollen bei jeder Veränderung und Weiterentwicklung von SCORM beachtet werden. [SCORM 2004a, S. 1-3 u.1-22] SCORM hat sich mittlerweile aufgrund seines verbreiteten Einsatzes in der Praxis als Industriestandard bzw. De-facto-Standard etabliert.

Die genannten Institutionen gingen zunächst getrennt ihren Standardisierungsbestrebungen nach, haben sie sich aber mittlerweile zur Zusammenarbeit entschlossen. Das daraus entstandene Kooperationsnetzwerk (vgl. Abb. 1.9) sieht folgendermaßen aus:

AICC, IMS und ARIADNE liefern Spezifikationen an das IEEE LTSC. Dieses besitzt das Recht, akkreditierte Standards zu definieren. Zudem stellen die Ergebnisse des IEEE Grundlagen für die nationale und internationale Normung dar.

Anhand des ADL Referenzmodells SCORM können konkrete Anforderungen an die Spezifikationen bzw. Standards abgeleitet werden. [Niegemann u.a. 2004, S. 271f]

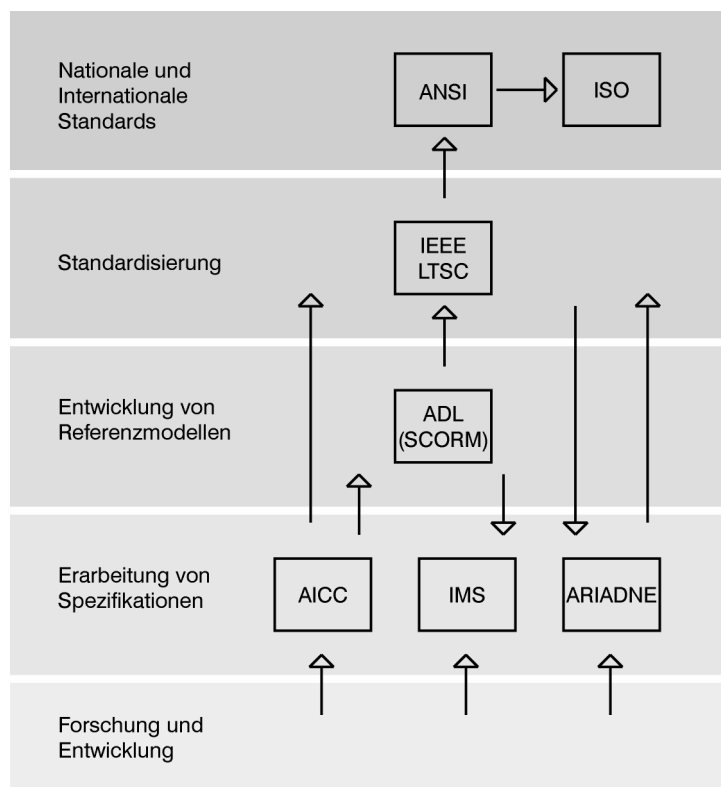


Abb. 1.9:
Kooperationsnetzwerk der E-Learning
Standardisierungsinitiativen,
nach [ANOVA o. J.]

1.4 Das SCORM Referenzmodell

Zu Beginn des Jahres 2000 begann die erste Test- und Evaluationsphase des SCORM Referenzmodells. Seitdem hat SCORM bereits drei Versionswechsel und eine Namensänderung erlebt. In seiner ersten Version (1.0) stand SCORM noch für Sharable *Courseware* Object Reference Model. Seit der Version 1.1 steht es für Sharable *Content* Object Reference Model. Diese Änderung wurde vorgenommen, da der Begriff ‚content‘ (Inhalt) besser beschreibt, worauf sich das Referenzmodell bezieht, nämlich nicht nur auf die Kursunterlagen (courseware) als Ganzes, sondern auch auf die verschiedenen Teile und Ebenen von Kursunterlagen bzw. Lerninhalten. [SCORM 2001a, S. 1-8]

Die aktuelle Version SCORM 2004 2nd Edition (1.3.1) wurde im Juli 2004 veröffentlicht. Sie stellt eine Überarbeitung der Version SCORM 2004 (1.3) dar, die im Januar 2004 erschienen ist. Viele Lernplattformen unterstützen momentan diese aktuellen Versionen³ noch nicht, so auch die Lernplattform des Bildungsportals Sachsen. Da zum Zeitpunkt der Erstellung des Lernmoduls „Typografie praktisch!“ noch kein Termin für die Umstellung von SCORM 1.2 auf SCORM 2004 feststand, wurde das Lernmodul nach SCORM 1.2 entwickelt. Aus diesem Grund beziehen sich die nachfolgenden Erläuterungen auf die Version 1.2, die im Oktober 2001 veröffentlicht wurde.

Das SCORM Referenzmodell setzt sich aus mehreren Büchern zusammen. Diese stellen abgesehen vom ersten Buch Sammlungen von Spezifikationen und Standards dar und werden auch als *technical books* bezeichnet. Mit jeder neuen SCORM Version werden die bereits vorhandenen Bücher aktualisiert oder bei Bedarf auch neue Bücher hinzugefügt. SCORM 1.2 besteht aus drei Büchern:

Das erste Buch besitzt den Titel *Overview*. Darin wird ein Überblick über ADL und SCORM sowie die technischen Spezifikationen und Richtlinien der folgenden Bücher gegeben.

Das zweite Buch beschreibt das *SCORM Content Aggregation Model*. Mit dem Ziel Interoperabilität zu gewährleisten, enthält dieses Buch Richtlinien zur Beschreibung und zum Aufbau von Lerninhalten. Das Content Aggregation Model basiert auf dem IEEE Standard *LOM* sowie den IMS Spezifikationen *Content Packaging* und *Meta-data XML Binding and Best Practice* und der *Content Structure Spezifikation* des AICC.

3. *SCORM 2004* und *SCORM 2004 2nd Edition* werden in dieser Arbeit zusammenfassend als *SCORM 2004* bezeichnet.

Das dritte Buch *The SCORM Run Time Environment* enthält die Beschreibung der Kommunikation zwischen Lernobjekt und Lernmanagementsystem. Das dazu verwendete Datenmodell (*Data Model*) basiert auf den AICC CMI Guidelines for Interoperability. Die Kommunikationsmechanismen (*Launch* und *Communication API*) sind durch Zusammenarbeit von AICC und ADL entstanden. [SCORM 2001a, S. 1-4f] u. [SCORM 2001c, S. 3-7]

Die Abbildung 1.10 stellt diese Gliederung in Bücher dar und zeigt, welche Standards bzw. Spezifikationen in welche SCORM-Bücher mit eingeflossen sind.

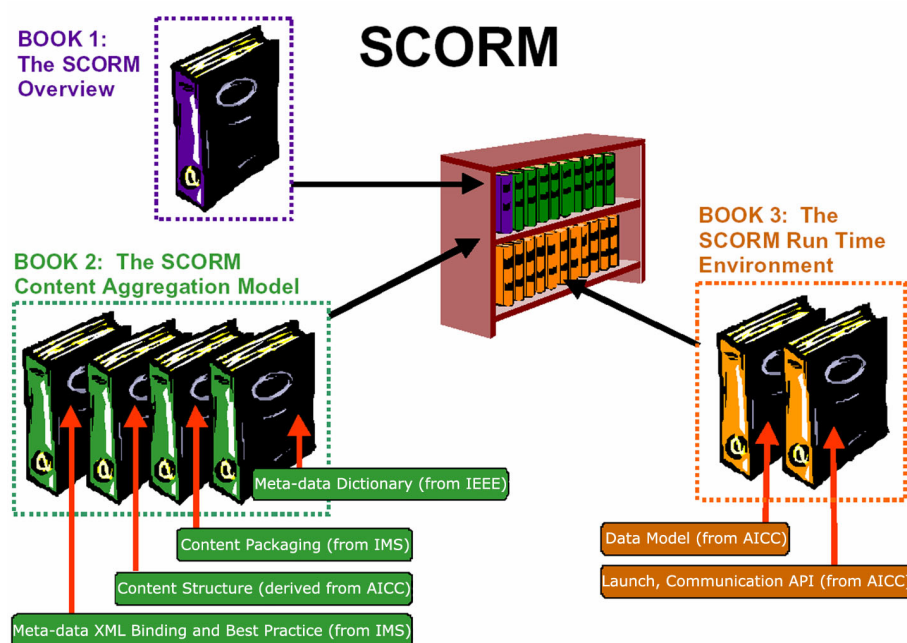


Abb. 1.10:
SCORM als eine
Sammlung von
Spezifikationen,
nach [SCORM
2001a, S. 1-5]

SCORM 2004 enthält bereits vier Bücher. Neu hinzugekommen ist das Buch *Sequenzierung und Navigation*, welches die Ablaufsteuerung der Lerninhalte und die Navigation genauer beschreibt. Zudem gab es verschiedene Änderungen im zweiten und dritten SCORM-Buch. Wie bereits erwähnt, soll in dieser Arbeit nicht weiter auf SCORM 2004 eingegangen werden. Die folgenden Abschnitte beschäftigen sich mit dem zweiten und dem dritten Buch der SCORM Version 1.2.

1.4.1 Das SCORM Content Aggregation Model

Das Content Aggregation Model (CAM) beschreibt ein Modell zur Erstellung von Lerneinheiten, die sich aus verschiedenen Komponenten zusammensetzen. Dieses Modells ist so angelegt, dass die einzelnen Komponenten *identifizierbar*, *wieder verwendbar*, *gemeinsam nutzbar* und *kompatibel* sind. Sie sollen in verschiedenen Lerneinheiten integriert und in verschiedenen Learning Management Systemen genutzt werden können. Unter SCORM 1.2 setzt sich das Content Aggregation Modell aus drei Teilen zusammen.

Im ersten Teil *SCORM Content Model Components* werden die verschiedenen Komponenten einer SCORM-Lerneinheit definiert. Zudem ist hier der Einsatz von Metadaten zur Beschreibung dieser Komponenten festgelegt.

Der zweite Teil mit dem Titel *Meta-Data* beinhaltet eine Auflistung mit genauer Beschreibung der Metadaten, die zur Beschreibung der Komponenten verwendet werden können.

Im dritten Teil, dem *Content Packaging*, wird genau definiert, wie die einzelnen Komponenten zu Lerneinheiten *verpackt* werden, und wie die Struktur des Pakets durch Metadaten beschrieben werden kann.

SCORM Content Model Components:

Das SCORM Content Model beinhaltet die drei Komponenten *Assets*, *Sharable Object Content (SCO)* und *Content Aggregation*:

Assets:

Ein Asset stellt die kleinste Lernressource dar, die von einem Learning Management System gestartet werden kann. Dabei kann es sich um jegliche Form elektronisch gespeicherter Daten handeln, die durch einen Internetbrowser geöffnet und dargestellt bzw. wiedergegeben werden können. Assets können demnach Texte, Bilder, Sound, Flashfilme oder auch HTML-Seiten (vgl. Abb. 1.11) repräsentieren. Mehrere Assets lassen sich zu einem weiteren Asset oder auch zu einem SCO zusammenfassen.

Sharable Object Content (SCO):

Ein Sharable Object Content setzt sich aus einem oder mehreren Assets zusammen. Dabei sollte es im Sinne der Wiederverwendbarkeit einen unabhängigen Lerninhalt repräsentieren. ADL empfiehlt daher, die Größe eines SCOs auf die kleinstmögliche, sinnvolle Lerneinheit zu begrenzen.

Der eigentliche Unterschied zwischen SCOs und Assets besteht jedoch darin, dass ein SCO von einem LMS nicht nur gestartet werden kann, sondern mit diesem über die Laufzeitumgebung (Run-Time Environment) auch kommunizieren kann. Daher muss ein SCO in der Lage sein, die Schnittstelle (Application Programming Interface/API), die vom LMS bereitgestellt wird, zu lokalisieren. Zudem muss es mindestens die API-Methoden zum Starten (`LMSInitialize("")`) und zum Beenden (`LMSFinish("")`) der Kommunikation aufrufen (vgl. Abb. 1.12). Alle weiteren Methoden, die zur Gestaltung der Kommunikation zu Verfügung stehen, sind optional. Im Abschnitt 1.4.2 werden die API-Methoden ausführlicher behandelt.

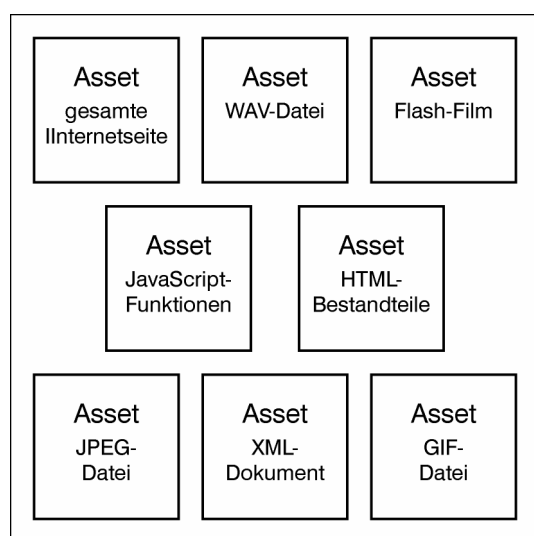


Abb. 1.11:
Beispiele für Assets, nach [SCORM 2001b, S. 2-4]

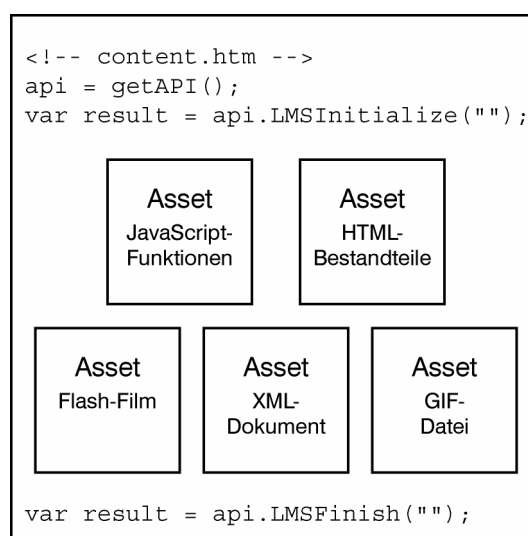


Abb. 1.12:
Beispiel für ein SCO, nach [SCORM 2001b, S. 2-5]

Content Aggregation:

Die Content Aggregation definiert die *Content Structure*, die einzelne Lernressourcen (Assets und SCOs) zu festen Lerneinheiten zusammenfasst. Je nach Bedarf stellen diese Lerneinheiten Kurse, Module oder auch nur einzelne Kapitel oder Themen dar.

In der Content Structure wird die Reihenfolge der Darstellung der Lernressourcen in einem LMS festgelegt. Zudem kann über die Content Structure die Definition von *prerequisites* erfolgen. Unter *prerequisites* versteht man Vorbedingungen, die erfüllt sein müssen, bevor die entsprechende Ressource dem Lernenden präsentiert werden kann.

Die Abbildung 1.13 skizziert einen möglichen Aufbau einer Content Aggregation. Der genaue Aufbau einer Content Structure wird im dritten Teil des zweiten Buches (*Content Packaging*) beschrieben.

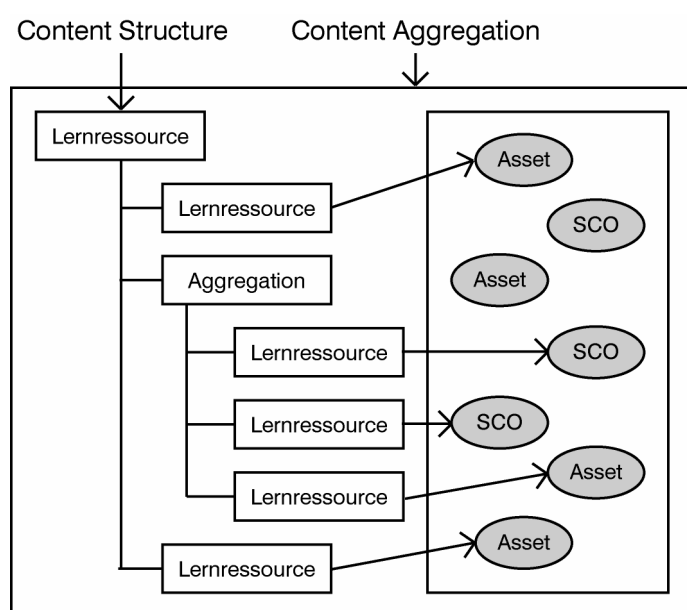


Abb. 1.13:
Content Aggregation,
nach [SCORM 2001b, S. 2-7]

Das Content Model beinhaltet außerdem Empfehlungen zur Beschreibung der drei Komponenten durch Metadaten, welche deren Wiederverwendbarkeit erleichtern sollen. Die genaue Beschreibung dieser Metadaten (*Asset Meta-data*, *SCO Meta-data* und *Content Aggregation Meta-data*) befindet sich im zweiten Teil des CAM und wird im Folgenden beschrieben.

Meta-Data:

Der Abschnitt *Meta-Data* des Content Aggregation Models besteht neben einem Überblick und praktischen Empfehlungen des ADL-Teams aus den Teilen *SCORM Meta-data Information Model*, *SCORM Meta-data XML-binding* und *SCORM Meta-data Application Profiles*:

SCORM Meta-data Information Model:

Das SCORM Meta-data Information Model basiert auf dem IMS Learning Resource Meta-data Information Model und besteht, genau wie dieses, aus neun Kategorien zur Beschreibung von Lernressourcen. Tabelle 1.2 gibt einen Überblick über die Kategorien und beschreibt diese kurz.

Tab. 1.2: Kategorien des SCORM Meta-data Information Model, nach [SCORM 2001a, S. 2-13ff]

Kategorie	Beschreibung
general	beinhaltet allgemeine Informationen zur Beschreibung der Lernressource als ganzes beschreiben (z.B. Titel, Sprache, Text oder Stichworte zur Beschreibung des Inhalts)
lifecycle	Kategorie zur Beschreibung von Informationen zur Entwicklung, zum Entwicklungsstand der Ressource (beinhaltet u.a. auch die an der Entwicklung beteiligten Personen)
metametadata	Metadaten zur Beschreibung der verwendeten Metadaten (z.B. Name und Version des Metadatenschemas)
technical	Beschreibung der technischen Voraussetzungen und Charakteristika der Ressource (z.B. Format und Größe der Datei)
educational	Enthält pädagogische Charakteristika der Lernressource (z.B. zur Interaktivität)
rights	Dient der Definition von Nutzungsrechten, -bedingungen und -einschränkungen
relation	Kategorie zur Beschreibung von Abhängigkeitsbeziehungen der Lernressource zu anderen, falls vorhanden
annotation	Beschreibung des pädagogischen Nutzens der Ressource
classification	Einordnung der Ressource in ein Klassifizierungsschema

Der Aufbau des Meta-data Information Models ist hierarchisch gestaltet. Die einzelnen Kategorien besitzen neben Datenelementen z.T. auch Unterkategorien. Das SCORM Meta-data Information Model enthält eine genaue hierarchisch aufgebaute Beschreibung der Kategorien und Unterkategorien sowie der Datenelemente. Darin wird u.a. auch definiert, wie häufig ein Element vorkommen kann, und welchem Datentyp es entsprechen muss. Das Meta-data Information Model unterscheidet dabei zwischen *Container*, *String*, *LangString*, *Vocabulary* und *Date*. Beim Datentyp *Vocabulary* werden verschiedene Werte vorgegeben, zwischen denen man wählen kann. Hierbei wird zwischen *Restricted* und *Best Practice* unterschieden. ‚Restricted‘ bedeutet, dass man sich bei der Auswahl an einen der vorgegebenen Werte zu halten hat. ‚Best Practice‘ gibt nur Empfehlungen, an die man sich nicht halten muss. Im Sinne der Interoperabilität wird aber empfohlen, sich an einen der vorgegebenen Werte zu halten.

SCORM Meta-data XML Binding:

Die SCORM Meta-data XML Binding basiert auf der IMS Learning Resource XML Binding Specification. Sie definiert, wie das Meta-data Information Model in der *Extensible Markup Language* (XML) darzustellen ist.

In der hierarchisch aufgebauten XML-Struktur stellt das Element `<lom>` immer das äußerste Element dar und deutet auf das verwendete Datenmodell ‚Learning Object Metadata‘ hin. Die in der Tabelle 1.2 vorgestellten Kategorien des Datenmodells (general, lifecycle, metametadata etc.) stellen Elemente (*Elements*) von `<lom>` dar und sind hierarchisch unterhalb angeordnet.

Die Meta-data XML Binding beschreibt jedes Element durch verschiedene Werte, wie z.B. *Multiplicity* (Häufigkeit des Auftretens eines Elements durch Festlegung eines minimalen und maximalen Werts) oder auch *Elements* (Elemente, die hierarchisch unterhalb liegen).

Zudem wird der Einsatz verschiedener Datentypen innerhalb der XML-Datei festgelegt. Im Folgenden werden Beispiele zu den *LangString*, *Vocabulary* und *Date* Datentypen dargestellt:

Datentyp *LangString*:

`<xml:lang>` definiert die Sprache (nach ISO-Standard) des Wortes oder Textes, z.B. ‚de‘ für Deutsch.

```
<classification>
  <keyword>
    <langstring xml:lang="de">Typografie</langstring>
  </keyword>
</classification>
```

Bsp. 1.1: Datentyp *LangString*

Datentyp *Date*:

Datumsangaben unter `<datetime>` erfolgen nach ISO-Standard; unter `<description>` kann das Datum ausgeschrieben werden .

```
<datetime>2000:12:01</datetime>
<description>
  <langstring>1. Dezember 2000</langstring>
</description>
```

Bsp. 1.2: Datentyp *Date*

Datentyp *Vocabulary*:

Der Wert ‚x-none‘ wird beim <vocabulary> Datentyp verwendet.

```
<classification>
<purpose>
  <vocabulary>
<source>
  <langstring xml:lang="x-none">LOMv1.0</langstring>
</source>
<value>
  <langstring xml:lang="x-none">Educational Objective</langstring>
</value>
</vocabulary>
</purpose>
</classification>
```

Bsp. 1.3: Datentyp *Vocabulary*

SCORM Meta-data Application Profiles:

Im dritten Teil der Meta-Data-Spezifikation, dem SCORM Meta-data Application Profile, wird der Zusammenhang zwischen den Komponenten des Modells und der IMS Learning Resource Meta-data Spezifikation hergestellt. Aufgrund der drei Komponenten wird zwischen den drei folgenden Application Profiles unterschieden:

- Die *Content Aggregation Meta-data* beschreibt die Anwendung der Learning Resource Meta-data auf die Content Aggregation Komponente.
- Die *SCO Meta-data* beschreibt die Anwendung der Meta-data auf SCOs.
- Die *Asset Meta-data* beschreibt die Anwendung der Meta-data auf Assets.

Für die drei Application Profiles sind verschiedene Voraussetzungen bezüglich der Datenelemente festgelegt. Dabei wird zwischen Datenelementen unterschieden, die als Metadaten einer Komponente zwingend (*mandatory*) vorhanden sein müssen, und Datenelementen, deren Verwendung optional (*optional*) ist.

Zur Zuordnung dieser Bedingungen steht eine Tabelle mit der Auflistung aller Elemente zur Verfügung. Tabelle 1.3 zeigt einen Ausschnitt daraus.

Tab. 1.3: *Mandatory* (M) und *optional* (O) Datenelemente, nach [SCORM 2001a, S. 2-96ff]

Name	Content Aggregation	SCO	Asset
2.0 lifecycle	M	M	O
2.1 version	M	M	O
2.2 status	M	M	O
2.3 contribute	O	O	O

Nach SCORM ist die Beschreibung von Komponenten einer Lerneinheit durch Metadaten nicht zwingend. Entscheidet man sich allerdings dazu, eine Komponente durch Metadaten zu beschreiben, um die Wiederverwendbarkeit zu erleichtern, müssen zumindest die durch *mandatory* ausgezeichneten Elemente definiert werden.

Content Packaging:

In diesem Abschnitt des Content Aggregation Models wird zunächst die *Content Structure* vorgestellt. Diese beschreibt verschiedene Möglichkeiten zur Strukturierung der Lernressourcen zu einer SCORM-Lerneinheit. In weiteren Teilen werden das *SCORM Content Packaging Information Model*, die *SCORM Content Package XML Binding* und die *SCORM Content Packaging Application Profiles* beschrieben.

SCORM Content Packaging Information Model:

Das SCORM Content Packaging Information Model basiert auf der IMS Content Packaging Spezifikation. Diese beschreibt Datenstrukturen, welche die Interoperabilität von Lerninhalten zu Learning Management Systemen gewährleisten sollen.

Ein IMS Content Package besteht zum einen aus einem XML Dokument mit der festen Bezeichnung *imsmanifest.xml*, welches die Struktur des Pakets beschreibt. Das äußerste Element dieser Struktur wird als `<manifest>` bezeichnet. Es beinhaltet die weiteren Elemente `<metadata>`, `<organizations>`, `<resources>` und kann zudem auch weitere `manifest-`Elemente besitzen.

Die Inhaltsstruktur des Pakets wird unter <organizations> festgelegt. Unter <resources> wird der Speicherort der verwendeten physischen Dateien innerhalb dieser Struktur definiert.

Die eigentlichen, physischen Dateien, auf die unter <resources> Bezug genommen wird, stellen den zweiten Teil eines Pakets dar.

ADL empfiehlt die Speicherung von Content Packages in Form von Archivdateien (wie z.B. ‚zip‘ oder ‚jar‘) als Package Interchange File (PIF). Abbildung 1.14 verdeutlicht den Aufbau eines IMS Content Packages.

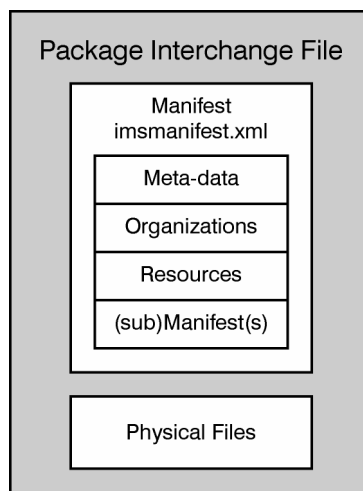


Abb. 1.14: IMS Content Packaging (PIF), nach [SCORM 2001b, S. 2-111]

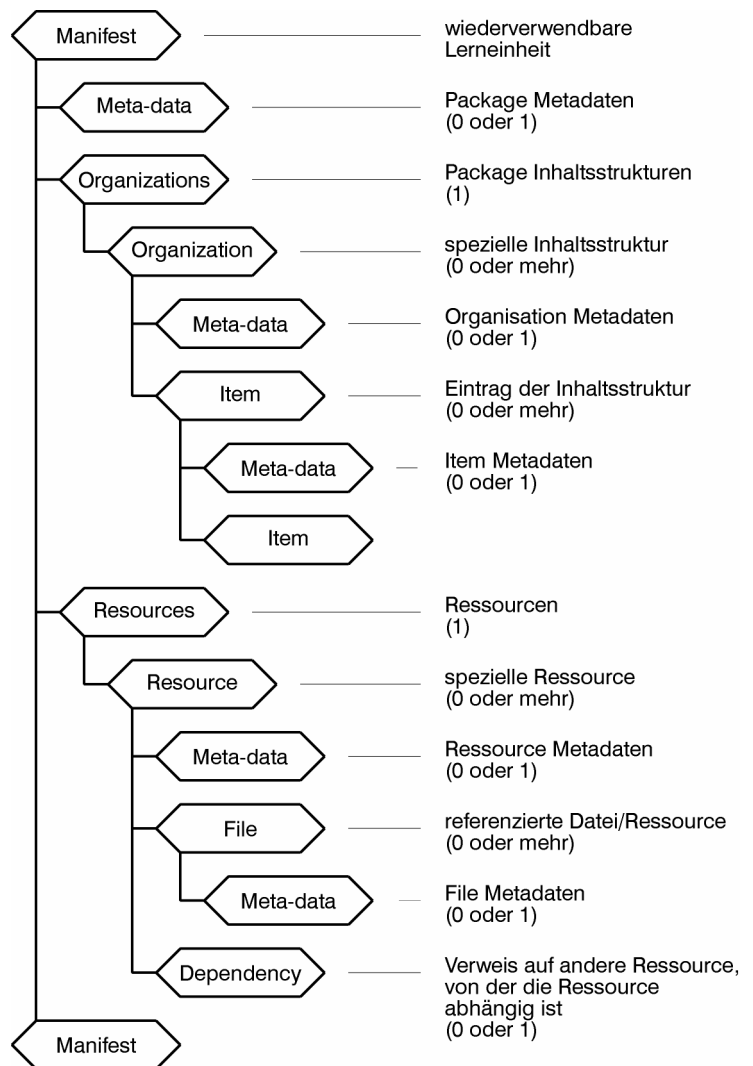


Abb. 1.15: Struktur einer Manifestdatei, nach [Kammer 2003, S. 28]

In Form einer Tabelle gibt ADL genaue Angaben zu den einzelnen Elementen des SCORM Content Information Models. Abbildung 1.15 gibt ein Beispiel zur Strukturierung einer Manifest Datei.

SCORM Content Packaging XML Binding:

Die SCORM Content Packaging XML Binding legt fest, wie das zuvor beschriebene SCORM Content Packaging Information Model in XML umgesetzt wird. Wie bereits erwähnt, wird die Manifestdatei (imsmanifest.xml) in die drei Teile Meta-data, Organizations und Resources unterteilt. Folgendes Beispiel verdeutlicht diese Dreiteilung.

```
<?xml version="1.0"?>
<manifest identifier="Manifest" version="1.0"
xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsproject.org/xsd/
imscp_rootv1p1p2imscp_rootv1p1p2.xsd
http://www.imsglobal.org/xsd/imsmd_rootv1p2p1_imsmd_rootv1p2p1.xsd
http://www.adlnet.org/xsd/adlcp_rootv1p2_adlcp_rootv1p2.xsd">
<metadata>
  <schema>ADL SCORM</schema>
  <schemaversion>1.2</schemaversion>
  <imsmd:lom>
    <imsmd:general>
      <imsmd:title>
        <imsmd:langstring xml:lang="de">Manifest Datei
</imsmd:langstring>
      </imsmd:title>
    </imsmd:general>
  </imsmd:lom>
</metadata>
<organizations default="ORG_1">
<organization identifier="ORG_1" structure="hierarchical">
  <title>default</title>
<item identifier="ITEM_1" identifierref="RESOURCE_1" isvisible="true">
  <title>Inhalt 1</title>
</item>
</organization>
</organizations>
<resources>
  <resource identifier="RESOURCE_1" type="webcontent" href="inhalt1.html">
    <file href="inhalt1.html"/>
  </resource>
</resources>
</manifest>
```

Bsp. 1.4: Manifestdatei, nach [SCORM 2001a, S. 2-127ff]

Im Bereich <organizations> lassen sich Organisationen festlegen. Innerhalb dieser Organisationen wird die Struktur der Lernressourcen durch die Anordnung von <items> festgelegt. Diese stellen Einträge dar, welche auf eine Ressource verweisen.

Im Bereich <resources> erhält jede Ressource einen Verweis auf eine physische Datei. SCORM legt fest, dass es sich bei einer solchen Datei um den Typ ‚webcontent‘ handeln muss.

Als *Webcontent* können sowohl HTML-Seiten als auch jeder andere beliebige Inhalt, der von einem Internetbrowser gestartet werden kann, definiert werden.

Content Packaging Application Profiles:

Die Content Packaging Application Profiles beschreiben, wie die IMS Content Packaging Spezifikation im Gesamtkontext von SCORM angewandt werden kann. Dabei wird zwischen zwei Paketformaten unterschieden. Das *Resource Package* stellt die einfachere der beiden Formen dar. Die Lernressourcen werden zwar verpackt, erhalten aber keine spezifische Struktur, da der Organizations-Bereich in der Manifestdatei leer bleibt. Ein Resource Package ist also nur eine Zusammenfassung verschiedener Lernressourcen, die von einem Lernsystem ins andere transportiert werden können.

Das Content Aggregation Package beinhaltet dagegen eine strukturierte Anordnung von Lernressourcen und kann dadurch Einfluss auf die Darstellung und die Abarbeitung der Lerninhalte nehmen.

1.4.2 Die SCORM Run-Time Environment

Das Ziel von SCORM, die Wiederverwendbarkeit und die Interoperabilität von Lernressourcen auf verschiedenen Lernplattformen zu gewährleisten, kann nur erfüllt werden, wenn es eine einheitliche Regelung zum Starten und zur Darstellung von Lernressourcen und für die Kommunikation zwischen Lernplattformen und Lernressourcen gibt. Die SCORM Run-Time Environment (RTE), das dritte Buch von SCORM 1.2, stellt mit drei verschiedenen Komponenten Mechanismen zur Erfüllung dieser Zielsetzungen zur Verfügung.

Der *Launch*-Mechanismus definiert eine einheitliche Vorgehensweise zum Starten und zur Darstellung von Lernressourcen. Für die Kommunikation zwischen Lernressourcen und Lernplattform sorgt das *Application Program Interface* (API), welches Mechanismen zur Statusabfragen und zum Datenaustausch unterstützt. Das *Data Model* stellt einen Standard für die Definition der Elemente, die zur Kommunikation genutzt werden, dar.

Die Spezifikationen für Launch, API und Data Model basieren auf den AICC CMI Guidelines for Interoperability (AICC CMI 001) und sind durch die Zusammenarbeit von ADL und AICC entstanden, wobei die AICC Richtlinien auf die Anforderungen einer webbasierter Kommunikation angepasst wurden. Die Abbildung 1.16 veranschaulicht die Arbeitsweise der RTE.

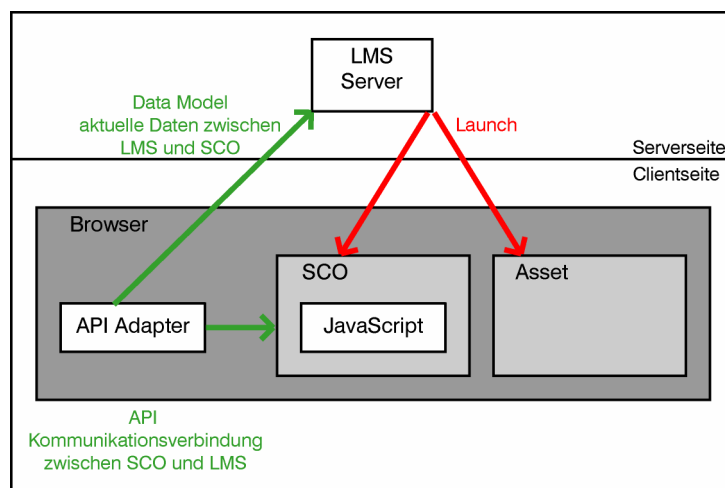


Abb. 1.16:
Launch, API und Data Model in der
SCORM RTE,
nach [SCORM 2001c, S. 3-3]

Launch:

Der Launch-Mechanismus beschreibt, wie Lernressourcen von einem LMS gestartet und dargestellt werden können. Dabei wird zwischen den beiden startbaren Lernressourcen SCO und Asset unterschieden.

Es liegt in der Verantwortung der Lernplattformen, die in der Content Structure festgelegten Struktur der SCOs und Assets umzusetzen und die Darstellungsreihenfolge der Lernressourcen festzulegen. Es kann dabei zwischen folgenden Möglichkeiten unterschieden werden:

- *sequential*: Darstellung der Lernressourcen erfolgt der Reihe nach, entsprechend der Reihenfolge in der Content Structure
- *user-directed*: Reihenfolge wird durch den Nutzer bestimmt, durch Auswahl aus einem Menü, welches vom LMS anhand der Content Structure generiert wird
- *adaptive*: Reihenfolge wird adaptiv angepasst, z.B. in Abhängigkeit zum Lernfortschritt des Nutzers

Fest vorgeschrieben wird hingegen von ADL, dass Lernplattformen das HTTP-Protokoll zum Anzeigen von Lernressourcen nutzen müssen. Für das Starten von Assets gibt es keine weiteren Richtlinien, da diese keine Kommunikation mit einem LMS eingehen können.

An den Launch-Mechanismus von SCOs stellt ADL dagegen weitere Anforderungen. So darf z.B. jeweils zu einem Zeitpunkt nur ein SCO dargestellt werden, und ein SCO auch kein weiteres SCO aus sich heraus starten.

Eine weitere Vorgabe besteht darin, dass ein LMS zur Gewährleistung der Kommunikation einen API-Adapter in Form eines *Document Object Models* (DOM) bereitstellen muss. Zudem ist vorgeschrieben, dass das LMS ein SCO in einem Kind-Fenster des Fensters, welches das DOM bereitstellt, öffnen muss.

Nachdem ein SCO gestartet wurde, muss dieses rekursiv nach dem Fenster suchen, welches den API-Adapter bereitstellt. Sobald der API-Adapter lokalisiert ist, kann die Kommunikation zwischen LMS und SCO beginnen.

Application Program Interface:

Das API kann man als eine Sammlung von Funktionalitäten beschreiben, die einem SCO in der RTE zur Verfügung stehen. Die Implementierung der Funktionalitäten bleibt einem SCO verborgen, wodurch die Wiederverwendbarkeit und die Interoperabilität von SCOs unterstützt wird.

API-Adapter stellen eine Software dar, welche die API-Funktionalitäten ausführen und nach außen hin zur Verfügung stellen. Wie bereits erwähnt, ist es die Aufgabe einer Lernplattform, einen API-Adapter bereitzustellen.

Die API-Funktionalitäten der SCORM Run-Time Environment kann man in drei Kategorien mit insgesamt neun verschiedenen Funktionen einteilen. Die Tabelle 1.4 stellt diese kurz vor. Die SCORM RTE enthält ausführliche Beschreibungen zu den einzelnen Funktionen.

Tab. 1.4: Funktionen zur Kommunikation zwischen SCO und LMS, nach [Kammer 2003, S. 34]

Kategorie	Kommando	Beschreibung
Execution State Funktionen zum Starten/Beenden der Kommunikation	<code>LMSInitialize("")</code>	Initialisierung der Kommunikation zwischen SCO und LMS
	<code>LMSFinish("")</code>	Beenden der Kommunikation
State Management Funktionen zur Fehlerbehandlung	<code>LMSGetLastError()</code>	gibt letzte Fehlermeldung zurück
	<code>LMSGetErrorString(errornumber)</code>	gibt Fehlernummer zurück
	<code>LMSGetDiagnostic(parameter)</code>	gibt Beschreibung zum Fehler zurück

Kategorie	Kommando	Beschreibung
Data Transfer Funktionen für den Datenaustausch	<code>LMSGetValue(data model element)</code>	liest den Wert des übergebenen Datenelements aus der LMS-Datenbank aus und gibt diesen zurück
	<code>LMSSetValue(data model element,value)</code>	schreibt den übergebenen Wert des übergebenen Datenelements in die LMS-Datenbank
	<code>LMSCommit("")</code>	sendet zwischengespeicherte Werte an das LMS

Ein Sharable Object Content durchläuft nach seinem Start durch eine Plattform verschiedene Stadien. Zu Beginn ist das SCO noch nicht initialisiert (*not initialized*). Erst nachdem das SCO den Funktionsaufruf `LMSInitialize("")` an den API-Adapter sendet, gilt dieses als initialisiert (*initialized*). Nun können Funktionen zur Kommunikation folgen. Während dieses Initialized-Zustands kann kein weiteres SCO gestartet werden. Zum Beenden der Kommunikation wird der Aufruf `LMSFinish("")` verwendet. Dadurch erhält das SCO den Zustand *Finished*. Nun kann ein neues SCO gestartet werden. Die Abbildung 1.17 verdeutlicht diesen Ablauf.

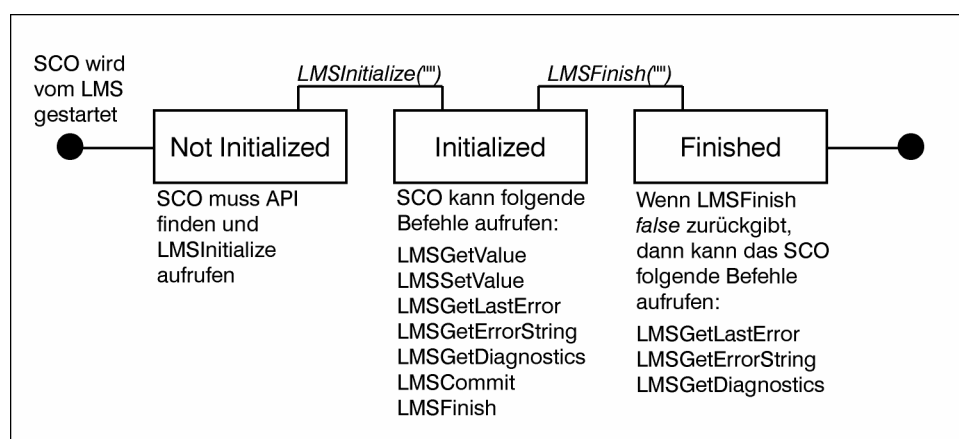


Abb. 1.17:
API-Adapter Zustandsübergänge,
nach [SCORM
2001c, S. 3-13]

Data Model:

Das SCORM Run-Time Environment Data Model liefert eine ausführliche Beschreibung der Datenelemente, die bei einer Kommunikation zwischen LMS und SCO gesendet bzw. abgefragt werden können. Dieses Datenmodell basiert in SCORM Version 1.2 alleine auf Datenelementen

des AICC CMI Data Models, welches in den bereits erwähnten AICC CMI Guidelines for Interoperability⁴ definiert ist.

Da angenommen wird, dass zukünftig auch noch andere Datenmodelle in Einsatz kommen werden, werden Elemente des CMI-Datenmodells mit dem Kürzel *cmi* am Anfang gekennzeichnet.

Ein Datenelement wird aus mehreren Teilen zusammengesetzt (z.B. *cmi.core.student_id*). Zu Beginn wird das Datenmodell definiert (z.B. *cmi*). Danach folgt die Definition der Datengruppe (z.B. *core*). An dritter Stelle wird das konkrete Datenelement definiert (z.B. *student_id*).

Beim CMI-Datenmodell wird zwischen acht verschiedenen Datengruppen unterschieden. Davon müssen mindestens die drei Datengruppen *cmi.core*, *cmi.suspend_data* und *cmi.launch_data* von einer scormkonformen Lernplattform unterstützt werden. Tabelle 1.5 gibt einen Überblick.

Tab. 1.5: CMI-Datengruppen, nach [SCORM 2001c, S. 3-20ff]

Datengruppe	Kindelemente (children)	Beschreibung
cmi.core	<i>student_id</i> , <i>student_name</i> , <i>lesson_location</i> , <i>credit</i> , <i>lesson_status</i> , <i>entry_score</i> , <i>total_time</i> , <i>lesson_mode</i> , <i>exit</i> , <i>session_time</i>	allgemeine Informationen zum Lernen und zum Lernablauf; mandatory
cmi.suspend_data		eindeutige Informationen, die benötigt werden, wenn ein Lernender zu einem späteren Zeitpunkt zum SCO zurückkehrt; mandatory
cmi.launch_data		eindeutige Informationen, die bei jedem Start eines SCOs benötigt werden; mandatory
cmi.comments cmi.comments_from_lms		ermöglicht das Sammeln und Setzen von Kommentaren über ein SCO
cmi.objectives	<i>id</i> , <i>score</i> , <i>status</i>	ermöglicht die Verfolgung der Leistung von Lernenden bezüglich bestimmter Lernziele
cmi.student_data	<i>mastery_score</i> , <i>max_time_allowed</i> , <i>time_limit_action</i>	ermöglicht die Anpassung eines SCOs an die Leistung eines Lernenden

4. Die AICC CMI Guidelines for Interoperability beinhalten sowohl ein Communication Data Model als auch ein Course Structure Data Model; das Data Model der SCORM 1.2 RTE basiert jedoch nur aus Teilen des Communication Data Models.

Datengruppe	Kindelemente (children)	Beschreibung
cmi.student_preferences	audio, language, speed, text	ermöglicht die Anpassung eines SCOs an die Präferenzen eines Lernenden
cmi.interactions	id, objectives, time, type, correct_response, weighting, student_response, result, latency	ermöglicht die Aufzeichnung von Interaktionsvorgängen eines Lernenden

Der Zugriff auf die CMI-Datenelemente geschieht vom Lernobjekt (SCO) aus durch die API-Kommandos `GetValue()` und/oder `SetValue()`.

2 Die Lernplattform des Bildungsportals Sachsen

2.1 Das Bildungsportal Sachsen

Das Bildungsportal Sachsen stellt ein gemeinsames Internetportal der sächsischen Hochschulen dar, welches über Angebote zur wissenschaftlichen Aus- und Weiterbildung informiert und gleichzeitig als Plattform für die Online-Nutzung virtueller Bildungsangebote fungiert. [Bildungsportal 2003, S. 7]

Zur technischen Realisierung der virtuellen Lernplattform nutzt das Bildungsportal das kommerzielle Learning Management System der Firma Saba, welches im Abschnitt 2.2 beschrieben wird. Momentan wird eine neue Lernplattform erprobt, die bereits ab dem Wintersemester 2005/06 parallel zu Saba eingesetzt wird und ab dem Sommersemester 2006 alleinige Lernplattform des Bildungsportals darstellen soll. Dabei handelt es sich um das an der Universität Zürich entwickelte Open Source Produkt OLAT (Online Learning And Training) [OLAT]. Dieses wird im Abschnitt 2.2.1 neben einer ausführlichen Beschreibung des LMS von Saba kurz vorgestellt.

2.1.1 Entstehungsgeschichte und Förderung

Als sich die sächsischen Hochschulen im Jahr 2001 auf Initiative des Sächsischen Staatsministeriums für Wissenschaft und Kunst zu dem Verbundprojekt *Bildungsportal Sachsen* entschieden haben, wurden in einigen anderen Bundesländern bereits ähnliche Kooperationsmodelle geplant oder waren bereits gestartet.¹

Seit Ende der neunziger Jahre sind durch Bund und Länder erhebliche finanzielle Mittel in die Entwicklung und die Erprobung computergestützter multimedialer Lehr- und Lernangebote geflossen. Dies geschah vorrangig mit dem Ziel, die Wettbewerbsfähigkeit der Hochschulen im

1. Weitere länderspezifische Projekte sind z.B.: Virtuelle Hochschule Baden-Württemberg (seit 1998), Virtuelle Hochschule Bayern (seit 2000), Multimedia Hochschulservice Berlin GmbH (seit 2000), E-Learning Academic Network Niedersachsen (seit 2001), Learning Lab Lower Saxony (seit 2001), Universitätsverbund MultiMedia Nordrhein-Westfalen (seit 1997), Virtueller Campus Rheinland-Pfalz (seit 2000), Virtuelle Saar Universität (seit 1999)

internationalen und auch im nationalen Vergleich zu stärken. So stellt auch die Initiierung des Bildungsportals Sachsen einen wichtigen Schritt zur Verbesserung der Wettbewerbsfähigkeit der sächsischen Hochschulen dar. [Bildungsportal 2003, S. 3ff]

Das Bildungsportal Sachsen wird als zweiphasiges Projekt im Rahmen des Hochschul- und Wissenschaftsprogramms durch das Sächsische Staatsministerium für Kunst und Wissenschaft (SMWK) gefördert.

Die Schwerpunkte während der ersten Phase (2001 bis 2003) waren zum einen der Aufbau und der prototypische Betrieb des *Internetportals* zum anderen die Entwicklung eines *regionalen Betreuungssystems* zur Unterstützung der Hochschulen bei der Erstellung und Nutzung der Online-Angebote. Gegen Ende dieser Projektphase wurde das Bildungsportal Sachsen in Betrieb genommen und somit die allgemeine Nutzung der Angebote möglich.

Die zweiten Projektphase (2004 bis 2006) setzt an den Ergebnissen der ersten an und setzt ihren Schwerpunkt auf *Kompetenzentwicklung und Nachhaltigkeit*. So sollen bis Ende 2006 Bedingungen geschaffen werden, die an den sächsischen Hochschulen nachhaltig die Integration von neuen Medien in der wissenschaftlichen Aus- und Weiterbildung fördern.

Zur Erfüllung der Zielsetzungen arbeiten die Trägerhochschulen *Technische Universität (TU) Chemnitz, Technische Universität (TU) Dresden, Universität Leipzig* und die *Hochschule Mittweida (FH)* eng zusammen. Diese stellen auch die Mitglieder des Projektrats, der das Bildungsportal Sachsen leitet. [Bildungsportal 2004, S. 1]

2.1.2 Ziele und Zielgruppen

Mit dem Schwerpunkt *Kompetenzentwicklung und Nachhaltigkeit* besteht die Arbeit des Verbundprojektes momentan vorrangig neben der Bereitstellung von Online-Angeboten auch in der Kompetenzentwicklung bezüglich der Erstellung und der Nutzung von E-Learning-Angeboten und der Schaffung von Strukturen, die die Einbindung von E-Learning in den Hochschulalltag fördern.

Im Speziellen lassen sich die Ziele des Bildungsportals Sachsen wie folgt definieren:

- Aufbau einer E-Learning Infrastruktur durch die Zusammenarbeit der Hochschulen mit dem Ziel einer nachhaltigen Aufbau- und Ablauforganisation
- Entwicklung von orts- und zeitunabhängigen Studien- und Weiterbildungsangeboten
- Entwicklung geeigneter pädagogischer und didaktischer Konzepte
- Förderung von Medienkompetenz als Schlüsselqualifikation

Die Arbeit des Bildungsportals Sachsen als Verbund verfolgt zum einen das Ziel, auch den kleineren Hochschulen des Landes Sachsen durch die Nutzung eines gemeinsamen Portals die Möglichkeit zu geben, E-Learning Angebote zu nutzen und zu veröffentlichen. Zum anderen liegt das Ziel dieser Verbundgemeinschaft auch darin, eine Kostenminimierung durch die gemeinsame Nutzung und Entwicklung von Angeboten zu erreichen.

So unterschiedlich wie die Zielsetzungen sind auch die Zielgruppen des Bildungsportals. Dieses richtet sich zum einen an die Nutzer von Online-Lernangeboten. Dies sind *Studenten* der sächsischen Hochschulen, die E-Learning begleitend oder weiterführend zu Präsenzveranstaltungen nutzen, und *Weiterbildungsnachfrager*, die sich berufsbegleitend weiterqualifizieren möchten. Zum anderen unterstützt das Bildungsportal *Lehrende*, die bereits entwickelte E-Learning Angebote in Präsenzveranstaltungen integrieren möchten, oder auch eigene Online-Angebote erstellen möchten.

Zudem steht das Bildungsportal für interessierte Schüler und Lehrer und auch für Kooperationspartner aus dem Hochschulbereich und aus der Wirtschaft zum Austausch und zur Ergänzung von Bildungsangeboten zur Verfügung. [Kleimann/Wannemacher 2004, S. 54]

2.1.3 Studienangebot

Derzeit stehen im Bildungsportal Sachsen über 300 E-Learning Angebote zur Verfügung, die von Studierenden der sächsischen Hochschulen kostenfrei genutzt werden können. Weiterbildungsangebote für Interessierten außerhalb der Hochschulen sind dagegen gebührenpflichtig. [Bildungsportal 2005b]

Die Produktion der E-Learning Angebote wird durch die Sächsischen Hochschulen mit der Unterstützung von hochschulischen Kompetenzzentren wie z.B. dem Media Design Center der TU Dresden oder auch dem WI-Medialab der TU Chemnitz selbst durchgeführt [Kleinmann/Wannemacher 2004, S. 56].

Um die inhaltliche Ausgestaltung des Bildungsportals Sachsen voranzutreiben, wurden vom Sächsischen Staatsministerium für Wissenschaft und Kunst (SMWK) in den Jahren 2001, 2003 und 2004 drei Ausschreibungen zur Entwicklung von multimedialen Lehr- und Lernangeboten an den Hochschulen Sachsens initiiert. Ein großer Teil der bisher entstandenen Angebote im Bildungsportal wurde im Rahmen dieser Ausschreibungen gefördert. Bis Ende des Jahres 2005 werden dies voraussichtlich über 120 Angebote sein. [Bildungsportal 2004, S. 2]

Folgende Darstellung zeigt die Verteilung dieser Angebote auf verschiedene Wissenschaftsbereiche.

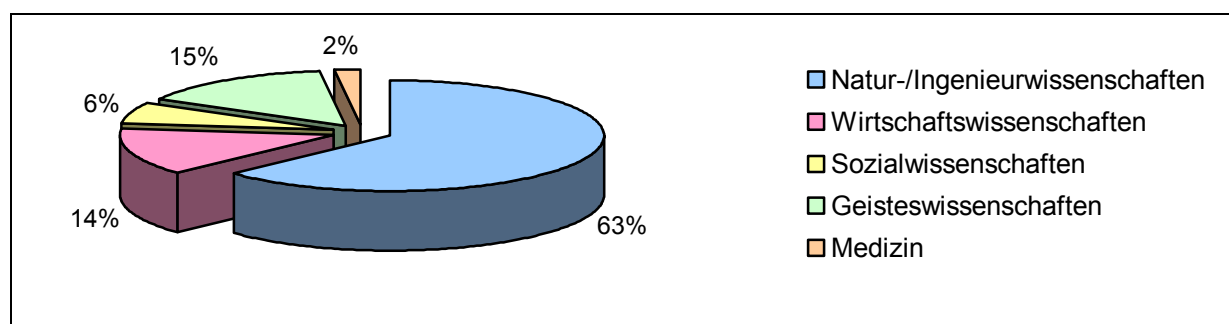


Abb. 2.1: Ausschreibungen 2001 – 2004, nach [Bildungsportal 2004, S. 2]

Es stehen jedoch nicht alle im Bildungsportal Sachsen vorhandenen Lernangebote allen Studenten bzw. Weiterbildungsnachfrager in gleichem Maße zur Verfügung.

Die Lernplattform, des Bildungsportal Sachsen ermöglicht es, den Nutzerkreis für ein Angebot z.B. durch ein Passwort einzuschränken. Dadurch besteht für Hochschullehrer die Möglichkeit, ein E-Learning Angebot beispielsweise nur den Studenten ihrer Lehrveranstaltung zur Verfügung zu stellen (vgl. 2.3.1).

2.2 Die Lernplattform der Firma Saba

Saba gehört zu den führenden Unternehmen im Bereich *Human Capital Management* (HCM), es entwickelt Systeme zur Schulung und zur Lernerfolgskontrolle, die weltweit von den verschiedensten Organisationen und Firmen genutzt werden.

Im Jahr 1997 wurde die Firma Saba gegründet und brachte ihr erstes Learning Management System auf den Markt. Seither sind viele zusätzliche Komponenten entwickelt worden, die zusammen eine Komplettlösung für das Management von Personal und Know-how in einem Unternehmen bieten.

Die Hauptzielgruppe von Saba liegt in der freien Wirtschaft, nicht im Hochschulbereich. Dennoch hat sich das Land Sachsen im Jahr 2001 im Rahmen einer beschränkten Ausschreibung nach eingehender Evaluierung für das damals aktuelle Produkt von Saba, *Saba3 Release4*, als LMS für das Bildungsportal entschieden [Kleimann/Wannemacher 2004, S. 53f]. Dieses wird auch heute noch vom Bildungsportal Sachsen genutzt.

Saba3 Release4 soll im Bildungsportal Sachsen ab dem Sommersemester 2006 vollständig durch das LMS OLAT ersetzt werden (vgl. 2.1). OLAT ist im Vergleich zu Saba deutlich besser an die Anforderungen für den Einsatz im Hochschulbereich angepasst.

Die folgenden Abschnitte befassen sich jedoch vorrangig mit der Lernplattform *Saba3 Release4*, welche zum Zeitpunkt der Erstellung und der Integration des Lernmoduls ‚Typografie praktisch!‘ in das BPS die aktuelle und alleinige Lernplattform des Bildungsportals darstellte.

2.2.1 Bestandteile der Saba Lernplattform

Saba3 Release4 ist im Jahr 2001 auf den Markt gekommen und setzt sich aus den beiden Hauptbestandteilen *Saba Learning* und *Saba Content* zur Veröffentlichung und Verwaltung von Lerninhalten zusammen. Zudem sind integrierte Funktionen zur Administration, Evaluation und Kommunikation vorhanden. Das aktuelle Produkt von Saba *Saba Enterprise 2005* stellt hingegen eine Gesamtlösung für HCM von Unternehmen dar und besteht aus mehreren kombinierbaren Modulen.

Das Bildungsportal hat sich dazu entschieden, kein Update für die aktuelle Lösung von Saba (Saba Enterprise 2005) zu erwerben, sondern Saba durch das frei nutzbare LMS OLAT zu ersetzen, welches, wie zuvor erwähnt, deutlich besser an die Bedürfnisse der Hochschullehrer und Studenten angepasst ist.

Um jedoch einen Einblick in den Umfang und die Möglichkeiten aktueller Learning Management Systeme zu geben, wird an dieser Stelle zunächst *Saba Enterprise 2005* vorgestellt. Nachfolgend an die Beschreibung der Bestandteile von Saba3 Release4 wird ein kleiner Einblick in das LMS OLAT gegeben.

Saba Enterprise 2005:

Saba Enterprise 2005 umfasst folgende Hauptprodukte (Module), die unabhängig voneinander lizenziert und genutzt werden können:

- *Saba Learning*: zur gezielten Förderung von Fähigkeiten und Wissen der Mitarbeiter
- *Saba Performance*: bietet einen systematischen und kontinuierlichen Prozess zur Abstimmung der täglichen Leistungen der Mitarbeiter mit dem Gesamtziel eines Unternehmens; wird auch zur Mitarbeitermotivation eingesetzt
- *Saba Talent*: unterstützt Unternehmen systematisch bei internen Stellenbesetzungen; dazu werden die Mitarbeiter je nach Fähigkeiten und Kenntnissen gezielt gefördert und geschult; z.B. werden so potenzielle Führungskräfte auf ihre mögliche zukünftige Position vorbereitet

Dabei dient die *Saba Foundation* als Basis für alle drei Produkte und beinhaltet u.a. Funktionen zum Managen der Nutzer-Profile oder auch zum Erstellen von Lern-/Lehrplänen. *Saba Collaboration* mit Funktionen für Kommunikation und Kollaboration stellt ein optionales Modul dar, welches für alle drei Hauptprodukte lizenziert und genutzt werden kann. Die Abbildung 2.2 stellt die drei Hauptprodukte mit ihren optionalen Modulen und der Saba Foundation vor.

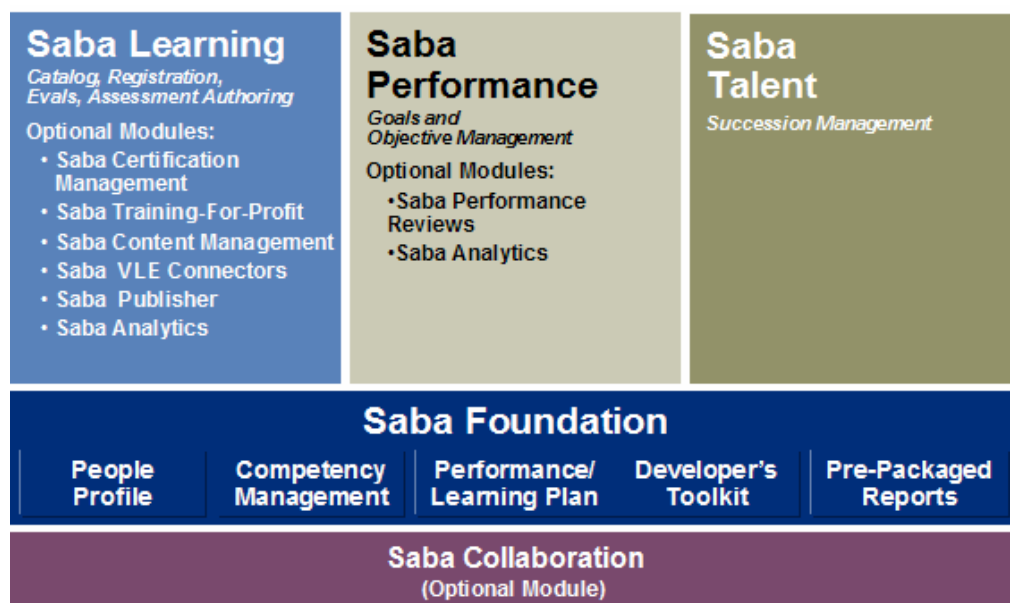


Abb. 2.2:
Saba Module
Saba Enterprise
2005
[Saba 2005]

Saba Learning, das eigentliche LMS, beinhaltet Funktionen für einen Auswahlkatalog, für die Nutzerregistrierung, zur Lernerevaluation und besitzt ein integriertes Autorenwerkzeug. Optional können folgende Module lizenziert und integriert werden:

- *Saba Certification Management*: Funktionen zum Managen von Zertifizierungen
- *Saba Learning Commerce*: E-Commerce-Funktionen (z.B. zur Abrechnung der Kursgebühren)
- *Saba Content Management*: ein Content Management System zum Verwalten von Lernressourcen
- *Saba Virtual Learning Environment (VLE) Connectors*: unterstützt die Integration von externen Virtual Classroom Anwendungen (wie z.B. Centra, WebEx, Microsoft Live Meeting)
- *Saba Publisher*: Autorenwerkzeug zur Erstellung von Lerninhalten und Tests
- *Saba Analytics*: bietet verschiedene Funktionalitäten zur Unterstützung des Lernmanagements; z.B. Messung und Analyse der Effektivität der Lernangebote
- *Saba Collaboration*: bietet Funktionalitäten für kollaboratives Arbeiten wie z.B. Instant Messaging

Saba3 Release4:

Saba3 Release4 besitzt keinen modularen Aufbau wie unter Saba Enterprise 2005 beschrieben, sondern stellt ein Produkt mit den beiden folgenden Hauptbestandteilen dar:

- *Saba Learning, Enterprise Edition*, einem internetbasierten Learning Management System (LMS)
- *Saba Content*, einem Learning Content Management System (CMS)

Saba3 Release4 stellt über diese beiden Bestandteile neben Funktionen zur *Veröffentlichung* (vgl. 2.2.2) und *Administration* von Lerninhalten auch *Kommunikationsfunktionen* (vgl. 2.2.3) zur Verfügung. Zudem lassen sich über ein integriertes *Autorenwerkzeug* Lerninhalten und Tests erstellen (vgl. 2.2.4). Als *Evaluations- und Bewertungshilfen* besitzt Saba Funktionen zur Protokollierung des Lernfortschritts und zur automatischen Bewertung von Tests. Dabei sind einige der optionalen Module der aktuellen Version 2005 unter Saba3 Release4 standardmäßig integriert, wie z.B. Saba Content oder der VLE Connector (vgl. 2.2.3).

Die Funktionalität von Saba3 Release4 lässt sich über folgende zusätzliche Produkte erweitern:

- *Saba Live!* und *Saba Dialog* bieten zusätzliche Funktionen im Bereich Kommunikation und Kollaboration (vgl. 2.2.3)
- *Saba Publisher* stellt ein externes Autorenwerkzeug dar und bietet zusätzliche Möglichkeiten bei der Erstellung von Lernmaterialien (vgl. 2.2.4)

Die Lizenzgebühren für Saba3 Release4 richten sich nach Anzahl der Nutzer und der Lizenzierung optionaler Produkte, wie z.B. Saba Live! oder Saba Publisher. Das Bildungsportal Sachsen besitzt zusätzliche Lizenzen, die zur Nutzung des Saba Publishers berechtigen. Die Kommunikations- und Kollaborationswerkzeuge SabaLive! und Saba Dialog werden dagegen vom Bildungsportal nicht eingesetzt.

Zur Bewältigung der anfallenden Aufgaben, die für eine reibungslose Funktion des gesamten Systems (LMS und CMS) zu erfüllen sind, besitzt Saba eine Dreiteilung in die Komponenten Saba Web, Saba Client und Saba Server.

Saba Web ist das Internetportal von Saba, über welches zahlreiche Vorgänge gesteuert werden können. Dies beinhaltet unter anderem den Import der Lerninhalte in die Inhaltsdatenbank, die Nutzerregistrierung, die Buchung und die Nutzung von Lernangeboten, sowie die Steuerung der Kommunikations- und Kollaborationsmöglichkeiten.

Die Komponente *Saba Client* verfügt über eine separate Nutzerschnittstelle und wird für verschiedene administrative Aufgaben eingesetzt, z.B. zur Systemadministration oder auch zum Verkaufsmanagement für gebührenpflichtige Lernangebote.

Saba Server dient hingegen der Speicherung und der Verwaltung der Daten, die von den Komponenten Web und Client verwendet werden. Dazu stellt Saba Server verschiedene Mechanismen, Tabellen und Datenelemente zur Verfügung.

OLAT:

Die Funktionalitäten von OLAT lassen sich folgenden Bereichen zuordnen [Hahn/Kosler 2005, S. 4ff]:

- Lernressourcen: Dieser Bereich dient der Verwaltung aller Lernressourcen (Kurse, Content Package Inhalte², Tests und Fragebögen). Lernressourcen können unter OLAT extern erstellt und importiert werden. Kurse, Tests und Fragebögen lassen sich auch intern mit dem *OLAT Course Editor* die durch die Verwendung von 14 verschiedenen Kursbaustein-Typen einfach, schnell und flexibel erstellen. Dabei handelt es sich u.a. um folgende Baustein-Typen [OLAT]:
 - *Local Content* (lokal gespeicherter Inhalt durch Hochladen von Dateien oder durch die Nutzung des im LMS integrierten WYSIWYG-Editors)
 - *External Content* (extern gespeicherter Inhalt, z.B. durch Zugriff auf externe Datenbanken)
 - *IMS QTI Test, Selftest, Questionnaire* (Autorenwerkzeug zur Erstellung von Tests, Selbsttests und Fragebogen, basierend auf der *IMS Question & Test Interoperability Specification* (IMS QTI))
 - *Discussion Forum, Chat* (Kommunikationsmöglichkeiten in Form von Diskussionsforum und Chat)
 - *File Share Folder* (Bereitstellung von Dateien in einem öffentlichen Ordner)
- Bookmarks: Lesezeichen können unter OLAT für Lernressourcen verwendet werden und ermöglichen direkten Zugriff auf den entsprechenden Inhalt.

2. OLAT unterstützt bis zur Version 4.0 IMS Inhaltspakete; ab 4.0 zusätzlich SCORM 1.2 Inhaltspakete

- Buddygroups: Dies sind Zusammenschlüsse von OLAT-Benutzern außerhalb des Kurskontextes, die unter Lernern als ‚virtuelle‘ Lerngruppen oder unter Autoren als Entwicklungsgruppen genutzt werden können. Buddygroups können von jedem OLAT-Benutzer angelegt werden.
- Verwaltung von Nutzerdaten: Jeder Nutzer besitzt die Möglichkeit, sich ein eigenes Profil anzulegen. Neben der Änderung des Passwortes und der Auswahl einer Sprache (z.B. Englisch oder Deutsch) können z.B. auch Dateien in einem eigenen Ordner abgelegt werden.
- Administration: verschiedenen administrative Funktionen, wie z.B. zur Vergabe von Nutzerrechten oder Funktionen zur Systemadministration (z.B. Verwaltung des verfügbaren Speicherplatzes)

Das System OLAT 3.2, welches in der ersten Hälfte des Jahres 2005 erschienen ist, wird vom Bildungsportal Sachsen bereits im Wintersemester 2005/06 zusätzlich zu Saba3 Release4 eingesetzt. OLAT 3.2 unterstützt die *IMS Content Packaging Specification* (vgl. 1.3.2) sowie die *IMS Question & Test Interoperability Specification*. SCORM wird jedoch von dieser Version nicht unterstützt. Erst mit der Version 4.0, die in der zweiten Hälfte des Jahres 2005 erscheinen soll, wird SCORM 1.2 unterstützt. Das Bildungsportal wird voraussichtlich ab dem Sommersemester 2006 OLAT 3.2 durch 4.0 ersetzen und dieses System als alleiniges LMS einsetzen.

Die folgenden Ausführungen zu Saba3 Release4 beschreiben verschiedene Funktionsbereiche, die für die Entwicklung von Lerninhalten, die unter Saba zum Einsatz kommen, von Bedeutung sein können. Dazu zählen sowohl die unterstützten Dateiformate und E-Learning Standards als auch die zur Verfügung stehenden Autorenwerkzeuge zur Erstellung und Bearbeitung von Lerninhalten bzw. Lernkontrollen.

Zudem werden die unter Saba3 Release4 zur Verfügung stehenden bzw. integrierbaren Kommunikationswerkzeuge vorgestellt.

2.2.2 Veröffentlichung von Lerninhalten

Unter Saba und somit auch im Bildungsportal Sachsen werden Lerninhalte zunächst in einer Inhaltsdatenbank (*repository*), die physische Dateien oder Links zu physischen Dateien enthält,

abgelegt. Sobald sich ein Lerninhalt in dieser Datenbank befindet, spricht man von einem Lernobjekt (learning object). Diese können durch Metadaten beschrieben und im Angebotskatalog (*Saba Learning Catalog*), der für Lernende sichtbar ist, veröffentlicht werden kann. Tabelle 2.1 zeigt die Dateiformate, die unter Saba3 Release4 importiert und veröffentlicht werden können.

Tab. 2.1: Unterstützte Dateiformate, nach [Saba 2001, S. 4-2ff]

Dateiformat	Kurzbeschreibung
AICC	Inhalt, der konform mit der AICC CMI 001 Spezifikation ist; dieser kann als URL oder im AICC Course Structure Format importiert werden
SCORM-CSF (SCORM 1.1)	Inhalt im Content Structure Format der SCORM Version 1.1; die CSF-Datei stellt eine Manifestdatei zur Beschreibung eines SCORM-Pakets dar; diese kann direkt oder als URL (Link auf eine CSF-Datei) importiert werden
SCORM Content Package (SCORM 1.2)	SCORM 1.2 Content Package; dieses muss als ZIP-Datei importiert werden
IMS Content Package	Inhalt, der konform mit der IMS Content Packaging Version 1.1.1 Spezifikation ist; dieser muss im ZIP-Format importiert werden
Learning Content Format	Inhalt im Saba Learning Content Format (LCF); die LCF-Datei stellt eine Manifestdatei zur Beschreibung eines Inhaltspakets dar; diese kann direkt oder als URL (Link auf eine LCF-Datei) importiert werden; das LCF ist Teil der Saba Universal Learning Format Version 1.0
File (Datei)	einzelne Dateien (PowerPoint-Format, Videoformate, Bildformate, etc.)
Directory (Verzeichnis)	Verzeichnisse (z.B. Dateien für eine HTML-Seite) können im ZIP-Format importiert werden
URL	Links, die auf Lerninhalte im Internet verweisen
Offline Content	Inhalt, der für Offline-Lernanwendungen konzipiert wurde, kann über den Saba Publisher bearbeitet und importiert werden
Netg / SkillSoft / SmartForce	Inhalte der Anbieter Netg, SkillSoft, SmartForce; diese können als URL importiert werden

Zudem lässt sich Lerninhalt direkt in Saba mit dem integrierten Autorenwerkzeug Content Builder (vgl. 2.2.4) erstellen und in der Inhaltsdatenbank abspeichern.

Bei der Veröffentlichung von Lerninhalten im Angebotskatalog wird zwischen drei Möglichkeiten unterschieden. Der Inhalt kann als *Product*, als *Course* oder als *Class* veröffentlicht werden. Angepasst an den Einsatz im Hochschulbereich lauten die entsprechenden Bezeichnungen im Bildungsportal *Selbstlernmodul* für *Product*, *Fach* für *Course*, *Lehrveranstaltung (LV)* für *Class* (vgl. 2.3.1).

Selbstlernmodule (Products) sind geschlossene und unabhängige Lerneinheiten. Sie können jederzeit vom Lernenden „selbst gesteuert“ genutzt werden und sind nicht an einen bestimmten Stundenplan bzw. Zeitplan gebunden. Ein Selbstlernmodul stellt jeweils einen Lerninhalt aus der Inhaltsdatenbank dar und kann jedes importierbare Dateiformat besitzen. Selbstlernmodule stehen den Lernenden im Angebotskatalog zur Einschreibung zur Verfügung.

Fächer (Courses) dienen als Vorlagen und als Rahmen für die Erstellung von Lehrveranstaltungen. Man muss ihnen keinen Lerninhalt aus der Inhaltsdatenbank zuweisen, es besteht aber die Möglichkeit, dies zu tun. Dabei kann es sich um jedes importierbare Dateiformat mit der Ausnahme von AICC- und Offline-Inhalten handeln. Die zugewiesenen Lerninhalte werden an die abgeleiteten Klassen weiter vererbt. Fächer können von Lernenden nicht direkt gebucht werden. Eine Einschreibung ist nur über eine dem Fach zugewiesene Lehrveranstaltung möglich.

Lehrveranstaltungen (Classes) stellen, wie gesagt, Instanzen von Fächern dar. Das heißt, sie besitzen die Eigenschaften des übergeordneten Fachs (z.B. Lerninhalte). Auch bei Lehrveranstaltungen kann es sich mit der Ausnahme von AICC- und Offline-Lerninhalten um beliebigen Inhalt aus der Inhaltsdatenbank handeln.

Zur Veröffentlichung im Angebotskatalog muss einer Lehrveranstaltung mindestens eine Sitzung (*Session*) zugewiesen werden, die für einen bestimmten Zeitpunkt und eine bestimmte Dauer festgelegt ist. Einem Fach lassen sich beliebig viele Lehrveranstaltungen zuweisen. [Saba 2001b, S. 5-5ff]

Beispielsweise kann dem Fach ‚Englisch für Informatiker‘ eine Lehrveranstaltung ‚Anfänger‘ und eine Lehrveranstaltung ‚Fortgeschrittene‘ zugewiesen werden. Dem Fach könnte z.B. als Lerninhalt ein Wörterbuch zugewiesen werden. Dieses kann sowohl von den Anfängern als auch von den Fortgeschrittenen genutzt werden. Dennoch besteht die Möglichkeit den beiden Lehrveranstaltungen zusätzlich unterschiedlichen Lernstoff zuzuweisen.

Abbildung 2.3 verdeutlicht den Zusammenhang zwischen Inhaltsdatenbank, Angebotskatalog, Fach (Course), Lehrveranstaltung (Class) und Selbstlernmodul (Product).

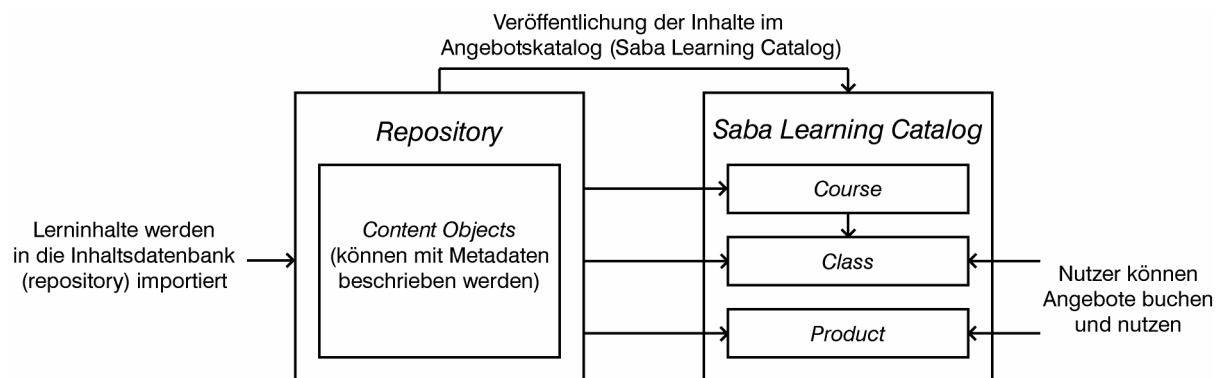


Abb. 2.3: Zusammenhang zwischen Inhaltsdatenbank und Angebotskatalog unter Saba

2.2.3 Funktionen im Bereich der Kommunikation

Wie bereits im ersten Kapitel beschrieben, stellen Kommunikationswerkzeuge einen wichtigen Funktionsbereich von Lernplattformen dar (vgl. 1.2.1). Die Kommunikation mit dem Lehrer oder mit anderen Lernenden erlaubt direkten Einfluss auf den Lernprozess und grenzt somit auch das Lernen über computergestützte Offline-Anwendungen vom E-Learning über das Internet oder über andere Kommunikationsnetzwerke ab.

Das Bildungsportal Sachsen stellt den Lernenden Kommunikationswerkzeuge des frei verfügbaren Systems ACollab der Universität Toronto zur Verfügung (vgl. 2.3.3). Die in diesem Abschnitt beschriebenen Kommunikationsmöglichkeiten unter Saba werden vom Bildungsportal zur Zeit nicht genutzt.

Der Zugang zu den Kommunikationsmöglichkeiten erfolgt unter Saba entweder über die Einschreibung in ein Lernangebot oder über die Zugehörigkeit zu einer bestimmten eine Gruppe (Community³), wie z.B. Student der Medieninformatik. Folgende Kommunikationswerkzeuge stehen unter Saba³ Release⁴ zur Verfügung:

Diskussionsforum (Discussion):

Diskussionsforen können unter Saba sowohl für Selbstlernmodule als auch für Lehrveranstaltungen genutzt werden und stehen den Lernenden nach der Einschreibung in das entsprechende Lernangebot zur Verfügung. Die Themen der Foren orientieren sich an den Inhalten der jeweili-

3. Communities stellen unter Saba Zugehörigkeitsgruppen dar, die explizit erstellt werden müssen.

gen Lernangebote. Auch für Communities gibt es Diskussionsforen, in welchen gruppenspezifischen Themen behandelt werden können.

Chat:

Chats lassen sich unter Saba für Selbstlernmodule, Lehrveranstaltungen und Communities nutzen. Über einen Instant Messaging Dienst besteht hier zusätzlich die Möglichkeit, Nachrichten sofort an bestimmte Teilnehmer zu senden.

Virtual Classroom:

Der Virtual Classroom stellt unter Saba eine spezielle Umgebung zur Präsentation von Lerninhalten dar, die sowohl für asynchrones als auch für synchrones Lernen verwendet werden kann.

Neben einem Inhaltsverzeichnis und Funktionen zur Navigation innerhalb des Lerninhalts besitzt der Virtual Classroom auch eine integrierte Darstellung der Lernergebnisse. Zudem ermöglicht die Umgebung je nach Art des Inhalts direkten Zugriff auf Chat, Diskussionsforen und auch auf Interaktions- und Bewertungswerkzeuge.

Beim synchronen Lernen kann der Lehrer beispielsweise jederzeit Nachrichten an die Teilnehmer schicken oder diese zur Bewertung ihres des Lernfortschritts befragen. Zudem besitzt der Lehrer die Kontrolle darüber, wann ein bestimmter Lerninhalt den Lernenden präsentiert wird. Weitere Funktionen, die häufig mit virtuellen Klassenzimmern in Zusammenhang gebracht werden, wie z.B. Application-Sharing, werden vom Saba Virtual Classroom nicht unterstützt. [Saba 2001b, S. 7-2]

Im Virtual Classroom wird zwischen zwei Modi für asynchrones und synchrones Lernen unterschieden [Saba 2001b, S. 7-2ff]:

- *self-paced mode (asynchrones Lernen)*: Dieser Modus beschreibt ein selbst gesteuertes Lernen, wobei der Lernende jederzeit Zugang zum Lerninhalt besitzt und diesen nach eigener Geschwindigkeit abarbeiten kann. Typisch für diesen Modus sind Lerninhalte, die als *Products* angeboten werden. Kommunikationswerkzeuge wie Chat und Diskussionsforum sind beim asynchronen Lernen nicht über die Umgebung des Virtual Classroom verfügbar. Diese dient in diesem Fall lediglich der Präsentation von Lerninhalten und –ergebnissen.
- *instructor-led mode (synchrones Lernen)*: In diesem Modus wird nach einem festen Stunden- bzw. Zeitplan gelernt. Alle Lernenden beschäftigen sich zum selben Zeitpunkt mit demselben

Lernmaterial, ein Lehrer (*instructor*) kann Anweisungen geben oder z.B. auch die Arbeitsgeschwindigkeit bestimmen. Typisch für diesen Modus sind Lerninhalte, die als *Classes* angeboten werden. Beim synchronen Lernen verfügt der Virtual Classroom über zwei verschiedenen Darstellungsumgebungen, dem *Learner Interface* für die Lernenden und dem *Instructor Interface* für den Lehrenden. Diese beinhalten Interaktionsmöglichkeiten wie Chat, Discussion sowie zur Befragung und Bewertung.

Zur Erweiterung dieser Virtual Classroom Funktionen beinhaltet Saba einen *VLE (Virtual Learning Environment) Connector*. Dieser stellt eine API-Spezifikation dar, mit der man Virtual Classrooms Produkte wie z.B. *Placeware*, *Centra*, *Interwise* oder auch *SabaLive!* anbinden kann.

SabaLive! ist ein separates Virtual Classroom Produkt von Saba, welches sich problemlos in Saba3 Release4 integrieren lässt und über zusätzliche Lizenzkosten erworben werden kann. Es bietet zusätzliche Virtual Classroom Funktionen wie z.B. Application-Sharing, Web-Tours und die Anleitung durch mehrere Instruktoren (Instructors).

SabaDialog:

Saba Dialog ist ein System für Nachrichten- und Informationsverwaltung und kann über zusätzliche Lizenzgebühren unter Saba3 Release4 verwendet werden. Es ermöglicht allen SabaNutzern, unabhängig von Lerninhalten und Communities, miteinander mittels Chat und Diskussionsforum in Verbindung zu treten. Dazu listet SabaDialog alle Online-Teilnehmer auf.

Zudem verwaltet das System eine Fragen-und-Antworten-Datenbank, die nach Schlagwörtern durchsucht werden kann. Die Nutzer haben die Möglichkeit, Antworten zu bewerten. [Saba 2001a, S. 20-11ff]

2.2.4 Autorenwerkzeuge

Autorenwerkzeuge zur Erstellung von Lerninhalten, Übungen und Tests bilden einen weiteren Funktionsbereich von Learning Management Systemen (vgl. 1.2.1). Saba verfügt sowohl über ein integriertes als auch über ein externes Autorenwerkzeug.

Beide Werkzeuge stehen über das Bildungsportal Sachsen zur Verfügung. Das integrierte Werkzeug *Content Builder* kann direkt über das Internetportal genutzt werden, der externe *Saba Publisher* ist auf einem Rechner im Media Design Center Dresden installiert und kann von dort aus genutzt werden.

Content Builder:

Der Content Builder ist ein in Saba integriertes Autorenwerkzeug, welches über das Internetportal des LMS genutzt werden kann. Der Content Builder kann zur Erstellung von einfach gestalteten Lerninhalten (*learning contents*) und Tests (*questionnaires*) im HTML-Format genutzt werden.

Zur Erstellung von Inhalten steht ein einfaches Textwerkzeug zur Verfügung. Zudem besteht die Möglichkeit, Lernmaterial über die Angabe einer URL als Link zur Verfügung zu stellen, oder Dateien direkt hochzuladen. Dabei kann es sich jegliche Art von Dateien handeln, diese werden vom Internetbrowser direkt angezeigt (wie z.B. HTML-Dateien, Flash-Filme, Bilder oder Sound) oder stehen den Lernenden zum Download zur Verfügung. Die einzelnen hochgeladenen Dateien lassen sich über den Content Builder jedoch nicht kombinieren, sondern werden jeweils getrennt voneinander (z.B. als ein Lernschritt) angezeigt.

Bei der Erstellung von Fragebögen und Tests unterstützt der Content Builder die Fragetypen *Multiple Choice*, *Wahr oder Falsch* und *Freitextantworten*. [Saba 2001b, S. 6-2ff]

Bereits importierte und in der Inhaltsdatenbank abgespeicherte Lerninhalte können mit dem Content Builder bearbeitet bzw. neu zusammengestellt werden. Dies ist neben Inhalten, die mit dem Content Builder erstellt wurden, auch bei den Inhaltsformaten SCORM-CSF, SCORM Content Package, IMS Content Package und Learning Content Format möglich [Saba 2001b, S. 4-54]. Dadurch wird die Wiederverwendbarkeit von einzelnen Komponenten möglich und bleibt nicht auf gesamte Lernmodule begrenzt (vgl. 1.2.2 u. 1.2.3).

Bei SCORM-Inhalten können beispielsweise neue Lernobjekte hinzugefügt bzw. vorhandenen entfernt oder durch die Veränderung der Inhaltseigenschaften bearbeitet werden. Beim Hinzufügen neuer Lernobjekte ist es möglich, neuen Inhalt mit dem Content Builder zu Erstellen oder auch bestehenden Inhalt aus der Inhaltsdatenbank zu kopieren und einzufügen. Bei den Inhaltseigenschaften eines Lernobjekts lässt sich u.a. der Name oder auch die URL zum Inhalt verändern. Zur Verdeutlichung zeigt die Abbildung 2.4 SCORM-Inhalt, der mit dem Content

Builder bearbeitet wird. Im Navigationsfeld links wird das Inhaltsverzeichnis des Lernmoduls dargestellt. Im mittleren Fenster kann man ein Werkzeug zur Bearbeitung auswählen (z.B. Kopieren, Ausschneiden, Löschen, Einfügen). Über das Hauptfenster (rechts) lassen sich Änderungen der Inhaltseigenschaften wie vornehmen.

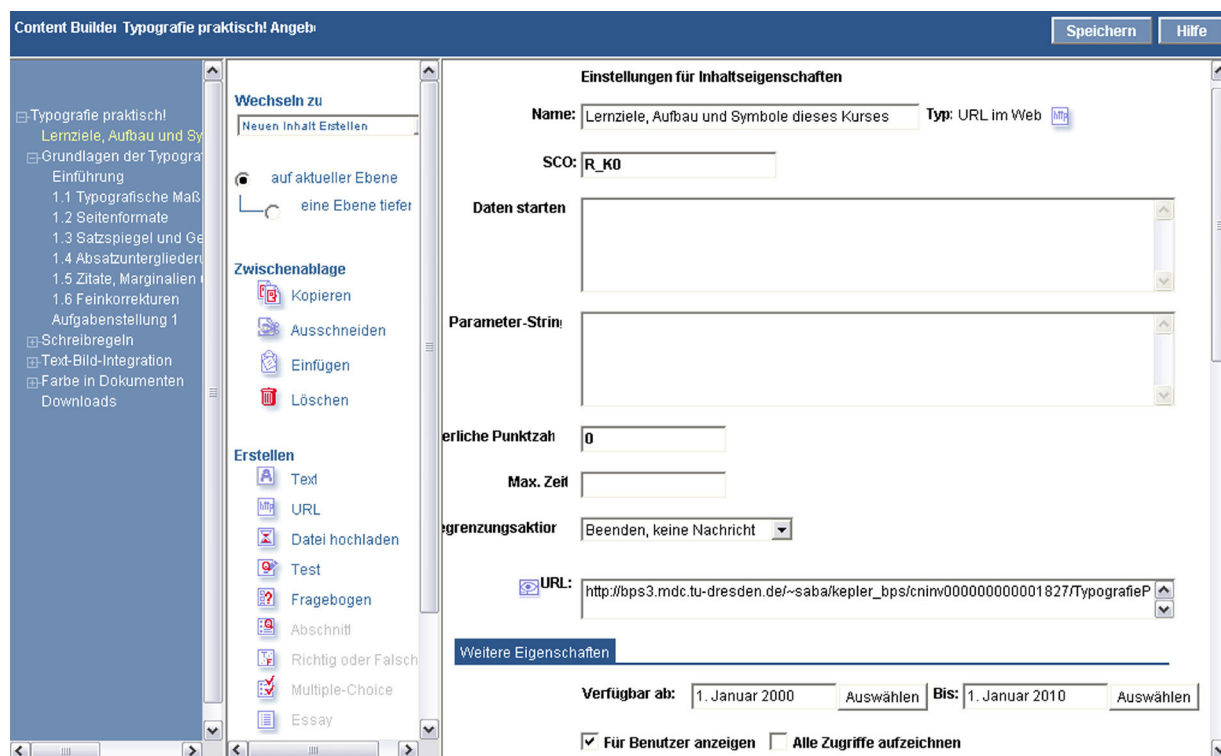


Abb. 2.4: Verwendung des Content Builders bei SCORM-Inhalt

Saba Publisher:

Saba bietet über den Saba Publisher ein externes Autorenwerkzeug an, welches über zusätzliche Lizenzkosten erworben werden kann. Der Saba Publisher ist nicht über das Internetportal verfügbar ist, sondern muss zur Nutzung auf einem lokalen Rechner installiert werden.

Der Saba Publisher ermöglicht eine einfache und schnelle Erstellung von Lerninhalten durch die Bereitstellung eines Assistenten und eine erweiterbare Vorlagen-Bibliothek.

Neben HTML- und Offline-Inhalten lassen sich mit dem Publisher auch SCORM- und AICC-kompatible Lerninhalte erstellen, dabei werden Internettechnologien wie Shockwave, Flash, HTML, Java, und JavaScript unterstützt. Die verschiedenen Datentypen (Text, Bild, Audio,

Video, Animation) und Schaltflächen lassen sich einfach (z.B. über ‚Drag and Drop‘) einfügen. Bei der Erstellung von Tests bzw. Übungen stehen folgende Fragetypen zur Verfügung [Baumgartner u.a. 2002, S. 237]:

- Multiple Choice
- Lückentext und Freitextantworten
- Wahr oder Falsch
- Zuweisungen / Verknüpfungen

Der Saba Publisher wird von der Firma Trivantis⁴ hergestellt, jedoch über Saba vertrieben und stellt eine spezielle Form des Autorenwerkzeugs *Lectora* von Trivantis dar. Die Unterschiede der beiden Produkte liegen in einzelnen, wenigen Funktionalitäten die Lectora zusätzlich in den selbst vermarkteten Produkten anbietet und in einer vorkonfigurierten Schnittstelle zum Saba System im Saba Publisher. Diese Schnittstelle ermöglicht den direkten Zugriff vom Saba Publisher aus auf die Inhaltsdatenbank des Saba Systems.

Folgende Abbildungen zeigen Ausschnitte aus der Demoversion des Programms Lectora [Trivantis].



Abb. 2.5:
Assistent (Wizard) zur Erstellung von Lerninhalten

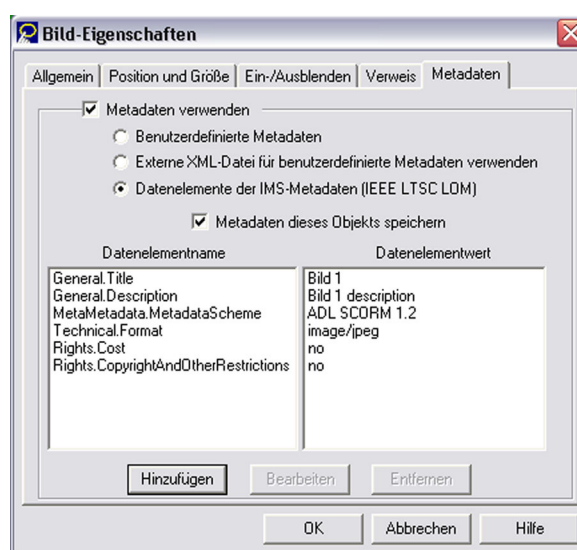


Abb. 2.6:
Fenster zur Erstellung von Metadaten

4. Die Firma Trivantis wurde im Jahr 1999 mit Hauptsitz in Ohio gegründet und hat sich auf die Herstellung von E-Learning Autorenwerkzeuge spezialisiert [Trivantis].

Abbildung 2.5 stellt einen Ausschnitt aus dem Assistenten dar, der zur Erstellung eines neuen Lernmoduls genutzt werden kann. Man hat die Wahl zwischen verschiedenen Designvorlagen und kann die Struktur eines Moduls leicht durch die Angabe der Anzahl der Kapitel und der zugehörigen Seiten erstellen. Abbildung 2.6 zeigt, wie den verwendeten Medien (z.B. einer Bilddatei) IMS-Metadaten nach dem Metadatenschema von SCORM 1.2 hinzugefügt werden können. Abbildung 2.7 zeigt die Arbeitsumgebung von Lectora bei der Verwendung einer Designvorlage.

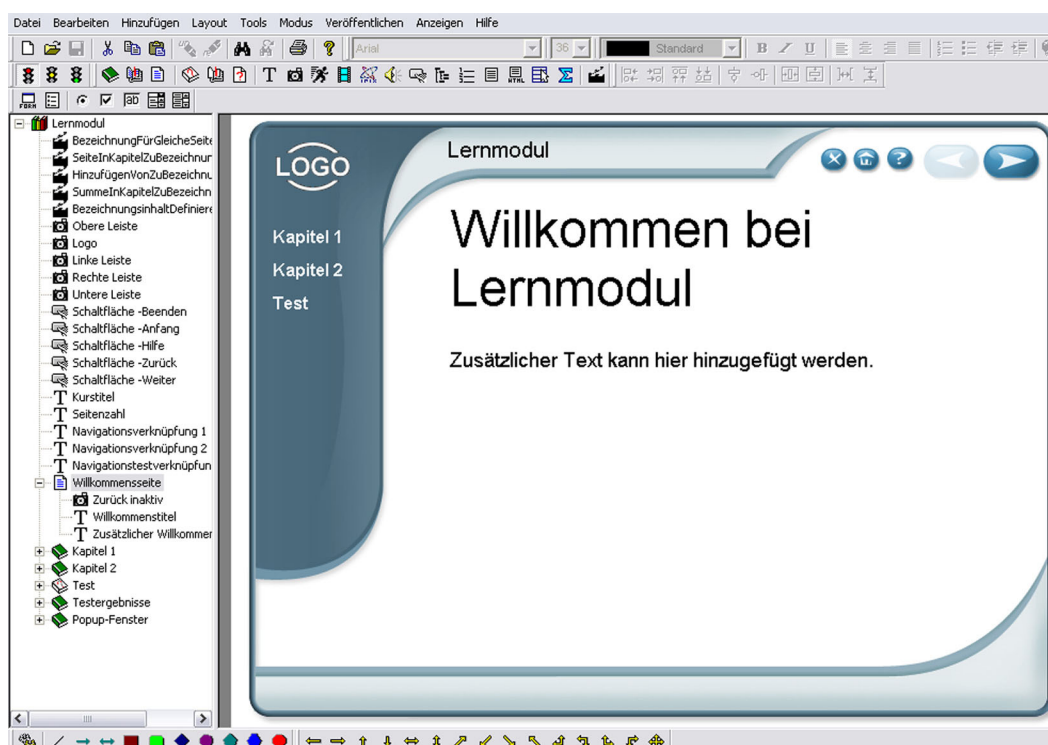


Abb. 2.7:
Ausschnitt aus
dem Programm
Lectora

2.2.5 Unterstützte E-Learning Standards

Die Unterstützung von E-Learning Standards kann als ein wichtiges Qualitätskriterium für Lernmanagement Systeme betrachtet werden. Verschiedene Evaluationen von Lernplattformen stuften die Unterstützung von internationalen E-Learning Standards als äußerst wichtig oder auch als K.O.-Kriterium ein.⁵

Für kommerzielle Anbieter wie Saba stellen Faktoren wie Interoperabilität, Portabilität, Wiederverwendbarkeit und Beständigkeit, die durch Standards gewährleistet werden können, entscheidende Verkaufsargumente dar. Im Folgenden wird die Unterstützung von Standards und Spezifikationen wichtiger Standardisierungsprojekte durch die Saba 3 Release4 beschrieben.

AICC:

Die *AICC-CMI Guidelines for Interoperability* stellt drei Kriteriengruppen auf, die von webbasierten Learning Management Systemen erfüllt werden müssen, damit diese sich AICC-konform nennen können. Dazu gehören folgende Fähigkeiten:

- *Launching Interoperability*: AICC-konformen Inhalt starten können
- *Communications Protocol Interoperability*: die Unterstützung des HTTP-basierten AICC CMI Protokolls zur Kommunikation zwischen LMS und Lerninhalt
- *Course Structure Interoperability*: die Möglichkeit, Kurstrukturen (definiert durch das Course Structure Data Model und die Course Structure Definition) zu importieren und zu interpretieren; dabei wird zwischen den Interoperabilitätsstufen 1, 2, 3a, 3b und 3 unterschieden

Um die Anforderungen für Stufe 1 der Course Structure Interoperability zu erfüllen, muss ein LMS in der Lage sein, die einzelnen Lerneinheiten einer AICC Kursstruktur zu erkennen, und diese in der korrekten hierarchischen Reihenfolge darzustellen. Auf dieser Stufe kann keine Abarbeitungsreihenfolge der Lerninhalte vorgeschrieben werden. Alle mandatory (zwingenden) Datenelemente des AICC CMI Course Structure Data Models müssen interpretiert werden können.

5. Dies ist z.B. der Fall bei der Evaluation für das österreichische Bundesministerium für Bildung, Wissenschaft und Kultur (bm:bwk) aus dem Jahr 2002, sowie bei EVA:LERN (einer Evaluation von Lernplattformen für die Behörde für Wissenschaft und Forschung des Landes Hamburg), und bei der Evaluation für die Lernplattformauswahl für das Bildungsportal Sachsen.

Die AICC Course Structure Interoperability der Stufen 2 und 3a, 3b und 3 stellt zusätzliche Anforderungen bezüglich der Verarbeitung von *prerequisites* (Voraussetzungen), *objectives* (Zielsetzungen) und von Vorgaben für die vollständige Abarbeitung/Erfüllung von Lerneinheiten/Tests. Dazu müssen die entsprechenden optionalen Datenelemente des Course Structure Data Models unterstützt werden. [AICC 2001, S. 137ff]

Saba ist für die Interoperabilität mit AICC-konformen Inhalten auf der Stufe 1 zertifiziert und besitzt folgende Fähigkeiten: [Saba 2003, S. 7f]

- Import von AICC Inhalten
- Export von AICC Inhalten
- Starten von AICC Inhalten
- Kommunikation mit AICC-Inhalten (Lernfortschrittsverfolgung bei AICC Inhalten und Speicherung von Lernergebnissen)
- Aufzeichnung von Interaktionen der Lernenden bei den einzelnen Fragen eines Tests

IMS:

Saba gehört zu den Mitgliedern des IMS Global Learning Consortium und beteiligt sich aktiv bei verschiedenen Arbeitsgruppen des Konsortiums. Folgende IMS-Spezifikationen werden von Saba unterstützt: [Saba 2002, S. 2f]

- *IMS Content Packaging*: Das Saba CMS kann *IMS Content Packages* (Inhaltspakete) importieren und mit einem Internetbrowser starten.
- *IMS Metadata*: Das Saba LMS nutzt IMS Metadaten für die Suche nach Lernressourcen; das Saba CMS nutzt IMS Metadaten zur Beschreibung von Inhalten in der Inhalts-Datenbank.
- *IMS Enterprise*: Das IMS Enterprise Datenmodell wird vom Saba LMS genutzt, um Daten mit anderen virtuellen Lernumgebungen (virtuellen Klassenzimmern) bei der Nutzung von Saba VLE Connector auszutauschen.

ADL SCORM:

Das Referenzmodell SCORM 1.2 liefert Richtlinien für den Aufbau und die Beschreibung von Lerneinheiten sowie die Kommunikation zwischen Lerneinheiten bzw. Lernobjekten und Learning Management Systemen (vgl. 1.4). Damit diese Kommunikation, wie sie in der SCORM

RTE beschrieben ist, stattfinden kann, muss nicht nur das Lernobjekt sondern auch das LMS bestimmte Anforderungen erfüllen. ADL beschreibt daher in SCORM 1.2 drei Kompatibilitätsstufen für Learning Management Systeme [Saba 2004, S. 8f]:

- **LMS Run-Time Environment Level I Conformance (LMS RTE-1)**

Um in diese erste Stufe zu erreichen, muss ein LMS folgende Fähigkeiten besitzen:

- Importieren und Verarbeiten von scorm-konformen Content Aggregation Content Packages
- Starten von scorm-konformen SCOs und Assets
- Ausstattung mit und Bereitstellung eines API-Adapters als Document Object Model (DOM); korrekte Implementierung aller API-Funktionen

- **LMS Run-Time Environment Level II Conformance (LMS RTE-2)**

Um in diese zweite Stufe zu erreichen, muss ein LMS die erste Stufe und zusätzlich folgende Anforderung erfüllen:

- Korrekte Implementierung eines oder mehrerer (nicht alle) optionalen Data Model Elemente der SCORM Version 1.2 RTE

- **LMS Run-Time Environment Level III Conformance (LMS RTE-3)**

Um in diese dritte Stufe zu erreichen, muss ein LMS die erste Stufe und zusätzlich folgende Anforderung erfüllen:

- Korrekte Implementierung aller optionalen Data Model Elemente der SCORM Version 1.2 RTE

Saba3 Release4 hat von ADL die Zertifizierung auf der dritten Konformitätsstufe für SCORM 1.2 erhalten. Es können SCORM-Inhalte importiert, exportiert und gestartet werden. Dabei werden alle Data Model Elemente der SCORM RTE Version 1.2, die zum Datenaustausch zwischen LMS und SCO zum Einsatz kommen, unterstützt (vgl. 1.4.2).

Diese Zertifizierung gibt jedoch keine Auskunft über die Unterstützung von Metadaten, die zur Beschreibung der Content Aggregation (*Content Aggregation Meta-data*) sowie von SCOs (*SCO Meta-data*) und Assets (*Asset Meta-data*) verwendet werden (vgl. 1.4.1). Unter Saba3 Release4 werden von diesen Metadaten jeweils nur die zwingenden, als ‚mandatory‘ bezeichneten Datenelemente unterstützt.

Für Lerninhalte, die nicht durch eine IMS-Manifestdatei strukturiert sind, sondern noch dem Course Structure Format (CSF) der SCORM Version 1.1 entsprechen, bietet Saba das Format *Deployed SCORM* mit der CSF-Dateien in Saba importiert werden können. [Saba, 2004, S. 9ff]

Zusammenfassend soll die Abbildung 2.8 den Zusammenhang zwischen verschiedenen Richtlinien von IMS, AICC und ADL und deren Integration in das Produkt Saba verdeutlichen.

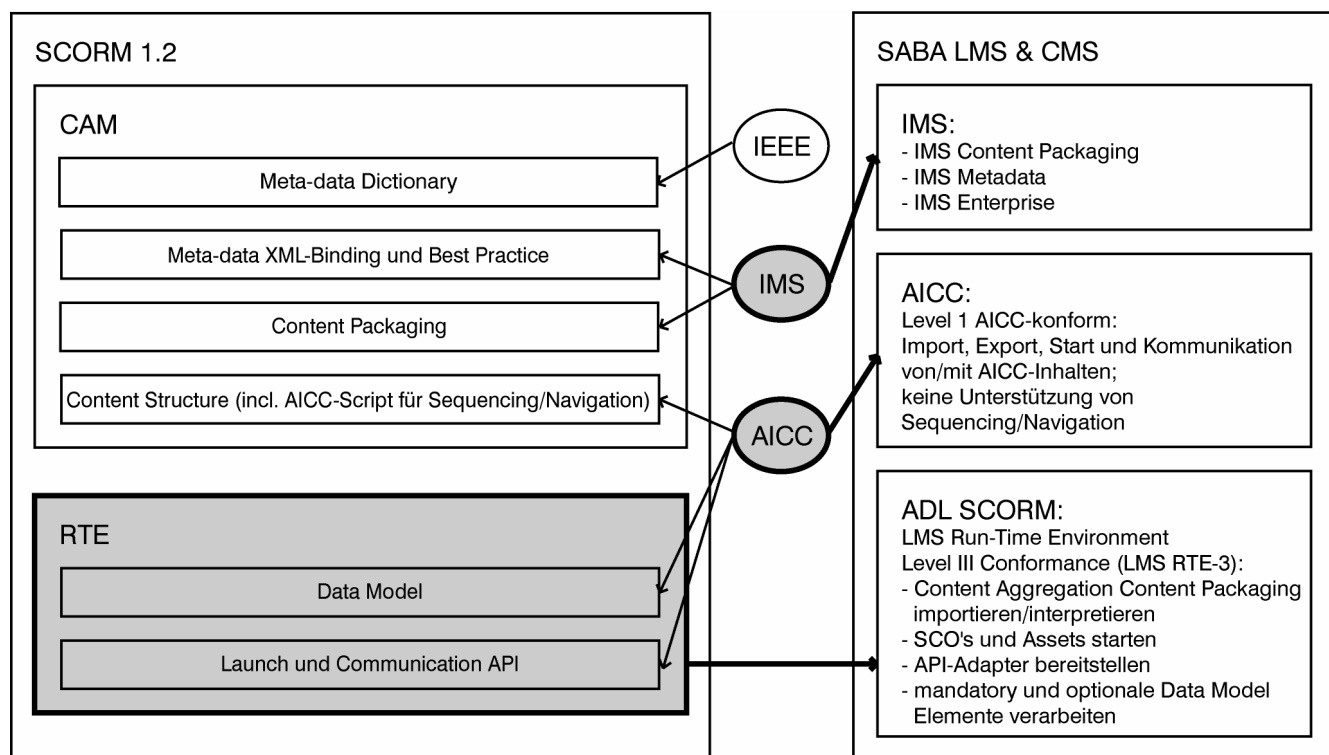


Abb. 2.8: Durch Saba3 Release4 unterstützte E-Learning Standards

Wie bereits erwähnt unterstützt Saba3 Release4 nicht die aktuelle Version SCORM 2004 (vgl. 1.4), dafür aber die SCORM 1.2 Run-Time Environment auf dem höchstmöglichen Level (LMS RTE-3). Dies setzt voraus, dass Saba SCORM-Inhaltspakete, die auf IMS Content Packaging beruhen, importieren sowie die Content Structure von SCORM-Paketen, die auf AICC Richtlinien basiert, interpretieren kann. Saba ist nach den AICC CMI Richtlinien für Interoperabilität auf der Stufe 1 zertifiziert und unterstützt somit nicht die Verarbeitung von prerequisites, welche unter SCORM 1.2 mit Hilfe eines AICC-Skripts [SCORM 2001b, S. 2-107ff] verwendet werden um *Sequencing und Navigation* zu unterstützen.

Unter Saba3 Release4 ist also die Ablaufsteuerung innerhalb eines SCORM-Pakets der Version 1.2 durch die Vorgabe von bestimmten Voraussetzungen bezüglich der Abarbeitung/Erfüllung von SCOs nicht möglich.

2.3 Die Nutzung der Lernplattform Saba3 Release4 für das Bildungsportal Sachsen

Das Bildungsportal Sachsen verwendet Saba3 Release4 als zentrales Learning Management System. Die folgenden Abschnitte stellen die Integration dieses Systems im BPS vor.

Zudem wird Einsatz des Systems ACollab zur Bereitstellung von Kommunikations- und Kollaborationsmöglichkeiten im Bildungsportal beschrieben. Dabei handelt es sich um ein frei verfügbares System der Universität Toronto, welches seit Frühjahr 2005 im Bildungsportal eingesetzt wird.

2.3.1 Das Internetportal

Die Eingangsseiten von Lernportalen können unter Saba individuell gestaltet werden. So besitzt auch das Bildungsportal Sachsen eine, den spezifischen Anforderungen entsprechende Begrüßungsseite [Bildungsportal].



Abb. 2.9: Startseite des Bildungsportals Sachsen

Auf dieser Eingangsseite befinden sich Hyperlinks zu folgenden Themen:

- Über uns: beinhaltet allgemeine Informationen zum Bildungsportal Sachsen, wie z.B. Zielsetzungen des Projekts, Neuigkeiten, Ausschreibungen, etc.
- Hochschulen: Auflistung aller beteiligten Hochschulen mit Hyperlinks zu den Hochschulen
- Ausbildung: über diesen Link gelangt man zur eigentlichen Lernplattform mit dem Angebotskatalog für Studenten und Angehörige der Sächsischen Hochschulen
- Weiterbildung: Link zum Weiterbildungsangebot mit meist kostenpflichtigen Angeboten

Im Bereich *Ausbildung* hat man die Möglichkeit, sich als Nutzer anzumelden bzw. zu registrieren. Zudem kann über den Angebotskatalog eines der *frei nutzbaren* Lernangebote ausgewählt werden. Jedoch steht erst nach der Anmeldung der volle Umfang des Angebotskatalogs zur Verfügung.

Nach einer Anmeldung im Bereich Ausbildung kann man folgenden Seiten nutzen:

- *Startseite*: mit allgemeinen und aktuellen Hinweisen
- *Meine Einschreibungen*: Auflistung der Einschreibungen und Zugang zu den Lerninhalten
- *Angebotskatalog*: Auflistung der Lernangebote

- *Communities*: Zugang zu Communities, die entsprechend der Zugehörigkeitsgruppe zugeordnet werden
- *Evaluationen*: über diesen Link besteht die Möglichkeit an allgemeine Umfragen und der Bewertung von Fragebögen teilzunehmen
- *Persönliche Daten*: Anzeige und Änderungsmöglichkeit der persönlichen Daten

2.3.2 Die Implementierung der Lernplattform Saba3 Release4 für das Bildungsportal Sachsen

Im Folgenden wird die Implementierung der Lernplattform Saba3 Release4 anhand der Seiten *Angebotskatalog* und *Meine Einschreibungen* sowie der Verwaltung von Angeboten und Inhalten erläutert.

Angebotskatalog:

Der Angebotskatalog kann nach den Kriterien *Hochschule* (Anbieter des Angebots), *Bereich* (Studienbereich), *Stichwort/Freitextsuche* und nach *Zugang* durchsucht werden. Unter *Zugang* wird zwischen folgenden Möglichkeiten unterschieden:

- *Demo*: frei zugängliche Demonstrationsversion eines Lerninhalts
- *frei nutzbar*: Lerninhalt ist ohne vorheriges Einschreiben verfügbar
- *Informationen*: Katalogeintrag mit Informationen zu einem Angebot; der Lerninhalt selbst ist über diesen Eintrag nicht verfügbar
- *nach Einschreibung*: Selbstlernmodul (SLM), welches nur nach Einschreibung zugänglich ist; ein Selbstlernmodul kann jederzeit und „selbst gesteuert“ genutzt werden
- *nach Einschreibung (LV)*: Lehrveranstaltung (LV), die nur nach Einschreibung zugänglich ist; eine Lehrveranstaltung ist an einen bestimmten Zeitplan gebunden.

Titel	Hochschule	Zugang	Anmelden
Abschlussarbeiten Seminar Sonntag	HTWD. Wirtschaftswissenschaften	nach Einschreibung (LV)	
Controlling - Vorlesungen zum Studienschwerpunkt Controlling	HTWD. Wirtschaftswissenschaften	nach Einschreibung (LV)	
Controlling-Fallstudien im Studienschwerpunkt Controlling	HTWD. Wirtschaftswissenschaften	nach Einschreibung (LV)	
Controlling-Seminar im Studienschwerpunkt Controlling	HTWD. Wirtschaftswissenschaften	nach Einschreibung (LV)	
Datensicherheit / Datenintegrität in Datenbanken	HTW. Informatik/Mathematik	frei nutzbar	
Diplomandenseminar Controlling	HTWD. Wirtschaftswissenschaften	nach Einschreibung (LV)	
Director Aufbaukurs	HTW. Informatik/Mathematik	nach Einschreibung	
Director Grundkurs	HTW. Informatik/Mathematik	nach Einschreibung	
Dramaturgie-Bibliografie	HMT Leipzig	nach Einschreibung	
Energieversorgung	TUD. Maschinenwesen	nach Einschreibung	
Entwurf einer Turbinenregelung	BPS.HS Zittau/Goerlitz	Informationen	
Fundamentals of Applied Computer Science	HTW. Informatik/Mathematik	nach Einschreibung	
IB MA Financial Management	HTWD. Wirtschaftswissenschaften	nach Einschreibung (LV)	
Innovationsmanagement in Klein- und Mittelunternehmen	Uni Leipzig	nach Einschreibung	
Interaktive Einführung in die Arbeitsweisen der Kartographie	HTWD. Vermessungswesen/Kartographie	frei nutzbar	

Abb. 2.10: Angebotskatalog des Bereichs Ausbildung

Lerninhalte können im Bildungsportal Sachsen als Selbstlernmodul (*Product*), als Lehrveranstaltung (*Class*) oder als Fach (*Course*) veröffentlicht werden (vgl. 2.2.2). Fächer werden verwendet, um bestimmte Rahmenbedingungen für Lehrveranstaltungen, wie z.B. maximale Teilnehmerzahl, Passwort, Kurzbeschreibungen oder auch Inhalte, festzulegen. Lehrveranstaltungen stellen Instanzen von Fächern mit festgelegtem Zeitplan dar.

Im Angebotskatalog stehen Selbstlernmodule und Fächer zur Auswahl. Einschreibungen können jedoch nur für Selbstlernmodule und Lehrveranstaltungen zu den Fächern vorgenommen werden. Dies geschieht im Angebotskatalog über die Symbole in der Spalte *Anmelden* (vgl. Abb. 2.10):

- Einschreibung für ein Selbstlernmodul: geschieht durch Mausklick auf das Symbol des Einkaufswagens ()
- Einschreibung in eine Lehrveranstaltung: geschieht durch Mausklick auf das Kalender-Symbol (); daraufhin werden alle dem angeklickten Fach zugeordneten Lehrveranstaltungen angezeigt, in diese man sich durch Mausklick auf das Einkaufswagen-Symbol () einschreiben kann

Meine Einschreibungen:

Im Bereich *Meine Einschreibungen* lassen sich die Einschreibungen zu Lehrveranstaltungen (*Meine Veranstaltungen*) und zu Selbstlernmodulen (*Meine Selbstlernmodule*) anzeigen. Von hier aus kann der Lerninhalt über das Symbol in der Spalte *Inhalt* gestartet werden. Zudem besitzt man von hier aus Zugang zu den Kommunikationsmöglichkeiten, kann Ergebnisse einsehen, oder sich vom Lernangebot abmelden. Bereits beendete Lernangebote kann man sich über den Hyperlink (*Beendete Lernangebote*) anzeigen lassen. Die Abbildung 2.11 zeigt ein Beispiel der Seite *Meine Selbstlernmodule*.

The screenshot shows the user interface of the Bildungsportal Sachsen. At the top, there is a navigation bar with links for 'Kalender', 'Kontakt', 'Hilfe', and 'Abmelden'. Below this is a header with the logo and name 'BILDUNGSPORTAL SACHSEN' and the tagline 'Aus- und Weiterbildung der Hochschulen des Freistaates Sachsen'. A dropdown menu 'Wechseln zu:' is set to 'Startseite'. The main navigation menu includes 'Startseite', 'Meine Einschreibungen', 'Angebotskatalog', 'Communities', 'Evaluationen', and 'Persönliche Daten'. The 'Meine Einschreibungen' section is active, showing a sidebar with links to 'Meine Veranstaltungen', 'Meine Selbstlernmodule', and 'Beendete Lernangebote'. The main content area is titled 'Meine Selbstlernmodule' and contains a text block explaining the page's purpose. Below this is a table with the following data:

Anmeldedatum	Titel	Ergebnis	Inhalt	Aktion
17.07.05	Typografie praktisch! Version: 1.0			
27.05.05	Director Grundkurs			

A legend titled 'Legende' explains the icons: a person icon for 'Inhalt aufrufen', a document icon for 'Ergebnisse einsehen', a speech bubble icon for 'Diskussionsforum öffnen', and a red 'X' icon for 'Vom Lernangebot abmelden'.

Abb. 2.11: Bereich *Meine Einschreibungen*

Inhalte und Angebote verwalten:

Das Auswahlfeld *Wechseln zu* ist in der Regel nur bei Hochschulmitarbeitern sichtbar. Studenten, die Lerninhalte ‚nur‘ nutzen, besitzen diese Auswahlmöglichkeit auf ihrer Seitendarstellung nicht.

Die Auswahl der angezeigten Menüeinträge ist von den Berechtigungen und den administrativen Aufgaben eines Mitarbeiters abhängig. Möchte man neue Lerninhalte hinzufügen und diese als Lernangebote veröffentlichen, benötigt man mindestens die Berechtigungen zum Verwalten von Angeboten und Inhalten, wie dies in der Abbildung 2.12 dargestellt ist.



Abb. 2.12:
Auswahlfeld *Wechseln zu*
(zu finden links neben dem Logo des
Bildungsportals)

Da eine ausführliche Beschreibung der Verwaltung von Inhalten und Angeboten im Rahmen dieser Arbeit zu umfangreich wäre, wird an dieser Stelle auf das Dokument „Hinweise und Dokumentation zum Lernmanagementsystem des Bildungsportal Sachsen“ vom Bildungsportal Sachsen verwiesen. Dieses beschreibt sehr genau die Vorgehensweise beim Import und der Veröffentlichung von Lerninhalten und ist über eine E-Mail mit Kontaktdaten und kurzem Betreff an training@bps.system.de als kostenlose PDF-Datei erhältlich.

2.3.3 Kommunikations- und Kollaborationsfunktionen des Systems ACollab

Bisher standen über das Bildungsportal Sachsen ausschließlich die im Lernmanagementsystem Saba integrierten Kommunikationswerkzeuge *Chat* und *Diskussionsforum* zur Verfügung (vgl. 2.2.3). Dabei war für Lehrveranstaltungen die Zuordnung von Chat und Forum möglich. Selbstlernmodulen konnte man ein Diskussionsforum zuordnen.

Der unter Saba3 Release4 zur Verfügung stehende Virtual Classroom wird vom Bildungsportal nicht eingesetzt, da dieser zwar eine Umgebung für synchrones Lernen zur Verfügung stellt, jedoch als virtuelles Klassenzimmer nur eingeschränkte Funktionen besitzt (vgl. 2.2.3). Auch zusätzliche kostenpflichtige Produkte von Saba zur Erweiterung der integrierten Kommunikations- bzw. Kollaborationsmöglichkeiten wie SabaDialog und Saba Live! werden vom Bildungsportal nicht genutzt. Dafür wird seit Frühjahr 2005 das frei verfügbare System ACollab des Adaptive Technology Resource Centre (ATRC) der Universität Toronto [ACollab] eingesetzt.

Dieses bietet mit einer kooperativen Lernumgebung (*Online-Community*) zusätzliche Funktionen zu Chat und Forum. Die Online-Community stellt die eigentliche Neuerung für das BPS dar. Die bisher bestehenden Kommunikationsmöglichkeiten Chat und Diskussionsforum wurden

jedoch auch vollständig durch die entsprechenden Funktionen von ACollab ersetzt und fest mit den Lehr- und Lernangeboten unter Saba verknüpft. Dabei wurden folgende Aspekte berücksichtigt [Bildungsportal 2005a, S. 11]:

- Beibehaltung der bisherigen Zuordnung von Kommunikationsmitteln zu Lehrveranstaltungen (Forum/Chat) bzw. Selbstlernmodulen (Forum)
- Lehrveranstaltungen kann die neue Gesamtfunktionalität *Online-Community* zugeordnet werden; einzelne Zugriffsmöglichkeiten auf Forum bzw. Chat bleiben aber erhalten

Die Nutzung der ACollab-Kommunikationswerkzeuge ist nach einem strengen Gruppenkonzept geregelt. Dies bedeutet, dass sich Forum, Chat oder Online-Community jeweils auf ein bestimmtes Lernangebot bezieht, woraus sich die Gruppenzugehörigkeit ergibt. Die Kommunikationsinhalte sind jeweils nur von Nutzern des jeweiligen Lernangebots einzusehen.

Der Zugang zu den Kommunikationsmöglichkeiten ist, wie zuvor erwähnt, über den Bereich *Meine Einschreibungen* geregelt (vgl. 2.3.2). Unter *Meine Lehrveranstaltungen* bzw. *Meine Selbstlernmodule* wird durch die entsprechenden Symbole angezeigt, welche Kommunikationsmöglichkeiten genutzt werden können. Abbildung 2.13 zeigt das Beispiel einer Veranstaltung, der eine Online-Community zugeordnet wurde (zu sehen in der Spalte *Kommunikation*).

Titel ▾	Veranstaltungsname	Hochschule	Datum	Status	Ergebnis	Inhalt	Kommunikation	Aktion
Testfach		HTW Dresden	18.07.05- 18.07.05	Offen - Bestätigt				 

Legende

 Inhalt aufrufen	 Vom Lernangebot abmelden
 E-Mail an Tutor	 Ergebnisse einsehen
 Diskussionsforum öffnen	 Als beendet kennzeichnen
 Chat	
 Online-Community	

Abb. 2.13: Kommunikationsmöglichkeiten unter *Meine Veranstaltungen*

Bei der Nutzung der Online-Community stehen folgende Bereiche mit Kommunikations- und Kollaborationsfunktionen zur Verfügung [Bildungsportal 2005a, S. 5ff]:

- Startseite: verschafft Überblick über aktuelle Vorgänge in der Gruppe, wie z.B. neue Forum-einträge oder neue Dateien, und ermöglicht Schnellzugriff auf diese
- Foren: dieser Bereich stellt den Zugang zu Diskussionsforen dar; Foren können vom Gruppenadministrator (Tutor) angelegt werden; auch die Zugriffsrechte (Lesen, Antworten, Erstellen neuer Threads) werden über den Gruppenadministrator geregelt; Teilnehmer haben die Möglichkeit, Threads zu abonnieren, sie werden dann per E-Mail über neue Nachrichten innerhalb dieses Threads benachrichtigt
- Chatraum: alle Nachrichten eines Chats werden chronologisch dargestellt und bleiben so lange erhalten, bis der letzte Teilnehmer den Chat verlässt; Tutoren können die Protokollierung eines Chats vornehmen; dadurch werden die Nachrichten in einer Datei gespeichert und können so auch nach Beendigung des Chats eingesehen werden; eine Benutzerliste verschafft Überblick über alle aktuellen Teilnehmer
- Posteingang: stellt ein einfaches, internes Nachrichtensystem zur Versendung von Nachrichten an einen oder mehrere Gruppenteilnehmer dar
- Kalender: dient der Gruppenterminverwaltung; die eingetragenen Termine sind immer für alle Teilnehmer sichtbar; die Termine können durch Zeitdauer, einen kurzen Text und den Anhang von Dateien aus der Bibliothek beschrieben werden
- Bibliothek: fungiert als zentrales Dateiarchiv der Gruppe; hier können Dateien, wie z.B. bearbeitete Übungsaufgaben, aus dem Entwurfsbereich gespeichert werden, wodurch der aktuelle Entwicklungsstand festgehalten wird
- Entwurfsbereich: stellt eine Arbeitsumgebung dar, über die Dateien zur gemeinsamen Bearbeitung zentral abgelegt werden können; die Bearbeitung erfolgt durch Download der aktuellen Dateiversion und dem Upload der bearbeiteten Datei; über ein Kommentar kann die Veränderung beschrieben werden; zur Speicherung der bearbeitbaren Dateien stehen Gruppenordner zur Verfügung, die vom Gruppenadministrator angelegt werden und von allen Gruppenmitgliedern genutzt werden können; zudem besitzt jeder Teilnehmer einen persönlichen Ordner
- Teilnehmer: zeigt die Teilnehmer der aktuellen Gruppe samt der persönlichen Profile an; jeder Teilnehmer hat hier die Möglichkeit sein Profil, z.B. durch das Hinzufügen eines Photos, zu

bearbeiten; der Tutor kann über die Spalte *Moderator* festlegen, welcher Teilnehmer im Forum als Moderator fungieren kann

- Systemverwaltung: diese Seite ist nur für den Gruppenadministrator zugänglich und wird auch nur in dessen Profil angezeigt; hier können neue Foren erstellt und Zugriffsrechte verwaltet werden, E-Mails an Gruppenmitglieder versendet und Neuigkeiten/Ankündigungen auf der Startseite eingetragen werden; zudem kann der Gruppenadministrator hier festlegen, welche der Kommunikations- bzw. Funktionsbereiche für die Teilnehmer verfügbar sein sollen



Abb. 2.14: Beispiel einer Startseite unter ACollab

3 Konzeption von Lerninhalten für das Bildungsportal Sachsen am Beispiel des Lernmoduls „Typografie praktisch!“

3.1 Vorgehensweise bei der Entwicklung von Lernmodulen

Der Entwicklungsprozess von Lernanwendungen lässt sich grob in die drei Phasen *Planung*, *Produktion* und *Postproduktion* unterteilen. Abbildung 3.1 gibt einen Überblick über diese Phasen und zeigt Zusammenhänge zwischen den einzelnen Schritten auf.

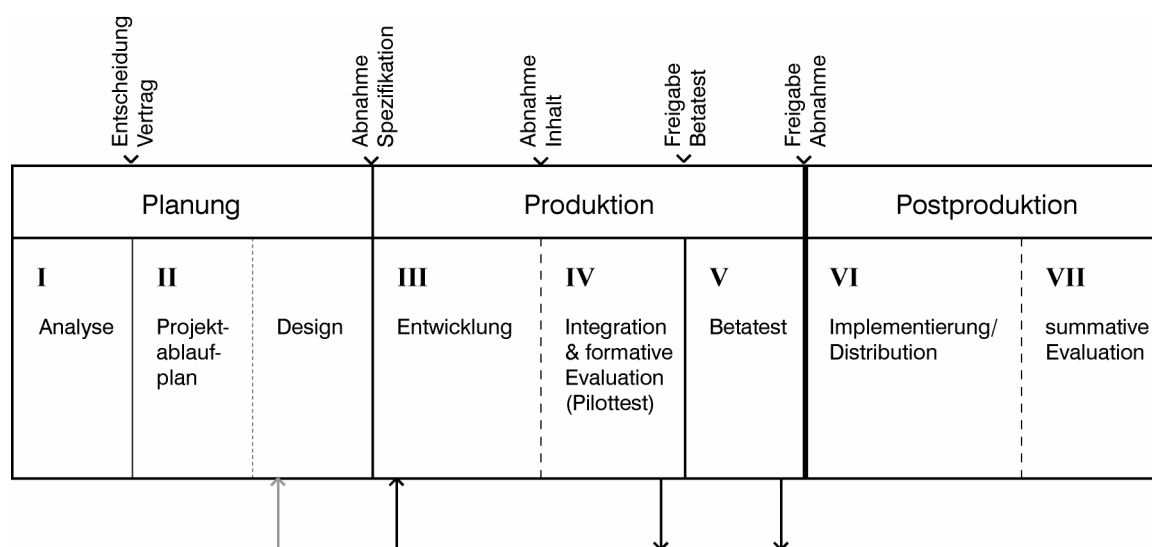


Abb. 3.1: Das CBT-Entwicklungsmodell im Überblick, nach [Schreiber 1998, S. 84]

Zu Beginn der *Planungsphase* sollte je nach Art und Größe des Projekts eine entsprechend umfangreiche Analyse der Ausgangsbedingungen stehen. Dieser lässt sich beispielsweise in folgende Bereiche unterteilen [Schreiber 1998, S. 53ff] u. [Niegemann u.a. 2004, S. 51ff]:

- Formulierung des Themas: Was soll das Thema der Lernanwendung sein? Wie lässt sich der Inhalt kurz beschreiben? Wie lautet der Titel bzw. Arbeitstitel?

- Adressaten (Zielgruppe): Welche Kompetenzen/Vorkenntnisse sind vorhanden? Welche Lernmotivation liegt vor? Welche allgemeinen Personenmerkmale wie z.B. Alter und Geschlecht treffen zu? (vgl. 3.2.1 *Zielgruppe und Einsatzbereich*)
- Lernziel(e): Welches Wissen soll vermittelt werden? Welche Fähigkeiten erlernt werden? (vgl. 3.2.2 *Lernziele*)
- Einsatzkontext: Wo und wie soll später mit dem Lernmodul gelernt werden? (siehe hierzu Abschnitt 3.2.1 *Zielgruppe und Einsatzbereich*) Welche technischen Voraussetzungen sind gegeben? (vgl. 3.2.3 *Technische Vorgaben*)
- Ressourcen: Welches Material und Personal, wie viel Zeit und Geld steht zur Verfügung? Welche Rechte am einzusetzenden Material sind vorhanden? (vgl. 3.2.5 *Rechtliche Aspekte*) Welche Hard- und Software steht zur Entwicklung zur Verfügung? (vgl. 3.2.3 *Technische Vorgaben*)

Die Analyse der Ausgangsbedingungen dient als Grundlage für die Entwicklung des *Designs*. Darunter ist sowohl die didaktische Konzeption als auch die visuelle und funktionelle Gestaltung der Benutzeroberfläche zu verstehen. Häufig wird zum Abschluss dieser Phase ein teilweise lauffähiger Prototyp erstellt. Die Entwicklung des *Designs* des Lernmoduls ‚Typografie praktisch!‘ wird in den Abschnitten 3.3 *Didaktische Konzeption* und 3.4 *Gestaltung der Benutzeroberfläche* beschrieben.

Nach der Festlegung des *Designs* kann die *Produktionsphase* beginnen. In dieser Phase werden die einzelnen Programmbausteine wie z.B. *Texte* (Basaltext, Drehbuch, Fachglossar, etc.), *Grafiken*, *audiovisuelles Material* und *Benutzeroberfläche* erstellt [Schreiber 1998, S. 81ff].

Es ist sinnvoll schon zur Beginn der *Produktionsphase* einzelne Komponenten zu lauffähigen Programmteilen zusammensetzen und dieses im Rahmen von *Pilottests* auf ihre technische, gestalterische und didaktische Qualität zu prüfen. Die Testergebnisse können so noch während der *Produktionsphase* berücksichtigt werden.

Nach Fertigstellung und Zusammensetzung der einzelnen Bausteine sollte ein Lernmodul, bevor es für Nutzer freigegeben wird, einem so genannten *Betatest* unterzogen werden. Dabei kann es sich sowohl um einen technischen Test als auch um eine praktisch-pädagogische Prüfung handeln. Nach dem erfolgreichen Bestehen des *Betatests* kann die Freigabe für die Nutzer stattfinden.

Die Implementierung (Veröffentlichung) bzw. die Distribution (Vervielfältigung, Vertrieb) wird bereits der *Postproduktion* zugeordnet. Auch so genannte summative Evaluationen werden der Phase der Postproduktion zugerechnet. Dabei kann es sich z.B. um Kosten-Nutzen-Analysen, Effektivitätsstudien oder um den Vergleich mit ähnlichen Produkten handeln [Schreiber 1998, S. 86ff]. Die Phasen der Produktion und der Postproduktion des Lernmoduls ‚Typografie praktisch!‘ werden im vierten Kapitel beschrieben.

3.2 Vorgaben

Das Lernmodul ‚Typografie Praktisch!‘ dient der Vermittlung von *typografischen Regeln* und deren *Umsetzung mit dem Programm InDesign CS*. Begleitend zu der Lehrveranstaltung ‚Typographie‘ im Studiengang Medieninformatik an der Hochschule für Technik und Wirtschaft (HTW) Dresden soll es Möglichkeiten zur selbständigen Vertiefung von Wissen und Fähigkeiten in den genannten Bereichen geben.

Der Auftrag zur Entwicklung dieses Lernmoduls stammt vom zuständigen Hochschullehrer, Prof. Dr.-Ing. Panajotov (Fachbereich Kartographie/Vermessungswesen), der auch das Thema, die Zielgruppe, die Lernziele, Lerninhalte und den Einsatzkontext vorgegeben hat.

3.2.1 Zielgruppe und Einsatzbereich

Die Zielgruppe des Lernmoduls ‚Typografie praktisch!‘ ergibt sich demnach aus Studenten des Studiengangs Medieninformatik an der HTW Dresden, die ihr Wissen aus der Lehrveranstaltung ‚Typographie‘ selbständig vertiefen möchten.

Die Zielgruppe besitzt durch die Lehrveranstaltung bereits Grundkenntnisse im Bereich Typografie und in dem Programm InDesign. Zudem ist aufgrund des Studiengangs Medieninformatik von guten Computerkenntnissen und einer aufgeschlossenen Haltung bezüglich des Lernens am Computer auszugehen. Erfahrungen mit selbst kontrolliertem Lernen bzw. Üben mit Computerprogrammen sind vorhanden.

Der Zugriff auf das Lernmodul geschieht als Selbstlernmodul über die Lernplattform des Bildungsportals Sachsen und ist nach dem Login auf der Plattform verfügbar. Da das Lernprogramm nicht durch ein zusätzliches Passwort geschützt ist, ist es allen weiteren im Bildungsportal registrierten Personen möglich, das Lernmodul zu nutzen.

Zudem besteht die Möglichkeit für Personen mit Rechten zur Verwaltung von Angeboten, das Lernmodul einem Fach oder einer Lehrveranstaltung zuzuordnen. Inhaber von Rechte zur Verwaltung von Inhalten haben z.B. die Möglichkeit, das Modul in der Inhaltsdatenbank zu duplizieren und mit dem Content Builder beliebig zu bearbeiten.

3.2.2 Lernziele

Lernziele lassen sich dem kognitiven, dem affektiven oder dem psychomotorischen Bereich zuordnen. Affektive Lernziele beziehen sich vorrangig auf Gefühle, Einstellungen und Verhaltensweisen, Lernziele im psychomotorischen Bereich beziehen sich auf Bewegungsabläufe wie z.B. Auto fahren oder Tanzen. Die meisten computerbasierten Lernanwendungen orientieren sich jedoch an Lernzielen aus dem kognitiven Bereich. Diese beziehen sich auf intellektuelle und geistige Fähigkeiten und sind an Tätigkeiten wie Wahrnehmung, Denken und Beurteilen gebunden. [Schreiber 1998, S. 20]

Auch die Lernziele des Moduls ‚Typografie praktisch!‘ lassen sich dem kognitiven Bereich zuordnen. Den Lernenden soll zum einen *theoretisches Wissen* aus dem Bereich der Typografie vermittelt werden. Zum anderen sollen sie mit *praktischen Umsetzungsmöglichkeiten* der typografischen Regeln in dem Programm InDesign CS vertraut gemacht werden und diese *praktisch umsetzen* können. Darüber hinaus ist die Entwicklung eines *gewissen Verständnisses* bezüglich der visuellen Gestaltung von Dokumenten wünschenswert.

Die beschriebenen *Groblernziele* beziehen sich auf folgende Themenbereiche, denen bereits konkrete Lerninhalte in Form von Stichpunkten vom Auftraggeber des Lernmoduls zugeordnet waren:

- Typografie und InDesign (Umsetzung typografischer Grundregeln mit InDesign): beinhaltet Maße und Einteilungen (Schriftgröße), Seitenproportionen, Festlegen des Satzspiegels, Absatzuntergliederung, Fußnoten, Zitate, Marginalien, Feinkorrekturen (Absatzumbruch, Wort-

- trennungen) und die praktische Umsetzung anhand der Erstellung eines 4-seitigen Dokuments
- Mikrotypografie (Schreibregeln): beinhaltet Zahlensatz, Zahlen vor Abkürzungen, Datumsangaben, DIN- und ISO-Nummern, Telefon-, Fax- und Postfachnummern, Bankverbindungen, Absatznummerierung (Absatznummern, Aufzählungen), Abkürzungen, Anführungszeichen, Auslassungszeichen, Auslassungspunkte, Textstriche (Bindestrich, Gedankenstrich) und die praktische Umsetzung anhand der Weiterbearbeitung des 4-seitigen Dokuments
 - Text-Bild-Integration: beinhaltet Anordnung von Abbildungen, Gestaltungsraster, optisches Gewicht von Abbildungen, Ausrichtung, Position, Schrift in Abbildungen, Bildunterschriften, Bildformate, Bildgrößen und die praktische Umsetzung anhand der der Weiterbearbeitung des 4-seitigen Dokuments
 - Farbe in Dokumenten: beinhaltet Funktionen und Einsatzbereiche von Farbe, Stimmung und Wirkung von Farben, Farbharmonie, Farbverläufe, Farbe in Abhängigkeit zum Ausgabemedium und die praktische Umsetzung anhand der farblichen Gestaltung und Fertigstellung des 4-seitigen Dokuments

3.2.3 Technische Vorgaben

Die Technischen Vorgaben lassen sich unterteilen in Vorgaben, die durch den *Einsatzkontext* des Lernmoduls definiert werden, und in Vorgaben, die in direktem Zusammenhang mit der *technischen Entwicklung* des Lernmoduls stehen. Zunächst soll der Einsatzkontext beschrieben werden.

Einsatzkontext:

Die Nutzung des Lernmoduls ist über die Internetplattform des Bildungsportals Sachsen und somit theoretisch von jedem beliebigen Computer mit Internetzugang möglich. Für die Festlegung der technischen Voraussetzungen wurde die Nutzung am Heimarbeitsplatz (privat oder Studentenwohnheim) und in den Computerlaboren der HTW Dresden in Betracht gezogen.

Die Computerlabore der HTW Dresden sowie die Studentenwohnheime in Dresden verfügen über eine Internetverbindungsrate von 10 MBit pro Sekunde. Zur Feststellung der Internetverbindungsrate der Heimarbeitsplätze wurden Studenten der Medieninformatik im Sommersemester 2005 dazu befragt. Das Ergebnis zeigte, dass die Mehrzahl mindestens über einen DSL-Anschluss, also über eine Internetverbindungsrate von mindestens 1 MBit pro Sekunde verfügt.

Das Lernmodul soll sowohl auf Macintosh-Rechnern als auch unter dem Betriebssystem Windows lauffähig sein. Dazu wird unter Windows der Internet Explorer ab der Version 6.0 und auf Macintosh das Programm Safari ab der Version 1.0 als Internetbrowser vorausgesetzt. Für das Abspielen von Shockwave-Filmen sollte mindestens ein Plug-In für Macromedia Flash 6 vorhanden sein.

Zur praktischen Umsetzung der Übungsaufgaben soll das Programm InDesign CS genutzt werden. Dieses steht den Studenten in den Computerlaboren des Fachbereichs Vermessungswesen/Kartographie zur Verfügung.

Technische Entwicklung:

Vorgaben bezüglich des Veröffentlichungsformats des Lernmoduls sind durch die Nutzung des Bildungsportals Sachsen als Veröffentlichungsplattform vorgegeben. Es können ausschließlich die im Abschnitt 2.2.1 *Import und Veröffentlichung von Lerninhalten* aufgelisteten Dateiformate verwendet werden. Im Sinne der Wiederverwendbarkeit des Lernmoduls und der einzelnen Lerneinheiten wurde die Entscheidung für das SCORM-Referenzmodell getroffen. Demnach sollte das Lernmodul nach der von Saba3 Release4 unterstützten SCORM Version 1.2 entwickelt werden, da eine Umstellung des Bildungsportals Sachsen auf SCORM 2004 bzw. auf ein System, das SCORM 2004 unterstützt, nach Aussagen des Bildungsportals zum Zeitpunkt der Lernmodulerstellung nicht in Aussicht stand.

Das System OLAT 4.0, welches ab dem Sommersemester 2006 vom Bildungsportal eingesetzt werden soll, unterstützt SCORM 1.2 (vgl. 2.2.1). Zudem wird davon ausgegangen, dass bei einer möglichen Umstellung auf ein System, welches die aktuelle SCORM-Version unterstützt, weiterhin die SCORM-Vorgängerversionen unterstützt werden. So kann z.B. Saba Enterprise 2005 neben der aktuellen Version SCORM 2004 auch SCORM 1.2 verarbeiten.

Die einzelnen Lernobjekte, aus denen sich das gesamte Lernmodul zusammensetzt, sollten aufgrund der zahlreichen Grafiken, Interaktionen und Animationen mit dem Programm

Flash MX¹ der Firma Macromedia realisiert werden. Dieses stellt durch eine grafische Benutzeroberfläche, eine Animationszeitleiste und eine eigene objektorientierte Programmiersprache ein komfortables Autorenwerkzeug zur Erstellung von Lerninhalten dar. Zudem ist dieses Werkzeug besonders für die Erstellung von Internetinhalten geeignet, da es vektorbasiert arbeitet und verschiedene Komprimierungsformate unterstützt. Die Erstellung der Lernobjekte und der einzelnen Bausteine erfolgte abgesehen von Sprachaufnahmen in den Computerlaboren der HTW Dresden und der dort zur Verfügung stehenden Software (vgl. Kapitel 4).

3.2.4 Vorgaben zum visuellen Design

Zur visuellen Gestaltung des Lernmoduls gab es vom Auftraggeber Vorgaben bezüglich der zu verwendenden Schrift und der Farbgebung. Als Vorlage dazu diente das bereits über das Bildungsportal Sachsen verfügbare Lernmodul ‚Interaktive Einführung in Arbeitsweisen der Kartographie‘ des Fachbereichs Vermessungswesen/Kartographie der HTW Dresden. Abbildung 3.2 zeigt die Startseite dieses Lernmoduls.

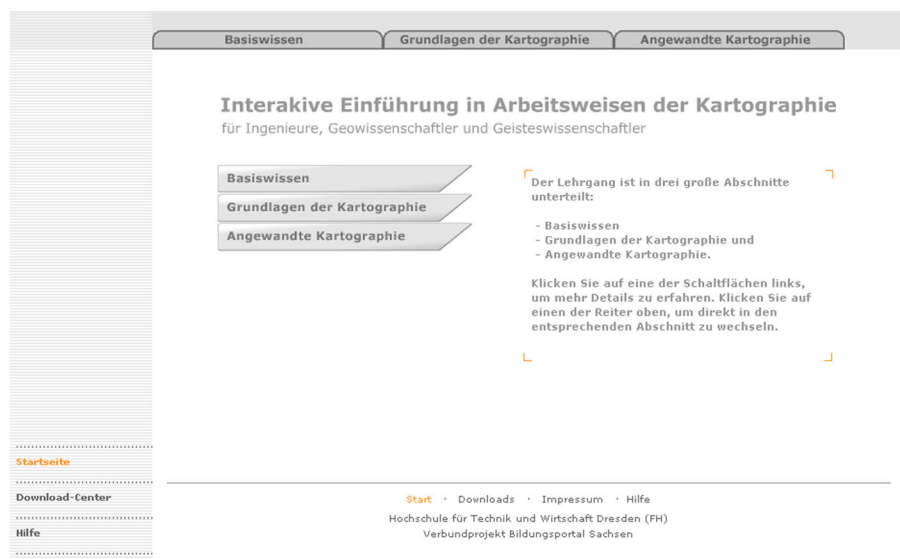


Abb. 3.2:
Startseite des Lernmoduls
'Interaktive Einführung in
die Arbeitsweisen der
Kartographie'

1. Flash MX 2004 ist die aktuelle Version von Flash. Das Lernmodul wurde jedoch mit der Flash MX-Version aus dem Jahre 2003 realisiert, welche zum Zeitpunkt der Erstellung in den Computerlaboren der HTW installiert war.

Die Vorgaben bezüglich der visuellen Gestaltung des Lernmoduls ‚Typografie praktisch!‘ setzen sich zusammen aus der Verwendung der Schriftart *Verdana* und der farblichen Gestaltung durch Grautönen und Orange. Die Farbe Orange, die auch im Logo der HTW Dresden vorkommt, soll dabei die Hochschule repräsentieren.

3.2.5 Rechtliche Aspekte zur Nutzung von Ressourcen

In diesem Abschnitt werden die rechtlichen Aspekte erläutert, die bei der Verwendung verschiedener Materialien beachtet werden müssen. Das in dem Lernmodul ‚Typografie praktisch!‘ vermittelte Wissen zur Typografie und zur praktischen Umsetzung beruht auf Fachliteratur, einem Vorlesungsskript und der eigenen Einarbeitung in das Programm InDesign CS. Das Lernmodul enthält neben selbst erstellten Texten und Grafiken auch Kleinzitate² aus der Fachliteratur. Zudem werden Screenshots und Bildschirmaufzeichnungen aus dem Programm InDesign CS der Firma Adobe verwendet.

Computerprogramme sind ebenso wie Schriftwerke durch das Urheberrechtsgesetz geschützt (§ 2 UrhG). Der Urheber besitzt das ausschließliche Recht zur Vervielfältigung, Verbreitung und Ausstellung des jeweiligen Werkes (§ 15 UrhG). Das Urheberrechtsgesetz regelt jedoch auch Ausnahmen, wie z.B. das Zitatrecht (§ 51 UrhG), welches eine erlaubnisfreie und vergütungsfreie Nutzung von urheberrechtlich geschützten Werken³ und Teilen eines Werks ermöglicht.

Demnach ist die Vervielfältigung, Verbreitung und öffentliche Wiedergabe von Stellen eines Werkes (Kleinzitat) nach der Veröffentlichung in einem selbständigen Sprachwerk⁴ zulässig, wenn dies in einem durch den Zweck gebotenen Umfang geschieht. Dabei ist zu beachten, dass das Werk, in welches ein Zitat integriert wird, selbst einer schöpferischen und urheberrechtlich geschützten Leistung entsprechen muss.

2. Kleinzitate stellen auszugsweise Stellen eines Werks dar, z.B. einzelne Sätze, Gedankengänge oder erklärende Abbildungen.

3. Die Veröffentlichung von ganzen Werken ist im Rahmen von wissenschaftlichen Arbeiten unter bestimmten Bedingungen möglich.

4. Der Begriff *Sprachwerk* beinhaltet Schriftwerke, Computerprogramme und Reden; also auch über den Computer genutzte *Lernprogramme*.

Unter die Definition des Kleinzitats fallen neben Textstellen und einzelnen Abbildungen eines Sprachwerks nach einem Urteil des Landesgerichts Berlin auch Screenshots oder sonstige Auszüge aus Film- oder Multimedia-Werken (LG Berlin, Urteil vom 16.03.2000 - 16 S 12/99). [eLearning and Law o.J.]

Demzufolge ist eine Verwendung von einzelnen Screenshots und Bildschirmmitschnitten aus dem Programm in Design CS sowie die Einbindung einzelner Stellen aus der Fachliteratur als Zitat zulässig. Eine zusätzliche Absicherung bei der Verwendung der Bildschirmbilder aus InDesign war durch die Information der Firma Adobe über die Erstellung des Lernmoduls gegeben [Adobe 2005]. Adobe erhielt dazu neben einer ausführlichen Beschreibung auch eine Version des fertig erstellten Lernmoduls.

3.3 Didaktische Konzeption

Die didaktische Konzeption eines Lernmoduls dient der genauen Festlegung *was* und *wie* gelernt werden soll. Dies beinhaltet neben der genauen Gliederung des Programminhalts unter anderem die Festlegung der Lehrstrategie, der Lernwegsstruktur, des Medieneinsatzes und der Interaktionsformen. [Schreiber 1998, S. 81]

Für die Erstellung des didaktischen Design des Lernmoduls ‚Typografie praktisch!‘ wurde zunächst anhand der vorgegebenen Lernziele und Lerninhalte eine Gliederung des Programminhalts erstellt. Daraus ergab sich eine Unterteilung in die vier Kapitel *Grundlagen der Typografie*, *Schreibregeln*, *Text-Bild-Integration* und *Farbe in Dokumenten*. Zudem gibt es eine *Einleitung*, welche die Lernziele, den Aufbau und die verwendete Symbolik des Lernmoduls beschreibt. Am Ende des Lernmoduls besteht über eine Seite mit *Downloadmöglichkeiten* die Möglichkeit, die Lerninhalte in Form eines PDF-Dokuments und eine Beispiellösung zur praktischen Aufgabe herunterzuladen. Die detaillierte Gliederung des Lernmoduls ist im Anhang A dieser Arbeit einzusehen.

Zudem galt es folgende Aspekte zu beachten:

- Welche Lehrstrategie ist zur Erreichung der Lernziele geeignet?
- Welchen Einfluss hat die Umsetzung nach der SCORM-Spezifikation auf die Programmstruktur? Und wie werden SCORM-Inhalte über das Bildungsportal dargestellt?

- Welche globalen Programmfunktionen werden vom Learning Management System bereitgestellt?

Die folgenden Abschnitte befassen sich mit diesen Fragestellungen und beschreiben die didaktische Konzeption des Lernmoduls.

3.3.1 Lehrstrategie

Lehrtheorien beschäftigen sich damit, mit welchen *Lehrstrategien* Wissen und Fähigkeiten am besten vermittelt werden können. Sie basieren auf Lerntheorien, geben aber konkrete Handlungsanweisungen und bilden somit das Bindeglied zwischen *Lerntheorie* und *Praxis*. [Minass 2002, S. 20]

Eine Lehrstrategie definiert also die konkrete Vorgehensweise zur Erreichung von Lernzielen. Sie legt z.B. fest, wie Lerninhalte unterteilt und in welcher Struktur und Reihenfolge sie dargestellt bzw. vermittelt werden. Bei der Wahl einer Lehrstrategie kann beispielsweise auf eine der klassischen Lerntheorien, wie Behaviorismus, Kognitivismus oder Konstruktivismus, zurückgegriffen werden. Da es sich beim Lernen aber um einen vielschichtigen Prozess handelt, sind zur Bestimmung der konkreten Lehrstrategie meist viele verschiedene Aspekte zu beachten.

Um eine geeignete Lernstrategie für das Lernmodul ‚Typografie praktisch!‘ zu entwickeln wurden zunächst die Lernziele und die Lerninhalte genauer untersucht.

Die *Lernziele* des Lernmoduls lassen sich dem kognitiven Bereich zuordnen (vgl. 3.2.2). Kognitive Lernziele betreffen den Erwerb intellektueller und geistiger Fähigkeiten und werden häufig anhand einer Skala abgestuft. Eine Unterteilung nach *Seufert u.a.* ordnet die kognitiven Lernziele folgenden drei Ebenen zu: [Minass 2002, S. 47]

- *Erarbeiten von Wissen*: dabei wird unterschieden zwischen deklarativem Wissen (wissen, was?) und prozessualem Wissen (wissen, wie?)
- *Anwenden von Wissen*: bedeutet die Reproduktion, in welcher deklaratives und/oder prozessuales Wissen in unveränderter Form und unter gleichen Bedingungen angewandt wird
- *Eigenkonstruktion von Wissen*: steht für die selbständige Schaffung neuen, subjektiven Wissens; Zusammenhänge werden erkannt und eigene Schlüsse gezogen

Diese Unterteilung lässt sich gut auf die Lernziele des Lernmoduls anwenden. In der ersten Ebene stellen die typografischen Regeln theoretisches (deklaratives) Wissen dar, zudem wird praktisches (prozessuales) Wissen anhand der Umsetzungsmöglichkeiten mit dem Programm InDesign beschrieben. Die Lernziele auf dieser ersten Ebene sollen den Lernenden in Form von Beschreibungen und Erklärungen, die sich aus Text, Grafiken und Animationen vermittelt werden.

Die Lernziele der zweiten und dritten Ebene stellen in diesem konkreten Fall die Umsetzung der typografischen Regeln in dem Programm InDesign dar. Dabei kann das vermittelte Wissen zum Teil in unveränderter Form angewandt werden, wie z.B. bei der Anwendung von Schreibregeln. Andere vermittelte Inhalte benötigen aber auch eine Eigenkonstruktion von Wissen, wie z.B. bei der Verwendung von Farbe in Dokumenten. Hier muss je nach Anwendungskontext subjektiv eine Entscheidung getroffen werden. Eine praktische Aufgabe, die direkt in dem Programm InDesign CS umgesetzt wird, soll der Anwendung und der Eigenkonstruktion von Wissen dienen.

Die zuvor beschriebene Gliederung der *Lerninhalte* entspricht in ihrer Reihenfolge im Großen und Ganzen der allgemeinen Vorgehensweise bei der Erstellung eines Dokuments in InDesign. So wird z.B. in der Regel zunächst das Seitenformat und der Satzspiegel (dies wird im ersten Kapitel erklärt) festgelegt und erst danach Text, Bildmaterial und Farbe integriert.

Jedes Kapitel wird am Ende mit einer praktischen Übungsaufgabe abgeschlossen (vgl. 3.2.2). Das Lernmodul besitzt also entsprechend den Kapiteln vier Aufgabenstellungen, die als gemeinsames Ziel die Erstellung eines Dokuments mit dem Programm InDesign CS haben.

Um diesen praktischen Bezug beizubehalten, wurde für die Erstellung der Programmstruktur die Reihenfolge der Gliederung der Lerninhalte übernommen. Das Lernmodul besitzt somit eine lineare Struktur, in der ein Kapitel nach dem anderen behandelt wird und jeweils mit einer praktischen Aufgabe abschließt. Die einzelnen Kapitel setzen sich aus Unterkapiteln (Lernobjekten) zusammen, in denen der theoretische Lernstoff präsentiert wird.

Die genaue Aufteilung der Lerninhalte in Lernobjekte und die Navigation innerhalb des Lernmoduls werden durch das SCORM-Modell Version 1.2 beeinflusst und in den Abschnitten 3.3.2 und 3.3.3 dieser Arbeit beschrieben.

3.3.2 Einteilung der Lerninhalte in Lernobjekte

SCORM beschreibt mit seinem Content Aggregation Model (vgl. 1.4.1) ein Modell zur Erstellung von Lernobjekten und zur Strukturierung dieser anhand der so genannten Content Structure. Dabei repräsentieren Assets und SCOs Inhalte, die von einem LMS gestartet bzw. dargestellt werden können. Jedoch können nur SCOs über die Laufzeitumgebung mit dem LMS kommunizieren. Dies bedeutet, dass nur bei der Verwendung von SCOs Lernerdaten über die Nutzung des Lerninhalts gespeichert werden können.

Aus diesem Grund wurden die einzelnen Teile des Lernmoduls als SCOs entwickelt. Diese sollen im Sinne der Wiederverwendbarkeit einen unabhängigen Lerninhalt repräsentieren. Dazu empfiehlt ADL, die Größe eines SCOs soll auf die kleinstmögliche, sinnvolle Lerneinheit begrenzt sein.

Bei der Festlegung von kleinstmöglichen, sinnvollen Lerneinheiten erschien es als angebracht, jeweils theoretische typografische Regeln und deren praktische Umsetzung in dem Programm InDesign als eine Einheit zusammenzufassen, da dadurch der Bezug von der Theorie zur Praxis gegeben wird.

Als Beispiel wurde das Thema *Seitenformate* (entspricht dem Abschnitt 1.2 der Inhaltsgliederung) als eine Lerneinheit bzw. ein Lernobjekt mit folgenden Lernschritten definiert:

- Schritt 1: Festlegung des Seitenformats (beinhaltet theoretische typografische Regeln)
- Schritt 2: Einrichten des Seitenformats in InDesign (beschreibt die praktische Umsetzung in InDesign)

In manchen Fällen besteht ein Lernobjekt gänzlich aus theoretischen Inhalten und Beispielen ohne konkrete praktische Umsetzungsmöglichkeit in InDesign. Dies ist meist dann der Fall, wenn es um allgemeine gestalterische Themen, wie z.B. *Farbharmonien und Farbkontraste* geht.

Um die Lerneinheiten möglichst klein zu halten, wurde die maximale Anzahl von Lernschritten pro Objekt auf vier begrenzt. Nicht nur aufgrund der ADL-Empfehlung sondern auch aus didaktischer Sicht erschien diese Begrenzung als angemessen. Pro Lernobjekt wird somit eine überschaubare Anzahl von Einzelschritten präsentiert, wodurch der Bezug vom ersten bis zum letzten Schritt leicht nachvollziehbar bleibt.

3.3.3 Navigation zwischen den Lernobjekten

Die Darstellungsstruktur bzw. Darstellungsreihenfolge der einzelnen Lernobjekte wird in einem SCORM-Modell durch die Content Structure (vgl. 1.4.1) festgelegt und durch das LMS über einen so genannten Launch-Mechanismus (vgl. 1.4.2) geregelt. Dabei liegt es in der Hand des Learning Management Systems wie die Darstellung und die Navigation umgesetzt werden. ADL unterscheidet grob zwischen einer sequentiellen, einer nutzer-definierten und einer adaptiven Darstellung.

Die sequentielle (sequential) Darstellung erfolgt der Reihe nach. Der Nutzer besitzt hier z.B. die Möglichkeit über Vor- und Zurück-Funktionen innerhalb dieser Reihenfolge zu navigieren. Bei einer nutzer-definierten (user-directed) Darstellungsreihenfolge hat der Nutzer die Möglichkeit aus einem Menü ein beliebiges Lernobjekt auszuwählen. Eine adaptive (adaptive) Darstellung passt die Reihenfolge der Lerneinheiten dem Lernfortschritt an (vgl. 1.4.2).

Auch schreibt ADL nicht vor, ob so genannte *prerequisites*, über die Einfluss auf die Navigation genommen werden kann, von einem LMS unterstützt werden müssen (vgl. 1.4.1).

Saba stellt SCORM-Inhalte in Form eines Auswahlmenüs, welches anhand der Content Structure generiert wird, dar. Da Saba keine *prerequisites* unterstützt (vgl. 2.2.4), hat der Nutzer somit die Möglichkeit, sich jedes beliebige Lernobjekt aus dem Menü anzeigen zu lassen, ohne dass eine bestimmte Reihenfolge vorgeschrieben werden kann.

Die im Abschnitt 3.3.1 beschriebene *lineare Struktur* des Lernmoduls kann somit durch die Content Structure definiert und angezeigt werden, der Nutzer besitzt aber die Möglichkeit, die Anzeigereihenfolge frei zu wählen.

Abbildung 3.3 verdeutlicht die Struktur des Lernmoduls. Lernobjekte stellen darin die vom Nutzer auswählbaren Lerneinheiten dar.

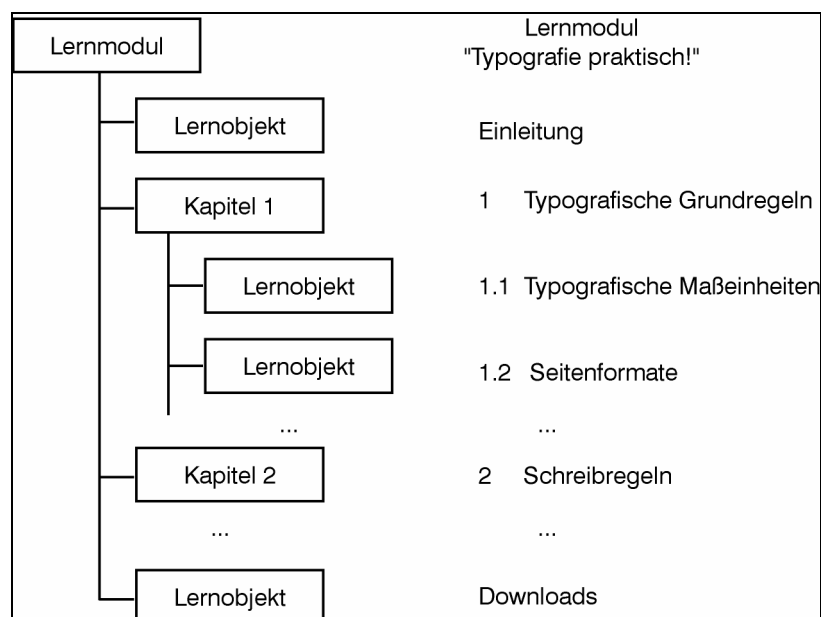


Abb. 3.3:
Struktur des Lernmoduls
'Typografie praktisch!'

3.3.4 Interaktionsmöglichkeiten

Die Interaktionsmöglichkeiten eines Lernmoduls stehen in engem Zusammenhang mit den Navigationsmöglichkeiten. Eine mögliche Definition des Begriffs Interaktivität in Zusammenhang mit E-Learning wurde bereits im Abschnitt *1.1 Begriffserklärung E-Learning* gegeben. Dabei wird zwischen folgenden Interaktivitätsformen unterschieden:

- *Steuernde Interaktivität*: zur Steuerung des Programmablaufs
- *Didaktische Interaktivität*: zur Gestaltung des Lernerlebnisses

Didaktische Interaktivität kann sowohl programmgesteuert als auch durch Feedback mit anderen Menschen (Lehrende und/oder Lernende) geschehen [Baumgartner u.a. 2002, S. 17]. Bei der Bereitstellung von Lerninhalten über ein LMS in Form des SCORM-Modells wird die didaktische Interaktivität auch durch das LMS mitbestimmt. So hat zum Beispiel der Entwickler eines SCORM-Lerninhalts keinen Einfluss darauf, welche Kommunikationsmöglichkeiten durch ein LMS bereitgestellt werden und ob eine didaktische Steuerung, z.B. durch prerequisites, vom LMS unterstützt wird.

Im Bildungsportal Sachsen werden SCORM-Inhalte als frei auswählbares Menü dargestellt, somit ist die programmgesteuerte didaktische Interaktivität auf die Ebene der einzelnen Lernobjekte begrenzt. Da im Lernmodul ‚Typografie praktisch!‘ auf dieser Ebene jedoch Lerninhalte nur präsentiert und nicht abgefragt werden, wurde auf eine programmgesteuerte didaktische Interaktivität gänzlich verzichtet.

Auch direktes Feedback durch Personen während des Lernens mit dem Lernmodul ist nicht vorgesehen, da das Lernmodul im Bildungsportal ein Selbstlernmodul darstellt, mit dem individuell und jederzeit gelernt werden kann. Dennoch besteht die Möglichkeit über ein Forum mit anderen Lernenden zu Kommunizieren.

Die Interaktionsmöglichkeiten des Lernmoduls ‚Typografie praktisch!‘ beschränken sich also somit auf eine steuernde Interaktivität, die wie bereits beschrieben, die Navigation zwischen den einzelnen Lernobjekten betrifft. Es sind aber auch Interaktionsmöglichkeiten innerhalb eines Lernobjekts vorhanden. Diese betreffen folgende Punkte:

- Navigation zwischen den einzelnen Lernschritten innerhalb eines Lernobjekts
- Auswählen und Anzeigen eines Themas bzw. einer Beschreibung innerhalb eines Lernschritts
- Anzeigen von Tipps, Tastaturkürzel und Beispielfilmen zu einem bestimmten Thema innerhalb eines Lernschritts

3.3.5 Medieneinsatz

Medien werden seit jeher zur Vermittlung von Informationen und Wissen eingesetzt. Im Bereich E-Learning stellen digitale Medien die Schnittstelle zwischen den zu vermittelten Lerninhalten und den Lernenden dar und gehören neben der Lehrstrategie, den Eigenschaften des Lerners und des Inhalts zu den wichtigsten Komponenten für erfolgreiches Lernen. Dabei ist die Auswahl geeigneter Medien genauso wichtig wie ein, den Lernzielen entsprechender, Einsatz.

Die einzelnen Lernobjekte des Lernmoduls ‚Typografie praktisch!‘ dienen der Vermittlung von deklarativem Wissen (wissen, was?) und prozeduralem Wissen (wissen, wie?). Es sollen also theoretische Sachverhalte vermittelt und praktische Anwendungsbeispiele aufgezeigt werden. Dazu wurden die im Folgenden beschriebenen Medien ausgewählt.

Text:

Text hat auch im Multimedia-Zeitalter noch eine große Bedeutung bei der Vermittlung von Wissen. Im Lernmodul ‚Typografie praktisch!‘ wird Text sowohl zur Vermittlung von deklarativem als auch von prozessuellem Wissen eingesetzt und stellt einen wichtigen Bestandteil dar.

Bei der Darstellung von Text über digitale Medien gelten jedoch andere Regeln als bei gedrucktem Text. Dies ist vor allem dadurch begründet, dass das Lesen am Bildschirm viel mühsamer ist als das Lesen eines gedruckten Textes, was vermutlich auf die Bildfrequenz (Flimmern) und auf eine niedrigere Auflösung als beim Druck zurückzuführen ist.

Aus diesem Grund und auch um eine gute Übersichtlichkeit zu gewährleisten wurde bei der Erstellung des Lernmoduls darauf geachtet, Text möglichst sparsam zu verwenden und in kleinen Textblöcken darzustellen. Zudem werden Texte möglichst häufig durch die Nutzung weiterer Medien unterstützt.

Grafiken:

Verschiedene Studien zeigten, dass Informationen aus Bildern und Grafiken schneller und einfacher entnommen werden können als aus Texten. Allerdings werden Bilder manchmal oberflächlicher als Text wahrgenommen und verarbeitet und lassen zudem häufig Mehrdeutigkeiten zu. [Niegemann u.a. 2004, S. 176]

Aus diesem Grund ist beim Einsatz von erklärenden Grafiken immer auf die richtige Kombination mit Text zu achten. Bei jeder zu vermittelnden Information ist aufs Neue zu prüfen, in wie weit eine Grafik oder Abbildung den Text unterstützen oder ersetzen kann. Im Lernmodul ‚Typografie praktisch!‘ werden schematische Grafiken sowohl zur Unterstützung von Text als auch als Ersatz für lange erklärende Texte eingesetzt. Da es sich bei dem theoretischen Lerninhalt zum Thema Typografie in der Regel um grafisch darstellbare Inhalte handelt, eignen sich Grafiken sehr gut zur Informationsvermittlung. So ist es z.B. für Lernende und Lehrer viel einfacher, wenn anhand einer Grafik gezeigt wird, wo sich der Satzspiegel auf einer Seite befinden sollte, als das anhand vieler Worte zu erklären.

Zudem wird nach *Ballstaedt (1997)* Lernmaterial mit Bildern von Lernenden positiver eingeschätzt als solches ohne Bilder. Auch zeigten *Levie und Lentz (1982)* und *Levin u.a. (1987)*, dass sich Bilder positiv auf das Lernen und die Lernleistung auswirken. [Niegemann u.a. 2004, S. 176]

Der Einsatz von Abbildungen und Grafiken kann somit auch neben der reinen Informationsvermittlung auch eine motivierende Funktion besitzen.

Screenshots:

Screenshots stellen Bildschirmfotos dar, die den Bildschirminhalt zu einem bestimmten Zeitpunkt darstellen. Sie werden an dieser Stelle separat behandelt, da sie im Lernmodul ‚Typografie praktisch!‘ eine besondere Funktion übernehmen.

Bis auf wenige Ausnahmen wurden Screenshots aus dem Programm InDesign CS als Interaktionsflächen eingesetzt um die Umsetzung der theoretischen typografischen Regeln in InDesign zu vermitteln. Bei Mausklick auf eine solche Interaktionsfläche wird eine Beschreibung der entsprechenden Funktion oder Einstellungsmöglichkeit angezeigt.

Abbildung 3.4 gibt ein Beispiel für diese Einsatzmöglichkeit und zeigt der Reihe nach die Zustände ‚Inaktiv‘, ‚Cursor darüber‘ und ‚Aktiviert‘.

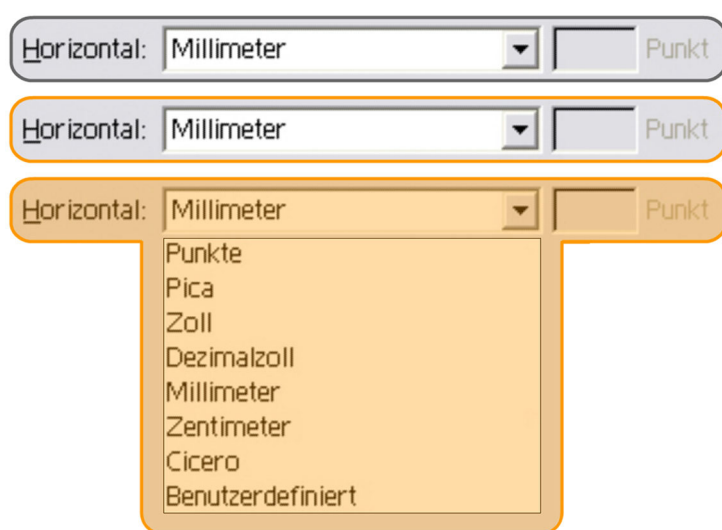


Abb. 3.4:
Einsatz von Screenshots als
Interaktionsflächen

Die Verwendung von Screenshots zur Erstellung der Interaktionsflächen soll den praktischen Bezug zum Programm InDesign CS gewährleisten.

Videosequenzen:

Dem Einsatz von Video als Informationsvermittler wird im Vergleich zu reinem Bild oder Ton eine Reihe von Vorteilen zugesprochen. Dazu zählen unter anderem die *Informationsdichte*, welche die Vermittlung vieler detaillierter Informationen ermöglicht, die *Realitätsnähe* und der *Zeitbezug*, welcher die Abbildung eines Vorgangs in Echtzeit ermöglicht. Zudem haben Studien gezeigt, dass der zusätzliche Einsatz von Video neben anderen Informationsträgern wie Text und Bild zu einer verbesserten Behaltens- und Verstehensleistung bei Lernenden führt. [Niegemann u.a. 2004, S. 148]

Im Lernmodul ‚Typografie praktisch!‘ werden kurze Videosequenzen mit erklärendem Sprechertext eingesetzt, um verschiedene Arbeitsschritten bei der Nutzung des Programms InDesign CD zu verdeutlichen. Die verwendeten Videosequenzen zeigen in Form von Bildschirmaufzeichnungen Funktionen und Einstellungsmöglichkeiten des Programms in ihrem realen Kontext. Der Einsatz der Videosequenzen soll die oben beschriebene Behaltens- und Verstehensleistung unterstützen.

Eine Optimierung des Lernprozesses durch Video ist jedoch in der Regel nur dann zu erwarten, wenn der Lernende die Videosequenz als Lernchance sieht und nicht eine passive ‚Konsumhaltung‘ einnimmt. Eine Möglichkeit dies zu verhindern, besteht darin, dem Lernenden ein gewisses Maß an Kontrolle über den Ablauf des Videos zu geben. [Niegemann u.a. 2004, S.151ff]

Aus diesem Grund wurden die Videosequenzen des Lernmoduls mit Funktionen zum *Start*, *Pause*, *Start von Anfang* und *Schieberegler* zum Abspielen ab einer bestimmten Position, sowie einem *Lautstärkeregler* ausgestattet.

3.3.6 Drehbuch

Das Drehbuch einer Lernanwendung dient der detaillierten Beschreibung des Inhalts und des Aufbaus der einzelnen Bildschirmseiten. Es enthält neben den ausformulierten Texten auch konkrete Angaben zu allen weiteren verwendeten Medien (Grafiken, Ton, Video, etc.), zu Interaktions- und Navigationsmöglichkeiten und, wenn vorhanden, auch zu Feedbackfunktionen [Schreiber 1998, S. 119f] und dient in der Regel als Arbeitsgrundlage für die technische Entwicklung eines Lernmoduls.

Vor der Erstellung des Drehbuchs des Lernmoduls ‚Typografie praktisch!‘ galt es, einen so genannten *Basaltext* zu entwickeln, welcher der *vollständigen* und *redundanzfreien* Darstellung des Inhalts dient. Dieser soll alle zielbezogenen oder in anderer Hinsicht für den Inhalt bedeutsamen Informationen festhalten [Schreiber 1998, S. 117] und somit als Grundlage für die Erstellung des Drehbuchs dienen.

Der Basaltext des Lernmodul ‚Typografie praktisch!‘ beschreibt dazu in Form einer Tabelle zu jedem Lernschritt kurz den entsprechenden Inhalt. Tabelle 3.1 zeigt einen Ausschnitt aus dem Basaltext.

Tab. 3.1: Ausschnitt aus dem Basaltext des Lernmoduls ‚Typografie praktisch!‘

1.1 Typografische Maßeinheiten	Schritt 1: <u>Wichtige Maßsysteme in der Typografie:</u>	Vorstellung der Maße Didot-Maß, Pica-Maß, DTP-Punkt und Metrisches Maß. Durch Interaktion wird auf Besonderheiten des ausgewählten Maßes und seine Größe (in Millimeter) eingegangen.
	Schritt 2: <u>Voreinstellungen für Einheiten und Einteilungen in InDesign:</u>	Erklärung, wo man in InDesign die Voreinstellungen für „Einheiten und Einteilungen“ treffen kann („ <i>Bearbeiten > Voreinstellungen > Einheiten und Einteilungen</i> “). Durch Interaktion kann man mehr zu den Eingabemöglichkeiten für Lineareinheiten (Ursprung, Horizontal, Vertikal) und 2.) und Tastaturschritte (Pfeiltasten, Schriftgrad/Zeilenabstand, Grundlinienversatz, Kerning) erfahren.

Zur Erstellung des Drehbuchs des Lernmoduls ‚Typografie praktisch!‘ wurden so genannte *Formblätter* genutzt, die den Inhalt und die Form einer Bildschirmseite repräsentieren. Formblätter setzen sich aus verschiedenen ausfüllbaren Bereichen zusammen, typisch dafür sind *Identifikation* (Einträge über Autor, Datum, Projekt, Bildschirmnummer, etc.), *Layout* (Texte,

Grafiken, Eingabemöglichkeiten, etc.) und *Anweisungen* (Angaben zur Ablaufsteuerung, Darstellungsart, Überblendung, etc.). Die Abbildung 3.5 zeigt den Aufbau eines Formblatts des Lernmoduls ‚Typografie praktisch!‘. Das gesamte Drehbuch des Lernmoduls samt Basaltext wurde dieser Arbeit als separater Anhang beigelegt.

Das Diagramm zeigt den Aufbau eines Formblatts. Oben befindet sich ein blauer Rahmen mit Metadatenfeldern: 'Projekt:', 'Autor:', 'Datum:', 'Kapitel:', 'Unter-kapitel:', 'Schritt:' und 'Bildschirm-seite:'. Darunter befindet sich ein Feld für 'aktueller Bildschirm:'. Der Hauptbereich ist in zwei farbige Blöcke unterteilt: einen großen roten Rahmen für 'Layout' und einen kleineren grünen Rahmen für 'An-weisungen:'.

Abb. 3.5:
Aufbau der Formblätter des Lernmoduls ‚Typografie praktisch!‘

3.4 Gestaltung und Evaluation der Benutzeroberfläche

Sobald die didaktische Konzeption feststeht, kann die Gestaltung der Benutzeroberfläche beginnen. Dazu gehört sowohl die grafische Gestaltung der Benutzeroberfläche (vgl. 3.4.2) als auch die Festlegung der Bedienfunktionen (vgl. 3.4.1).

Im Anschluss an die Gestaltung und einer prototypischen Umsetzung ist es sinnvoll, im Rahmen von Usability Tests und Evaluationen die Benutzeroberfläche auf ihre Tauglichkeit zu überprüfen. Das Wort *Usability* (Gebrauchstauglichkeit) bezeichnet die Eignung einer Sache oder eines Gutes in Bezug auf seinen Verwendungszweck und wird in Zusammenhang mit Computeranwendungen meist mit Effektivität, Effizienz und auch der Benutzerfreundlichkeit der grafischen Mensch-Computer-Schnittstelle in Zusammenhang gebracht. [Wikipedia 2005]

Usability stellt eine der Grundvoraussetzungen für erfolgreiches E-Learning dar. So ergab z.B. eine Studie von *Hara & Kling (2000)*, dass sich Probleme des Interfacedesigns negativ auf die Motivation beim webbasierten Lernen auswirken und die Interaktion erschweren. Ziel eines Usabilitytests ist es, die Schwachstellen eines Systems bzw. die Probleme, die Nutzer mit ihm haben, herauszufinden [Niegemann u.a. 2004, S. 314f]. Anhand der Testergebnisse lassen sich somit gezielte Korrekturen vornehmen.

In den Abschnitten 3.4.3 und 3.4.4 werden ein Nutzertest und eine Evaluation durch Experten beschrieben, die sich mit der Benutzerfreundlichkeit der grafischen Schnittstelle des Lernmoduls ‚Typografie praktisch!‘ beschäftigen. Diese wurden zu einem frühen Zeitpunkt der Produktionsphase vorgenommen und bewerten einen Prototyp des Lernmoduls, welcher zu jedem der vier Inhaltskapitel mindestens ein lauffähiges und mit Inhalten gefülltes Lernobjekt mit allen Lernschritten und Interaktionsmöglichkeiten (abgesehen von den Beispielfilmen) besaß. Zudem war die Einleitungsseite mit der Beschreibung der Lernziele, dem Aufbau und der verwendeten Symbole vorhanden. Der Zugriff auf die einzelnen Lernobjekte (Flash-Filme) erfolgte über eine HTML-Seite, welche als Ersatz für die vom LMS generierte Inhaltsstruktur diente.

Zunächst folgt aber die Beschreibung der funktionellen und grafischen Gestaltung der Benutzeroberfläche.

3.4.1 Funktionelle Gestaltung

Bei der funktionellen Gestaltung des Lernmoduls ‚Typografie praktisch!‘ kann zwischen verschiedenen Ebenen unterschieden werden. Dieser Abschnitt beschreibt zuerst *globale Programmfunktionen*, die durch das Learning Management System bereitgestellt werden und für den Lernenden immer global verfügbar sind. Im Anschluss daran werden Funktionen erläutert, die in den einzelnen Lernobjekten selbst realisiert wurden. Dabei handelt es sich um Funktionen zur *Navigation*, *Interaktion* und *Filmsteuerung*.

Globale Programmfunktionen:

Das LMS des Bildungsportals übernimmt für des Lernmodul ‚Typografie praktisch!‘ folgende Funktionen:

- Darstellung eines *Inhaltsverzeichnisses* in Form eines *Auswahlmenüs* zum Starten der einzelnen Lernobjekte (vgl. 3.3.3)
- Lernfortschrittsanzeige
- Kommunikationsforum zum Austausch mit anderen Lernenden

Abbildung 3.6 zeigt einen Ausschnitt aus einem durch das LMS erzeugten Inhaltsverzeichnisses und der dazugehörenden Lernfortschrittsanzeige. Nutzer des Lernmoduls ‚Typografie praktisch!‘ können über die Werte für *Abschlussstatus*, *Benötigte Zeit* und *Versuche gesamt* nachvollziehen, ob schon alle Lernschritte eines Lernobjekts angezeigt wurden, wie lange mit einer Lernobjekt gearbeitet wurde und wie oft eine Lernobjekt bereits gestartet wurde.

Inhalte	Abschlussstatus	Ergebnis	Zum Bestehen erforderlich	Benötigte Zeit (in Sek)	Versuche gesamt
Typografie praktisch!					
<u>Lernziele, Aufbau und Symbole dieses Kurses</u>	Abgeschlossen		0	20 sec.	2
Grundlagen der Typografie					
<u>Einführung</u>	Abgeschlossen		0	9 sec.	2
<u>1.1 Typografische Maßeinheiten</u>	Abgeschlossen		0	10 min. 43 sec.	9
<u>1.2 Seitenformate</u>	Unvollständig		0	18 sec.	1
<u>1.3 Satzspiegel und Gestaltungsraster</u>	Unvollständig		0	42 sec.	2
<u>1.4 Absatzuntergliederung</u>	Unvollständig		0	1 min. 10 sec.	1
<u>1.5 Zitate, Marginalien und Fußnoten</u>	Abgeschlossen		0	1 min. 41 sec.	2

Abb. 3.6: Ausschnitt des Inhaltsverzeichnisses und der Lernfortschrittsanzeige

Navigationsfunktion:

Wie im Abschnitt 3.3.2 *Einteilung der Lerninhalte in Lernobjekte* bereits beschrieben, kann sich ein Lernobjekt aus bis zu vier Lernschritten zusammensetzen. Auch wurde, wie im Abschnitt 3.3.4 *Interaktionsmöglichkeiten* beschrieben, festgelegt, dass im Lernmodul gänzlich auf didaktische Interaktion verzichtet wird. Das heißt, es wird kein Lernweg zwischen den einzelnen Schritten festgelegt, sondern der Nutzer kann sich die Lernschritte nach Belieben anzeigen lassen. Diese sind durch kleine runde Schaltflächen, die zur Orientierungshilfe mit einer Ziffer gekennzeichnet sind, zu erreichen. Bei Mouse-Over über eine Schaltfläche, verändert diese ihre

Größe und wird bei Aktivierung farblich gekennzeichnet. Abbildung 3.7 zeigt eine Navigationsleiste, bei welcher der zweite Lernschritt aktiviert wurde.



Abb. 3.7: Navigationsleiste

Interaktionsfunktionen:

Die im Folgenden beschriebenen Interaktionsfunktionen beziehen sich auf die Funktionen innerhalb eines Lernschritts. Diese wurden im Abschnitt 3.3.4 *Interaktionsmöglichkeiten* bereits erwähnt. Innerhalb eines Lernschritts gibt es die Möglichkeit sich zu verschiedenen Themenpunkten jeweils mehr Informationen anzeigen zu lassen. Dabei lässt sich zwischen theoretischen Themen für die Vermittlung typografischer Regeln und der praktischen Umsetzung mit dem Programm InDesign unterscheiden. Ein Lernschritt behandelt entweder ein theoretisches oder ein praktisches Thema.

Bei theoretischen Themen werden die Interaktionsmöglichkeiten in Textform dargestellt. Zur Verdeutlichung, dass es sich um eine Interaktionsmöglichkeit handelt, ist der Text, wie es für Hyperlinks typisch ist, unterstrichen (vgl. Abb. 3.8). Zudem verändert der Text seine Farbe, wenn die Maus über ihn fährt und wenn er aktiviert wurde.

Bei praktischen Themen dienen Schaltflächen, die Screenshots aus dem Programm InDesign CS darstellen, dem Anzeigen der entsprechenden Themenpunkte (vgl. Abb. 3.9). Auch diese werden bei Mouse-Over und bei ihrer Aktivierung farblich gekennzeichnet (vgl. 3.3.5).

1:1,414 (DIN-A-Serie)

1:1,5

1:1,538

1:1,618 (Goldener Schnitt)

1:1,732

1:2,236



Abb. 3.8:
Interaktionmöglichkeit
bei theoretischem Inhalt

Abb. 3.9:
Interaktionsflächen
bei praktischem Inhalt

Zudem soll es für die einzelnen Themenpunkte die Möglichkeit geben, sich zusätzliche Tipps, Tastaturkürzel und/oder einen Beispielfilm anzeigen zu lassen. Diese Funktionen sind nicht für jeden Themenpunkt in gleichem Maße verfügbar. Um jedoch ein einheitliches Design zu gewährleisten bleiben alle drei Symbole immer an selber Position auf dem Bildschirm sichtbar, sind aber durch ihre farbliche Gestaltung als ‚aktivierbar‘ oder ‚nicht aktivierbar‘ gekennzeichnet. Der nicht aktivierbare Zustand wird in einem heller grau gekennzeichnet, und fällt dadurch kaum auf. Der aktivierbare Zustand verändert bei Mouse-Over seine Größe und bei Aktivierung seine Farbe. Abbildung 3.10 zeigt die verwendete Symbolik in einem Beispiel, in dem Tastaturkürzel und Beispielfilm aktivierbar sind. Das ganz links abgebildete Symbol für zusätzliche Tipps ist in diesem Fall nicht aktivierbar.

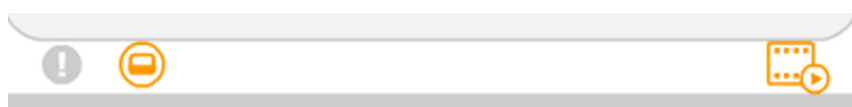


Abb. 3.10:
Schaltflächen zum Anzeigen
zusätzlicher Informationen

Filmsteuerung:

Für die Filmsteuerung von Videosequenzen gibt es wie bereits im Abschnitt 3.3.5 *Medieneinsatz* kurz erwähnt folgende Funktionen:

- Start/Pause: Start- und Pausefunktion sind in einer Schaltfläche realisiert, die je nach Zustand ihre Funktion wechselt
- Start von Anfang: startet das Abspielen erneut vom Anfang der Videosequenz an
- Schiebereglern: ermöglicht das Abspielen ab der ausgewählten Position
- Lautstärkereglern: ermöglicht die Regulierung der Lautstärke des Sprechertextes

Zusätzlich gibt es eine Schaltfläche für das Schließen des Filmfensters, das nur zum Abspielen von Videosequenzen eingesetzt wird. Abbildung 3.11 zeigt einen Ausschnitt aus einem solchen Fenster.

Abgesehen von den Reglern für Lautstärke und Abspielposition reagieren die Funktionen auf Mouse-Over durch die Veränderung ihrer Größe.

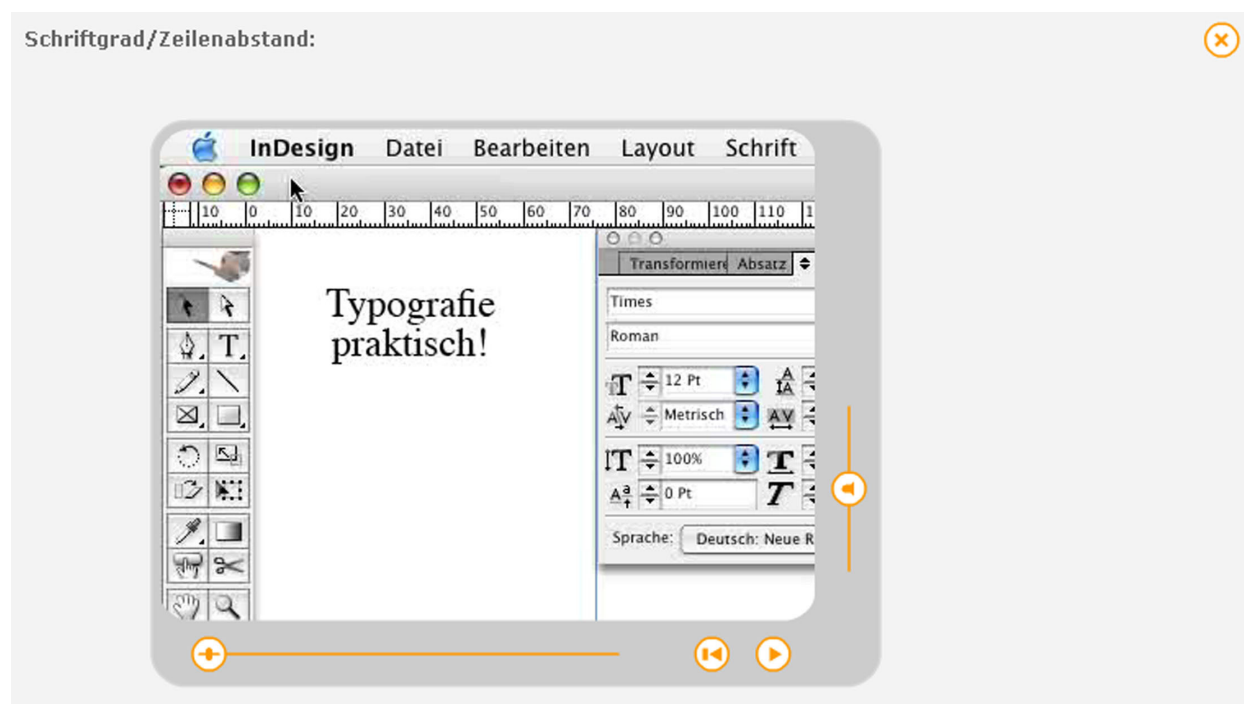


Abb. 3.11: Ausschnitt aus einem Fenster zum Abspielen von Filmsequenzen

3.4.2 Screendesign

Die Entwicklung des Screendesigns beginnt in der Regel mit der Einteilung des Bildschirms in verschiedene Funktionsbereiche. Dabei ist in der Regel die Einteilung in folgende drei Bereiche ausreichend: [Schreiber 1998, S. 334f]

- Arbeitsbereich: dient der eigentlichen Darstellung des Inhalts oder der Interaktionsmöglichkeiten; nimmt den größten Bildschirmbereich ein
- Orientierungsbereich: dient der Orientierung des Lerners; macht die aktuelle Position innerhalb der Programmstruktur deutlich
- Steuerungsbereich: dient der Darstellung globalen Funktionen und der Navigation

Da bei der Veröffentlichung des Lernmoduls ‚Typografie praktisch!‘ die Funktion für das Navigieren zwischen den einzelnen Lernobjekten vom Lernmanagement übernommen wird, muss diese Funktion nicht in den Steuerbereich realisiert werden. Dieser Bereich enthält in dem konkreten Fall also nur die Funktion zur Navigation zwischen den einzelnen Lernschritten.

Bei der Erstellung des Orientierungsbereichs des Lernmoduls muss beachtet werden, dass es sich bei den einzelnen Lernobjekten um SCOs, also um wiederverwendbare Lerneinheiten handelt. Es würde gegen die Wiederverwendbarkeit sprechen, innerhalb eines SCOs dessen genaue Position innerhalb eines Lernmoduls anzuzeigen, da das SCO in einem anderen Zusammenhang vermutlich auch eine andere Position besitzt. Aus diesem Grund werden als Orientierungshilfen innerhalb eines Lernobjekts jeweils nur das Thema des Lernobjekts sowie der aktuelle Titel des Lernschritts angezeigt.

Die Abbildung 3.12 zeigt die Einteilung der Fläche des Lernmoduls ‚Typografie praktisch!‘ in Arbeits-, Orientierungs- und Steuerungsbereich. Dabei übernehmen die einzelnen Bereiche folgende Funktionen:

- 1a: Arbeitsbereich zur Darstellung von groben Informationen und Interaktionsmöglichkeiten; Vertiefung der Informationen erfolgt durch Interaktion
- 1b: Darstellung von weiterführenden Informationen; Anzeige erfolgt durch Interaktion im Bereich 1a
- 1c: Darstellung von Schaltflächen zur Anzeige von zusätzlichen Tipps, Tastaturkürzeln und zum Öffnen des Beispielfilms

- 2a: Orientierungsbereich zur Anzeige des Themas des Lernobjekts
- 2b: Anzeige des Themas und der Nummer (1.,2.,3. oder 4.) des Lernschritts
- 3: Bereich zur Darstellung der Navigationsschaltflächen; dient zusätzlich zur Orientierung

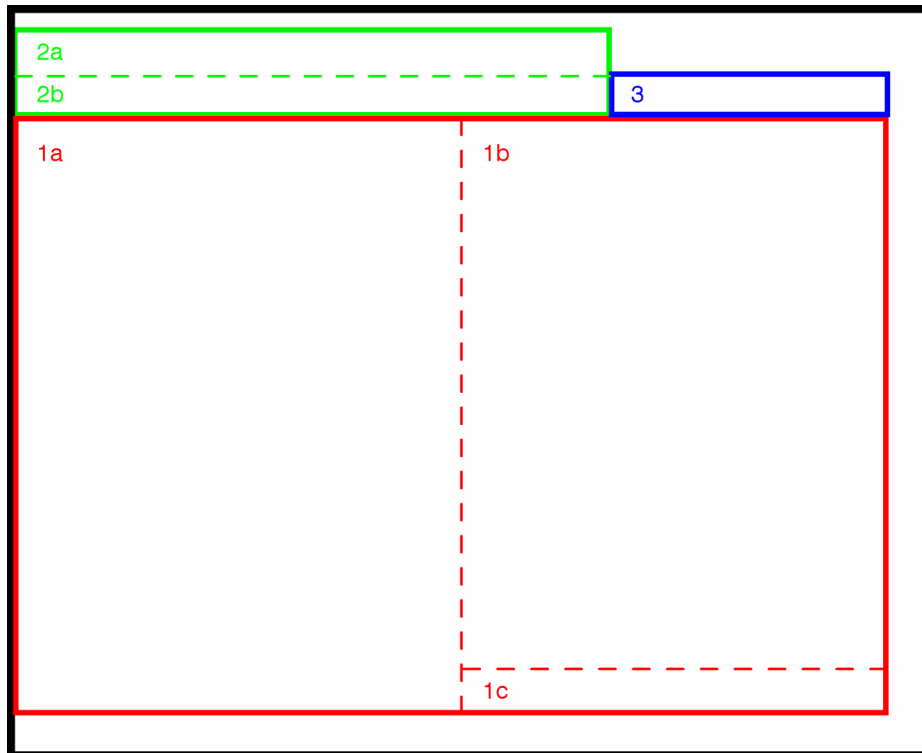


Abb. 3.12:
Grobe Einteilung des Lernmoduls in Arbeitsbereich (1), Orientierungsbereich (2) und Steuerungsbereich (3)

Basierend auf dieser Bereichseinteilung, den umzusetzenden Funktionalitäten und den Vorgaben bezüglich des visuellen Designs und auch der Zielgruppe wurde das Screendesign entwickelt.



Abb. 3.13: Screendesign des Lernmoduls ‚Typografie praktisch!‘

Die Idee der farblichen Gestaltung durch Grautöne und Orange und die Verwendung der Schriftart *Verdana* wurden dem Lernmodul ‚Interaktive Einführung in die Arbeitsweisen der Kartographie‘ entnommen und durch den Auftraggeber vorgegeben (vgl. 3.2.4). Weitere gestalterische Elemente wie die Linien im Hintergrund und die Verwendung von abgerundeten Formen sind auch in dem Kartographie-Lernmodul enthalten. Wobei im Lernmodul ‚Typografie praktisch!‘ die Verwendung von abgerundeten Formen konsequent durchgezogen wurde.

Die folgenden Zielsetzungen hatten einen maßgebenden Einfluss auf die der gestalterischen Entwicklung des Lernmoduls:

- *zeitgemäßes und ansprechendes, aber dennoch ruhiges und stabiles Design*: ein ansprechendes Design soll zum einen zum Lernen bzw. zur Nutzung des Lernmoduls anregen zum anderen sollte das Design auch nicht zu sehr vom Lerninhalt ablenken; die konsequente Ver-

- wendung von abgerundeten Formen passt sowohl in den derzeitigen Zeitgeschmack, sorgt aber auch dafür, dass das Lernmodul eine Einheit ergibt und somit ein ruhiges und stabiles Erscheinungsbild besitzt; kleine Nuancen, wie z.B. die feinen Linien im Hintergrund, Farbabstufungen und Farbverläufe lassen das Design dennoch interessant aussehen
- übersichtliche Gliederung des Bildschirms: der Arbeitsbereich stellt den größten und hellsten Bereich des Lernmoduls dar und zieht dadurch die Aufmerksamkeit auf sich; die anderen Bereiche sind klar vom Arbeitsbereich getrennt
 - einfache Bedienbarkeit und konsistente Darstellungsweise: eine konsistente Darstellungsweise unterstützt die Bedienbarkeit eines Programms, da der Lernende sich nicht immer wieder aufs Neue mit einer veränderten oder neuen Darstellung befassen muss; im Lernmodul ‚Typografie praktisch!‘ soll beispielsweise die konsequente Nutzung von Mouse-Over-Effekten bei Schaltflächen die Bedienbarkeit erleichtern

3.4.3 Nutzertest

Dieser Abschnitt beschreibt die Vorgehensweise bei der Durchführung von Usabilitytests an drei Testpersonen, die nacheinander das Lernmodul unter Beobachtung nutzten.

Da das Lernmodul ‚Typografie praktisch!‘ zwar für eine bestimmte Zielgruppe entwickelt wurde, im Bildungsportal Sachsen aber auch allgemein zur Nutzung zur Verfügung steht, wurden als Testpersonen Studenten mit unterschiedlichen Kenntnissen bezüglich des Programms InDesign, dem Lernen am Computer und auch der Nutzung von Computern im Allgemeinen ausgewählt.

Für den Test wurden fünf verschiedenen Fragen zu den Themen Typografie, InDesign und zum Lernmodul entwickelt, welche durch die Nutzung der vorhandenen Lernobjekte zu beantworten waren. Dabei wurde darauf geachtet, dass es sich um Fragen handelt, die durch wenige Worte zu beantwortet werden können.

Zudem wurde ein Formular für den Tester⁵ entwickelt, welches während der Testdurchführung für Notizen genutzt wurde. Dieses Formular enthält für jede Aufgabe folgende Bereiche:

- Zeitaufwand
- angeklickte Symbole (beinhaltet Symbole für Lernschritte, Tipps, zusätzliche Informationen, Tastaturkürzel)
- Fehlinteraktionen (z.B. wenn eine Testperson versucht, ein geöffnetes Fenster wieder zu schließen und dafür mehrfach ohne Erfolg mit der Maustaste klickt)
- Bereich für sonstige Notizen

Den Testpersonen wurde zur Beantwortung des Fragebogens ein ausreichender Zeitrahmen von maximal 45 Minuten eingeräumt. Sie wurden angewiesen die Fragen schriftlich in Form von Stichpunkten zu beantworten und bei der Nutzung des Lernmoduls laut zu Denken. Die Methode des *lauten Denkens* wird bereits seit vielen Jahren in der Forschung über Lernprozesse eingesetzt und ermöglichte in diesem Kontext die Sammlung zusätzlicher Daten zur Beurteilung des Lernmoduls, die nur durch eine Beobachtung der Testpersonen eventuell verborgen geblieben wären.

Im Anschluss an die Beantwortung der Fragen diente ein kurzes Gespräch der Richtigstellung der Interpretation der Beobachtungsergebnisse und ermöglichte Kritik und Verbesserungsvorschläge seitens der Testpersonen.

Insgesamt kamen die Testpersonen gut mit der Handhabung des Lernmoduls zurecht. Zu beobachten war jedoch, dass manche Testpersonen nicht auf Anhieb die geöffneten Fenster für Zusatzinformationen (Tipps, Tastaturkürzel) wieder schließen konnten. Dasselbe gilt für die Interaktionsflächen, die Screenshots aus dem Programm InDesign darstellen. Manchmal gelang es erst nach mehrmaligem ‚Herumklicken‘ die entsprechenden Fenster bzw. Schaltflächen zu schließen. Zudem wurden die als Screenshots dargestellten Interaktionsschaltflächen sowie die Schaltflächen zum Navigieren zwischen den Lernschritten von einer Testperson nicht sofort als Schaltflächen wahrgenommen.

5. Der Tester ist im konkreten Fall der Entwickler des Lernmoduls.

Aufgrund der Testergebnisse wurden folgende Änderungen an der Benutzeroberfläche des Lernmoduls vorgenommen:

- Veränderung im Kapitels 0 (Lernziele, Aufbau und Symbole dieses Kurses): Um die Navigationsmöglichkeit zwischen den einzelnen Lernschritten anhand der Schaltflächen ①, ②, ③ und ④ zu verdeutlichen, wird auf den ersten beiden Seiten der Einleitung mit dem Text ‚Weiter mit Schaltfläche ②‘ bzw. ‚Weiter mit Schaltfläche ③‘ auf die Schaltflächen hingewiesen.
- Schließen von Fenstern für Zusatzinformationen (Tipps, Tastaturkürzel): Diese Fenster waren zunächst nur durch einen wiederholten Mausklick auf die entsprechende Schaltfläche zu schließen. Um den Nutzern die Bedienung des Lernmoduls zu erleichtern, wurde die Programmierung so geändert, dass auch ein Mausklick außerhalb des geöffneten Fensters zum Schließen des Fensters führt.
- Schließen von Interaktionsflächen, die InDesign-Screenshots darstellen: Diese Interaktionsflächen konnten zunächst nur durch erneuten Mausklick auf den aktiven Screenshot deaktiviert werden. Zur Vereinfachung der Handhabung des Lernmoduls wurde die Programmierung so verändert, dass eine Interaktionsfläche auch durch Mausklick außerhalb der aktiven Interaktionsfläche deaktiviert werden kann.
- Hinweis auf Interaktionsflächen: Über den Hinweis ‚Mehr zu den Einstellungsmöglichkeiten durch Mausklick auf die Buttons (im linken Fensterbereich)‘ soll der Nutzer auf die aktivierbaren Interaktionsflächen hingewiesen werden.

Alle Fragen dieses Nutzertests und die zugehörigen Ergebnisse befinden sich im Anhang B dieser Arbeit.

3.4.4 Usability Evaluation

Eine Usability Evaluation durch Experten wurde von zwei Studenten der Medieninformatik, die sich durch eine Projektarbeit in das Thema Usability eingearbeitet hatten, in Form einer *heuristischen Evaluation* durchgeführt.

Dazu wurde ein Kriterienkatalog entwickelt, der folgende Bereiche umfasst:

- *Grundlagen der ‚guten‘ Gestaltung von Anwendungen (nach Nielsen u.a.):*
umfasst die Punkte Dialoggestaltung, Sprachgestaltung, Systemrückkopplung, Konsistenz, wiedergegebene Inhalte durch die Medien, ästhetischer Anspruch
- *Benutzermotivation und Interaktionsgestaltung nach dem ARCS-Modell (Keller 1983):*
umfasst die Punkte Aufmerksamkeit, Relevanz, Erfolgsvorsicht und Zufriedenheit





Das Lernmodul ‚Typografie praktisch!‘ wurde in vielen Punkten positiv bewertet, dennoch gab es einige Kritikpunkte, die zu Nachbesserungen am Lernmodul geführt haben. Im Folgenden werden die Kritikpunkte und die resultierenden Veränderungen aufgelistet:



Dialoggestaltung:

- *„Die Farbe Orange hat zu viele Bedeutungen bezüglich Orientierung und Interaktion (Rollovereffekt, Anzeigen des gerade aktiven Abschnitts, Anzeigen von Benutzbarkeit einer Schaltfläche)“:*

Aufgrund dieser Feststellung wurden die in Tabelle 3.2 dargestellten Veränderungen an den Interaktionsflächen des Lernmoduls vorgenommen:

Tab. 3.2: Veränderungen an den Interaktionsflächen

Ursprüngliche Variante:	Veränderte Variante:
<p>Navigationsleiste:</p>  <p><u>inaktiver Zustand:</u> grau mit weißer Ziffer <u>aktiver Zustand:</u> orange mit weißer Ziffer <u>Mouse-Over-Effekt:</u> orange mit weißer Ziffer, vergrößert</p>	<p>Navigationsleiste:</p>  <p><u>inaktiver Zustand:</u> weiß mit orangefarbener Ziffer <u>aktiver Zustand:</u> orange mit weißer Ziffer <u>Mouse-Over-Effekt:</u> orange mit weißer Ziffer, vergrößert</p>
<p>Schaltflächen für zusätzliche Hinweise:</p>  <p><u>nicht aktivierbarer Zustand:</u> grau <u>inaktiver Zustand:</u> orange mit weißer Ziffer <u>aktiver Zustand:</u> weiß mit orangefarbener Ziffer <u>Mouse-Over-Effekt:</u> orange mit weißer Ziffer, vergrößert</p>	<p>Schaltflächen für zusätzliche Hinweise:</p>  <p><u>nicht aktivierbarer Zustand:</u> grau <u>inaktiver Zustand:</u> weiß mit orangefarbener Ziffer <u>aktiver Zustand:</u> orange mit weißer Ziffer <u>Mouse-Over-Effekt:</u> orange mit weißer Ziffer, vergrößert</p>

Ursprüngliche Variante:	Veränderte Variante:
<p>Interaktionsflächen – Screenshots:</p>  <p><u>inaktiver Zustand:</u> grau, halbtransparent <u>aktiver Zustand:</u> orangefarbener Rahmen <u>Mouse-Over-Effekt:</u> orange</p>	<p>Interaktionsflächen – Screenshots:</p>  <p><u>inaktiver Zustand:</u> grau, halbtransparent <u>aktiver Zustand:</u> orange <u>Mouse-Over-Effekt:</u> orangefarbener Rahmen</p>

- „Die aktuelle Position innerhalb des gesamten Lernmoduls ist nur schwer ersichtlich. Der komplette Navigationspfad sollte auch in den Unterkapiteln ersichtlich sein. Dies unterstützt die Orientierung im Lernmodul.“:

Das Lernmodul ist im Sinne der Wiederverwendbarkeit der einzelnen SCOs bewusst so gestaltet, dass ein SCO auch in einem anderen Lernkontext Verwendung finden kann (vgl. 3.3.2). Dafür stellt das LMS eine anhand des SCORM-Pakets generierte Inhaltsstruktur mit einer Lernfortschrittsanzeige zur Verfügung.

Systemrückkopplung:

- „Der vom Lernenden durchlaufene bzw. abgearbeitete Lernweg ist nicht nachvollziehbar. Die abgearbeiteten Kapitel und Unterkapitel werden nicht als „bearbeitet“ markiert bzw. es gibt keine Verlaufsinformationen („History“).“:

Wie bereits erwähnt stellt das LMS eine Inhaltsstruktur mit einer Lernfortschrittsanzeige zur Verfügung. Da zum Zeitpunkt der Usabilitytests das Lernmodul jedoch noch nicht über das LMS verfügbar war, konnte dieser Aspekt bei der Auswertung nicht berücksichtigt werden.

- „Bei langsamen Internetverbindungen kann es zu Verzögerungen beim Laden von Flashfilmen kommen. Daher wäre es angebracht in einem solchen Fall den Nutzer über das Laden und dessen Fortschritt zu informieren.“:

Dem Lernmodul wurden zusätzliche Preloader, die bei längerem Laden der einzelnen Inhaltsfilme erscheinen, hinzugefügt.

Inhaltliche Synchronität angewandter Medien:

- „Bei Monitoren mit schlechterer Qualität führt der geringe Kontrast zwischen Text und Hintergrund (mit den horizontalen Linien) zu einer schlechten Lesbarkeit.“:

Das ursprünglich hellgrau-weiß gestreifte Fenster zur Anzeige von Inhalten wurde durch einen hellgrauen Hintergrund ersetzt. Abbildung 3.14 zeigt das ursprüngliche Aussehen dieses Fensters.

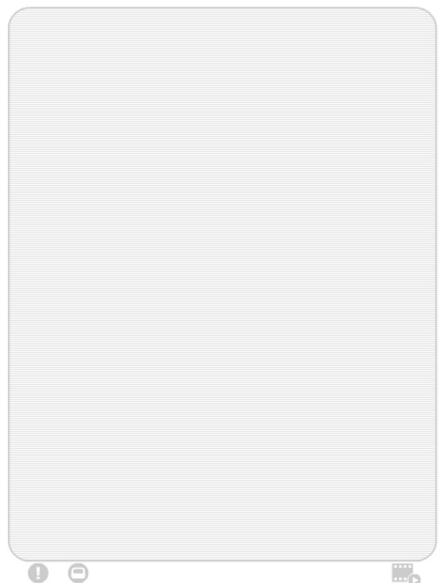


Abb. 3.14:
Ursprüngliches Aussehen
Des Inhaltsfenster

Aufmerksamkeit:

- „Der sprachliche Still ist sehr sachlich und nüchtern. Es ist zu empfehlen einen etwas personalisierteren Sprachstil zu verwenden, um eine Beziehung zum Lernenden zu verstärken. Ein direktes Ansprechen des Lernenden würde dies unterstützen.“:

Der Sprachstil wurde auf Wunsch des Auftraggebers nicht verändert.

Relevanz:

- „Das Lernmodul bietet keine Anpassung an Vorkenntnisse, Ansprüche und Ziele des Lernenden (z.B. Grundkurs, Aufbaukurs).“:

Eine Unterteilung des Lerninhalts in Grundkurs und Aufbaukurs erscheint in dem konkreten Fall nicht sinnvoll, da es sich beim gesamten Lernmodul um einen ‚Aufbaukurs‘ handelt, der im Hochschulbereich zusätzlich zum festen Lehrplan eine Vertiefungsmöglichkeit bieten soll.

Erfolgszuversicht:

- *„Der Lernende sollte über den Inhalt und die Lernziele am Anfang eines jeden Kapitels informiert werden.“:*

Um den Lernenden einen besseren Überblick über den Inhalt und die Lernziele zu geben, wurde jedem Kapitel eine einführende Seite (Einleitung) vorangestellt, die von den Lernenden, wenn gewünscht, gelesen werden kann.

Zufriedenheit:

- *„Bei den Aufgaben ist keine Erfolgskontrolle möglich. Hier wäre zu überlegen, ob dem Lernenden eine Musterlösung zu Verfügung gestellt wird. Dies fördert die Zufriedenheit beim Lernenden. Bei fehlender Zufriedenheit verursacht durch ein Abweichen der Anstrengungen des Lernenden gegenüber den Erwartungen kann es schnell zur Demotivation kommen.“:*

Das Lernmodul ‚Typografie praktisch!‘ bietet den Lernenden über die Seite ‚Downloads‘ die Möglichkeit, sich eine in InDesign erstellte Musterlösung herunterzuladen. Die Möglichkeit eine Musterlösung zur Verfügung zu stellen, war von Anfang an geplant, wurde aber bei dieser Evaluation nicht berücksichtigt, da die Musterlösung und die Seite ‚Downloads‘ zum Zeitpunkt der Evaluation noch nicht entwickelt war.

Der gesamte Kriterienkatalog (Anhang C) und die Auswertung der Usability Evaluation (Anhang D) sind im Anhang dieser Arbeit einsehbar.

4 Technische Entwicklung des Lernmoduls ‚Typografie praktisch!‘

4.1 Erstellung der Lernobjekte

Das vierte Kapitel beschreibt ausführlich die *technische Entwicklung* des Lernmoduls ‚Typografie praktisch!‘. Dazu wird in diesem Abschnitt die *Entwicklung* der Programmbausteine, wie z.B. Benutzeroberfläche und Grafiken, und die *Zusammensetzung* der Bausteine zu Lernobjekten beschrieben.

4.1.1 Erstellung der Lernobjekte mit dem Programm Flash MX

Die einzelnen Lernobjekte des Lernmoduls ‚Typografie praktisch!‘ wurden mit dem Programm *Flash MX Version 6.0* als Flash-Filme entwickelt (vgl. 3.2.3). Dieses stellt mit einer grafische Benutzerschnittstelle, einer Animationszeitleiste und der objektorientierten Programmiersprache *ActionScript 1* ein komfortables Autorenwerkzeug zur Erstellung von Lerninhalten dar.

Die Erstellung eines Flash-Films erfolgt über ein Flash-Dokument mit der Dateinamenerweiterung ‚.fla‘. Ein solches Flash-Dokument enthält neben der Programmierung die gesamten (unkomprimierten) Daten. Bei der Veröffentlichung eines Flash-Dokuments wird ein Flash-Film mit der Erweiterung ‚.swf‘ erstellt, welcher nur die für die Filmwiedergabe erforderlichen Daten enthält. Dabei werden umfangreiche Datenpakete durch einen Kompressionsalgorithmus um bis zu 90 Prozent minimiert [Wendt 2003, S.252].

Um einen groben Überblick über die Funktionsweise von Flash MX zu geben, werden die Elemente des Arbeitsbereichs im Folgenden kurz beschrieben:

- *Bühne*: repräsentiert den sichtbaren Bereich einer Flashanwendung; Text, Grafiken und Sound lassen sich entweder direkt oder durch ActionScript gesteuert über die Bühne veröffentlichen

- Zeitleiste mit Ebenen: über die Zeitleiste kann der Bühneninhalt mit Hilfe von einzelnen Frames im Zeitablauf strukturiert bzw. animiert werden; die Zeitleiste besitzt verschiedene Ebenen, welche die Anordnung auf der Bühne (vorderste bis hinterste Ebene) definieren
- Bibliothek: in der Bibliothek lassen sich wieder verwendbare Elemente als so genannte Symbole (*Movieclip*, *Schaltflächen* oder *Grafik*) speichern; Symbole werden in Form von Instanzen auf der Bühne dargestellt; aus einem Symbol lassen sich beliebig viele Instanzen, die modifizierte Eigenschaften besitzen können, erstellen; die Verwendung von Symbolen ist dann sinnvoll, wenn das entsprechende Element mehrmals auf der Bühne vertreten ist, denn somit wird das eigentliche Symbol nur einmal gespeichert, und für jede Instanz nur die Abweichungen zum Symbol festgehalten
- ActionScript: stellt die Programmiersprache von Flash MX dar; zwei verschiedene Bearbeitungsmodi (Normalmodus und Expertenmodus) ermöglichen sowohl eine einfache Nutzung ohne tiefer gehende Programmierkenntnisse als auch die Lösung komplexer und individueller Problemstellungen; dazu stellt ActionScript verschiedene Elemente wie Aktionen (z.B. `play` zur Filmsteuerung) und Objekte (z.B. `Movieclip` oder `TextFormat`) samt Methoden und Eigenschaften bereit
- Bedienfelder: Bedienfelder (wie z.B. die Werkzeugleiste oder der Eigenschafteninspektor) dienen der Erstellung bzw. Bearbeitung von einzelnen Elementen auf der Bühne (wie Text oder Grafiken), der Zeitleiste und des gesamten Dokuments

Für die Entwicklung der Lernobjekte wurden die Benutzeroberfläche des Lernmoduls ‚Typografie praktisch!‘ in die folgenden drei Bereiche unterteilt:

- Hintergrund: enthält alle gleich bleibenden Elemente im Hintergrund des Lernobjekts; dies beinhaltet den gesteihten Hintergrund sowie das InDesign-Logo
- Navigation: dieser Bereich repräsentiert den Steuerungs- und Orientierungsbereich (vgl. 3.4.2); hier befinden sich alle Elemente, die die Navigation zwischen den einzelnen Lernschritten steuern sowie die Überschriften für Lernobjekt und Lernschritt
- Inhalt: repräsentiert den Arbeitsbereich (vgl. 3.4.2); enthält die Lerninhalte eines einzelnen Lernschritts

Die Unterteilung ermöglichte die separate Entwicklung der einzelnen Bereiche. Somit war es ausreichend für den Hintergrund einen einzigen Flash-Film zu entwickeln, der von jedem Lernobjekt genutzt wird. Auch für die Steuerung der Navigation war es ausreichend nur einen Film

zu entwickeln, der die Navigation über die Anzahl der Lernschritte dynamisch generiert. Lediglich für den Bereich zur Darstellung der recht unterschiedlich aufgebauten Lerninhalte erschien es sinnvoll, für jeden Lernschritt einen spezifischen Flash-Film zu entwickeln.

In Flash lässt sich eine solche Unterteilung in Einzelfilme über das Objekt Movieclip realisieren. Dieses besitzt die Methode `loadMovie`, wodurch mehrere Flash-Filme zeitgleich abgespielt werden können.

Dadurch ist es möglich, dass der Hintergrund-Film den Navigations-Film lädt, welcher wiederum, je nachdem welcher Lernschritt aktuell ist, den entsprechenden Inhalts-Film lädt. Abbildung 4.1 verdeutlicht diese Ladereihenfolge. Dabei dient eine *Start-Film* der Definition globaler Variablen, die z.B. festlegen, um welches Lernobjekt es sich handelt und wie die Überschriften für das Lernobjekt und die Lernschritte lauten.

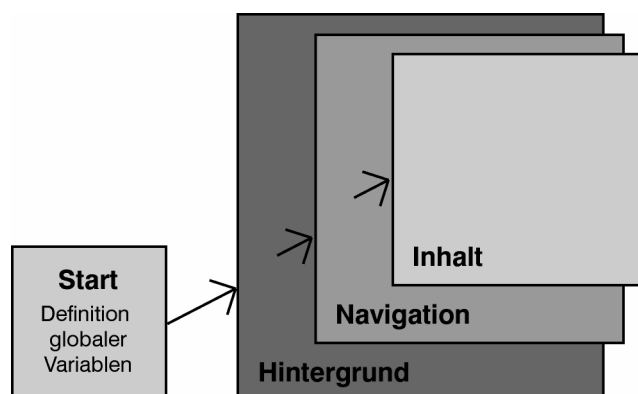


Abb. 4.1:
Ladereihenfolge der Flash-Filme des Lernmoduls ‚Typografie praktisch!‘

Die Speicherung der Flash-Filme erfolgte in einem Ordnersystem innerhalb des übergeordneten Ordners ‚TypografiePraktisch‘. Dieser enthält folgende Ordner und Unterordner:

- js: enthält JavaScript-Funktionen, die für die Umsetzung von SCORM am Lernmodul relevant sind
- Kapitel0 bis Kapitel5: diese Ordner enthalten die Flash-Filme zum Starten eines Lernobjekts sowie die Unterordner:
 - 0, 1, 2 usw.: repräsentieren die Unterkapitel und enthalten die Inhalts-Filme

- sharedAssets: beinhaltet die Unterordner:
 - clips: beinhaltet die Beispielfilme im untergeordneten Ordner ‚movieclips‘ und den Film *clips.swf* zur Steuerung der einzelnen Beispielfilme
 - downloads: enthält die Dateien zum Herunterladen, wie z.B. das Skript
 - faltblatt: hier befinden sich Bildmaterial und Texte, die zur Erstellung der Übungsaufgabe benötigt werden
 - hintergrund: enthält den Film *hintergrund.swf*
 - navigation: beinhaltet den Film *navigation.swf*
 - shared: Speicherort des Films *shared.swf*, dieser stellt Elemente, die von verschiedenen Flash-Filmen genutzt werden, zur Verfügung
- upgradeFlash: dieser Ordner enthält eine HTML-Seite, die aufgerufen wird, falls der benötigte Flash-Player nicht vorhanden ist

Im Folgenden werden die Erstellung und die Programmierung der einzelnen Flash-Filme beschrieben:

Start:

Dieser Film dient, wie bereits erwähnt, vor allem der Definition globaler Variablen bezüglich des jeweiligen Lernobjekts, welche für die weitere Filmsteuerung notwendig sind (vgl. Tab. 4.1). Demnach besitzt jedes Lernobjekt einen Start-Film, der nach Kapitel und Unterkapitel benannt ist. So wird beispielsweise das Lernobjekt *1.1 Typografische Maßeinheiten* über den Flash-Film *kl_1.swf* gestartet. Dieser befindet sich in dem Ordner *Kapitel1*.

Tab. 4.1: Globale Variablen eines *Start-Films* am Beispiel von *kl_1.swf*

Variablendefinition	Beschreibung
<code>_global.number_steps = 2;</code>	Anzahl der Lernschritte
<code>_global.title_chapter = "../Kapitel1/1/";</code>	Definition des Lernobjekts (beinhaltet Pfadangabe für Lerninhalte)
<code>_global.unterkapitel = "Typografische Maßeinheiten";</code>	Überschrift Unterkapitel
<code>_global.title_step = new Array();</code>	Array für Überschriften der Lernschritte

Variablendefinition	Beschreibung
<pre>_global.title_step[0] = "1. Wichtige Maßsys- teme in der Typografie"; _global.title_step[1] = "2. Voreinstellungen für Einheiten und Einteilungen in InDesign";</pre>	Definition der Überschriften für Lern- schritte
<pre>_global.bg_size = 1;</pre>	Definition der Art des Lernobjekt (1 entspricht kleinem/getrenntem Hintergrund für Lerninhalte; 0 entspricht großem Hintergrund für Einleitungen, Aufgabenstellungen und Downloadseiten)

Des Weiteren startet der Start-Film die Ladevorgänge für den Hintergrund (*hintergrund.swf*), die Navigation (*navigation.swf*) und lädt einen Flash-Film, der Elemente beinhaltet, die von verschiedenen Filmen genutzt werden (*shared.swf*). Dabei wird der Hintergrund in die unterste Ebene (`_root`) des Start-Films geladen und stellt somit den Basisfilm jedes Lernobjekts dar.

Über einen *Preloader* wird der sichergestellt, dass der Hintergrund erst dann angezeigt wird, sobald dieser vollständig geladen ist.

Hintergrund:

Der Hintergrund dient zum einen der grafischen Darstellung des Hintergrunds zum anderen wird aus diesem Film heraus der Navigations-Film geladen.

Die grafische Darstellung des Hintergrunds besteht aus einem schräg verlaufenden Farbverlauf, der von der linken oberen Ecke aus mit einem dunklen Grauton in einen hellen Grauton am unteren und rechten Rand des Films übergeht. Der Farbverlauf wurde so gestaltet, dass dieser nahtlos in die HTML-Seite übergeht, über die das Lernmodul vom Webbrowser gestartet wird. Diese besitzt den helleren Grauton als Hintergrundfarbe.

Die hellgrauen Hintergrundlinien werden per Actionscript über den Farbverlauf gezeichnet und gehen in diesen optisch über. Auch per Actionscript wird das Adobe InDesign-Logo aus der Bibliothek in der rechten oberen Ecke platziert. Zuletzt wird, wie im Beispiel 4.1 dargestellt, auf der Bühne ein leeres Movieclip erzeugt, in welches der Film *navigation.swf* geladen wird.

```
this.createEmptyMovieClip("navigation",4); //Bezeichnung und Ebene
this.navigation._x = 12.5;
this.navigation._y = 65.8;
this.navigation.loadMovie("../sharedAssets/navigation/navigation.swf",1);
```

Bsp. 4.1: Laden des Films *naviagtion.swf*

Navigation:

Im Flash-Dokument *navigation.fla* befinden sich zunächst keine grafischen Elemente auf der Bühne. Der gesamte Inhalt wird per Actionscript platziert. Die Abbildungen 4.2 und 4.3 zeigen Lernobjekte nachdem Navigationsfunktionen hinzugefügt wurden. Die dargestellten Filme unterscheiden sich nicht nur in der Anzahl der Lernschritte und in ihren Überschriften sondern auch in der Aufteilung des Hintergrunds im Arbeitsbereich. Nur bei Lernobjekten, die Lerninhalte präsentieren, wird der Arbeitsbereich anhand eines kleineren Fensters unterteilt. Lernobjekte ohne Interaktionsmöglichkeiten wie z.B. Aufgabenstellungen besitzen keine Unterteilung.

Die globale Variable `_global.bg_size` legt fest, welcher Hintergrund dargestellt werden soll. Auch die Überschriften werden anhand von globalen Variablen bestimmt (vgl. Tab. 4.1).



Abb. 4.2:
Lernobjekt bestehend aus Hintergrund und Navigation mit unterteiltem Arbeitsbereich für die Präsentation von Lerninhalten mit Interaktionsmöglichkeiten



Abb. 4.3:
Lernobjekt bestehend aus Hintergrund und Navigation ohne Unterteilung des Arbeitsbereichs für die Darstellung von Einleitungen, Aufgabenstellungen und Downloadmöglichkeiten

Im Skript der Datei *navigation.fla* werden zunächst die fünf Grautöne definiert, die für die Farb-
abstufungen hinter den Navigations-Schaltflächen genutzt werden. Dabei wird immer der
dunkelste Grauton (`greyShade[1]`) für den Balken hinter der Überschrift des Lernschritts (z.B.
1. Elemente des Satzspiegels) verwendet. Der nächste Grauton (hinter der Schaltfläche des ersten
Lernschritts) entspricht dem nächst helleren Ton (`greyShade[2]`). Diese Abstufung ist in
Abbildung 4.4 im rot umrandeten Bereich dargestellt.



Abb. 4.4:
Grautöne im Navigationsbereich

Je nachdem wie viel Lernschritte vorhanden sind, werden alle fünf definierten Grautöne einge-
setzt, oder es kommt nur ein Teil zum Einsatz. Um das Design der verschiedenen Lernobjekte
dennoch relativ einheitlich zu erhalten, wird der unterste Balken am Rande der Arbeitsfläche
immer im hellsten Grauton (`greyShade[5]`) dargestellt (vgl. grün umrandeter Bereich in Abb.
4.4). Besitzt ein Lernobjekt weniger als vier Lernschritte muss demnach ein Übergang zwischen
dem hellen Ton aus der unteren Leiste und dem Grauton des letzten Lernschritts (z.B. `greyS-
hade[3]` in Abb. 4.4) hergestellt werden. Dies erfolgt anhand eines Farbverlaufs wie in
Abbildung 4.4 blau umrandet dargestellt.

Dazu wurde zunächst an entsprechender Position eine Movieclip erstellt, welches mit der
Movieclip-Methode `beginGradientFill` je nach Anzahl der Lernschritte ausgefüllt wird.
Anhand der Parameter dieser Funktion lässt sich festlegen, ob ein Verlauf linear oder radial
(`fillType`) gestaltet sein soll, welche Farben verwendet werden sollen (`colors`), welche
Transparenzwerte die Farben besitzen sollen (`alphas`), wie das Verteilungsverhältnis sein soll
(`ratios`) und in welchem Bereich der Verlauf dargestellt werden soll (`matrix`).

Zur Definition des Parameters `colors` wurde die globale Variable `_global.number_steps` wie folgt verwendet:

```
colors = [greyShade[_global.number_steps+1],greyShade[5]];
```

Besitzt also ein Lernobjekt zwei Lernschritte, wird ein Farbverlauf aus den Farben `greyShade[3]` und `greyShade[5]` erstellt.

Auch mit Hilfe der Variable `_global.number_steps` werden die Graustufen hinter den Schaltflächen durch eine *for-Schleife* dynamisch erzeugt. Dabei wird jeweils das Symbol `buttonBgShape`, welches die abgerundete Form repräsentiert, aus der Bibliothek geladen und als Instanz auf der Bühne dargestellt.

Die Schaltflächen selbst werden auch mit einer *for-Schleife* aus der Bibliothek (`btnStep`) heraus als Instanzen auf der Bühne dargestellt. Die Bezeichnung einer Instanz wird dabei dynamisch, dem Lernschritt entsprechend, erzeugt. Dasselbe gilt für die Ziffer auf einer Schaltfläche. Diese wird entsprechend dem Lernschritt innerhalb der Schleife einem dynamischen Textfeld zugeordnet. Das folgende Beispiel verdeutlicht dies:

```
this.createEmptyMovieClip("buttons",5);
...
for (var y=0;y<_global.number_steps;y++){
    this.buttons.attachMovie("btnStep","btnStep"+y,y+1);
    ...
    this.buttons['btnStep'+y].btnStepNumber.text = y+1;
}
```

Bsp. 4.2: Darstellung der Schaltflächen auf der Bühne

Die Schaltfläche, die den ersten Lernschritt repräsentiert erhält somit die Bezeichnung `btnStep0` und wird der ersten Ebene des Movieclips `buttons` zugeordnet.

Da es sich beim Symbol `btnStep` um ein Movieclip handelt, stellen die Instanzen automatisch auch Movieclips dar und besitzen Movieclip-Methoden wie `onRollOver`, `onRollOut`, `onRelease` und `onReleaseOutside`¹. Den `btnStep`-Instanzen wurden jeweils diese vier Methoden zugeordnet, welche weitere Funktionen zur Ereignisbehandlung aufrufen.

1. `onRollOver`, `onRollOut`, `onRelease` und `onReleaseOutside` stellen Methoden zur Ereignisbehandlung dar und kommen zum Einsatz, wenn die Maus über die Schaltfläche fährt, diese verlässt, gedrückt oder außerhalb der Schaltfläche gedrückt wird

Dazu wurden die Funktionen `over(i)` (`onRollOver`), `out(i)` (`onRollOut`, `onReleaseOutside`) und `released(i)` (`onRelease`) definiert. Diese besitzen jeweils den Parameter `i` zur Identifikation der auslösenden Schaltfläche.

Sobald die Maus auf einer Schaltfläche losgelassen wird, wird die Funktion `released(i)` aufgerufen, welche veranlasst, dass der entsprechende Inhalt geladen und dargestellt wird. Beispiel 4.3 zeigt dies.

```
function released(i) {
    ...
    this.contentFrame.loadMovie(_global.title_chapter+(i+1)+".swf",1);
    this.onEnterFrame=this.preload;
    ...
}
function preload() {
    ...
}
```

Bsp. 4.3: Ereignisbehandlung beim Loslassen der Maus auf einer Schaltfläche

Der Lerninhalt wird in den Movieclip `contentFrame` geladen. Anhand der globalen Variable `_global.title_chapter` und dem Parameter `i` wird der zu ladende Inhalt bestimmt. Als Beispiel wurde im Start-Film `kl_1.swf` die Variable `_global.title_chapter` mit `../Kapitel1/1/` gleichgesetzt (vgl. Tab. 4.1). Wird in diesem Fall `btnStep0` losgelassen, so wird der Film mit der Pfadangabe `../Kapitel1/1/1.swf` geladen. Im nächsten Schritt wird sofort die Funktion `preload()` gestartet, die dafür sorgt, dass der Inhalt erst dann sichtbar wird, wenn dieser vollständig geladen ist.

Inhalt:

Die Bezeichnungen der Filme, die den eigentlichen Inhalt des Lernmoduls repräsentieren, entsprechen jeweils dem Lernschritt, den sie innerhalb eines Lernobjekts repräsentieren. Die Filme `1.swf`, `2.swf`, `3.swf` oder `4.swf` stehen demnach für den ersten, zweiten, dritten oder vierten Lernschritt. Alleine durch ihren Speicherort, z.B. `../Kapitel1/1/` lassen sich diese Filme dem jeweiligen Lernobjekt zuordnen.

Es lässt sich grob zwischen drei Varianten der Inhalts-Filme unterscheiden. Zum einen gibt es Lernobjekte, die bestimmte Funktionen übernehmen, aber keinen Lerninhalt darstellen. Dabei handelt es sich um Einleitungen, Aufgabenstellungen und Downloadmöglichkeiten (vgl. Abb. 4.3). Die Inhalte dieser Lernobjekte besitzen abgesehen von Downloadmöglichkeiten keine

Interaktionsmöglichkeiten, stellen reine Textinhalte dar und besitzen größtenteils keine Programmierung. Daher wird auf diese Art von Filmen nicht weiter eingegangen.

Umfangreicher dagegen war die Erstellung der Inhalte für Lernobjekte, die Lerninhalte mit Interaktionsmöglichkeiten repräsentieren (vgl. Abb. 4.2). Diese Flash-Filme besitzen eine Unterteilung in einen linken und einen rechten Bildschirmbereich. Der rechte Bereich dient der Darstellung von Inhalten, die durch die Interaktionsmöglichkeiten auf der linken Seite angezeigt werden können. Auch auf der rechten Seite befinden sich Schaltflächen zur Darstellung zusätzlicher Hinweise und Inhalte (Tipps, Tastaturkürzel und Beispielfilm). Beim Start besitzt zunächst nur der linke Bereich Inhalt, dieser setzt sich aus Interaktionsmöglichkeiten und in der Regel einem kurzen einführenden Text zusammen. Bei der Entwicklung dieser Flash-Filme wurde zwischen Lerninhalten, die theoretischen Inhalt darstellen, und Inhalten, welche die praktische Umsetzung in InDesign mit Hilfe von Screenshots vermitteln sollen, unterschieden (vgl. Abb. 3.7 und 3.8). Zunächst wird die Entwicklung der Filme mit theoretischem Lernstoff beschrieben.

Theoretischer Lerninhalt:

Eine solche Flash-Datei enthält zunächst auf der linken Seite der Bühne einen einführenden kurzen Text und am unteren Rand des rechten Bühnenbereichs die Interaktionsschaltflächen zum Anzeigen von Tipps, Tastaturkürzel und Beispielfilmen (vgl. Abb. 4.5). Diese Interaktionsschaltflächen befinden sich in dem Movieclip `contentBackground`, welches zur Laufzeit aus der Datei `shared.swf` geladen und gesteuert wird.



Abb. 4.5: Beispiel für theoretischen Lerninhalt mit Interaktionsmöglichkeiten im linken Bereich und der Darstellung von Inhalt mit einer Interaktionsfläche für zusätzliche Tipps im rechten Bereich

Das Grundgerüst der Interaktionsmöglichkeiten im linken Bildschirmbereich wurde in einer separaten Datei (*linkInteraktion.as*) festgehalten. Darin wird die Klasse `interactionLink` definiert, welche so allgemein gehalten ist, dass sie von allen Dateien zur Darstellung theoretischen Lerninhalts genutzt werden kann.

Über den Verweis `#include „linkInteraktion.as“` wird diese Datei in das Skript integriert. Folgendes Beispiel stellt die Funktionen aus dieser Datei kurz vor:

```
function interactionLink(){
    //Festlegung der Darstellung für inaktiven Text
    this.textFormat = new TextFormat();
    ...
    //Festlegung der Darstellung für inaktiven Text
    this.textFormatActiv = new TextFormat();
    ...
    //Erstellung eines Textfelds mit der Bezeichnung 'textLink'
    this.createTextField("textLink",1,0,0,0,0);
}

// die Funktion ,interaktionLink' wird vom Objekt ,MovieClip' abgeleitet
// und erhält somit alle Methoden von ,MovieClip'
interactionLink.prototype = new MovieClip();

// Funktion zum Festlegen von Eigenschaften, wie Text, Höhe, Breite
// und TextFormat
interactionLink.prototype.setProperties = function(properties){
    this.textLink.text = properties[0];
}
```

```

    this.textLink._width = properties[1];
    this.textLink._height = properties[2];
    this.textLink.setTextFormat(this.textFormat);
}

// Funktionen zur Steuerung der Maus-Interaktion
interactionLink.prototype.onRollOver = function(){
    this.textLink.setTextFormat(this.textFormatActiv);
}
interactionLink.prototype.onRollOut = function(){
    this.textLink.setTextFormat(this.textFormat);
}
interactionLink.prototype.onReleaseOutside = function(){
    this.textLink.setTextFormat(this.textFormat);
}

// Funktionen zur Bestimmung von Eigenschaften bei aktiviertem bzw.
// inaktiviertem Zustand
interactionLink.prototype.activate = function(){
    this.textLink.setTextFormat(this.textFormatActiv);
    this.enabled = false;
}
interactionLink.prototype.inactivate = function(){
    this.textLink.setTextFormat(this.textFormat);
    this.enabled = true;
}

```

Bsp. 4.4: Datei ‚linkInteraktion.as‘

Im eigentlichen Inhalts-Film werden die Link-Eigenschaften zunächst anhand eines mehrdimensionalen Arrays² festgelegt. Danach wird ein leeres Objekt aus der Bibliothek (`linkDummy`) mit der Klasse `interactionLink` verknüpft, wodurch dieses Objekt alle Methoden und Eigenschaften der Klasse `interactionLink` erhält. Die Links zur Interaktion werden als Instanzen von `linkDummy` auf der Bühne dargestellt. Über eine zusätzliche Release-Funktion wird die individuelle Behandlung eines Inhalts, z.B. die Zuweisung von Schaltflächen für zusätzliche Hinweise gesteuert. Folgendes Beispiel stellt diese Vorgehensweise kurz vor:

```

// Array zur Definition der Links
this.linkProperties = new Array();
// Zuweisung von Text, Breite, Höhe und aktivierbaren Schaltflächen für
// zusätzliche Tipps, Tastaturkürzel, Beispielfilm
this.linkProperties[0] = ["Das DTP-Maß", 110,18,1,0,0];
this.linkProperties[1] = ["Das Didot-Maß",120,18,1,0,0];
...

// Verbindung zwischen "interactionLink" und "linkDummy"
Object.registerClass("linkDummy",interactionLink);

```

2. Ein *Array* dient der Definition eines Objekts mit mehreren Elementen

```

// Links auf Bühne darstellen und Aktionen steuern
for (x=0;x<4;x++){ // Beispiel für 4 Links
    // Bezeichnung
    this.attachMovie("linkDummy","link"+(x+1),x+1);
    // Position
    ...
    // Identifikation
    this['link'+(x+1)].id = x+1;
    //Eigenschaften
    this['link'+(x+1)].setProperty(this.linkProperties[x]);
    //Aktionen
    this['link'+(x+1)].onRelease = function(){
        linkReleased(this.id);
    }
}

// zusätzliche, individuelle Behandlung der Release-Funktion eines Links
linkReleased = function(i){
    ...
    // Darstellung des Inhalts aus der Bibliothek (content1, content2, etc.)
    // auf der rechten Seite der Bühne
    this.contentBg.contentFrame.attachMovie("content"+i,"content",1);
    ...
    // Aktivierung von Schaltflächen für zusätzliche Hinweise
    for (y=0;y<3;y++){
        if (this.linkProperties[i-1][y+3] == 1){
            this.contentBg['btn'+(y+1)].enabled = true;
            ...
            // die Movieclips befinden sich in der Bibliothek mit
            // Bezeichnungen ,btnContent1', ,btnContent2', etc.
            // oder auch ,btnContent1_1', ,btnContent1_2' etc.
            // bei mehreren Hinweisen pro Inhalt
            this.contentBg['btn'+(y+1)].content = "btnContent"+i;
            ...
            if (y == 2){
                // Funktion ,ExampleMovie' wird aufgerufen,
                //wenn der Inhalt einen Beispielfilm besitzt
                ExampleMovie(i);
            }
        }
    }
}

// Funktion legt Beispielfilm fest
function ExampleMovie(){
    this.contentBg.btn3.content = "4.3.3.2";
    _global.movieTitle = "Farbverlauf:";
}

```

Bsp. 4.5: Darstellung und Interaktion der Links bei theoretischem Inhalt

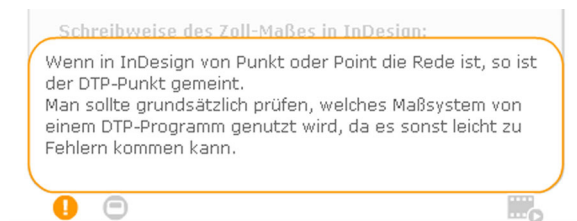


Abb. 4.6:
Beispiel für zusätzlichen
Hinweis im rechten Bereich
des Lernmoduls

Zusätzliche Hinweise (Tipps und Tastaturkürzel) werden durch kleine Fenster dargestellt (vgl. Abb. 4.6), die sich entweder durch nochmaliges Klicken auf die entsprechende Schaltfläche oder durch Mausklick außerhalb des Fensterrahmens schließen lassen. Dazu wird über die Movieclip-Methode `hitTest` überprüft, ob die Maus außerhalb des Fensters gedrückt wurde:

```
this.onMouseDown = function () {  
    ...  
    (!this.contentBg.contentButtonFrame.btnContent.hitTest  
    (_root._xmouse,_root._ymouse,true)) {  
        this.contentBg.contentButtonFrame.btnContent.removeMovieClip();  
        ...  
    }  
}
```

Bsp. 4.6: ‚hitTest‘ wird bei jedem Maus-Druck aufgerufen

Praktischer Lerninhalt:

Ähnlich wie bei theoretischem Lerninhalt befindet sich zunächst auf der Bühne in der Regel ein kurzer einführender Text, strukturierende Überschriften und grafische Elemente sowie die drei Interaktionsschaltflächen am unteren Rand der rechten Seite (vgl. Abb. 4.7).

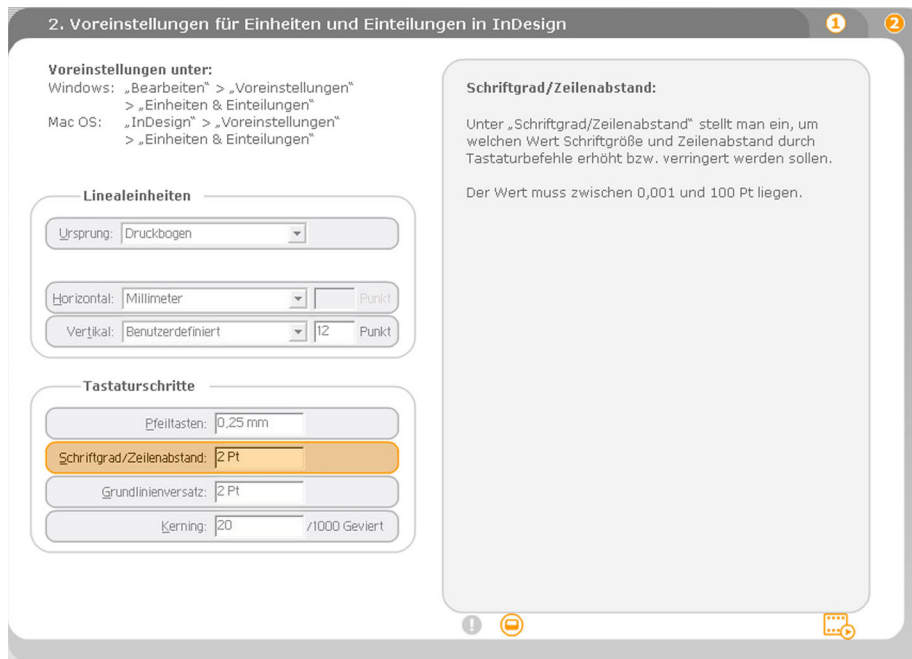


Abb. 4.7: Beispiel für praktischen Lerninhalt mit Interaktionsmöglichkeiten im linken Bereich und der Darstellung von Inhalt mit zwei Interaktionsflächen für zusätzliche Tipps und Beispielfilm im rechten Bereich

Auch bei praktischem Lerninhalt befindet sich die Grundstruktur für die Interaktionen in einer separaten Datei namens *buttonInteraction.as*. Folgendes Beispiel stellt diese Datei mit der Klasse `interactionButton` kurz vor:

```
// Festlegung der Starteigenschaften einer Schaltfläche
function interactionButton() {
    this.state = 0;
    this._alpha = 50;
}

// die Funktion ‚interaktionButton‘ wird vom Objekt ‚MovieClip‘ abgeleitet
// und erhält somit alle Methoden von ‚MovieClip‘
interactionButton.prototype = new MovieClip();

// Festlegung der Position der Schaltfläche
interactionButton.prototype.setProperty = function(properties) {
    this._x = properties[0];
    this._y = properties[1];
}

// Funktionen zur Steuerung der Maus-Interaktion
interactionButton.prototype.onRollOver = function() {
    if (this.state == 0) { // Schaltfläche ist nicht aktiviert
        this.gotoAndStop(2);
        this._alpha = 100;
    }
}
interactionButton.prototype.onRollOut = function() {
    if (this.state == 0) {
```

```

        this._alpha = 50;
        this.gotoAndStop(1);
    }
}
interactionButton.prototype.onReleaseOutside = function(){
    if (this.state == 0){
        this._alpha = 50;
        this.gotoAndStop(1);
    }
}

// Funktionen zur Bestimmung von Eigenschaften bei aktiviertem bzw.
// inaktiviertem Zustand
interactionButton.prototype.activate = function(){
    this.gotoAndStop(3);
    this._alpha = 100;
    this.state = 1; // Schaltfläche ist aktiviert
}
interactionButton.prototype.inactivate = function(){
    this._alpha = 50;
    this.gotoAndStop(1);
    this.state = 0;
}
}

```

Bsp. 4.7: Datei ‚buttonInteraktion.as‘

Im eigentlichen Flash-Film werden die Eigenschaften der Schaltflächen auch hier zunächst anhand eines mehrdimensionalen Arrays festgelegt. Danach werden in einer Schleife die Schaltflächen-Movieclips mit den Bezeichnungen 1, 2, 3 usw. mit der Klasse `interactionButton` verknüpft und als Instanzen (IA1, IA2, IA3 usw.) auf der Bühne dargestellt. Das folgende Beispiel stellt die wichtigsten Funktionen kurz vor:

```

// Array zur Bestimmung der Eigenschaften der Schaltflächen
this.buttonProperties = new Array();
// Zuweisung der Position (x, y) und der aktivierbaren Schaltflächen für
// zusätzliche Tipps, Tastaturkürzel, Beispielfilm
this.buttonProperties[0] = [14.2,148.3,0,0,0];
this.buttonProperties[1] = [14.2,208.3,0,0,0];
...
this.numberBtns = 7; // Variable für Anzahl der Schaltflächen

for (x=1;x<(this.numberBtns+1);x++){
    // Verbindung zwischen "interactionButton" und dem Movieclip x
    // (entspricht ‚1‘, ‚2‘, ‚3‘ etc.) aus der Bibliothek
    Object.registerClass(x,interactionButton);

    // Schaltflächen auf Bühne darstellen und Aktionen steuern
    this.attachMovie(x,"IA"+x,(this.numberBtns+1)-x);
    this['IA'+x].id = x; // Identifikation
    this['IA'+x].setProperties(this.buttonProperties[x-1]);
    this['IA'+x].onRelease = function(){
        buttonReleased(this.id);
    }
}

```

```

}

// zusätzliche, individuelle Behandlung der Release-Funktion
// einer Schaltfläche
buttonReleased = function(i){
    if (this['IA'+i].state == 0){ // Schaltfläche ist nicht aktiv
        ...
        // Darstellung des Inhalts aus der Bibliothek (content1, content2, etc.)
        // auf der rechten Seite der Bühne
        this.contentBg.contentFrame.attachMovie("content"+i,"content",1);
        ...

        // Aktivierung von Schaltflächen für zusätzliche Hinweise
        for (y=0;y<3;y++){
            if (this.buttonProperties[i-1][y+2] == 1){
                this.contentBg['btn'+(y+1)].enabled = true;
                ...
                // die Movieclips befinden sich in der Bibliothek mit
                // Bezeichnungen ,btnContent1', ,btnContent2', etc.
                // oder auch ,btnContent1_1', ,btnContent1_2' etc.
                // bei mehreren Hinweisen pro Inhalt
                this.contentBg['btn'+(y+1)].content = "btnContent"+i;
                if (y == 2){
                    // Funktion ,ExampleMovie' wird aufgerufen,
                    //wenn der Inhalt einen Beispielfilm besitzt
                    ExampleMovie();
                }
                ...
            }
        }
    }
}

// Funktion legt Beispielfilm fest
function ExampleMovie(movieNumber){
    this.contentBg.btn3.content = "1.1.2.5";
    _global.movieTitle = "Schriftgrad/Zeilenabstand:";
}
}

```

Bsp. 4.8: Darstellung und Interaktion der Schaltflächen bei praktischem Inhalt

Die Interaktions-Schaltflächen besitzen zwei Zustände, *aktiv* und *nicht aktiv*. Beim Loslassen der Maustaste auf einer nicht aktiven Schaltfläche wird diese aktiviert und der entsprechende Inhalt wird dargestellt. Befindet sich eine Schaltfläche im Zustand *aktiv*, wird beim Loslassen der Maus der Inhalt entfernt und die Schaltfläche bekommt wieder den Zustand *nicht aktiv*.

Bei der Aktivierung mancher Schaltflächen öffnet sich ein zusätzlicher Screenshot mit der Abbildung eines Dropdown-Menüs aus dem Programm InDesign (vgl. Abb. 4.8).

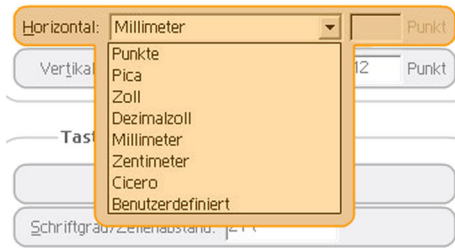


Abb. 4.8:
Beispiel für ein
Dropdown-Menü

Beim Loslassen der Maustaste auf einem solchen Menü soll die Schaltfläche jedoch aktiviert und das Menü weiter sichtbar bleiben. Nur beim Mouse-Release auf der eigentlichen Schaltfläche bzw. außerhalb der Schaltfläche soll der Inhalt und das Dropdown-Menü entfernt und der Schaltfläche deaktiviert werden.

Zur Umsetzung dieser Anforderung wird bei der Aktivierung einer solchen Schaltfläche ein unsichtbarer Dummy, der eine Instanz des Dropdown-Menüs darstellt auf der Bühne platziert. Das Skript dazu befindet sich im entweder im ersten Frame der Hauptszene, in den meisten Fällen jedoch im dritten Frame der aktivierten Buttons und kann wie folgendes Beispiel aussehen.

```
// Dummy für 'DropDown-Menü'
_parent.createEmptyMovieClip("dummy",10);
_parent.dummy._x = 112.8;
_parent.dummy._y = 396.4;
_parent.dummy._alpha = 0;
_parent.dummy.attachMovie("5a","a",1);
_parent.dummy.a.onRollOver = function(){
    this.useHandCursor = false;
}
```

Bsp. 4.9: Erstellung eines Dummys für ein Dropdown-Menü

Wie bei den Filmen für theoretischen Inhalt wird auch bei praktischem Inhalt über die Methoden `onMouseDown` und `hitTest` überprüft, ob die Maustaste innerhalb eines aktiven Fensters, sei es ein Dropdown-Menü oder ein Fenster für zusätzliche Inhalte auf der rechten Bildschirmseite, gedrückt wurde. Bei Klick außerhalb werden die Fenster geschlossen.

Shared:

Der Film *shared.swf* beinhaltet, wie bereits erwähnt, Elemente, die von anderen Filmen zur Laufzeit geladen werden können. Dabei handelt es sich um die Schaltflächen für zusätzliche Hinweise und der Programmierung der Interaktion dieser Schaltflächen.

Die drei Schaltflächen für Tipp, Tastaturkürzel und Beispielfilm sind in der Datei *shared.fla* in dem Movieclip `contentBackground` positioniert. Dieser Movieclip enthält neben den Schaltflächen auch noch eine transparente Fläche, die sich bei der Aktivierung eines Hinweisfensters halbtransparent über das rechte Inhaltsfenster legt.

Das Skript zur Steuerung der Schaltflächen befindet sich im ersten Frame des Movieclips. Das folgende Beispiel stellt die Release-Funktion grob vor, die beim Loslassen der Maustaste auf einer Schaltfläche aufgerufen wird:

```
function released(i){
    ...
    if (this['btn'+i].state == 0){ // Schaltfläche ist nicht aktiv
        ...
        this.transparency._alpha = 70;
        ...
        if (i!=3){ // Schaltfläche für Tipp oder Tastaturkürzel
            // this['btn'+i].content wird im Inhalts-Film (parent) definiert
            this.contentButtonFrame.attachMovie
                (this['btn'+i].content,"btnContent",1);
        }
        ...
        if (i == 3){ // Schaltfläche für Beispielfilm
            _parent.createEmptyMovieClip("movieBackground",20);
            _parent.movieBackground._x = 0;
            _parent.movieBackground._y = 0;
            _parent.movieBackground._alpha = 0;
            _parent.movieBackground.loadMovie("../sharedAssets/clips/clip.swf");
        }
    } else if (this['btn'+i].state == 1){ // Schaltfläche ist aktiv
        ...
    }
}
```

Bsp. 4.10: Ereignisbehandlung beim Loslassen der Maus auf einer Schaltfläche

Die Darstellung von zusätzlichen Hinweisfenstern sowie des Beispielfilm wird also im Film *shared.swf* geregelt. Die Definition dieser Hinweise bzw. des Beispielfilms erfolgt jedoch im aufrufenden Inhalts-Film.

Clip:

Der Film *clip.swf* stellt Funktionen zum Laden und Abspielen von Beispielfilmen zur Verfügung. Die Flash-Datei besteht aus zwei Frames in der Hauptzeitleiste. Im ersten Frame wird der im Inhalts-Film festgelegte Beispielfilm mit Hilfe eines *Preloaders* aus dem Ordner *movieclips* auf die Bühne geladen. Beispiel 4.11 verdeutlicht dies.

```
// leeren Movieclip erzeugen
this.createEmptyMovieClip("movie",1);
...
// Beispielfilm laden
this.movie.loadMovie("../sharedAssets/clips/movieclips/"
+ parent.ContentBG.btn3.content+".swf");
```

Bsp. 4.11: Laden des Beispielfilms

Sobald der Beispielfilm vollständig geladen ist, wird dieser auf der Bühne angezeigt und der Film *clip.swf* springt in seinen zweiten Frame, in dem die Abspielsteuerung geregelt wird:

```
// preload2 dient dem Vorausladen des Beispielfilms
function preload2(){
    // geladene Frames des Beispielfilms
    this.loadedFrames = this.movie._framesloaded;
    // gesamte Frames des Beispielfilms
    this.totalFrames = this.movie._totalframes;
    ...
    if((this.loadedFrames >= (this.totalFrames))){
        // Beispielfilm wird sichtbar
        this.movie._alpha = 100;
        // clip.swf springt in den zweiten Frame
        this.gotoAndStop(2);
        delete this.onEnterFrame;
    }
    ...
}
```

Bsp. 4.12: Preloader zur Steuerung der Anzeige des Beispielfilms

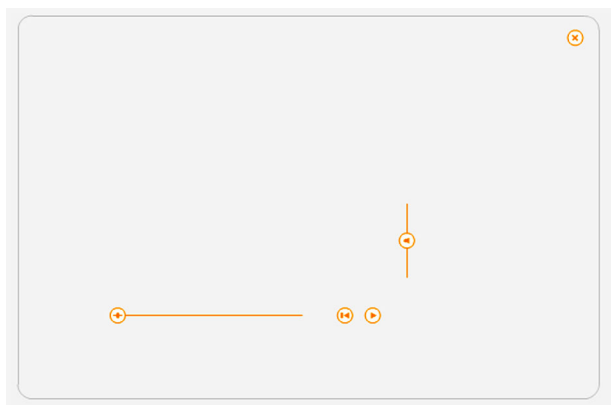


Abb. 4.9: Filmsteuerung in *clip.swf*

Abbildung 4.9 zeigt die Steuerfunktionen im zweiten Frame. Die Programmierung der Interaktionsfunktionen für die Schaltflächen zum Abspielen, des Films (*playBtn*), zum Rückspulen & Abspielen (*backBtn*) und zum Schließen des gesamten Fensters (*closeBtn*) befinden sich

direkt in der Hauptzeitleiste. Dabei beeinflussen sich der *Play*- und der *Back*-Button gegenseitig. Wird z.B. der *Back*-Button aktiviert, erhält der *Play*-Button die Funktion und das Aussehen eines *Stopp*-Buttons.

Die Programmierung der Schieberegler basiert auf den Aktionen `startDrag` und `stopDrag`. Über `startDrag` lässt sich das *Zielobjekt*, welches gezogen werden soll, angeben, der *Einrastepunkt* auf dem Zielobjekt definieren und die Koordinaten für ein Rechteck, innerhalb dessen *gezogen* werden kann, festlegen.

Die beiden Schieberegler stellen Movieclips dar, welche als Basis jeweils eine Schaltflächeninstanz besitzen, die folgendes Skript beinhalten:

```
on (press) {
    // ,left', ,up', ,right', ,down' zur Definition des Rahmenrechtecks
    // werden im eigentlichen Movieclip definiert
    startDrag(this, false, left, up, right, down);
    ...
}
on (release, releaseOutside) {
    stopDrag();
    ...
}
```

Bsp. 4.13: Basisskript der Schaltflächeninstanz für Schiebereglerfunktion

Weiteren Aktionen, die auf `startDrag` bzw. `stopDrag` folgen, werden über die Movieclip-Instanzen `regler_film` bzw. `regler_volume` geregelt.

Flash Detection:

Zuletzt wurden Flash-Filme erstellt, die überprüfen, ob der aufrufende Internetbrowser das benötigte Flash Plug-In zum Starten der Lernobjekte besitzt. Dazu bietet Macromedia eine vorgefertigte Flash-Datei mit entsprechendem ActionScript Programmcode. Diese kann zusammen mit einer ausführlichen Anleitung im Rahmen des *Flash Player Detection Kit* kostenlos von Macromedia [Flash Detection] heruntergeladen werden.

Die Flash-Datei muss nur noch an wenigen Stellen auf den spezifischen Anwendungskontext angepasst werden. Die folgenden Programmzeilen zeigen die individuellen Änderungen:

```
//the url that the visitor should be sent to if they do not have the required
//version of Flash.
altContentURL = "../upgrade_flash/upgrade_flash.html";
//-----
//the url that the visitor should be sent to if they have the required
//version of Flash.
flashContentURL = "kl_0.html";
//-----
//The required player version necessary to view the content.
contentVersion = 6;
//-----
//The 'dot' release of the player necessary to view the content.
contentMajorRevision = 0;
//-----
//The revision number of the player necessary to view the content.
contentMinorRevision = 0;
```

Bsp. 4.14: Flash-Detection Skript von Macromedia

Mit der Variablen `altContentURL` wird der Pfad zu alternativem Inhalt angegeben, der gestartet werden soll, falls die benötigte Flash-Version nicht vorhanden ist. Über `flashContentURL` wird die Datei (Flash/HTML) definiert, die gestartet werden soll, wenn das benötigte Plug-In vorhanden ist. Die weiteren drei Variablen `contentVersion`, `contentMajorRevision` und `contentMinorRevision` bestimmen die benötigte Flash-Player Version, z.B. 6.0r0.

Vor dem Start jedes Lernobjekts wird ein Flash-Detection-Film mit der entsprechenden Bezeichnung z.B. `kl_0checkFlash.swf` aufgerufen. Sollte der benötigte Flash-Player nicht vorhanden sein, so wird die Datei `upgrade_flash.html` gestartet, welche einen Link zum Downloaden des benötigten Players enthält.

4.1.2 Erstellung von integrierten Medien

Das Lernmodul ‚Typografie praktisch!‘ setzt sich aus *Text*, *Grafiken*, *Bildern*, *Screenshots* und *Videosequenzen* zusammen (vgl. 3.3.5). Im vorigen Abschnitt *Erstellung der Lernobjekte mit dem Programm Flash MX* wurde vorrangig auf die Programmierung mit ActionScript beschrieben. Nun soll genauer auf die Erstellung der genutzten Medien eingegangen werden.

In Flash MX lassen sich *Text* und *Grafiken* sowohl durch ActionScript als auch durch die grafische Benutzerschnittstelle erstellen. Bei der Erstellung des Lernmoduls wurden beide Möglichkeiten genutzt.

Text:

Zur Erzeugung von Text bietet Flash MX zum einen die Möglichkeit, eingebettete Schriftarten zu verwenden, zum anderen kann der Flash Player zur Laufzeit auf Geräteschriftarten des Systems zurückgreifen. Bei der Verwendung von *eingebetteten Schriftarten* werden die benötigten Schriftartinformationen im Flash-Film gespeichert und können so auf jedem Computersystem exakt wiedergegeben werden. Dabei wird unter Flash MX automatisch *Anti-Aliasing* verwendet. Dies ist eine Art *Weichzeichner* der störende, sichtbare Pixel („Treppeneffekt“) entfernt. Leider führt Anti-Aliasing vor allem bei kleinen Schriftgrößen dazu, dass wichtige Details verloren gehen können, was natürlich zu einer schlechteren Lesbarkeit führt [Blatz 2002, S.339].

Daher ist bei kleineren Schriftgrößen die Verwendung von *Geräteschriftarten* angebracht, da für diese kein Anti-Aliasing genutzt wird. Ein weiterer Vorteil von Geräteschriftarten liegt darin, dass die Dateigröße etwas kleiner ist als bei eingebettetem Text.

Im Lernmodul ‚Typografie praktisch!‘ wird vorwiegend auf Geräteschriftarten zurückgegriffen. Somit sind die Dateigröße und die Ladezeit der Flash-Filme geringer. Ausschlaggebend für die Verwendung von Geräteschriftarten war aber die bessere Lesbarkeit. Eingebettete Schriftarten werden lediglich zur Darstellung von Textbeispielen genutzt. Abbildung 4.10 zeigt ein Beispiel für die Nutzung von Geräteschriftarten sowie einer eingebetteten Schriftart im Lernmodul.



Abb. 4.10:
Nutzung von Geräteschriftarten
(hier Verdana) und eingebetteten
Schriftarten (hier Times New
Roman)

Dabei wurde die Schriftart *Times New Roman* zur Darstellung von Beispielen als eingebettete Schriftart verwendet. Der restliche Text wird durch *Verdana* entsprechend der Designvorgaben (vgl. 3.3.5) als Geräteschriftart dargestellt. Der Flash-Player verwendet somit den Schriftsatz von Verdana, der bereits auf dem System vorhanden ist. Sollte Verdana möglicherweise nicht verfügbar sein, wird nach einer ähnlichen Schriftart gesucht.

Überschriften für Lernobjekte (`_global.unterkapitel`) und Lernschritte (`_global.title_step[i]`) sowie die Nummerierung der Schaltflächen für die Lernschritte werden im Lernmodul ‚Typografie praktisch!‘ dynamisch per ActionScript zur Laufzeit generiert (vgl. 4.1.1).

Die meisten Textkomponenten im Lernmodul wurden hingegen über das Textwerkzeug direkt auf der Bühne bzw. in Movieclips erstellt. Dabei läßt sich bei *statischem Text* über das das Kontrollkästchen ‚Schriftarten des Geräts verwenden‘ die Verwendung von Geräteschriftarten festlegen. Beim Einsatz von *dynamischem Textfeldern* wird in Flash MX hingegen automatisch eine Geräteschriftart verwendet.

Grafiken, Bilder und Screenshots:

Die *schematischen Grafiken*, die im Lernmodul zur Unterstützung und zum Ersatz von Text eingesetzt werden, wurden in Flash MX mit den zur Verfügung stehenden Zeichenwerkzeugen erstellt und als Movieclips abgespeichert. Grafikelemente, die mehrfach in einem Flash-Film auftreten wurden, um die Datengröße möglichst gering zu halten, als einzelne Movieclips abgespeichert und im entsprechenden Kontext eingesetzt.

Screenshots aus dem Programm InDesign CS wurden unter Windows und Mac OS über die Drucktaste bzw. die entsprechende Tastenkombination (Apfel-Shift-3) erstellt und mit dem Programm Adobe Photoshop zugeschnitten und als JPEG gespeichert.

Im Lernmodul ‚Typografie praktisch!‘ kommen abgesehen von den Screenshots nur wenige *Bitmapbilder* zur Unterstützung des Lerninhalts zum Einsatz. Zur Komprimierung von Bitmapbildern bietet Flash MX das JPEG-Format mit verschiedenen Qualitätsstufen an. Die Komprimierung der eingesetzten Bilder erfolgte je nach Bildart und Qualitätsanspruch.

Videosequenzen:

Die Videosequenzen, die im Lernmodul als Beispielfilme eingesetzt werden, stellen die Nutzung bestimmter Programmfunktionen von InDesign CS dar. Dazu wurden Bildschirmmitschnitte unter Mac OS mit dem Programm Snapz Pro X der Firma Ambrosia Software erstellt und im QuickTime-Format (‚.mov‘) gespeichert. Zur Vertonung dieser Videosequenzen wurden erklärende Texte mit einer professionellen Sprecherin in einem Tonstudio aufgenommen. Das Ergebnis waren 82 Tracks auf einer Musik-CD (cda-Format).

Aus den vorhandenen Materialien wurden mit dem Programm *Adobe After Effects* elf Beispielfilme zusammengesetzt und als Audio Video Interleave Dateien (‚.avi‘) gespeichert. AVI-Filme lassen sich problemlos in Flash-Dateien als eingebettetes Video integrieren. Dabei wird die Video-Datei wie z.B. auch Bitmap-Dateien, zu einem Teil des Flash-Films mit entsprechend vielen Frames.

Für ein möglichst gutes Resultat sollten die Bildrate des importierten Videos und die Bildrate des Flash-Films übereinstimmen. Aus diesem Grund wurde bereits bei der Erstellung der Quick-Time-Filme mit Snapz Pro sowie bei der Erstellung AVI-Filme mit Adobe After Effects auf eine einheitliche Bildrate von 25 Bildern pro Sekunde geachtet. Dies entspricht der Bildrate, welche für die Flash-Filme des Lernmoduls ‚Typografie praktisch!‘ verwendet wurde.

Beim Import von eingebetteten Videos wird in Flash MX der *Sorenson Spark Codec* als Kompressionsalgorithmus genutzt. Dieser beruht auf zwei Kompressionsmethoden. Zum einen werden über die so genannte *zeitlichen Komprimierung* die Inhalte der einzelnen Bilder analysiert und nur die Unterschiede zum Vorbild gespeichert. Dabei dienen Schlüsselbilder mit vollständigem Bildinhalt, die in regelmäßigen Abständen gesetzt werden, als Referenzbilder. Zum anderen besteht die Möglichkeit, im Rahmen einer *räumlichen Komprimierung* die einzelnen Bilder unabhängig vom vorhergehenden oder nachfolgenden Bild zu komprimieren, wie dies z.B. auch bei der JPEG-Komprimierung der Fall ist.

Unter Flash MX stehen dazu folgende Einstellungsmöglichkeiten zu Verfügung:

- Qualität: bestimmt die Komprimierungsqualität (0 bis 100) der einzelnen Bilder; die Ergebnisse sind mit der JPEG-Komprimierung vergleichbar
- Schlüsselbildintervall: der Wert (z.B. 25) bestimmt nach wie vielen Bildern jeweils ein Schlüsselbild, d.h. ein vollständiges Bild, gespeichert werden soll; ein kleines Intervall bewirkt, dass sich gesuchte Bilder rascher auffinden lassen, die Dateigröße wird dadurch aber auch geringer komprimiert
- Skalieren: hiermit lässt sich anhand einer Skala von 1 bis 100 die Breite und Höhe des Videos skalieren
- Video mit Bildrate des Macromedia Flash-Dokuments synchronisieren: hiermit lässt sich die Wiedergabegeschwindigkeit des importierten Videos im Verhältnis zur Bildrate des Flash-Films bestimmen

Da das Lernmodul ‚Typografie praktisch!‘ über das Internet verfügbar ist, wurde bei der Erstellung der Beispielfilme versucht, die Datenmenge bei einer akzeptablen Filmqualität möglichst klein zu halten. Dazu wurde beim Import der AVI-Filme jeweils die Qualität auf 70 und das Schlüsselbildintervall auf 25 eingestellt.

Die zeitliche Komprimierung anhand des Schlüsselbildintervalls von 25 bewirkte bereits eine starke Reduzierung der Datenmenge. Beispielsweise konnte ein eingebetteter AVI-Film mit einer Originalgröße von 338 MB auf 14,7 MB ohne erkennbare Qualitätsverluste reduziert werden (vgl. Abb. 4.11). Durch die Herabsetzung der Qualität von 100 auf 70 konnte die Dateigröße zusätzlich verringert werden. Dies bewirkte bei dem zuvor genannten Beispiel eine weitere Reduzierung auf 3,48 MB, wobei leichte Komprimierungsspuren erkennbar wurden (vgl. Abb. 4.12). In Hinblick auf die Reduzierung der Datenmenge stehen diese Qualitätsverluste jedoch durchaus in einem vertretbaren Verhältnis.

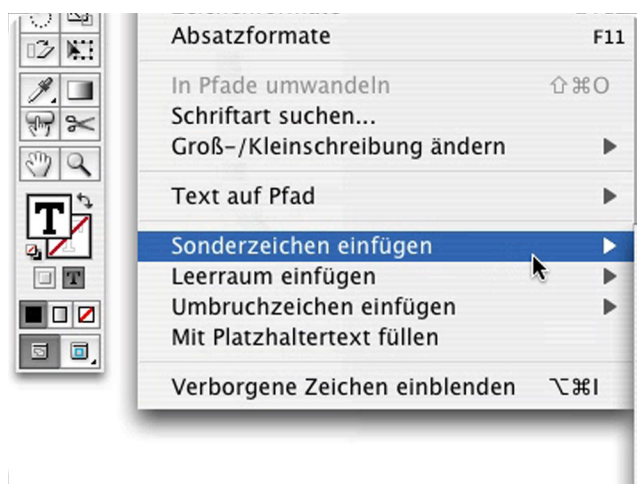


Abb. 4.11:
Eingebettetes Video in Flash MX mit einem Schlüsselbildintervall von 25, einer Qualität von 100 und einer Dateigröße von weniger als 5% der Originalvideodatei

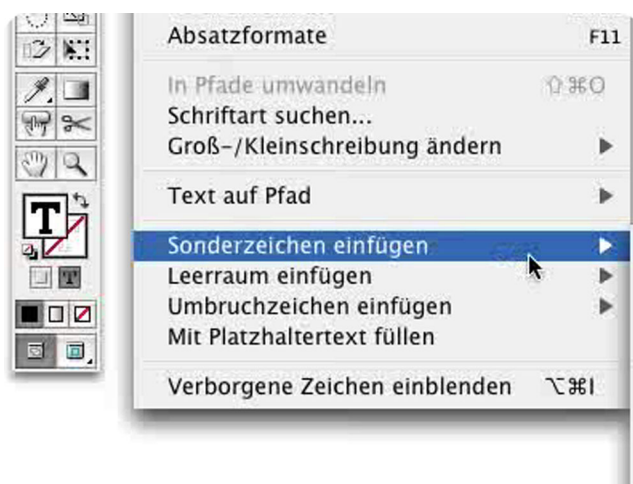


Abb. 4.12:
Eingebettetes Video in Flash MX mit einem Schlüsselbildintervall von 25, einer Qualität von 70 und einer Dateigröße von ca. 1% der Originalvideodatei

Nach dem Import wurden die Flash-Filme in das SWF-Format exportiert und im Ordner *movieclips* gespeichert. Die Benennung der Filme erfolgte anhand ihrer Position im Lernmodul. Beispielsweise erfolgt der Zugriff auf den Film *1.1.2.5.swf* über Kapitel *1.1*, Lernschritt *2* und die Interaktions-Schaltfläche *5*.

4.2 Umsetzung von SCORM 1.2 am Lernmodul

Das Bildungsportal Sachsen unterstützt durch das Learning Management System Saba3 Release4 Lerninhalte der SCORM Version 1.2 oder 1.1. Die aktuelle Version SCORM 2004 wird von dem eingesetzten LMS nicht unterstützt (vgl. 1.4 u. 2.2.5). Da das Lernmodul für den Einsatz im Bildungsportal konzipiert ist, wurde dieses entsprechend der Anforderungen von SCORM 1.2 entwickelt.

Unter SCORM stellen sowohl Assets als auch SCOs Inhalte dar, die von einem LMS gestartet bzw. dargestellt werden können. Jedoch können nur SCOs über die Laufzeitumgebung mit dem LMS kommunizieren. So ist nur über die Verwendung von SCOs die Speicherung von Lernerdaten über die Nutzung des Lerninhalts wie z.B. *Abschlussstatus*, *Benötigte Zeit* und *Versuche gesamt* (vgl. 3.4.1) möglich.

Da den Nutzern des Lernmoduls ‚Typografie praktisch!‘ Rückmeldung über den Abschlussstatus eines Lernobjekts gegeben werden soll, wurden diese als SCOs entwickelt (vgl. 3.3.2). Um die dazu erforderliche Kommunikation über das Application Program Interface (API) gewährleisten zu können, müssen Flash-Filme in ein HTML-Dokument mit entsprechenden Funktionen eingebunden werden [Brown 2002, S. 8].

Die folgenden Abschnitte beschreiben die Erstellung der SCOs für das Lernmodul ‚Typografie praktisch!‘ sowie die Beschreibung der Bestandteile des Lernmoduls durch Metadaten und die Erstellung der Programmstruktur.

4.2.1 Kommunikation zwischen Lernobjekten und Learning Management System

Das SCORM-Referenzmodell legt fest, dass die Kommunikation über das API von Seiten des SCOs ausschließlich über JavaScript-Funktionen zu erfolgen hat [SCORM 2001c, S. 3-16]. Dabei liegt es in der Verantwortung eines SCOs den API-Adapter, der vom LMS bereitgestellt wird, ausfindig zu machen und die Kommunikation mit der Funktion `LMSInitialize("")` zu starten. Sobald dies geschehen ist, kann weitere Kommunikation wie z.B. Austausch von Lernerdaten (Data Transfer) oder Daten zur Fehlerbehandlung (State Management) stattfinden. Um die Kommunikation zu beenden, muss das SCO die Funktion `LMSFinish("")` aufrufen (vgl. Tab. 1.4 u. Abb. 1.15).

Zur Erstellung der SCOs des Lernmoduls ‚Typografie praktisch!‘ wurden die Flash-Filme zunächst im HTML-Format (z.B. *k1_1.html*) veröffentlicht. Um die Kommunikation über den API-Adapter zu gewährleisten, wurden den HTML-Seiten JavaScript-Funktionen über die beiden Dateien *APIWrapper.js* und *SCOFunctions.js* zugewiesen (vgl. Bsp. 4.15).

```
<HTML>
<HEAD>
<scriptlanguage=javascript src="../js/APIWrapper.js"></script>
<scriptlanguage=javascript src="../js/SCOFunctions.js"></script>
<TITLE>k1_1</TITLE>
</HEAD>
...
</HTML>
```

Bsp. 4.15: Einschließen von JavaScript-Funktionen in HTML-Datei

Diese JavaScript-Funktionen werden von ADL zur freien Nutzung über die *Sample RTE 1.2.2*³ zur Verfügung gestellt und besitzen alle wichtigen Funktionen, die für die Kommunikation zwischen SCO und LMS benötigt werden. Dabei sind individuelle Veränderungen möglich, um die Dateien den spezifischen Anforderungen anzupassen. Die im Lernmodul ‚Typografie praktisch!‘ verwendeten Funktionen der Dateien *APIWrapper.js* und *SCOFunctions.js* werden im Folgenden beschrieben.

APIWrapper.js stellt folgende Funktionalitäten zur Verfügung:

- Ausfindigmachen des APIs
- Aufrufen von API-Funktionen (Initialisieren und Beenden der Kommunikation, Datenaustausch)
- Behandlung von Fehlern, die beim Aufrufen von API-Funktionen auftreten können

SCOFunctions.js enthält Funktionen, die von einem SCO direkt zur Steuerung der Kommunikation aufgerufen werden können. Dazu werden Funktionen von *APIWrapper.js* entsprechend ihrer Verwendung durch SCOs gebündelt und bereitgestellt.

Zum Datenaustausch unter SCORM 1.2 wird das CMI-Datenmodell verwendet (vgl. 1.4.2). Somit nutzen *APIWrapper.js* und *SCOFunctions.js* CMI-Datenelemente um Daten zwischen LMS und SCO auszutauschen. Für das Lernmodul ‚Typografie praktisch!‘ werden die

3. Sample RTE 1.2.2 (Sample Run-Time Environment) stellt eine Umgebung zum Starten/Testen von SCORM-Paketen dar; *APIWrapper.js* und *SCOFunctions.js* sind im Beispiel *ADL_Maritime_Navigation* enthalten; die *SampleRTE1_2_2* kann kostenlos von ADL heruntergeladen werden [ADL 2004b].

CMI-Datenelemente `cmi.core.session_time` zum Festhalten des Zeitaufwands für ein Lernobjekt und `cmi.core.lesson_status` zum Speichern des Abschlussstatus eines SCOs eingesetzt. Dabei kann das Datenelement `cmi.core.lesson_status` kann folgende Werte annehmen:

- *completed*: der Inhalt eines SCOs wurde vollständig durchgearbeitete
- *incomplete*: der Inhalt eines SCOs wurde nicht vollständig durchgearbeitete
- *failed*: nicht alle Voraussetzungen zum Bestehen eines SCOs sind erfüllt
- *passed*: alle Voraussetzungen zum Bestehen eines SCOs sind erfüllt
- *browsed*: SCO wurde gestartet
- *not attempted*: SCO wurde (noch) nicht gestartet

Funktionen der Datei *APIWrapper.js*:

- *Funktionen zum Ausfindigmachen des APIs*: die Funktion `getAPIHandle()` wird beim Initialisieren der Kommunikation aufgerufen

```
function getAPIHandle() {
  if (apiHandle == null) {
    apiHandle = getAPI();
  }
  return apiHandle;
}
function getAPI() {
  var theAPI = findAPI(window);
  if ((theAPI == null) && (window.opener != null) &&
    (typeof(window.opener) != "undefined")) {
    theAPI = findAPI(window.opener);
  }
  if (theAPI == null) {
    alert("Unable to find an API adapter");
  }
  return theAPI
}
function findAPI(win) { // Suche nach dem Fenster mit API-Adapter
  while ((win.API == null) && (win.parent != null) && (win.parent != win)) {
    findAPIAttempts++;
    // Note: 7 is an arbitrary number, but should be more than sufficient
    if (findAPIAttempts > 7) {
      alert("Error finding API -- too deeply nested.");
      return null;
    }
    win = win.parent;
  }
  return win.API;
}
```

Bsp. 4.16: Funktionen zum Ausfindigmachen des APIs

- *Funktionen zum Initialisieren und Beenden der Kommunikation:* die Variable `api` steht für den API-Adapter; die Funktionen `LMSInitialize("")` und `LMSFinish("")` stellen Funktionen des API-Adapters dar und werden über das LMS bereitgestellt

```
function doLMSInitialize() {
    var api = getAPIHandle();
    if (api == null) {
        ...
    }
    var result = api.LMSInitialize("");
    ...
    return result.toString();
}

function doLMSFinish() {
    var api = getAPIHandle();
    if (api == null) {
        ...
    } else {
        // call the LMSFinish function that should be implemented by the API
        var result = api.LMSFinish("");
        ...
    }
    return result.toString();
}
```

Bsp. 4.17: Starten und Beenden der Kommunikation

- *Funktionen zum Datenaustausch:* die Funktionen `LMSGetValue(n)`, `LMSSetValue(n,v)` und `LMSCommit()` stellen Funktionen des API-Adapters dar und werden vom LMS bereitgestellt; die CMI-Datenelemente (z.B. `cmi.core.lesson_status`) werden über den Parameter `name` übergeben

```
function doLMSGetValue(name) {
    var api = getAPIHandle();
    if (api == null) {
        ...
    } else {
        var value = api.LMSGetValue(name);
        ...
    }
}

function doLMSSetValue(name, value) {
    var api = getAPIHandle();
    if (api == null) {
        ...
    } else {
        var result = api.LMSSetValue(name, value);
        ...
    }
    return;
}
```

Bsp. 4.18: Funktionen zum Datenaustausch

Funktionen der Datei *SCOFunktions.js*:

- *Funktion, die beim Laden eines SCOs aufgerufen wird:* dient dem Aufruf für Funktionen zum Finden des API-Adapters und zum Initialisieren der Kommunikation; falls das SCO vom Lernenden noch nie besucht wurde, wird der Status der Lerneinheit (*lesson*) auf unvollständig (*incomplete*) gesetzt; zum Festhalten der Lernzeit wird ein *Timer* gestartet (*startTimer*)

```
function loadPage(){
  var result = doLMSInitialize();
  var status = doLMSGetValue( "cmi.core.lesson_status" );
  if (status == "not attempted"){
    // the student is now attempting the lesson
    doLMSSetValue( "cmi.core.lesson_status", "incomplete" );
  }
  exitPageStatus = false;
  startTimer();
}
```

Bsp. 4.19: Aufruf von Funktionen beim Laden eines SCOs

- *Funktion, die zum Beenden eines SCOs aufgerufen wird:* dient dem Beenden der Kommunikation (*doLMSFinish()*); zudem wird die Lernzeit berechnet (*computeTime()*) und der Status der Lerneinheit aktualisiert (*doLMSCommit()*)

```
function doQuit(){
  computeTime();
  exitPageStatus = true;
  var result;
  result = doLMSCommit();
  result = doLMSFinish();
}
```

Bsp. 4.20: Funktionen zum Beenden der Kommunikation

- *Funktion, die zum Ändern des Status eines SCOs aufgerufen wird:* diese Funktion wurde der Datei *SCOFunktions.js* für den Einsatz im Lernmodul ‚Typografie praktisch!‘ hinzugefügt; sie ersetzt die vorhandene Funktion *doContinue(status)*, welche verschiedene Abfragemechanismen bereitstellt, die für das Lernmodul nicht benötigt werden

```
function doChangeStatus(status){
  var result;
  result = doLMSSetValue("cmi.core.lesson_status", status);
}
```

Bsp. 4.21: Funktion zum Ändern des Status einer Lerneinheit

Zur Steuerung der Kommunikation im Lernmodul ‚Typografie praktisch!‘ wurden die JavaScript-Funktionen `loadPage()` und `doQuit()` in die HTML-Seiten der SCOs (z.B. *kl_1.html*) integriert. Sie werden beim Laden bzw. Entladen der HTML-Seite aufgerufen:

```
<BODY bgcolor="#E3E3E3" onload="loadPage()" onunload="doQuit()">
```

Mit diesen beiden Funktionen (`loadPage()` und `doQuit()`) sind bereits die Mindestanforderungen an ein SCO, nämlich der Aufruf der Funktionen `LMSInitialize("")` und `LMSFinish("")`, gewährleistet. Zudem wird der Abschlussstatus beim Laden des SCOs auf *unvollständig* (`incomplete`) gesetzt und ein ‚Timer‘ zum Festhalten der Lernzeit gestartet bzw. gestoppt. Beim Beenden wird die Lernzeit berechnet und der Abschlussstatus aktualisiert.

Die Funktion `doChangeStatus(s)` dient dem Verändern des Abschlussstatus eines SCOs und wird im Lernmodul ‚Typografie praktisch!‘ mit dem Parameter `completed` zum Setzen des Abschlussstatus auf *vollständig* verwendet. Dazu wird diese Funktion (`doChangeStatus('completed')`) im Flash-Film *navigation.swf* beim Öffnen des letzten Lernschritts aufgerufen.

Zum Aufruf von JavaScript-Funktionen aus Flash MX heraus stehen die Aktionen `getURL` und `fscommand` zur Verfügung. Der Vorteil von `fscommand` besteht darin, dass diese Methode sowohl die Kommunikation von Flash nach JavaScript als auch die Kommunikation von JavaScript nach Flash unterstützt. Die Aktion `getURL` unterstützt lediglich die Kommunikationsrichtung Flash zu JavaScript, besitzt jedoch den Vorteil, von mehr Internetbrowsern unterstützt zu werden [Macromedia 2005]. Da für das Lernmodul ‚Typografie praktisch!‘ nur die Kommunikation von Flash nach JavaScript zum Einsatz kommt, wurde hierfür `getURL` verwendet (vgl. Bsp. 4.2.2).

```
function completeLesson() {
    getURL("javascript:doChangeStatus('completed');");
}
var statusLesson = 0;
...
// Schaltfläche zu Lernschritt (i+1) wird aktiviert
function released(i){
    ...
    if ((i==( _global.number_steps-1)) && (statusLesson==0)) {
        completeLesson();
        statusLesson=1;
    }
}
```

Bsp. 4.22: Programmausschnitt aus der Datei *navigation fla*

Der Befehl `doChangeStatus('completed')` wird an Javascript weitergegeben und über *SCOFunktions.js* weiterverarbeitet. Der vom LMS bereitgestellte API-Adapter wird letztendlich über folgenden Befehl angesprochen:

```
api.LMSSetValue("cmi.core.lesson_status", "completed");
```

Bereits während der Erstellung von Lernobjekten nach SCORM empfiehlt es sich regelmäßige Tests zur Überprüfung, ob die Objekte scormkonform sind, durchzuführen. Dazu wurde die ADL SCORM Conformance Test Suite Version 1.2.7 genutzt. Diese stellt für Learning Management Systeme, Sharable Content Objects, Meta Data XML Dokumente und Content Packages eine Testumgebung zur Überprüfung der Einhaltung von SCORM Version 1.2 zur Verfügung. ADL weist aber ausdrücklich daraufhin, dass ein erfolgreiches Bestehen eines solchen Tests nicht mit einer ADL Zertifizierung durch einen dazu befugten Prüfer gleichzusetzen ist.

Die Test Suite kann über die ADL Webseite [ADL 2004a] heruntergeladen werden und benötigt für eine korrekte Funktionsweise die Java Runtime Environment (JRE) 1.3., welche von Java Sun zum Download [Java Sun] zur Verfügung gestellt wird [ADL 2002].

Die SCOs des Lernmoduls ‚Typografie praktisch!‘ wurden über den *Sharable Content Object (SCO) Run-Time Environment Conformance Test* dieser Test Suite erfolgreich geprüft. Die Tabelle 4.2 zeigt einen Ausschnitt aus einem Testprotokoll.

Das Lernmodul ‚Typografie praktisch!‘ ist durch *Content Aggregation Meta-data* und durch *SCO Meta-data* beschrieben, d.h. es wurde eine Meta-data XML-Datei zur Beschreibung des gesamten Lernmoduls sowie für jedes Lernobjekt eine Meta-data XML-Datei erstellt.

SCORM unterscheidet zwischen zwingenden (*mandatory*) und optionalen (*optional*) Datenelemente innerhalb einer solchen XML-Datei. Content Aggregation und SCO Meta-data Dateien besitzen unter SCORM 1.2 dieselben ‚mandatory‘ Elemente. Die Abbildung 4.13 zeigt einen Ausschnitt aus einer XML-Datei zur Beschreibung der Content Aggregation bzw. eines SCOs mit den ‚mandatory‘ Kategorien `general`, `lifecycle`, `metametadata`, `technical`, `rights` und `classification`.

Für das Lernmodul ‚Typografie Praktisch!‘ wurden neben den Angaben zu den ‚mandatory‘ Elementen dieser Kategorien auch optionale Angaben im Bereich `lifecycle` zum Autor und zum Erscheinungsdatum des Lernmoduls bzw. des SCOs gemacht. Zudem beinhaltet die *Content Aggregation Meta-data* zur Beschreibung der Inhaltsstruktur im Bereich `general` über das Element `aggregationlevel` Angaben zur Zusammenstellung der Lerninhalte als *Kurs* (entspricht Level 3).

Die Erstellung der XML-Dateien erfolgte mit dem Programm Adobe GoLive, welches drei verschiedene Bearbeitungsmodi (Layout-, Quelltext-, Gliederungs-Editor) für XML-Dateien zur Verfügung stellt (vgl. Abb. 4.13).

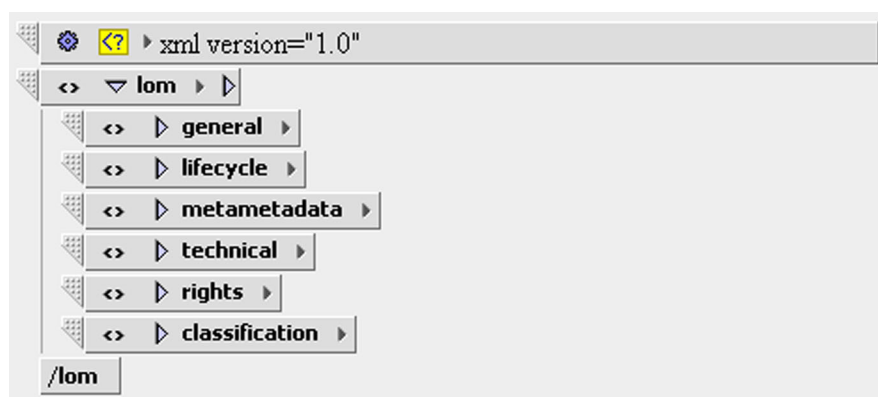


Abb. 4.13:
Ausschnitt aus der Gliederungs-Ansicht einer Meta-Data XML-Datei

Das folgende Beispiel zeigt einen Ausschnitt eines Meta-data XML-Quelltextes mit den ‚mandatory‘ Elementen der Kategorie **general**: title, catalogentry (samt catalog und entry), description und keyword.

```
<?xml version="1.0"?>
<lom xmlns="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.imsglobal.org/xsd/imsmd_rootv1p2p1
imsmd_rootv1p2p1.xsd">
  <general>
    <title>
      <langstring>Typografische Masseinheiten</langstring>
    </title>
    <catalogentry>
      <catalog>URL</catalog>
      <entry>
        <langstring xml:lang="de">TypografiePraktisch/Kapitel1</langstring>
      </entry>
    </catalogentry>
    <description>
      <langstring>Wichtige Masssysteme der Typografie und Voreinstellungen
        fuer Einheiten und Einteilungen in InDesign </langstring>
    </description>
    <keyword>
      <langstring>Masssysteme</langstring>
    </keyword>
    <keyword>
      <langstring>Typografie</langstring>
    </keyword>
    ...
  </general>
  ...
</lom>
```

Bsp. 4.23: XML-Quelltext mit ‚mandatory‘ Elementen der Kategorie general für SCOs/Content Aggregation

Über den *Meta-data Conformance Test* der ADL SCORM Conformance Test Suite Version 1.2.7 lässt sich die korrekte Umsetzung der XML-Dateien nach SCORM 1.2 testen. Dabei wird überprüft, ob alle ‚mandatory‘ Elemente vorhanden und richtig implementiert sind. Zudem werden optionale Elemente auf ihre korrekte Implementierung getestet. Es besteht sowohl die Möglichkeit einzelnen XML-Dateien für Assets, SCOs oder Content Aggregations als auch gesamte SCORM-Pakete zu testen.

Die Metadaten-Dateien des Lernmoduls ‚Typografie praktisch!‘ wurden sowohl einzeln, als auch nach Fertigstellung des gesamten SCORM-Pakets als Ganzes getestet.

4.2.3 Content Packaging

In einem letzten Schritt galt es, den einzelnen Komponenten des Lernmoduls eine Struktur (*Content Structure*) zu geben und diese zu einem Paket (*Content Packaging*) zusammenzufassen, welches vom LMS geöffnet und verarbeitet werden kann (vgl. 1.4.1).

Ein solches *Content Paket* wird nach SCORM anhand einer so genannten *Manifestdatei* (*imsmanifest.xml*) beschrieben, welche die Struktur des Pakets beschreibt und auf physische Dateien verweist (vgl. 1.4.1). Abbildung 4.14 verdeutlicht eine mögliche Gliederung der Manifestdatei durch die Kategorien `metadata`, `organizations` und `resources`.

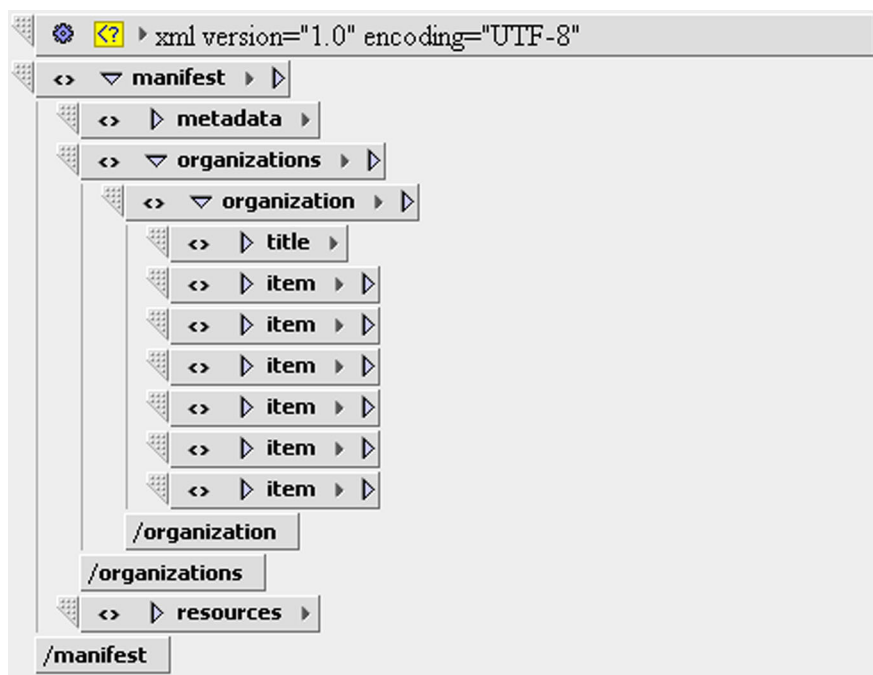


Abb. 4.14:
Ausschnitt aus der Gliederungs-Ansicht der ‚ims.manifest.xml‘-Datei

Der Bereich `metadata` dient der Definition der Content Aggregation Meta-data. Diese kann entweder direkt in der Manifestdatei definiert werden, oder wie es im Falle des Lernmoduls ‚Typografie praktisch!‘ gemacht wurde, in einer separaten Datei erfolgen. Ist dies der Fall, wird im Bereich `metadata` auf diese separate XML-Datei verwiesen.

Im Bereich `organizations` wird die Struktur des Lernmoduls festgelegt. Das Lernmodul ‚Typografie praktisch!‘ wurde durch eine `organization` und der Verwendung von `items` zur Definition der einzelnen Kapitel strukturiert. Auch die einzelnen Lernobjekte (SCOs) innerhalb der Kapitel wurden als `Items` deklariert. Diese besitzen jeweils einen Verweis auf ein Element (`resource`) aus dem Bereich `resources`.

Eine `resource` beinhaltet Informationen über die verwendete physische Datei. Dabei handelt es sich um Pfadangaben zum SCO (`file`), Angaben über SCO Metadaten wie z.B. Speicherort (`adlcp:location`), Schema und Schemaversion sowie um Angaben über Abhängigkeiten (`dependency`). Abhängigkeiten stehen wiederum für eine ‚Resource‘, welche Informationen (Pfadangaben) über Assets enthält, die von einem SCO genutzt werden. Wie z.B. die Beispiel-filme im Lernmodul ‚Typografie praktisch!‘, die separat und nicht im SCO selbst gespeichert werden.

Die Erstellung der Manifestdatei des Lernmoduls ‚Typografie praktisch!‘ und die Speicherung des Lernmoduls als Paket (Packet Interchange File) erfolgte mit dem Reload-Editor Version 1.3 der Firma Reload (Reusable eLearning Object Authoring & Delivery), der kostenlos über die Internetseite von Reload heruntergeladen werden kann [Reload 2004a]

Dieser Editor ermöglicht über ein strukturiertes Manifest-Fenster (*Manifest Pane*) die Erstellung der Manifestdatei per Mausklick bzw. Drag and Drop. Über ein weiteres Fenster ‚Attribut Pane‘ lassen sich die Eigenschaften der einzelnen Elemente der Manifestdatei bearbeiten. Abbildung 4.15 zeigt diese beiden Fenster zur Bearbeitung der Manifestdatei.

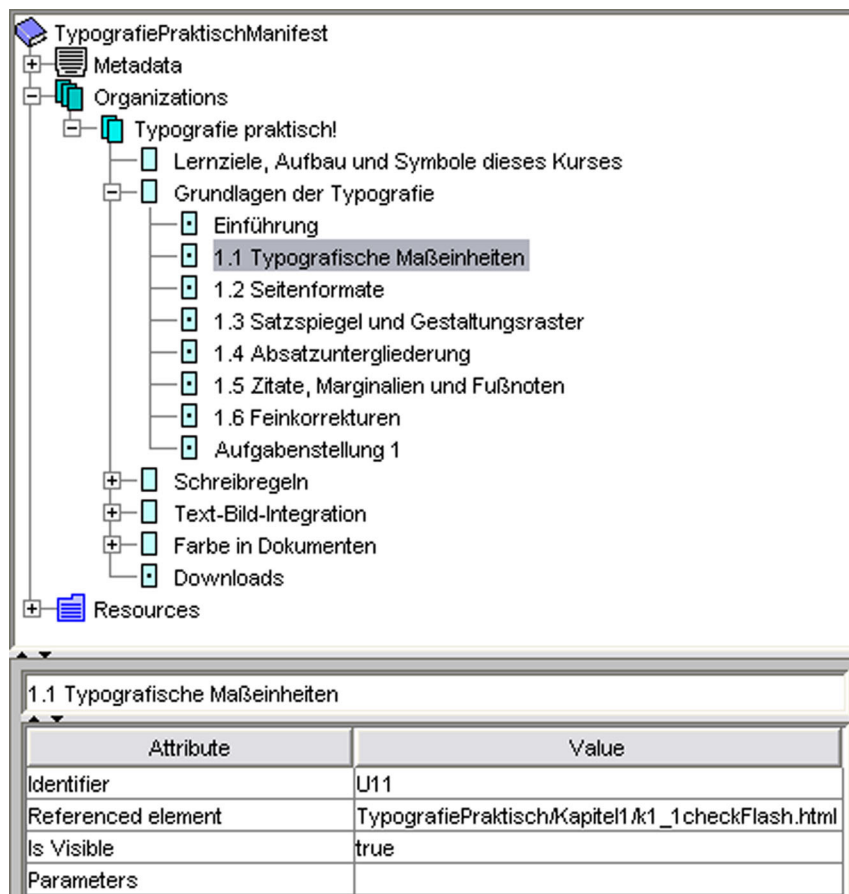


Abb. 4.15:
Manifest und Attribut
Pane des Reload Editors

Der Reload Editor erzeugt automatisch anhand der Manifeststruktur und der angegebenen Eigenschaften eine ‚imsmanifest.xml‘-Datei (vgl. Bsp. 1.1), die zu Testzwecken über einen Webbrowser gestartet werden kann.

Aus der Manifestdatei und den physischen Dateien lässt sich über den Reload Editor ein so genanntes Package Interchange File (PIF) im ZIP-Format erzeugen. Zum Testen einer PIF- oder ‚imsmanifest.xml‘-Datei kann der Reload SCORM Player verwendet werden. Auch dieser kann kostenlos über Reload heruntergeladen werden [Reload 2004b]. Dieser ermöglicht das Importieren und Starten von SCORM-Paketen. Die *Sample RTE 1.2.2* stellt eine entsprechende Testumgebung von ADL dar [ADL 2004b].

Die PIF-Datei des Lernmoduls ‚Typografie praktisch!‘ wurde sowohl dem *Content Packaging Conformance Test* der *ADL Test Suite 1.2.7* unterzogen als auch über den Reload SCORM Player und die *Sample RTE 1.2.2* erfolgreich getestet. Erst danach wurde diese auf den Server des Bildungsportal Sachsen geladen.

4.3 Einstellen und Veröffentlichen des SCORM-Inhalts im System des Bildungsportal Sachsens

Das Bildungsportal Sachsen stellt zu Testzwecken ein Trainingssystem zur Verfügung. Dieses stellt vom Aufbau und von den Möglichkeiten das Pendant zum Bereich ‚Ausbildung‘ (<http://bessel.hrz.tu-chemnitz.de/bps>) des Bildungsportals dar. Die Testumgebung ist aber nicht über das offizielle Internetseite des Bildungsportal [Bildungsportal] zu erreichen, sondern über die Adresse <http://messier.hrz.tu-chemnitz.de/training>. Das Trainingssystem wird in regelmäßigen Abständen mit der Datenbank des Produktionssystems abgeglichen um eine möglichst identische Situation zum eigentlichen Bildungsportal zu gewährleisten.

Das Lernmodul ‚Typografie praktisch!‘ wurde zunächst in das Trainingssystem des BPS gestellt und technischen Tests (Betatests) unterzogen (vgl. 3.1). Nach einer kurzen Überarbeitungsphase wurde das Lernmodul in das eigentliche Produktionssystem des BPS gestellt und ist dort als Selbstlernmodul *nach Einschreibung* verfügbar.

Der nächste Abschnitt beschreibt die Vorgehensweise bei der Einstellung und der Veröffentlichung des Lernmoduls in das Trainingssystem, welche identisch mit der Vorgehensweise im Produktionssystem ist.

4.3.1 Einstellen und Veröffentlichung des Lernmoduls im Trainingssystem

Zur Veröffentlichung neuer Lerninhalte müssen diese zunächst in die Inhaltsdatenbank (*repository*) eingestellt werden. Erst danach ist die Veröffentlichung über den Angebotskatalog möglich (vgl. 2.2.1 u. 2.3.1). Im Folgenden werden die beiden Schritte beschrieben:

Einstellen von SCORM-Inhalt in die Inhaltsdatenbank:

Nach der Anmeldung im Trainingssystem ist zunächst der Wechsel in den Bereich *Inhalt verwalten* notwendig (vgl. Abb. 2.12). Daraufhin wird die Struktur des *Repository* angezeigt. Dort besteht die Möglichkeit direkt *Inhalt hinzuzufügen* oder einen neuen Ordner anzulegen bzw. in einen der bestehenden Ordner zu wechseln. Befindet man sich in einem Ordner, z.B. *HTW Dresden*, stellt man über *Neues Element hinzufügen* neuen Inhalt ein.

Abbildung 4.16 stellt das daraufhin erscheinende Fenster *Inhalt hinzufügen* dar. Dort wählt man *Inhalt importieren – Nach Inhaltsformat – SCORM-Paket* und klickt anschließend auf *Weiter*.



Abb. 4.16:
Inhalt hinzufügen

Im nächsten Schritt ist die Definition von *Inhaltsobjektdetails* notwendig. Diese Details dienen der Definition von Eigenschaften, die für das Speichern von Lerninhalten in der Inhaltsdatenbank relevant sind.

Abbildung 4.17 zeigt die Einstellungsmöglichkeiten. Dabei müssen die rot dargestellten Felder *Name*, *Inhaltsformat*, *Sicherheitsdomäne* und *Inhaltsserver* ausgefüllt werden. Als Hilfestellung befindet sich rechts neben einigen Eingabefeldern (z.B. bei Sicherheitsdomäne und Inhaltsserver) eine kleine Lupe (🔍), über die sich ein Popup-Fenster mit entsprechenden Auswahlmöglichkeiten öffnen lässt.

Darüber hinaus muss der Speicherort des hochzuladenden SCORM-Pakets angegeben werden. Dazu kann man über die Schaltfläche *Durchsuchen* das entsprechende ZIP-Archiv im lokalen Dateisystem auswählen. Sobald alle notwendigen Angaben gemacht sind, lassen sich diese über den Link *Speichern* (in Abb. 4.17 grün umrandet) bestätigen. Daraufhin wird das SCORM-Paket auf den Server des BPS hochgeladen, entpackt und steht über die Inhaltsdatenbank zur Veröffentlichung zur Verfügung.

Inhaltsobjektdetails		Speichern
<p>Definieren Sie die Attribute des Inhaltsobjekts, wählen Sie das zu importierende Objekt und klicken Sie auf Speichern.</p>		
<p>Inhaltsdetails</p> <p>Name * <input type="text" value="Beispiel"/></p> <p>Inhaltsformat * SCORM-Paket</p> <p>Kommunikationsprotokoll Player SCORM 1.1 - 1.2 <input type="button" value="Saba Player"/></p> <p>Sprache <input type="text"/></p> <p>Quellobjekt <input type="text"/> Version 1</p> <p>Sicherheitsdomäne * HD.intern <input type="checkbox"/> Bereit zum Veröffentlichen</p> <p>Anbieter <input type="text"/> Inhaltstyp <input type="text"/></p> <p>Verfügbar ab <input type="text"/> Verfügbar bis <input type="text"/></p> <p>Zielordner HTW Dresden <input type="text"/></p>	<p>Inhaltsserver * BPS-Content1 <input type="button"/></p> <p>Geben Sie das ZIP-Archiv an, das auf dem lokalen Inhaltsserver extrahiert werden soll: <input type="text" value="C:\Dokumente und Einste"/> <input type="button" value="Durchsuchen..."/></p> <p>Dieses Objekt besitzt zurzeit keine benutzerdefinierten Attribute.</p>	

Abb. 4.17: Inhaltsobjektdetails - Speichern

Inhaltsobjektdetails		Veröffentlichen	Duplizieren	Exportieren	Speichern
<p>Bearbeiten Sie im Folgenden die Attribute und klicken Sie auf Speichern, um die Änderungen zu speichern. Veröffentlichen Sie den Inhalt, indem Sie rechts oben auf Veröffentlichen klicken, oder exportieren Sie ihn, indem Sie auf Exportieren klicken.</p>					
<p>Inhaltsdetails</p> <p>Name * test</p> <p>Inhaltsformat * SCORM-Paket</p> <p>Kommunikationsprotokoll Player SCORM 1.1 - 1.2 <input type="button" value="Saba Player"/></p> <p>Sprache Englisch <input type="button"/></p> <p>Quellobjekt <input type="text"/> Version 1</p> <p>Sicherheitsdomäne * HD.intern <input type="checkbox"/> Bereit zum Veröffentlichen</p> <p>Anbieter <input type="text"/> Inhaltstyp <input type="text"/></p> <p>Verfügbar ab <input type="text"/> Verfügbar bis <input type="text"/></p> <p>Erstellt von * htw.wagner</p> <p>Erstellt am * 15.08.2005</p> <p>Änderungsdatum * 15.08.2005</p> <p>Zielordner HTW Dresden <input type="text"/></p>	<p><u>Zu übergeordnetem Ordner wechseln</u></p> <p>Inhaltsserver * BPS-Content1 <input type="button"/></p> <p>Geben Sie das ZIP-Archiv an, das auf dem lokalen Inhaltsserver extrahiert werden soll: test.zip <input type="text"/> <input type="button" value="Durchsuchen..."/></p> <p><u>Mit Content Builder bearbeiten</u></p> <p>Dieses Objekt besitzt zurzeit keine benutzerdefinierten Attribute.</p>				

Abb. 4.18: Inhaltsobjektdetails - Veröffentlichen

Abbildung 4.18 stellt den nächsten Schritt dar. Nachdem die Angaben gespeichert und der Inhalt erfolgreich hochgeladen wurde, stehen über die Seite *Inhaltobjektdetails* zusätzlich zu dem Link *Speichern* weitere Möglichkeiten zur Verfügung (in Abb. 4.18 grün umrandet). Über *Speichern* lassen sich Änderungen bezüglich der Inhaltobjektdetails abspeichern, zudem lässt sich der Inhalt aus dem LMS *exportieren* oder innerhalb des LMS *duplizieren*. Über den Link *Veröffentlichen* kann man den Inhalt in den Angebotskatalog stellen.

Veröffentlichen des Selbstlernmoduls in den Angebotskatalog:

Über den Link *Veröffentlichen* unter *Inhaltobjektdetails* öffnet sich das Fenster *Veröffentlichen* (vgl. Abb. 4.19). Hier kann man zwischen den Formaten *Fach*, *Veranstaltung* und *Selbstlernmodul* zur Veröffentlichung des Inhalts wählen. Für das Lernmodul ‚Typografie praktisch!‘ wurde *Neues Selbstlernmodul* gewählt und über den Link *weiter* bestätigt.

Veröffentlichen

Sie haben ein Inhaltsobjekt zur Veröffentlichung im Angebotskatalog ausgewählt.

Wo möchten Sie veröffentlichen?

Fach-

- Neues Fach
- Vorhandenes Fach

Veranstaltung-

- Neue Veranstaltung
- Vorhandene Veranstaltung

Selbstlernmodul-

- Neues Selbstlernmodul
- Vorhandenes Selbstlernmodul

[Abbrechen](#) [Weiter](#)

Abb. 4.19:
Veröffentlichen

Details zum Selbstlernmodul

Definieren Sie den/die/das product zum Veröffentlichen der Inhalte und klicken Sie auf Speichern, um das Ziel zu erstellen.

<p>Hochschule/Bereich * <input type="text" value="HDVER"/> 🔍</p> <p>Modul-Nr. * <input type="text" value="HDVER001"/></p> <p>Name * <input type="text" value="Testmodul"/></p> <p>Angebots-/Zugangstyp * <input type="text" value="nach Einschreibung"/> 🔍</p> <p>Im Katalog anzeigen ab * <input type="text" value="15.08.2005"/> 📅</p> <p><input checked="" type="checkbox"/> Im Katalog anzeigen (aktiviert) <input checked="" type="checkbox"/> Für Mitarbeiter <input checked="" type="checkbox"/> Für Studenten</p> <p>Kurzbeschreibung <input style="width: 100%; height: 50px;" type="text" value="Testmodul"/></p>	<p>Passwort <input type="text"/></p> <p>Eigentümer/Verantwortlicher <input type="text"/></p> <p>Sicherheitsdomäne <input type="text" value="HD.intern"/> 🔍</p> <p>Version <input type="text"/></p> <hr/> <p>Kommunikation</p> <p><input checked="" type="checkbox"/> Forum aktivieren</p>
--	--

Speichern

Abb. 4.20: Details zum Selbstlernmodul

Im letzten Schritt werden *Details zum Selbstlernmodul* festgelegt (vgl. Abb. 4.20). Dies dient der Definition von Eigenschaften, die für die Veröffentlichung des Inhalts im Angebotskatalog relevant sind. Auch hier müssen die rot dargestellten Felder ausgefüllt werden. Dabei handelt es sich um folgende Eingabefelder:

- *Hochschule/Bereich*: zur Festlegung des Zuordnungsbereichs im Angebotskatalog
- *Modul-Nr.*: dient als eindeutiger Bezeichner zur Speicherung im Datensatz; bei Festlegen von *Hochschule/Bereich* wird hierfür automatisch ein fünfstelliges Präfix generiert, welches durch zusätzliche Beschreibung oder Nummerierung erweiterbar ist
- *Name*: Titel des Selbstlernmoduls, der im Angebotskatalog angezeigt wird
- *Angebots-/Zugangstyp*: hier kann man zwischen *Demo*, *frei nutzbar*, *Informationen* und *nach Einschreibung* wählen (vgl. 2.3.2)
- *Im Katalog anzeigen ab*: dient der Definition des Veröffentlichungsdatums

Für das Lernmodul ‚Typografie praktisch!‘ wurde zusätzlich das Kommunikationsforum aktiviert und eine *Kurzbeschreibung* über das Lernangebot gemacht, die über den Angebotskatalog eingesehen werden kann. Über die Schaltfläche *Speichern* wurde das Selbstlernmodul im Angebotskatalog veröffentlicht.

4.3.2 Technische Tests und Einstellen ins Produktionssystem

Wie bereits erwähnt wurde das Selbstlernmodul ‚Typografie praktisch!‘ im Trainingssystem des BPS auf seine technische Tauglichkeit unter verschiedenen Systemen und mit verschiedenen Internetbrowsern getestet. Die Tests wurden von einer zusätzlichen Testperson, nicht dem Entwickler des Lernmoduls, durchgeführt. Tabelle 4.3 zeigt die Ergebnissen der Tests entsprechend der verschiedenen technischen Voraussetzungen.

Tab. 4.3: Testbericht – technische Tests

	Browser/Player	Version	Internetverbindung	Ergebnis
PC	Internet Explorer	6.0	10,0 MBit/s (Standleitung)	ohne Probleme
	Mozilla Firefox	1.0 Preview Release	10,0 MBit/s (Standleitung)	ohne Probleme
	Mozilla Firefox	1.0 Preview Release	DSL	allgemein ohne Probleme leichtes Ruckeln bei einigen Videosequenzen (Film wird nicht im Vorfeld geladen)
	Flash Player	6		ohne Probleme
MAC	Safari	1.0	10,0 MBit/s	allgemein ohne Probleme teilweise leichtes Ruckeln bei einigen Videosequenzen (Film wird nicht immer im Vor- feld geladen) Im Inhaltsverzeichnis wird der Abschlussstatus nicht angezeigt
	Firefox	1.0	10,0 MBit/s	e-Lernmodul konnte nicht gestartet werden Liegt nicht am e-Lernmodul, da dieses Problem auch bei ande- ren e-Lernmodulen vorlag. Daher keine Kenntnisse über Lauffähigkeit.

Die technischen Tests zeigten, dass es noch Probleme beim Vorausladen der langen Beispielfilme gab. Daraufhin wurden Veränderungen am Preloader im Flash-Film ‚clip fla‘ vorgenommen (vgl. 4.1.2 u. Abb. 4.12). Dieser nutzte zunächst die Movieclip-Methoden `getBytesLoaded()` bzw. `getBytesTotal()` zur Berechnung Verhältnisses zwischen geladenem und noch nicht geladenem Filmanteil. Solange der Dateiheder der zu ladenden Datei vom Flash-Player noch nicht ausfindig gemacht wurde, und somit auch noch kein Wert für `getBytesTotal()` vorliegt, wird `getBytesTotal()` automatisch auf den Wert ‚-1‘ gesetzt. Das Problem lag vermutlich an diesem negativen Wert in Kombination mit der Abfrage `if(getBytesLoaded() >= getBytesTotal())`, die zur Beendigung der Preload-Funktion verwendet wurde. Solange der Header der zu ladenden Datei noch nicht ausfindig gemacht werden konnte, wurde der Ladevorgang durch die Werte ‚0‘ für `getBytesLoaded()` und ‚-1‘ für `getBytesTotal()` vorzeitig abgebrochen.

Der Austausch dieser Methoden durch die Movieclip-Eigenschaften `_framesloaded` bzw. `_totalframes` führte zu einer vollen Funktionsfähigkeit des preloaders für die Beispielfilme. Der Wert für `_totalframes` wird, solange dieser nicht bestimmt werden kann, auf ‚0‘ gesetzt; `_framesloaded` erhält hingegen den Wert ‚-1‘ Die Bedingung zur Beendigung der Preload-Funktion wird also nicht erfüllt.

Eine weitere Veränderung wurde bei der Kommunikation zwischen LMS und SCO vorgenommen. Zunächst wurde die Übermittlung der JavaScript-Befehle aus dem Flash-Film durch die Nutzung der Aktion `fscommand` realisiert. Diese Methode wird aber von deutlich weniger Internetbrowsern unterstützt und führte auch auf dem System Mac OS in Kombination mit dem Webbrowser Safari dazu, dass der Abschlussstatus eines SCOs vom LMS nicht angezeigt wurde. Daraufhin wurde die Methode `getURL` zur Übermittlung von JavaScript-Befehlen aus Flash eingesetzt (vgl. 4.2.1), wodurch der Abschlussstatus auch unter Safari angezeigt wird.

Das Problem unter Mac OS mit dem Internetbrowser Firefox („e-Learnmodul konnte nicht gestartet werden“) konnte leider nicht behoben werden. Das Problem liegt nicht am Lernobjekt (Flash-Film bzw. HTML-Seite) sondern an der Java-Programmierung, welche verwendet wird, um ein Lernobjekt aus dem LMS heraus über ein Java-Applet gestartet wird. Nach Angaben des Bildungsportals kann dieses Problem momentan vom Bildungsportal nicht gelöst werden.

Das Lernmodul funktioniert jedoch unter dem Internetbrowser Safari auf Macintosh-Systemen, welcher unter den technischen Vorgaben als Standardbrowser für Mac OS festgelegt wurde (vgl. 3.2.3).

Anhang A – Inhaltliche Gliederung des Lernmoduls

0 Einleitung

0.1 Lernziele, Aufbau und Symbole dieses Kurses

0.1.1 Lernziele

0.1.2 Aufbau dieses Kurses

0.1.3 Symbole dieses Kurses

1 Grundlagen der Typografie

1.1 Typografische Maßeinheiten

1.1.1 Wichtige Maßsysteme in der Typografie

1.1.2 Voreinstellungen für Einheiten und Einteilungen in InDesign

1.2 Seitenformate

1.2.1 Festlegung des Seitenformats

1.2.2 Einrichten des Seitenformats in InDesign

1.3 Satzspiegel und Gestaltungsraster

1.3.1 Elemente des Satzspiegels

1.3.2 Festlegung des Satzspiegels

1.3.3 Festlegung des Gestaltungsrasters

1.3.4 Einrichten des Satzspiegels und des Gestaltungsrasters in InDesign

1.4 Absatzuntergliederung

1.4.1 Absatzlänge und Absatztrennung

1.4.2 Formatieren von Absätzen in InDesign

1.4.3 Festlegen von Absatzformaten in InDesign

1.5 Zitate, Marginalien und Fußnoten

1.5.1 Zitate und Marginalien

1.5.2 Fußnoten

1.5.3 Zeicheneinstellungen in InDesign

1.5.4 Einstellungen für Fußnotenzeichen in InDesign

1.6 Feinkorrekturen

1.6.1 Absatzumbruch

1.6.2 Absatzumbruchoptionen in InDesign

- 1.6.3 Silbentrennung
- 1.6.4 Silbentrennung in InDesign
- 2 Schreibregeln**
- 2.1 Zahlen und Nummern**
- 2.1.1 Schreibregeln für Zahlen
- 2.1.2 Länderspezifische Schreibregeln für Dezimalzahlen
- 2.1.3 Schreibregeln für Nummern
- 2.1.4 Absatz-Nummerierungen und Aufzählungen
- 2.2 Abkürzungen**
- 2.2.1 Schreibregeln für Abkürzungen
- 2.3 Anführungs- und Auslassungszeichen**
- 2.3.1 Sprachspezifischen Anführungszeichen
- 2.3.2 Sprachspezifische Anführungszeichen in InDesign
- 2.3.3 Auslassungszeichen
- 2.4 Textstriche**
- 2.4.1 Bindestrich
- 2.4.2 Gedankenstrich
- 2.5 Leerraum und Sonderzeichen in InDesign**
- 2.5.1 Einfügen von Leerraum
- 2.5.2 Einfügen von Sonderzeichen
- 3 Text-Bild-Integration**
- 3.1 Gewicht und Ausrichtung von Abbildungen**
- 3.1.1 Gewicht von Abbildungen
- 3.1.2 Ausrichtung von Abbildungen
- 3.2 Bildanordnung im Gestaltungsraster**
- 3.2.1 Bildanordnung in einem einspaltigen Layout
- 3.2.2 Bildanordnung in einem mehrspaltigen Layout
- 3.2.3 Bildanordnung und Konturenführung in InDesign
- 3.3 Abbildungen und Schrift**
- 3.3.1 Schrift in Abbildungen
- 3.3.2 Bildlegenden

3.4 Bildformate in InDesign

3.4.1 Importieren und Exportieren von Bildern in InDesign

3.4.2 Komprimierung von Bildern in InDesign beim PDF-Export

3.4.3 Einstellungen der Komprimierungsformate auf der Seite „Komprimierung“

3.5 Hyperlinks und Lesezeichen

3.5.1 Die Hyperlink-Palette

3.5.2 Hyperlinkoptionen

3.5.3 Die Lesezeichen-Palette

3.6 Schaltflächen

3.6.1 Schaltflächen und deren Zustände

3.6.2 Schaltflächen-Optionen

4 Farbe in Dokumenten

4.1 Funktionen und Stimmungen von Farbe

4.1.1 Funktionen von Farbe

4.1.2 Stimmung und Wirkung von Farbe

4.2 Farbkontraste und Farbharmonien

4.2.1 Farbkontraste

4.2.2 Farbharmonien

4.3 Farbverläufe

4.3.1 Einsatzmöglichkeiten für Farbverläufe

4.3.2 Zwischenstufen bei Farbverläufen

4.3.3 Farbverläufe in InDesign

4.4 Farbe in Abhängigkeit zum Ausgabemedium

4.4.1 Allgemeines zu Farbe und Ausgabemedium

4.4.2 Farbmodelle, Farbumfang und Farbmanagement

4.4.3 Farbmanagement in InDesign

4.4.4 Farbmanagement-Profil zuweisen

5 Downloads

5.1 Downloadmöglichkeiten

5.1.1 Skript und Beispiellösung

Anhang B – Nutzertest

Testpersonen:

Zielgruppe des Lernmodul ‚Typografie praktisch!‘ sind Studenten mit Grundlagenwissen über das Programm InDesign und typografischen Vorkenntnissen. Da das Modul aber über das Bildungsportal Sachsen für jedermann frei zugänglich sein wird, kann es auch von Nutzern ohne entsprechende Vorkenntnisse genutzt werden.

Daher wurde der Test an drei Testpersonen mit verschiedenen Wissensständen in Bezug auf InDesign und Typografie durchgeführt:

- Testperson 1 (T1):

Studentin der Medieninformatik (keine Vorkenntnisse in Typografie oder InDesign, aber geübte Computernutzerin)

- Testperson 2 (T2):

Architekturstudentin (Vorkenntnisse in Typografie und InDesign, geübte Computernutzerin)

- Testperson 3 (T3):

Studentin der Sozialpädagogik (keine Vorkenntnisse in Typografie oder InDesign, eher ungeübte Nutzerin, seltenes Erlernen von neuen Computerprogrammen)

Aufgabenstellungen und Durchführung des Tests:

Die Aufgaben sind so formuliert, dass sie eine möglichst große Spannbreite an Navigationsschritten durchlaufen werden müssen, um die Fragen zu beantworten. So soll festgestellt werden, ob die Navigationsmöglichkeiten von den Nutzern auch als solche erkannt werden. Die Fragen sind nicht nach Schwierigkeitsgrad oder Kapiteln geordnet. Zu jedem der vier Kapitel wurde eine Frage formuliert, zusätzlich gibt es eine allgemeine Frage.

Während die Testpersonen die Aufgaben bearbeiten, macht der Tester Notizen zum Zeitaufwand für die jeweilige Frage, zu den verwendeten Navigationssymbolen (beinhaltet Symbole für Lernschritte, Tipps, zusätzliche Informationen, Tastaturkürzel) und zu Fehlinteraktionen (d.h., wenn eine Testperson z.B. versucht, ein geöffnetes Fenster wieder zu schließen und dafür mehrfach ohne Erfolg ‚herumklickt‘).

Die Testpersonen haben in einem Zeitrahmen von max. 45 Minuten die Möglichkeit, die Fragen schriftlich (in Stichworten) zu beantworten. Dabei wird keine bestimmte Reihenfolge vorgegeben, sondern die Testperson kann so vorgehen, wie sie es im ‚Normalfall‘ bei der Nutzung des Lernmoduls auch machen würde.

Frage 1:

Sie möchten ein Versicherungsunternehmen gründen. Welche Farbe wählen Sie, wenn Sie Ihren Kunden vermitteln wollen, dass Ihr Unternehmen auf *Stabilität* setzt?

Die Lösung ist im Kapitel 4 (Farbe in Dokumenten) unter Punkt 4.1 (Funktionen und Stimmungen von Farben) unter Schritt ② zu finden:

Folgenden Navigationsbereichen müssen zur Beantwortung genutzt werden:

- Hauptmenü (bereitgestellt durch das Bildungsportal)
- Navigation zwischen den einzelnen Lernschritten (1-2)
- Navigation innerhalb eines Lernschritts (durch Hyperlinks)

Ergebnisse:

Zeitaufwand:

5 Minuten (T1), 4 Minuten (T2), 7 Minuten (T3)

Genutzte Navigationssymbole:

Alle drei Testpersonen nutzen die erforderlichen Navigationsbereiche.

Fehlinteraktionen:

keine

Korrekte Antworten:

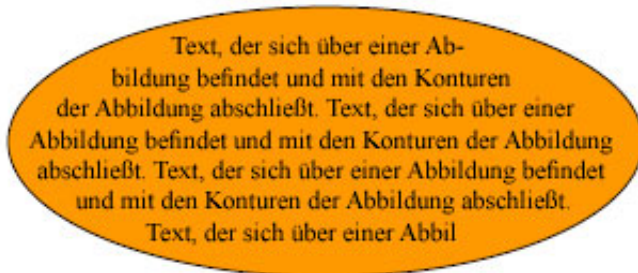
3 (alle Testpersonen beantworten die Frage korrekt)

Zusätzliche Notizen:

T1: entdeckt die Buttons mit den Schritten z.B. ② nicht auf Anhieb.

Frage 2:

Sie wollen eine Grafik in InDesign einfügen, und deren Eigenschaften so festlegen, dass sich folgendes Bild ergibt:



Anmerkung: Der Fließtext befindet sich über der Grafik und wird von den Konturen der Grafik begrenzt.

Die Lösung ist im Kapitel 3 (Text-Bild-Integration) unter Punkt 3.3 (Einfügen von Bildern in InDesign):

Folgenden Navigationsbereichen müssen zur Beantwortung genutzt werden:

- Hauptmenü (bereitgestellt durch das Bildungsportal)
- Navigation innerhalb eines Lernschritts (durch Interaktionsflächen)

Ergebnisse:

Zeitaufwand:

7 Minuten (T1), 7 Minuten (T2), 17 Minuten (T3)

Genutzte Navigationssymbole:

Zu den erforderlichen Navigationssymbolen wurden zusätzlich folgende genutzt:

T2: Auf der Suche nach dem erforderlichen Inhalt öffnet T2 verschiedene Lernobjekte und navigiert auch innerhalb der Lernschritte (① und ②).

T3: Auch T3 öffnet verschiedene Filme, navigiert innerhalb dieser (zwischen ① und ②) und klickt das Symbol (!) an.

Fehlinteraktionen:

T3: beim Schließen des Fensters (!). Nach einigen Versuchen, durch Klicken außerhalb des Symbols, entdeckt T3, dass sich das Fenster durch wiederholte Aktivierung des Symbols (!) schließen lässt.

Korrekte Antworten:

3 (alle Testpersonen beantworten die Frage korrekt)

Zusätzliche Notizen:

T2: nimmt die aktivierbaren Interaktionsflächen zunächst nicht als Schaltflächen wahr. Erst nach längerem Zögern bewegt T2 die Maus über die Flächen und klickt diese an.

Frage 3:

Welches Zeichen benutzt man um ... darzustellen?

Wie heißt dieses Zeichen in InDesign?

Wie kann man es einfügen? Gibt es mehrer Möglichkeiten zum Einfügen dieses Zeichens?

Die Lösungen sind im Kapitel 2 (Mikrotypografie) unter Punkt 2.5 (Anführungs- und Auslassungszeichen) unter Schritt ② und unter 2.6 (Anführungs- und Auslassungszeichen in InDesign) unter Schritt ② zu finden:

Folgenden Navigationsbereichen müssen zur Beantwortung genutzt werden:

- Hauptmenü (bereitgestellt durch das Bildungsportal)
- Navigation zwischen den einzelnen Lernschritten (1-2)
- Navigation innerhalb eines Lernschritts (durch Hyperlinks bzw. Interaktionsflächen)
- Buttons für Tastaturkürzel/Tipps

Ergebnisse:

Zeitaufwand:

4 Minuten (T1), 5 Minuten (T2), 10 Minuten (T3)

Genutzte Navigationssymbole:

Buttons für Tastaturkürzel/Tipps werden nur von T1 und T2 genutzt. Ansonsten verwenden allen drei Testpersonen die erforderlichen Navigationssymbole.

Fehlinteraktionen:

T2: beim Schließen der Fenster für Tipps und Tastaturkürzel. Nach einigen Versuchen, durch Klicken außerhalb der entsprechenden Symbole, gibt T2 es auf und geht zum nächsten Punkt über ohne das Fenster explizit zu schließen.

T3: beim Schließen der Fenster der Interaktionsflächen (mit Dropdown-Menü). Erst durch mehrmaliges ‚Herumklicken‘ wird Fenster geschlossen.

Korrekte Antworten:

T1: Keine vollständige Antwort. Da diese Testperson die Buttons für Tipps und Tastaturkürzel nicht wahrnimmt, kann sie den letzten Teil der Aufgabe nicht korrekt lösen.

T2 und T3: Korrekte Lösung der Aufgaben.

Zusätzliche Notizen:

T2: liest die Überschriften der jeweiligen Lernschritte nicht.

Frage 4:

Finden Sie heraus, welches dieser Maßsysteme in InDesign genutzt wird:

- **Didot-Maß**
- **Pica-Maß**
- **DTP-Maß**

Die Lösungen sind im Kapitel 1 (Typografie) unter Punkt 1.1 (Typografische Maßeinheiten) zu finden:

Folgenden Navigationsbereichen müssen zur Beantwortung genutzt werden:

- Hauptmenü (bereitgestellt durch das Bildungsportal)
- Navigation innerhalb eines Lernschritts (durch Hyperlinks)

Ergebnisse:

Zeitaufwand:

2 Minuten (T1), 3 Minuten (T2), 1 Minute (T3)

Genutzte Navigationssymbole:

Alle drei Testpersonen verwenden die erforderlichen Navigationssymbole.

T2: nutzt zusätzlich die Symbole (!) und (+)

Fehlinteraktionen:

Keine

Korrekte Antworten:

3 (alle Testpersonen beantworten die Frage richtig)

Zusätzliche Notizen:

Alle 3 Testpersonen lösen die Aufgabe mit der ‚richtigen‘ Vorgehensweise und ohne Schwierigkeiten.

Frage 5:

Wofür stehen folgende Symbole?



Die Lösungen sind im Kapitel 0 (Lernziele, Aufbau und Symbole dieses Kurses) zu finden:

Es soll hiermit herausgefunden werden, ob dem Tester diese Symbole schon während der Beantwortung der vorigen Aufgaben aufgefallen sind, und ob die Funktion bekannt sind bzw. bekannt ist, wo man diese nachlesen kann.

Ergebnisse:

Zeitaufwand:

2 Minuten (T1), 1 Minute (T2), 1 Minute (T3)

Genutzte Navigationssymbole:

Alle drei Testpersonen verwenden die erforderlichen Navigationssymbole.

T2: nutzt zusätzlich die Symbole (!) und (+)

Fehlinteraktionen:

keine

Korrekte Antworten:

3 (alle Testpersonen beantworten die Frage richtig)

Zusätzliche Notizen:

Alle 3 Testpersonen haben die Aufgabe mit der ‚richtigen‘ Vorgehensweise ohne Schwierigkeiten gelöst.

T1: bemerkt wird erst durch diese Fragestellung darauf aufmerksam gemacht, dass es Schaltflächen für zusätzliche Tipps und Hinweise gibt.

Allgemeine Feststellungen zu den Vorgehensweisen der Testpersonen:

Alle drei Testpersonen haben sich vor Beantwortung der Fragen Zeit genommen, um das Kapitel 0 (Lernziele, Aufbau und Symbole dieses Kurses) zu lesen. Dabei waren die Vollständigkeit und die Gründlichkeit beim Durchlesen dieses Kapitels für das weitere Zurechtfinden innerhalb des Lernmoduls von Bedeutung:

Vorgehensweise von T1:

T1 hat zwar das Kapitel 0 geöffnet und angefangen zu lesen, hat aber den Hinweis am Ende der Seite („weiter mit der Schaltfläche „2““) wie auch die Schaltflächen zu den jeweiligen Schritten (①, ②, ③) nicht wahrgenommen. Dadurch hat T1 übersehen, dass dieses Kapitel noch mehr Lernschritte besitzt.

T1 ist somit auch nicht zu der Beschreibung der Symbole gelangt, die sich im Schritt 2 des Kapitels 0 befindet. Dies hatte zur Folge, dass T1 die Buttons für Tipps, Tastaturkürzel oder zusätzliche Hinweise nicht wahrgenommen hat und dadurch nicht den gesamten Inhalt des Moduls nutzen konnte.

Vorgehensweisen von T2 und T3:

T2 und T3 haben alle drei Schritte innerhalb des Kapitels 0 geöffnet und gelesen. Beiden Testpersonen sind bei der Beantwortung der Fragen die Buttons für Zusatzinfos (Tipps, Tastaturkürzel, Hinweise) aufgefallen.

Anhang C – Kriterienkatalog zur Usability Evaluation

Beurteilungskriterien für die Untersuchung des Lernmoduls „Typografie praktisch!“

Grundlagen der “guten” Gestaltung von Anwendungen (nach Nielsen u.a.):

Dialoggestaltung

- Einfachheit: zu jeder Zeit nur tatsächlich notwendige Informationen für Nutzer zugänglich
- enge räumliche Anordnung und farbliche Identität zusammengehöriger Informationen
z.B.: eindeutige Trennung der Bildschirmbereiche (Steuerung und Arbeitsbereich) und Darstellung eines Orientierungsbereichs, sowie Gruppierung nach Nähe
- sparsame Verwendung von Bildelementen
- eindeutige Identifizierbarkeit der verschiedenen Interaktions-/ Dialogelemente in Form und Funktionalität;
z.B.: Ist eine interaktive Grafik als solche erkennbar?
- schnelle Erreichbarkeit häufig genutzter Interaktionselemente
z.B.: Ist Menü stets sichtbar?

Sprachgestaltung

- natürlich-sprachliche Begriffe aus Sprachwortschatz des Nutzers in Interaktions-/ Dialogelementen
- Funktionalität des Elementes aus sprachlicher Benennung erkennbar
- standardisierte Begriffe und Benennung für Funktionsbeschreibung
z.B.: Sind wiederkehrende Schaltflächen mit gleicher Funktionalität konsistent dargestellt bzw. bezeichnet (Tooltip)

Systemrückkopplung

- Zustand der Vorgänge im System zu jeder Zeit erkennbar
z.B.: Heißt Stillstand Absturz oder Abarbeitung eines rechenintensiven Vorganges im Hintergrund?
- optimale Systemreaktionszeiten
z.B.: Wie lange dauert es, bis nach Klick auf Menü die Einträge darin sichtbar werden?

Konsistenz

- Erfolgt stets gleiche Reaktion des Systems nach gleicher Aktion des Nutzers?
- Kann Nutzer zu jedem Zeitpunkt die Abläufe des Systems kontrollieren und steuern?
- Kann nach Aktion stets zur Ausgangsposition wie vor der Aktion zurückgekehrt werden?

inhaltliche Synchronität angewandter Medien

- Passen zusammen präsentierte Medien formal und inhaltlich zusammen?

wiedergegebene Inhalte durch die Medien

- Sind die wiedergegebenen Inhalte interpretierbar und erkennbar?(Lesbarkeit)

ästhetischer Anspruch

- Ist das Layout “passend” zum Thema/ Inhalt gewählt und ästhetisch ansprechend?
- Unterstützt die Gestaltung der Designelemente bzw. des Gesamtlayouts den Zweck der Lernumgebung

Benutzermotivation und Interaktionsgestaltung nach dem ARCS-Modell (Keller 1983):

Aufmerksamkeit

- Gewinnen bzw. Aufrechterhalten der Aufmerksamkeit des Lernenden über den gesamten Kurs
z.B. Verwenden neuer, überraschender widersprüchlicher oder ungewisser Ereignisse

Relevanz

- Bedeutsamkeit des Lehrstoffs vermitteln (Lehrzielorientierung, Vertrautheit, Anpassung an Motivationsprofile)

Erfolgszuversicht

- Fördern von positiven Erfolgserwartungen bei dem Lernenden (Lernanforderungen, Erfolgchancen)

Zufriedenheit

- Vermeidung von Frustration bzw. Demotivation und Förderung von Zufriedenheit des Lernenden
z.B. Anwendung des Gelernten, Lob, gerechte Beurteilungsmaßstäbe

Anhang D – Ergebnisse der Usability Evaluation

Auswertung der Heuristische Evaluation des Lernmoduls „Typografie praktisch!“

Grundlagen der “guten” Gestaltung von Anwendungen (nach Nielsen u.a.):

• *Dialoggestaltung*

Die Dialoggestaltung ist einfach gehalten und es sind nur die relevanten Informationen sichtbar. Die Benennung der Menüpunkte ist inhaltlich klar und eindeutig.

Die notwendige Aufteilung des Bildschirms in Steuerungs- und Arbeitsbereich ist zwar vorhanden, jedoch ist die Benutzung des Popup-Systems des Browsers aus folgenden Gründen als kritisch zu betrachten: Es kann nicht vorausgesetzt werden, dass bei den Browsern der Nutzer das Popup-System so funktioniert, wie es im Design angelegt war bzw. Popups überhaupt möglich sind. Weiterhin ist zu bemängeln, dass beim Starten eines weiteren Unterkapitels, dessen Inhalt nicht in das bestehende Popup-Fenster geladen wird, sondern ein neues zusätzliches Popup-Fenster geöffnet wird. So kann es passieren, dass eine Unzahl von Popup-Fenstern gleichzeitig geöffnet ist, was nicht zur Übersichtlichkeit beiträgt. Die Kapitel und ihre Unterkapitel sollten jeweils eine stärkere gestalterische Einheit bilden, d.h. die Kapitel sollten optisch deutlicher voneinander getrennt sein.

Bildschirmelemente wie Grafiken und Schaltflächen sind angemessen und sparsam verwendet.

Die Benutzung von Farbe als gestalterisches Mittel zur Unterstützung von Orientierung und Interaktion ist unzureichend eingesetzt und nicht eindeutig. Die Farbe orange hat zu viele Bedeutungen bezüglich Orientierung und Interaktion:

- Rollovereffekt
- Anzeigen des gerade aktiven Abschnitts
- Anzeigen von Benutzbarkeit einer Schaltfläche

Bei Schaltflächen ist zu bemerken, dass die Farbe grau hauptsächlich dazu verwendet wird, um eine Schaltfläche als unbenutzbar zu definieren. Dies wird nicht konsequent durchgehalten. Die Buttons für die jeweiligen Lernschritte sind benutzbar, obwohl sie mit der Farbe grau markiert sind. Dies ist widersprüchlich.

Die aktuelle Position innerhalb des gesamten Lernmoduls ist nur schwer ersichtlich. Der komplette Navigationspfad sollte auch in den Unterkapiteln ersichtlich sein. Dies unterstützt die Orientierung im Lernmodul.

- ***Sprachgestaltung***

Interaktions- und Dialogelemente sind sprachliche unmissverständlich und eindeutig, sowie auf den Sprachwortschatz der potentiellen Nutzer zugeschnitten.

- ***Systemrückkopplung***

Der vom Lernenden durchlaufene bzw. abgearbeitete Lernweg ist nicht nachvollziehbar. Die abgearbeiteten Kapitel und Unterkapitel werden nicht als „bearbeitet“ markiert bzw. es gibt keine Verlaufsinformationen („History“).

Bei langsamen Internetverbindungen kann es zu Verzögerungen beim Laden von Flashfilmen kommen. Daher wäre es angebracht in einem solchen Fall den Nutzer über das Laden und dessen Fortschritt zu informieren.

- ***inhaltliche Synchronität angewandter Medien***

Bei Monitoren mit schlechterer Qualität führt der geringe Kontrast zwischen Text und Hintergrund (mit den horizontalen Linien) zu einer schlechten Lesbarkeit.

- ***ästhetischer Anspruch***

Das Layout ist als zum Inhalt passend und als qualitativ hochwertig einzuschätzen.

Die Gestaltung der Designelemente bzw. des Gesamtlayouts erfüllt den Zweck der Lernumgebung.

Benutzermotivation und Interaktionsgestaltung nach dem ARCS-Modell (Keller 1983):

• ***Aufmerksamkeit***

Der Umfang der einzelnen Lektionen ist angemessen dosiert, dadurch wird eine Konzentration des Lernenden auf den wesentlichen Inhalt erleichtert.

Der sprachliche Still ist sehr sachlich und nüchtern. Es ist zu empfehlen einen etwas personalisierteren Sprachstil zu verwenden, um eine Beziehung zum Lernenden zu verstärken. Ein direktes Ansprechen des Lernenden würde dies unterstützen. Durch eine Konfrontation des Lernenden mit Fragen oder zu lösenden Problemen direkt während der Lektionen könnte Neugier und Fragehaltungen angeregt bzw. informationssuchendes Verhalten stimuliert werden. Dies fördert nicht nur die Aufmerksamkeit sondern erhöht auch den Lerneffekt.

• ***Relevanz***

Das Lernmodul bietet keine Anpassung an Vorkenntnisse, Ansprüche und Ziele des Lernenden (z.B. Grundkurs, Aufbaukurs).

• ***Erfolgszuversicht***

Die gute Dosierung der Lektionen fördert die positiven Erfolgserwartungen beim Lernenden. Die Aufgaben beziehen sich direkt auf das vorher Gelernte und wenden es an. Sie sind nicht zu schwer oder zu leicht.

Der Lernende sollte über den Inhalt und die Lernziele am Anfang eines jedem Kapitels informiert werden. Es wäre denkbar, dass die einzelnen Kapitel im Hauptmenü auch mit einem Informationsfenster hinterlegt werden, das eine kurze allgemeine Einführung in die jeweilige Thematik und die zu erwartenden Lernziele liefert.

• **Zufriedenheit**

Bei den Aufgaben ist keine Erfolgskontrolle möglich. Hier wäre zu überlegen, ob dem Lernenden eine Musterlösung zu Verfügung gestellt wird. Dies fördert die Zufriedenheit beim Lernenden. Bei fehlender Zufriedenheit verursacht durch ein Abweichen der Anstrengungen des Lernenden gegenüber den Erwartungen kann es schnell zur Demotivation kommen.

Abkürzungsverzeichnis

ADL	Advanced Distributed Learning
AICC	Aviation Industry Computer-Based Training Committee
ANSI	American National Standards Institute
API	Application Programming Interface
ARIADNE	Alliance of Remote Instructional Authoring and Distribution Networks for Europe
CEN	Comité Européen de Normalisation
CMI	Computer Managed Instruction
DIN	Deutsche Institut für Normung
HCM	Human Capital Management
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electric and Electronic Engineers
IMS	Instructional Management Systems
ISO	International Organisation for Standardization
LCMS	Learning Content Management System
LMS	Learning Management System
LOM	Learning Object Metadata
LTSC	Learning Technology Standards Committee
LO	Learning Object
RLO	Reusable Learning Object
RTE	Run Time Environment
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model
SMWK	Sächsische Staatsministerium für Kunst und Wissenschaft
XML	Extensible Markup Language

Glossar

Die folgenden Begriffserklärungen beruhen auf den Definitionen von [Baumgartner u.a. 2002, S. 296ff] sowie [Niegemann u.a. 2004, S. 341ff] und dem Online-Lexikon [Wikipedia 2005].

Application Programming Interface Programm- bzw. Anwendungsschnittstelle; definiert Klassen und Funktionen und stellt diese nach außen hin zur Verfügung

Application-Sharing Gemeinsame oder gleichzeitige Nutzung einer Softwareanwendung über Datennetze. Application-Sharing ist Bestandteil von Desktop-Konferenzsystemen und Virtual-Classroom-Tools.

Assessment-Tools Werkzeuge durch Durchführung von Beurteilungsverfahren zur systematischen Bewertung von Fähigkeiten, Fertigkeiten und Wissensstand von Lernenden.

Asynchron Nicht gleichzeitig, ungleichzeitig; beim asynchronen Lernen findet der Wissensvermittlungsprozess zwischen Lehrern und Lernenden zeitlich versetzt statt, z.B. durch die Nutzung von E-Mails.

Blackboard ‚Virtueller Aushang‘; Elektronische Form der Pinnwand mit der man Mitteilungen einer bestimmten Gruppe (z.B. Mitlernende) machen kann.

Chat Synchrone, computergestützte Kommunikationsform über Tastatur zwischen zwei oder mehreren Personen.

Communities Zugehörigkeitsgruppen, denen man sich z.B. durch eine bestimmte Rollenzugehörigkeit (z.B. Student der Medieninformatik) anschließen kann; durch die Zugehörigkeit zu einer Community kann man z.B. in einem Lernmanagementsystem Zugang zu bestimmten Kommunikationsgruppen (z.B. Foren, Chats, etc.) erhalten

Curriculums-Verwaltung Verwaltung des Lehrplans bzw. der Lehrzielvorgabe.

Diskussionsforen → siehe *Forum*

E-Business Electronic Business bzw. Elektronischer Handel (auch Internetverkauf, Elektronischer Marktplatz, Virtueller Marktplatz, E-Commerce, E-Business); umfasst im weiteren Sinne jede Art von geschäftlicher Transaktion, bei der elektronische Kommunikationstechniken zur Leistungsanbahnung, -vereinbarung oder -erbringung eingesetzt werden.

Elektronische Agenda ‚Virtueller Terminplaner‘; webbasiertes Werkzeug zur Verwaltung von Terminen und Aufgaben.

E-Mail Asynchrones elektronisches Kommunikationswerkzeug; „elektronische Post“.

Expertensuche Suchverfahren, welches die Suche digital gespeicherter Daten durch die Angabe bestimmter Suchkriterien konkretisiert.

FAQ / FAQ Listen Frequently Asked/Answered Questions; was soviel bedeutet wie *häufig gestellte/beantwortete Fragen*. Bei FAQ Listen handelt es sich also um die Auflistung von Antworten zu häufig gestellten Fragen.

Forum / Diskussionsforum Kommunikationsplattform zum Austausch von Gedanken und Erfahrungen. Die Kommunikation findet dabei asynchron und schriftlich über Datennetze statt.

Group Calendaring Ermöglicht die Planung und Abstimmung von Terminen in einer Gruppe und kann als Ergänzung zur Elektronischen Agenda verwendet werden.

Human Capital Management Personalinformationssysteme/Personalwirtschaftssysteme; steht für die konsequente Ausrichtung von Mitarbeiterprozessen und -kompetenzen an den Erfordernissen der Geschäftsprozesse und -strategie.

Information Retrieval Informationswiedergewinnung oder Informationsbeschaffung; wird eingesetzt um aus großen elektronisch verfügbaren Informationsmengen passende Information herauszufinden.

Instant Messaging Kommunikationsform, zum Feststellen für Internetnutzer, ob „Freunde“ zum selben Zeit online sind. Ist dies der Fall, können Nachrichten an diese versendet werden.

JavaScript Objektorientierte Programmiersprache, die von der Firma Netscape entwickelt wurde, um statische HTML-Seiten optimieren und dynamisch gestalten zu können

Java-Applet Kleines Computerprogramm, das in der Programmiersprache Java verfasst wurde; der Begriff Applet bedeutet im eigentlichen Sinne ‚Anwendungs-Schnipsel‘ und setzt sich aus **Application** und **snippet** zusammen.

Kollaboration Zusammenarbeit mehrerer Einzelpersonen oder einer Gruppe; unter kollaboratives Lernen versteht man das Lernen in einer Gruppe.

Lerntheorien Lernen steht zum einen für den Erwerb von bereits existentem Wissen zum anderen kann Lernen aber auch für die Entstehung neuen Wissens durch Erfahrung, Entdeckung oder Forschung bedeuten. Verschiedene Lerntheorien versuchen zu beschreiben, wie Lernprozesse ablaufen. Bei der Gestaltung von Lernsystemen nehmen folgende Lerntheorien grundlegende Rollen ein:

Behavioristisch: Lernprozess beruht auf Erfahrungen, gesteuert durch äußere Reize, positive und negative Verstärkung.

Kognitivistisch: Lernen als vielschichtiger, komplexer Prozess der durch Aufnahme, Verarbeitung und Speicherung von Informationen; dabei spielen persönlichen Zielen, individuellen Einstellungen und Vorwissen des Lernenden ein bedeutende Rolle.

Konstruktivistisch: Lernen als aktiver und individueller Prozess der Wissenskonstruktion; basiert auf subjektiven Entscheidungsstrukturen und ist ohne aktive Beteiligung des Lernenden nicht möglich.

Logfile-Analyse Ein Logfile (Logdatei) enthält ein automatisch erstelltes Protokoll über Aktionen während der Computernutzung. Bei der Logfile-Analyse wird die Logdatei eines Zeitraumes nach bestimmten Kriterien untersucht und bewertet.

Logging Protokollierung von Ereignissen oder Aktionen des Benutzers eines Systems oder einer Anwendung.

Mailing Liste Eine Mailing Liste bietet Nachrichtenaustausch in Briefform (E-Mail) innerhalb einer geschlossenen Gruppe. Innerhalb der Gruppe ist dieser Nachrichtenaustausch öffentlich.

Metadaten Unter Metadaten ("Daten über Daten") versteht man strukturierte Daten, die eine Informationsressource beschreiben und diese dadurch besser auffindbar machen.

Multimedia-Authoring Integration von multimedialen Elementen in Offline- oder Online-Anwendungen mit Autorenwerkzeugen.

New Economy ‚Neue Wirtschaft‘; steht für die Idee, dass durch Computern und neue Kommunikationsmedien im Zuge einer *digitalen Revolution* eine radikal neue Wirtschaftsform entsteht. Im Sinne der New Economy liegen die Prioritäten im weltweiten Wettbewerb um innovative Ideen, in der Verarbeitung von Informationen.

Newsgroups Dienen dem Austausch und dem Veröffentlichen von Nachrichten, die über einen Newsserver weiteren Benutzern zur Verfügung gestellt werden.

Newsletter Meist elektronisches Rundschreiben.

Online-Teilnehmerlisten Listen der Personen einer bestimmten (Teilnehmer-)Gruppe die zu dem aktuellen Zeitpunkt ins System eingeloggt sind.

Open Source ‚Quelloffenheit‘ wird meist auf Computer-Software angewandt und bedeutet, dass es jeder die Möglichkeit hat, den Einblick in den Quelltext eines Programms zu haben.

Online-Voting Abstimmung bzw. Stimmabgabe über das Internet.

Pädagogischer Agent Charaktere, die über den Bildschirm durch Lernprogramme führen und den Lernprozess unterstützen. Sie agieren als Tutoren, Coaches, Kritiker und Co-Lerner.

Plug-In Zusatzmodul für ein Programm, welches weitere Funktionen zur Verfügung stellt. Web-Browser nutzen Plug-Ins um bestimmte Datentypen abspielen bzw. darstellen zu können.

Prozessdaten Daten, die durch einen Rechenprozeß aus gespeicherten Attributen gewonnen werden. Ein Beispiel dafür ist die Häufigkeit der Nutzung eines Online-Kurses. Diese wird mit jeder Nutzung um den Wert 1 erhöht.

Reporting-Tool Bewertungswerkzeug; dient im E-Learning Bereich der Bewertung von Fähigkeiten, Fertigkeiten und Wissensstand von Lernenden.

Spezifikation Detaillierte Dokumentation zur Lösung eines Sachverhaltes. Spezifikationen für Software enthalten z.B. konkrete Angaben zur Syntax, Semantik oder zur Implementierung.

Synchron Gleichzeitig; synchrones Lernen findet unter anderem im Klassenzimmer, im virtuellen Klassenzimmer oder auch im Chat statt.

Thread Diskussionsfaden in einem Diskussionsforum → siehe *Forum*

Virtual Classroom / Virtuelles Klassenzimmer Vernetzung geographisch getrennter Schüler über das Internet. Vorwiegend synchrone Kommunikationsformen wie z.B. Whiteboard oder Chat ermöglichen den Austausch zwischen Lernenden und Lehrern.

Web-Course-Tool Werkzeug zur Erstellung und Präsentation von Online-Kursen.

Web-Safari / Web-Tours Gleichzeitiges Surfen räumlich voneinander getrennter Nutzer durch ein Webangebot unter Anleitung.

Whiteboarding Nutzung einer virtuellen Tafel, welche Text- und Zeichenwerkzeuge besitzt. Die Verwendung eines Whiteboards über ein Netzwerk ermöglicht gemeinsames Entwerfen und Betrachten von Skizzen und Zeichnungen.

Abbildungsverzeichnis

Abbildung 1.1:	Beispiele für Lerntechnologien	11
Abbildung 1.2:	Die fünf hauptsächlichen Funktionsbereiche von Lernplattformen	12
Abbildung 1.3:	Learning Management System	12
Abbildung 1.4:	Die Familie der E-Learning-Content Autorenwerkzeuge	14
Abbildung 1.5:	Content Management System	16
Abbildung 1.6:	Hierarchie der RLOs	17
Abbildung 1.7:	Learning Content Management System	18
Abbildung 1.8:	Funktionelle Kategorisierung von E-Learning Standards	21
Abbildung 1.9:	Kooperationsnetzwerk der E-Learning Standardisierungsinitiativen ...	26
Abbildung 1.10:	SCORM als eine Sammlung von Spezifikationen	28
Abbildung 1.11:	Beispiele für Assets	30
Abbildung 1.12:	Beispiel für ein SCO	30
Abbildung 1.13:	Content Aggregation	31
Abbildung 1.14:	IMS Content Packaging (PIF)	36
Abbildung 1.15:	Struktur einer Manifestdatei	36
Abbildung 1.16:	Launch, API und Data Model in der SCORM RTE	39
Abbildung 1.17:	API-Adapter Zustandsübergänge	41
Abbildung 2.1:	Ausschreibungen 2001 – 2004	47
Abbildung 2.2:	Saba Module <i>Saba Enterprise 2005</i>	50
Abbildung 2.3:	Zusammenhang zwischen Inhaltsdatenbank und Angebotskatalog unter Saba	56
Abbildung 2.4:	Nutzung des Content Builders	60
Abbildung 2.5:	Assistent (Wizard) zur Erstellung von Lerninhalten	61
Abbildung 2.6:	Fenster zur Erstellung von Metadaten	61
Abbildung 2.7:	Ausschnitt aus dem Programm Lectora	62
Abbildung 2.8:	Durch Saba3 Release4 unterstützte E-Learning Standards	66
Abbildung 2.9:	Startseite des Bildungsportals Sachsen	68
Abbildung 2.10:	<i>Angebotskatalog</i> des Bereichs Ausbildung	70
Abbildung 2.11:	Bereich <i>Meine Einschreibungen</i>	71

Abbildung 2.12:	Auswahlfeld <i>Wechseln zu</i>	72
Abbildung 2.13:	Kommunikationsmöglichkeiten unter <i>Meine Veranstaltungen</i>	73
Abbildung 2.14:	Beispiel einer Startseite unter ACollab	75
Abbildung 3.1:	Das CBT-Entwicklungsmodell im Überblick	76
Abbildung 3.2:	Startseite des Lernmoduls ‚Interaktive Einführung in die Arbeitsweisen der Kartographie‘	82
Abbildung 3.3:	Struktur des Lernmoduls ‚Typografie praktisch!‘	89
Abbildung 3.4:	Einsatz von Screenshots als Interaktionsflächen	92
Abbildung 3.5:	Aufbau der Formblätter des Lernmoduls ‚Typografie praktisch!‘	95
Abbildung 3.6:	Ausschnitt des Inhaltsverzeichnisses und der Lernfortschrittsanzeige ..	97
Abbildung 3.7:	Navigationsleiste	98
Abbildung 3.8:	Interaktionsmöglichkeit bei theoretischem Inhalt	99
Abbildung 3.9:	Interaktionsflächen bei praktischem Inhalt	99
Abbildung 3.10:	Schaltflächen zum Anzeigen zusätzlicher Informationen	99
Abbildung 3.11:	Ausschnitt aus einem Fenster zum Abspielen von Filmsequenzen	100
Abbildung 3.12:	Grobe Einteilung des Lernmoduls in Arbeitsbereich (1), Orientierungsbereich (2) und Steuerungsbereich (3)	102
Abbildung 3.13:	Screenedesign des Lernmoduls ‚Typografie praktisch!‘	103
Abbildung 3.14:	Ursprüngliches Aussehen des Inhaltsfensters	109
Abbildung 4.1:	Ladereihenfolge der Flash-Filme des Lernmoduls ‚Typografie praktisch!‘	113
Abbildung 4.2:	Lernobjekt bestehend aus Hintergrund und Navigation mit unterteiltem Arbeitsbereich für die Präsentation von Lerninhalten mit Interaktionsmöglichkeiten	116
Abbildung 4.3:	Lernobjekt bestehend aus Hintergrund und Navigation ohne Unterteilung des Arbeitsbereichs für die Darstellung von Einleitungen, Aufgabenstellungen und Downloadmöglichkeiten	116
Abbildung 4.4:	Grautöne im Navigationsbereich	117
Abbildung 4.5:	Beispiel für theoretischen Lerninhalt mit Interaktionsmöglichkeiten im linken Bereich und der Darstellung von Inhalt mit einer Interaktionsfläche für zusätzliche Tipps im rechten Bereich	121

Abbildung 4.6:	Beispiel für zusätzlichen Hinweis im rechten Bereich des Lernmoduls	124
Abbildung 4.7:	Beispiel für praktischen Lerninhalt mit Interaktionsmöglichkeiten im linken Bereich und der Darstellung von Inhalt mit zwei Interaktionsflächen für zusätzliche Tipps und Beispielfilm im rechten Bereich	125
Abbildung 4.8:	Beispiel für ein Dropdown-Menü	128
Abbildung 4.9:	Filmsteuerung in <i>clip.swf</i>	130
Abbildung 4.10:	Nutzung von Geräteschriftarten (hier Verdana) und eingebetteten Schriftarten (hier Times New Roman)	133
Abbildung 4.11:	Eingebettetes Video in Flash MX mit einem Schlüsselbildintervall von 25, einer Qualität von 100 und einer Dateigröße von weniger als 5% der Originalvideodatei	136
Abbildung 4.12:	Eingebettetes Video in Flash MX mit einem Schlüsselbildintervall von 25, einer Qualität von 70 und einer Dateigröße von ca. 1% der Originalvideodatei	136
Abbildung 4.13:	Ausschnitt aus der Gliederungs-Ansicht einer Meta-Data XML-Datei	145
Abbildung 4.14:	Ausschnitt aus der Gliederungs-Ansicht der ‚ims.manifest.xml‘-Datei	147
Abbildung 4.15:	Manifest und Attribut Pane des Reload Editors	149
Abbildung 4.16:	Inhalt hinzufügen	151
Abbildung 4.17:	Inhaltobjektdetails – Speichern	152
Abbildung 4.18:	Inhaltobjektdetails – Veröffentlichen	152
Abbildung 4.19:	Veröffentlichen	153
Abbildung 4.20:	Details zum Selbstlernmodul	154

Tabellenverzeichnis

Tabelle 1.1: Beispiele für Basistechnologien	10
Tabelle 1.2: Kategorien des SCORM Meta-data Information Model	32
Tabelle 1.3: <i>Mandatory</i> (M) und <i>optional</i> (O) Datenelemente	35
Tabelle 1.4: Funktionen zur Kommunikation zwischen SCO und LMS	40
Tabelle 1.5: CMI-Datengruppen	42
Tabelle 2.1: Unterstützte Dateiformate	54
Tabelle 3.1: Ausschnitt aus dem Basaltext des Lernmoduls ‚Typografie praktisch!‘	94
Tabelle 3.2: Veränderungen an den Interaktionsflächen	107
Tabelle 4.1: Globale Variablen eines <i>Start-Films</i> am Beispiel von <i>k1_1.swf</i>	114
Tabelle 4.2: Protokollausschnitt eines SCO Conformance Test durch die Test Suite 1.2.7	144
Tabelle 4.3: Testbericht – technische Tests	155

Verzeichnis der Codebeispiele

Beispiel 1.1:	Datentyp <i>LangString</i>	33
Beispiel 1.2:	Datentyp <i>Date</i>	33
Beispiel 1.3:	Datentyp <i>Vocabulary</i>	34
Beispiel 1.4:	Manifestdatei	37
Beispiel 4.1:	Laden des Films <i>naviagtion.swf</i>	116
Beispiel 4.2:	Darstellung der Schaltflächen auf der Bühne	118
Beispiel 4.3:	Ereignisbehandlung beim Loslassen der Maus auf einer Schaltfläche	119
Beispiel 4.4:	Datei ‚linkInteraktion.as‘	121
Beispiel 4.5:	Darstellung und Interaktion der Links bei theoretischem Inhalt	122
Beispiel 4.6:	‚hitTest‘ wird bei jedem Maus-Druck aufgerufen	124
Beispiel 4.7:	Datei ‚buttonInteraktion.as‘	125
Beispiel 4.8:	Darstellung und Interaktion der Schaltflächen bei praktischem Inhalt	126
Beispiel 4.9:	Erstellung eines Dummys für ein Dropdown-Menü	128
Beispiel 4.10:	Ereignisbehandlung beim Loslassen der Maus auf einer Schaltfläche	129
Beispiel 4.11:	Laden des Beispielfilms	130
Beispiel 4.12:	<i>Preloader</i> zur Steuerung der Anzeige des Beispielfilms	130
Beispiel 4.13:	Basisskript der Schaltflächeninstanz für Schiebereglerfunktion	131
Beispiel 4.14:	Flash-Detection Skript von Macromedia	132
Beispiel 4.15:	Einschließen von JavaScript-Funktionen in HTML-Datei	138
Beispiel 4.16:	Funktionen zum Ausfindigmachen des APIs	139
Beispiel 4.17:	Starten und Beenden der Kommunikation	140
Beispiel 4.18:	Funktionen zum Datenaustausch	140
Beispiel 4.19:	Aufruf von Funktionen beim Laden eines SCOs	141
Beispiel 4.20:	Funktionen zum Beenden der Kommunikation	141
Beispiel 4.21:	Funktionen zum Ändern des Status einer Lerneinheit	141
Beispiel 4.22:	Programmausschnitt aus der Datei <i>navigation fla</i>	142
Beispiel 4.23:	XML-Quelltext mit ‚mandatory‘ Elementen der Kategorie <i>general</i> für SCOs bzw. Content Aggregation	146

Literaturverzeichnis

- [ACollab]** *ACollab*. ATRC.
<http://www.atutor.ca/acollab/index.php> [Stand: 25.09.05]
- [ADL 2002]** *How to Convert Content into a Sharable Content Object (SCO)*. Academic ADL Co-Lab. 2002.
<http://www.academiccollab.org/resources/publications.html>
[Stand: 25.09.05]
- [ADL 2004a]** *SCORM 1.2 Conformance Test Suite*.
Advanced Distributed Learning (ADL). 2004.
<http://www.adlnet.org/scorm/history/12/scts.cfm>
[Stand:25.09.2005]
- [ADL 2004b]** *SCORM 1.2 Sample Run-Time Environment*.
Advanced Distributed Learning (ADL). 2004.
<http://www.adlnet.org/scorm/history/12/srte.cfm>
[Stand:25.09.2005]
- [Adobe 2005]** *Mit Adobe für's Leben lernen. Fachhochschule Dresden setzt Branchenstandards für die Ausbildung ein*. Adobe Systems GmbH. 2004.
http://www.adobe.de/education/pdfs/2005_HTWDresden.pdf
[Stand: 25.09.05]
- [AICC]** *Aviation Industry CBT Committee*.
<http://www.aicc.org> [Stand: 25.09. 05]
- [AICC 2001]** *DOCUMENT NO. CMI001. CMI Guidelines for Interoperability. Revision 3.5*. AICC. 2001.
<http://www.aicc.org/docs/tech/cmi001v3-5.pdf>
[Stand: 25.09.05]
- [ANOVA o.J.]** *Standardisierung durch wen? – Zusammenarbeit verschiedener Standardisierungsgremien*.
ANOVA Multimedia Studios GmbH. O.J.
<http://elearning.anova.de/de/standards/index.html>
[Stand: 25.09.05]
- [ARIADNE]** *ARIADNE Foundation for the European Knowledge Pool*.
<http://www.ariadne-eu.org> [Stand: 25.09. 05]

- [Baumgartner u.a. 2002]** Baumgartner, Peter, Hartmut Häfele und Kornelia Maier-Häfele: *E-Learning Praxishandbuch. Auswahl von Lernplattformen. Marktübersicht – Funktionen – Fachbegriffe*. StudienVerlag, Innsbruck 2002.
- [Bildungsportal]** *Bildungsportal Sachsen. Aus- und Weiterbildung der Hochschulen des Freistaates Sachsen*.
<http://www.bildungsportal-sachsen.de> [Stand: 25.09.05]
- [Bildungsportal 2003]** *Bildungsportal Sachsen. Verbundprojekt der Hochschulen Sachsens zur virtuellen Aus- und Weiterbildung. Projekte 2001 – 2003*. Bildungsportal Sachsen, 2003.
- [Bildungsportal 2004]** *Bildungsportal Sachsen. Verbundprojekt der Hochschulen Sachsens zur virtuellen Aus- und Weiterbildung. Projekte 2004 – 2005*. Bildungsportal Sachsen, 2004.
- [Bildungsportal 2005a]** *Das neue Kommunikations- und Kooperationswerkzeug des Bildungsportals Sachsen. Grundlagen – Funktionen – Integration*. Bildungsportal Sachsen, 2005.
- [Bildungsportal 2005b]** *Bildungsportal Sachsen – Pressemeldung zur CeBIT 2005*. Bildungsportal Sachsen, 2005.
<http://www.htwm.de/bps/project/project.html>
[Stand: 25.09.05]
- [Blatz 2002]** Blatz, Carlo u. die Powerflasher: *Flash MX professionell*. Galileo Design, Bonn 2002.
- [contentmanager]** *contentmanager.de. Das Content Management Portal*. F&P GmbH – FEIG & PARTNER.
<http://www.contentmanager.de> [Stand: 25.09.05]
- [eLearning and Law o.J.]** *Erlaubnisfreie Nutzung urheberrechtlich geschützter Leistungen*. eLearning and Law. Center of eLearning.
Kuratorium OFFIS e.V. Oldenburg o.J.
<http://ella.offis.de> [Stand: 25.09.05]
- [Epolos o.J.]** *Was standardisieren Standards?*
Epolos ELAN-Pilot. Oldenburg/Osnabrück. o.J.
<http://www.epolos.de/index.php?n=Main.Standard>
[Stand: 25.09.05].
- [Flash Detection]** *Flash Player Detection Kit*. Macromedia, Inc.
http://www.macromedia.com/software/flashplayer/download/detection_kit [Stand: 25.09.05]

- [Gries 2003]** Gries, Volker: *Nutzung von Standards bei der Entwicklung von e-Learning Content*. ANOVA Multimedia Studios GmbH. 2003. <http://elearning.anova.de/de/fachwissen/index.html> [Stand: 25.09.05]
- [Hahn/Kosler 2005]** Hahn, Daniel und Monique Kosler: *Funktionelle und software-technische Analyse sowie Dokumentation der Lernplattform OLAT Version 3.1.4*. Projektarbeit an der Technischen Universität Chemnitz, 2005. <http://www.olat.org/public/doc.html> [Stand: 25.09.05]
- [IEEE]** *IEEE*. <http://www.ieee.org> [Stand: 25.09.05]
- [IMS]** *Global Learning Consortium, Inc.* <http://www.imsproject.org> [Stand: 25.09.05]
- [Java Sun]** *Sun Developer Network*. Sun Microsystems, Inc. <http://java.sun.com> [Stand: 25.09.05]
- [Kammer 2003]** Kammer, Katja: *SCORM-konforme Lernobjekte mit Macromedia Flash MX - Konzeption und Implementierung eines Lernmoduls zum Thema "Heuristik" für die Lernumgebung SIMLA*, Diplomarbeit an der Hochschule für Technik und Wirtschaft Dresden, 2003.
- [Kleimann/Wannemacher 2004]** Kleinmann, Bernd und Klaus Wannemacher: *E-Learning an deutschen Hochschulen. Von der Projektentwicklung zur nachhaltigen Implementierung*. HIS Hochschul-Informationssystem GmbH, Hannover 2004.
- [Macromedia 2005]** *Flash Player Tech Note*. Macromedia, Inc. 2005. http://www.macromedia.com/go/tn_14159 [Stand: 25.09.05]
- [Minass 2002]** Minass, Erik: *Dimensionen des E-Learning - Neue Blickwinkel und Hintergründe für das Lernen mit dem Computer*. SmartBooks Publishing AG, Kilchberg 2002.
- [Niegemann u.a. 2004]** Niegemann, Helmut M., Silvia Hessel, Dirk Hochscheid-Mauel, Kristina Aslanski, Markus Deimann und Gunther Kreuzberger: *Kompendium E-Learning*. Springer Verlag, Berlin Heidelberg 2004.

- [opensourceCMS]** *opensourceCMS. Try Before You ... Install.*
The Open Source Collective, Inc.
<http://www.opensourcecms.com> [Stand: 25.09.05]
- [OLAT]** *Open Source LMS OLAT - eLearning made easy.*
<http://www.olat.org/public/index.html> [Stand: 25.09.05]
- [Reload 2004a]** *Reload Editor.* Reload.
<http://www.reload.ac.uk/editor.html> [Stand: 25.09.05]
- [Reload 2004b]** *Reload Player.* Reload.
<http://www.reload.ac.uk/scormplayer.html> [Stand: 25.09.05]
- [Saba 2001a]** *Saba Provider Guide. Saba3 Release4. Version 3.4.0.2.*
Saba Software Inc. 2001.
- [Saba 2001b]** *Saba Content Administrator Guide. Saba3 Release4. Version 3.4.0.2.* Saba Software Inc. 2001.
- [Saba 2002]** *Saba Standards Support.* Saba Software Inc. 2002.
<http://www.learningstore.com/sabapartners> [Stand: 25.09.05]
- [Saba 2003]** *Creating AICC-Compliant Content for Interoperability with Saba. Revision 2.3.* Saba Software Inc. 2003.
<http://www.learningstore.com/sabapartners> [Stand: 25.09.05]
- [Saba 2004]** *Using SCORM Content with Saba. Version 1.0.*
Saba Software Inc. 2004.
<http://www.learningstore.com/sabapartners> [Stand: 25.09.05]
- [Schreiber 1998]** Schreiber, Alfred: *CBT-Anwendungen professionell entwickeln.* Springer Verlag, Berlin Heidelberg 1998.
- [SCORM 2001a]** *Sharable Object Reference Model. Version 1.2. The Overview.*
Advanced Distributed Learning (ADL). 2001.
<http://www.adlnet.org/SCORM/History/12/documents.cfm>
[Stand: 25.09.05]
- [SCORM 2001b]** *Sharable Object Reference Model. Version 1.2. The SCORM Content Aggregation Model.*
Advanced Distributed Learning (ADL). 2001.
<http://www.adlnet.org/SCORM/History/12/documents.cfm>
[Stand: 25.09.05]

- [SCORM 2001c]** *Sharable Object Reference Model. Version 1.2. The SCORM Run Time Environment.*
Advanced Distributed Learning (ADL). 2001.
<http://www.adlnet.org/SCORM/History/12/documents.cfm>
[Stand: 25.09.05]
- [SCORM 2004a]** *SCORM 2004. 2nd Edition. Overview.*
Advanced Distributed Learning (ADL). 2004.
<http://www.adlnet.org/downloads/70.cfm> [Stand:25.09.05]
- [SCORM 2004b]** *Sharable Content Object Reference Model (SCORM). Run-Time Environment (RTE). Version 1.3.1.*
Advanced Distributed Learning (ADL). 2004.
<http://www.adlnet.org/downloads/70.cfm> [Stand: 25.09.05]
- [Strähle 2004]** Strähle, Dominik: *Evaluierung von Nutzungsmöglichkeiten eines Content Management Systems für die Mitarbeiterschulung im Schienenfahrzeugbereich.*
Diplomarbeit an der Hochschule für Technik und Wirtschaft Dresden (FH), 2004.
- [Trivantis]** *Trivantis.*
<http://www.trivantis.com> [Stand: 25.09.05]
- [MASIE 2003]** *Making Sense of Learning Specifications & Standards: A Decision Maker's Guide to their Adoption. 2nd Edition.*
The Masie Center. 2003.
http://www.masie.com/standards/s3_2nd_edition.pdf
[Stand: 25.09.05]
- [virtual-learning]** *virtual-learning.at. bm:bwk. bildung.at.*
<http://virtual-learning.qualifizierung.com> [Stand: 25.09.05]
- [Wendt 2003]** Wendt, Matthias: *Praxishandbuch CBT und WBT – Konzipieren, entwickeln, gestalten.*
Carl Hanser Verlag, München Wien 2003.
- [Wikipedia – Interoperabilität]** *Interoperabilität. Die freie Enzyklopädie. 2005.*
<http://de.wikipedia.org/wiki/Interoperabilität>
[Stand: 25.09.05]
- [Wikipedia - Standard]** *Standard. Die freie Enzyklopädie. 2005.*
<http://de.wikipedia.org/wiki/Standard> [Stand: 25.09.05]

Selbständigkeitserklärung

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

Dresden, den 07.10.2005

Kerstin Wagner