

Architektur eines webbasierten Lernsystems

Diplomarbeit im Fach Informatik

vorgelegt von
Florian Gnägi
<florian@gnaegi.ch>
geb. Zürich, Schweiz
Matrikelnummer 93-725-208

4. Dezember 2001

Angefertigt am
Institut für Informatik
der Universität Zürich
Prof. Dr. H. Schauer

INHALTSVERZEICHNIS

1. Einleitung	2
1.1 Zielsetzung und Aufbau der Arbeit	2
1.2 Begriffe	3
1.3 Vorgehensweise	3
1.3.1 Object Engineering Process	3
1.3.2 Abgrenzung zum Systementwurf	5
1.3.3 Unified Modeling Language	5
1.3.4 Technologiefrage: Java, J2EE, EJB	6
2. Ausgangslage: Erfahrungen mit OLAT 1	8
2.1 Hintergründe für die Entwicklung von OLAT 1	8
2.2 Einsatz von OLAT 1 am Institut für Informatik	9
2.3 Erweiterung des Einsatzgebietes am OLAT-Zentrum	9
2.4 Limitationen von OLAT 1	10
2.4.1 Skalierbarkeit	10
2.4.2 Erweiterbarkeit	11
3. Systemidee für OLAT 2	12
3.1 Allgemeine Anforderungen	12
3.2 Kompatibilität zu OLAT 1	12
3.3 Universitätsweiter Einsatz	12
3.4 Anbindung an universitäre Informationssysteme	13
3.5 Berücksichtigung von Internationalen Standards	13
3.6 Interoperabilität	13
4. Internationale Standards	15
4.1 IEEE Learning Technology Standards Committee	15
4.1.1 IEEE 1484.1: Architecture and Reference Model	17
4.2 IMS	20
4.2.1 IMS Learning Resource Meta-Data Specification	21
4.2.2 IMS Content Packaging Specification	22
4.2.3 IMS Question & Test Interoperability Specification	24
4.2.4 IMS Learner Information Package Specification	26
4.2.5 IMS Enterprise Specification	27
4.3 SCORM	28
4.3.1 SCORM Content Aggregation Model	29
4.3.2 SCORM Run-Time Environment	32

5. Grobanalyse	35
5.1 Anforderungsbeitragende	35
5.1.1 Reale Anforderungsbeitragende	35
5.1.2 Abstrakte Anforderungsbeitragende	37
5.2 Geschäftsprozesse identifizieren	37
5.2.1 Contentverwaltung	38
5.2.2 Kursverwaltung	40
5.2.3 Klassenverwaltung	41
5.2.4 Evaluationsverwaltung	43
5.2.5 Gruppenverwaltung	44
5.2.6 Betreuung und Bewertung	45
5.2.7 Kurs belegen	46
5.2.8 Lernen	46
5.2.9 Kommunizieren	48
5.2.10 Dateien verwalten	49
5.2.11 Personenverwaltung	51
5.2.12 Systemverwaltung	53
6. Anwendungsfälle	54
6.1 Contentverwaltung	54
6.2 Kursverwaltung	63
6.3 Klassenverwaltung	66
6.4 Evaluationsverwaltung	72
6.5 Gruppenverwaltung	75
6.6 Betreuung und Bewertung	81
6.7 Kurs belegen	84
6.8 Lernen	85
6.9 Kommunizieren	88
6.10 Dateien verwalten	92
6.11 Personenverwaltung	95
6.12 Systemverwaltung	100
7. Anforderungsspezifikation	103
7.1 Funktionale Anforderungen	103
7.2 Anforderungen an die Benutzbarkeit	105
7.3 Anforderungen an die Zuverlässigkeit, Effizienz und Sicherheit	106
7.4 Supportanforderungen	108
8. Geschäftsklassen	109
9. Anwendungsarchitektur	112
9.1 Anwendungsschichten	112
9.2 Komponenten	113
9.3 Architektur	116
9.3.1 Client Tier	116
9.3.1.1 SAP Campus Client	116
9.3.1.2 Webbrowser Client	116
9.3.2 Presentation Tier	117
9.3.2.1 Webservice Controller	117
9.3.2.2 Front Controller	117

9.3.3	Business Tier	118
9.3.3.1	Control-Komponenten	118
9.3.3.2	Entity-Komponenten	118
9.3.4	Integration und Resource Tier	119
10.	Zusammenfassung	121
	Glossar	123
	Abkürzungen	128
	Bibliographie	130
	Index	136

ABBILDUNGSVERZEICHNIS

4.1	Komponenten der Learning Technology Systems Architecture (Quelle: IEEE 1484.1/D8 [T ⁺ 01])	18
4.2	IMS Content Framework (Quelle: IMS Content Packaging Specification Best Practice and Implementation Guide [AM01a])	23
4.3	ASI Datenstruktur der IMS QTI (Quelle: IMS Question & Test Interoperability: An Overview [SSBL01a])	24
4.4	SCORM Overview (Quelle: The SCORM Overview 1.2 [D ⁺ 01b])	29
4.5	SCORM Content Aggregation (Quelle: SCORM Content Aggregation Model 1.2 [D ⁺ 01a])	30
4.6	Das SCORM Run-Time Environment (Quelle: The SCORM Run-Time Environment 1.2 [D ⁺ 01c])	32
5.1	Anforderungsbeitragende des OLAT 2 Systems	36
5.2	Die wichtigsten Geschäftsprozesse im Überblick	38
5.3	Übersicht über den Geschäftsprozess Contentverwaltung	39
5.4	Übersicht über den Geschäftsprozess Kursverwaltung	41
5.5	Übersicht über den Geschäftsprozess Klassenverwaltung	42
5.6	Übersicht über den Geschäftsprozess Evaluationsverwaltung	43
5.7	Übersicht über den Geschäftsprozess Gruppenverwaltung	44
5.8	Übersicht über den Geschäftsprozess Betreuung und Bewertung	45
5.9	Übersicht über den Geschäftsprozess Kurs belegen	46
5.10	Übersicht über den Geschäftsprozess Lernen	47
5.11	Übersicht über den Geschäftsprozess Kommunizieren	49
5.12	Übersicht über den Geschäftsprozess Dateien verwalten	50
5.13	Übersicht über den Geschäftsprozess Personenverwaltung	51
5.14	Übersicht über den Geschäftsprozess Systemverwaltung	52
8.1	Die wichtigsten Geschäftsklassen im Überblick	110
9.1	Die J2EE Anwendungsschichten (Quelle: Core J2EE Patterns [ACM01])	113
9.2	Entitätskomponenten von OLAT 2	114
9.3	Anwendungsarchitektur von OLAT 2	120

TABELLENVERZEICHNIS

4.1	Beispiele der IMS Learning Resource Spezifikation und des IEEE LOM Standards	21
6.1	Systemanwendungsfall <i>Content erstellen</i>	54
6.2	Systemanwendungsfall <i>Content löschen</i>	55
6.3	Systemanwendungsfall <i>Content importieren</i>	55
6.4	Systemanwendungsfall <i>Content Metadata editieren</i>	55
6.5	Systemanwendungsfall <i>Content exportieren</i>	56
6.6	Systemanwendungsfall <i>Contentstruktur editieren</i>	56
6.7	Systemanwendungsfall <i>Blockelemente erstellen</i>	56
6.8	Systemanwendungsfall <i>Block Metadaten editieren</i>	56
6.9	Systemanwendungsfall <i>Element in Block hinzufügen</i>	57
6.10	Systemanwendungsfall <i>Element von Block entfernen</i>	57
6.11	Systemanwendungsfall <i>Blockstruktur editieren</i>	57
6.12	Systemanwendungsfall <i>Block löschen</i>	58
6.13	Systemanwendungsfall <i>Block vollständig löschen</i>	58
6.14	Systemanwendungsfall <i>SCO erstellen</i>	58
6.15	Systemanwendungsfall <i>SCO Metadaten editieren</i>	59
6.16	Systemanwendungsfall <i>Asset erstellen</i>	59
6.17	Systemanwendungsfall <i>Asset Metadaten editieren</i>	59
6.18	Systemanwendungsfall <i>Asset löschen</i>	59
6.19	Systemanwendungsfall <i>Asset editieren</i>	60
6.20	Systemanwendungsfall <i>Assessment erstellen</i>	60
6.21	Systemanwendungsfall <i>Assessment löschen</i>	60
6.22	Systemanwendungsfall <i>Assessment importieren</i>	60
6.23	Systemanwendungsfall <i>Assessment exportieren</i>	61
6.24	Systemanwendungsfall <i>Assessment Metadaten editieren</i>	61
6.25	Systemanwendungsfall <i>Item erstellen</i>	61
6.26	Systemanwendungsfall <i>Item löschen</i>	61
6.27	Systemanwendungsfall <i>Item editieren</i>	62
6.28	Systemanwendungsfall <i>Section erstellen</i>	62
6.29	Systemanwendungsfall <i>Section löschen</i>	62
6.30	Systemanwendungsfall <i>Section komplett löschen</i>	62
6.31	Systemanwendungsfall <i>Section editieren</i>	63
6.32	Systemanwendungsfall <i>Kurs erstellen</i>	63
6.33	Systemanwendungsfall <i>Kurs löschen</i>	63
6.34	Systemanwendungsfall <i>Kurs importieren</i>	64
6.35	Systemanwendungsfall <i>Kurs Metadaten editieren</i>	64
6.36	Systemanwendungsfall <i>Kursstruktur editieren</i>	64
6.37	Systemanwendungsfall <i>Kursnachricht zustellen</i>	65

6.38	Systemanwendungsfall <i>Kursstruktur exportieren</i>	65
6.39	Systemanwendungsfall <i>Kursdaten exportieren</i>	65
6.40	Systemanwendungsfall <i>Kursperformance überwachen</i>	66
6.41	Systemanwendungsfall <i>Lernerverhalten beobachten</i>	66
6.42	Systemanwendungsfall <i>Klasse erstellen</i>	66
6.43	Systemanwendungsfall <i>Klasse löschen</i>	67
6.44	Systemanwendungsfall <i>Klassenbelegung exportieren</i>	67
6.45	Systemanwendungsfall <i>Einschreibung überwachen</i>]	67
6.46	Systemanwendungsfall <i>Platzangebot verändern</i>	68
6.47	Systemanwendungsfall <i>Klassen Metadaten editieren</i>	68
6.48	Systemanwendungsfall <i>Raum zuweisen</i>	68
6.49	Systemanwendungsfall <i>Raumzuweisung löschen</i>	69
6.50	Systemanwendungsfall <i>Klassennachricht zustellen</i>	69
6.51	Systemanwendungsfall <i>Lerner Klasse zuweisen</i>	69
6.52	Systemanwendungsfall <i>Coach Klasse zuweisen</i>	70
6.53	Systemanwendungsfall <i>Stellvertretercoach Klasse zuweisen</i>	70
6.54	Systemanwendungsfall <i>Stellvertretercoach entfernen</i>	70
6.55	Systemanwendungsfall <i>Coach von Klasse entfernen</i>	71
6.56	Systemanwendungsfall <i>Lerner von Klasse entfernen</i>	71
6.57	Systemanwendungsfall <i>Lerner umteilen</i>	71
6.58	Systemanwendungsfall <i>Evaluation erstellen</i>	72
6.59	Systemanwendungsfall <i>Evaluation löschen</i>	72
6.60	Systemanwendungsfall <i>Evaluation importieren</i>	72
6.61	Systemanwendungsfall <i>Evaluation exportieren</i>	73
6.62	Systemanwendungsfall <i>Evaluationsresultate exportieren</i>	73
6.63	Systemanwendungsfall <i>Evaluation überwachen</i>	73
6.64	Systemanwendungsfall <i>Evaluation Metadaten editieren</i>	74
6.65	Systemanwendungsfall <i>Evaluationsstruktur editieren</i>	74
6.66	Systemanwendungsfall <i>Evaluationsfrage erstellen</i>	74
6.67	Systemanwendungsfall <i>Evaluationsfrage löschen</i>	75
6.68	Systemanwendungsfall <i>Evaluationsfrage editieren</i>	75
6.69	Systemanwendungsfall <i>Gruppentyp erstellen</i>	75
6.70	Systemanwendungsfall <i>Gruppentyp löschen</i>	76
6.71	Systemanwendungsfall <i>Grupentyp Metadaten editieren</i>	76
6.72	Systemanwendungsfall <i>Kursgruppen automatisch erstellen</i>	76
6.73	Systemanwendungsfall <i>Klassengruppen automatisch erstellen</i>	77
6.74	Systemanwendungsfall <i>Kursgruppe manuell erstellen</i>	77
6.75	Systemanwendungsfall <i>Klassengruppe manuell erstellen</i>	77
6.76	Systemanwendungsfall <i>Gruppe Metadaten editieren</i>	78
6.77	Systemanwendungsfall <i>Gruppe löschen</i>	78
6.78	Systemanwendungsfall <i>Lerner Gruppe zuweisen</i>	78
6.79	Systemanwendungsfall <i>Coach Gruppe zuweisen</i>	79
6.80	Systemanwendungsfall <i>Lerner von Gruppe entfernen</i>	79
6.81	Systemanwendungsfall <i>Coach von Gruppe entfernen</i>	79
6.82	Systemanwendungsfall <i>Gruppencontent zuweisen</i>	80
6.83	Systemanwendungsfall <i>Gruppennachricht zustellen</i>	80
6.84	Systemanwendungsfall <i>Bewertungsinformationen lesen</i>	81
6.85	Systemanwendungsfall <i>Lerner überwachen</i>	81
6.86	Systemanwendungsfall <i>Bewertung erstellen</i>	81
6.87	Systemanwendungsfall <i>Bewertung editieren</i>	82

6.88	Systemanwendungsfall <i>Gruppenbewertung erstellen</i>	82
6.89	Systemanwendungsfall <i>Gruppenbewertung editieren</i>	83
6.90	Systemanwendungsfall <i>Gruppe überwachen</i>	83
6.91	Systemanwendungsfall <i>Anfrage beantworten</i>	83
6.92	Systemanwendungsfall <i>Kurs belegen</i>	84
6.93	Systemanwendungsfall <i>Belegung löschen</i>	84
6.94	Systemanwendungsfall <i>Status beobachten</i>	85
6.95	Systemanwendungsfall <i>Coach kontaktieren</i>	85
6.96	Systemanwendungsfall <i>Suchen</i>	85
6.97	Systemanwendungsfall <i>Lernpfad betrachten</i>	86
6.98	Systemanwendungsfall <i>Lernpfad weiterführen</i>	86
6.99	Systemanwendungsfall <i>Content lesen</i>	86
6.100	Systemanwendungsfall <i>Test lösen</i>	87
6.101	Systemanwendungsfall <i>Lösung abgeben</i>	87
6.102	Systemanwendungsfall <i>Nachricht lesen</i>	88
6.103	Systemanwendungsfall <i>Nachricht erstellen</i>	88
6.104	Systemanwendungsfall <i>Adressbucheintrag einfügen</i>	88
6.105	Systemanwendungsfall <i>Adressbucheintrag löschen</i>	89
6.106	Systemanwendungsfall <i>Blackboard erstellen</i>	89
6.107	Systemanwendungsfall <i>Blackboard Metadaten editieren</i>	89
6.108	Systemanwendungsfall <i>Blackboard löschen</i>	90
6.109	Systemanwendungsfall <i>Blackboard exportieren</i>	90
6.110	Systemanwendungsfall <i>Blackboardnachricht erstellen</i>	90
6.111	Systemanwendungsfall <i>Blackboardnachricht lesen</i>	91
6.112	Systemanwendungsfall <i>Blackboardnachricht bewerten</i>	91
6.113	Systemanwendungsfall <i>Blackboardnachricht löschen</i>	91
6.114	Systemanwendungsfall <i>Externe Datei erstellen</i>	92
6.115	Systemanwendungsfall <i>Externe Datei editieren</i>	92
6.116	Systemanwendungsfall <i>Interne Datei erstellen</i>	92
6.117	Systemanwendungsfall <i>Interne Datei editieren</i>	93
6.118	Systemanwendungsfall <i>Datei Metadaten editieren</i>	93
6.119	Systemanwendungsfall <i>Datei löschen</i>	93
6.120	Systemanwendungsfall <i>Datei kopieren</i>	94
6.121	Systemanwendungsfall <i>Datei verschieben</i>	94
6.122	Systemanwendungsfall <i>Datei referenzieren</i>	94
6.123	Systemanwendungsfall <i>Person selbständig registrieren</i>	95
6.124	Systemanwendungsfall <i>Person registrieren</i>	95
6.125	Systemanwendungsfall <i>Person automatisch registrieren</i>	96
6.126	Systemanwendungsfall <i>Person automatisch löschen</i>	96
6.127	Systemanwendungsfall <i>Person löschen</i>	97
6.128	Systemanwendungsfall <i>Persönliche Daten editieren</i>	97
6.129	Systemanwendungsfall <i>Passwort setzen</i>	97
6.130	Systemanwendungsfall <i>Fremdes Passwort setzen</i>	98
6.131	Systemanwendungsfall <i>Rolle imitieren</i>	98
6.132	Systemanwendungsfall <i>Rollenimitierung akzeptieren</i>	99
6.133	Systemanwendungsfall <i>Rolle zuweisen</i>	99
6.134	Systemanwendungsfall <i>Rolle verändern</i>	100
6.135	Systemanwendungsfall <i>System starten</i>	100
6.136	Systemanwendungsfall <i>System stoppen</i>	101
6.137	Systemanwendungsfall <i>Systemdaten importieren</i>	101

6.138	Systemanwendungsfall <i>System Metadaten editieren</i>	101
6.139	Systemanwendungsfall <i>Systemdaten exportieren</i>	102
6.140	Systemanwendungsfall <i>System updaten</i>	102
6.141	Systemanwendungsfall <i>System überwachen</i>	102
7.1	Funktionale Anforderungen	105
7.2	Anforderungen an die Benutzbarkeit	106
7.3	Anforderungen an die Zuverlässigkeit, Effizienz und Sicherheit . . .	108
7.4	Supportanforderungen	108

1. EINLEITUNG

1.1 Zielsetzung und Aufbau der Arbeit

In der vorliegenden Arbeit wird die Architektur eines webbasierten Lernsystems entwickelt. Die folgenden Aspekte werden dabei berücksichtigt:

- Einschreibung und Administration der Teilnehmer
- Gruppenbildung und Zuordnung der Gruppen zu Tutoren
- Kommunikation und kollaboratives Lernen innerhalb und ausserhalb einzelner Gruppen
- Betreuung der Gruppen durch Tutoren (distance coaching)
- Einbindung von interaktiven multimedialen Lernmaterialien (content management)
- Automatisiertes Testen (assessment)
- Einbindung einer Datenbank zur Speicherung aller relevanten Informationen

In der Architektur werden die wichtigsten Komponenten des Lernsystems identifiziert und die Schnittstellen aufgezeigt. Die Komponenten weisen dabei ein hohes Abstraktionsniveau auf, um die Komplexität und Grösse eines solchen Lernsystems als Ganzes erfassen zu können. Die Architektur soll detailliert genug sein, um als Grundlage für einen Entwurf dienen zu können.

Die Arbeit beginnt mit einer Analyse der Ausgangslage und geht kurz auf das bestehende Lernsystem OLAT 1 ein. Dabei wird gezeigt, warum die bei OLAT 1 verwendete Architektur den Ansprüchen nicht mehr genügt und eine neue Architektur entwickelt werden muss. Es folgen dann die Identifikation und Beschreibung der wichtigsten Prozesse und den damit verbundenen Anforderungen an das Lernsystem. Dies wird der gewichtigste Teil der Arbeit sein. Ein spezielles Kapitel widmet sich den internationalen Standards im Bereich E-Learning, sie tragen einen entscheidenden Anteil zu den Anforderungen an das Lernsystem bei. Aus den Anforderungen werden anschliessend die wichtigsten Geschäftsklassen herausgearbeitet und eine Architektur abgeleitet. Die Arbeit schliesst mit der Zusammenfassung.

1.2 Begriffe

Architektur:

In dieser Arbeit wird der Begriff *Architektur* gemäss der IEEE 610.12 Definition verstanden. Er beschreibt die Organisationsstruktur eines Systems oder einer Komponente. Die Arbeit selbst ist das Lösungskonzept.[Gli00]

Web:

Der Begriff *Web* wird als Abkürzung für *World Wide Web* verstanden und steht für ein internetbasiertes und verteiltes Hypertext-Informationssystem. Es ist diejenige Teilmenge des Internets, welche über einen Webbrowser zugänglich ist.¹

Webbasierten:

Unter einem *webbasierten* System versteht man demzufolge ein Client-Server System, welches als Client lediglich einen HTML-Webbrowser verwendet.

Lernsystem:

Unter einem *Lernsystem* ist ein technisches System zu verstehen, welches mindestens eine der IEEE LTSA Systemkomponenten (learning preferences, behavior, performance and preference information, assessment information, query, catalog information, locator, learning content, multimedia, interaction context) zum Inhalt hat.[T⁺01]

1.3 Vorgehensweise

1.3.1 Object Engineering Process

Die Entwicklung der Architektur wird geleitet von dem von der oose.de GmbH² vorgeschlagenen Object Engineering Process³. Der OEP ist ein praxisorientierter und praxiserprobter Vorgehensleitfaden für die objektorientierte Softwareentwicklung. Er zeichnet sich dadurch aus, dass er verhältnismässig einfach an unternehmensspezifische Bedürfnisse angepasst werden kann. Er gliedert sich in die fünf Phasen *Vorbereitung, Entwurf, Konstruktion, Einführung und Betrieb*. Auf die ausführlichen Beschreibungen der Phasen, Meilensteinen, Iterationen, Prozesse, Aktivitäten, Akteure und Ergebnisse des Prozesses wird hier nicht näher eingegangen, ebensowenig auf das komplexe Metamodell des Entwicklungsprozesses. Da in dieser Arbeit lediglich die Systemarchitektur beschrieben werden soll, beschränkt sich die verwendete Methodik auf den Teil des OEP, welcher in dem Buch *Objektorientierte Softwareentwicklung* von Bernd Oestereich beschrieben wird.[Oes01]

Oestereich nennt die folgenden Arbeitsschritte:

- *Systemidee und Zielsetzung:*

Die Systemidee ist in Kapitel 3 auf Seite 12 beschrieben. Sie zeigt in groben Zügen die Funktionalität des Systems auf.

¹ <http://www.nue.org/foldoc/foldoc.cgi?World-Wide+Web>

² <http://www.oose.de>

³ <http://www.oose.de/oep>

- *Anforderungsbeitragende identifizieren:*
Die verschiedenen Stakeholder und Ansprechpartner werden beschrieben. Neben Systembetroffenen und Anforderungsbeitragenden sind auch Standards oder andere zu beachtende Regelungen als Stakeholder zu betrachten. Dies wird in Kapitel 5.1 *Anforderungsbeitragende* ab Seite 35 dargelegt.
- *Geschäftsprozesse identifizieren:*
Das System wird mit einem Anwendungsfalldiagramm näher beschrieben. Die identifizierten Geschäftsprozesse werden in einem weiteren Schritt zusätzlich möglichst abstrakt mit einem Aktivitätsdiagramm aufgezeichnet. Die beteiligten Akteure werden festgelegt. Dieser Schritt wird in Kapitel 8 *Geschäftsklassen* ab der Seite 109 bearbeitet.
- *Interessen der Anforderungsbeitragenden identifizieren:*
Die Ziele und Interessen der Anforderungsbeitragenden werden notiert um Konflikte, Probleme und Schwachstellen zu entdecken. Auf diesen Schritt wird in der vorliegenden Arbeit nicht weiter eingegangen.
- *Geschäftsanwendungsfälle identifizieren:*
Alle konsistenten, zielgerichteten und zeitlich ununterbrochenen Aktivitäten werden zu Geschäftsanwendungsfällen zusammengefasst. Diese werden von einem Akteur ausgelöst und führen zu einem Ergebnis mit fachlichem Wert.
- *Anwendungsfälle essentiell beschreiben:*
Die identifizierten Geschäftsanwendungsfälle werden auf ihre Essenz hin untersucht. Die Ereignisse, eingehenden Informationen, Vorbedingungen und Nachbedingungen werden notiert.
- *Systemanwendungsfälle identifizieren:*
Die zuvor essentiell beschriebenen Anwendungsfälle werden in diesem weiteren Schritt konkretisiert und in allen Ausprägungen innerhalb ihres Umfeldes betrachtet. Dabei soll auch das Risiko, die Wichtigkeit, der geschätzte Aufwand und die Dringlichkeit des jeweiligen Systemanwendungsfalles bestimmt werden. Die Systemanwendungsfälle sind im Kapitel 6 *Anwendungsfälle* ab Seite 54 zu finden. Auf die Aufteilung nach Geschäfts- und essentiellen Anwendungsfällen wird hier verzichtet.
- *Materialsammlung und -studie:*
Mit der Materialsammlung sollen vorhandene Materialien, wie zum Beispiel Formulare, gesammelt werden. Dies spielt in dem vorliegenden Fall keine Rolle, da es keine solche Formulare gibt.
- *Anforderungen beschreiben:*
Im Kapitel 7 *Anforderungsspezifikation* ab Seite 103 werden die Anforderungen an das System beschrieben, welche nicht schon in den Anwendungsfällen zum Ausdruck kommen. Es wird unterschieden zwischen funktionalen Anforderungen, Anforderungen zur Benutzbarkeit, Effizienz, Zuverlässigkeit, Änderbarkeit, Erweiterbarkeit, Wartbarkeit und Adminstrierbarkeit des Systems.
- *Geschäftsklassen identifizieren:*
Nun werden die wichtigsten fachlichen Gegenstände identifiziert und als Klassen mit ihren strukturellen Zusammenhängen in Klassendiagrammen modelliert. Dies ist in Kapitel 8 *Geschäftsklassen* ab Seite 109 zu finden.

- *Fachliches Glossaranlegen:*
Das Glossar beinhaltet die fachlichen Begriffe, die Klassen und Assoziationsrollen wie alle übrigen Gegenstände, Konzepte und Prozesswörter. Sie werden kontinuierlich gesammelt und im Anhang 10 *Glossar* ab Seite 123 aufgelistet.
- *Anwendungsfall-Ablaufmodell entwickeln:*
Aus den Anwendungsfallbeschreibungen werden die elementaren Aktivitäten extrahiert und mit Hilfe von Aktivitätsdiagrammen die Abläufe modelliert. Es werden alle Ausnahmen und Verzweigungen dargestellt. Es würde den Rahmen dieser Arbeit sprengen, die Anwendungsfälle als Aktivitätsdiagramme darzustellen, daher wird dieser Schritt ausgelassen.
- *Systemschnittstelle beschreiben:*
In einem letzten Schritt wird zu jedem Anwendungsfall die Schnittstelle mit den ein- und ausgehenden Daten, Objekten und Ereignissen genau definiert. Dieser Schritt wird ebenfalls ausgelassen werden.
- *Exploratives Schnittstellen-Prototyping:*
Als Schnittstellenprototyp kann das ganze OLAT 1 System angesehen werden. Auf zusätzliche Schnittstellenprototypen wird daher verzichtet.
- *Anwendungsarchitektur definieren:*
Aus den vorliegenden Informationen kann nun eine Anwendungsarchitektur mit ihren Verantwortlichkeiten, Aufgaben und Besonderheiten entwickelt werden. Die Architektur ist in Kapitel 9 *Anwendungsarchitektur* ab Seite 112 zu finden.

1.3.2 Abgrenzung zum Systementwurf

Oestereich fährt nach der Definition der Anwendungsarchitektur weiter mit der Identifikation der fachlichen Komponenten. Für jede Komponente folgt dann die Entwicklung der eigentlichen Klassenmodelle mit ihren Zustandsmodellen, Abhängigkeiten, Schnittstellen und den Zusammenarbeitsmodellen. Diese Schritte gehören bereits in den Systementwurf und würden den Rahmen dieser Arbeit bei weitem sprengen. Sie werden hier daher nicht behandelt. Ebenso wenig wird auf die anschließende Implementation eingegangen.

1.3.3 Unified Modeling Language

UML⁴ (Unified Modeling Language) ist eine Sprache und Notation zur Spezifikation, Konstruktion, Visualisierung und Dokumentation von Modellen für Softwaresysteme und stellt ein Industriestandard dar. Auf die einzelnen Elemente⁵ von UML werde ich hier nicht weiter eingehen. In dieser Arbeit werden alle Spezifikationen gemäss UML Version 1.4 dargestellt.[Oes01]

⁴ <http://www.uml.org>

⁵ http://www.job-agent.de/download/OOAD-UML_5A_Glossar_und_Notation.pdf

1.3.4 Technologiefrage: Java, J2EE, EJB

Natürlich soll bei der Spezifikation der Anforderungen und der Entwicklung der Systemarchitektur die Implementationsfrage grundsätzlich offen gelassen werden. Trotzdem ist es hilfreich, sich schon zu Beginn darüber Gedanken zu machen. Die Wahl einer Plattform und Programmiersprache kann auch die Architektur eines Systems beeinflussen oder in eine bestimmte Richtung treiben. OLAT 2 soll eine zukunftsgerichtete Lernplattform werden, welche auf den modernsten Technologien aufbaut.

Die von Sun Microsystems⁶ entwickelte Programmiersprache Java⁷ hat sich im Ausbildungs- wie im kommerziellen Bereich als Standard etabliert. Gegenüber herkömmlichen Programmiersprachen wird ein in Java geschriebenes Programm nicht direkt auf dem Prozessor ausgeführt, sondern innerhalb einer virtuellen Maschine. Diese wandelt den sogenannten Bytecode durch einen Interpreter in nativen Code um, welcher dann von der Hardware verarbeitet werden kann. Durch die öffentliche Spezifikation der Sprache sowie der virtuellen Maschine ist eine in Java erstellte Applikation unabhängig von der gewählten Hardwareplattform und dem Hersteller des Betriebssystems. Es gibt diverse weitere Java-Implementationen neben der Referenzimplementation von Sun, auch solche, welche unter einer Open Source Lizenz stehen.[G⁺00]

Die J2EE (Java 2 Enterprise Edition) definiert eine ganze Systeminfrastruktur für grose und verteilte Applikationen. In sogenannten Containern werden komponentenorientierte Java-Applikationen ausgeführt, welche nach diesen Containerspezifikationen gebaut wurden. J2EE definiert zwei Ebenen: Die Schnittstelle zum Applikationsentwickler und die Schnittstelle zum Serviceprovider. Die Idee der J2EE ist es, den Systementwickler vom Runtimeprovider zu trennen und diesen wiederum von den verschiedenen Serviceprovidern abzukapseln. Ein Serviceprovider ist beispielsweise ein Datenbankhersteller, welcher seine Datenbank durch das JDBC⁸-Interface dem Runtimeprovider anbietet. Der Runtimeprovider ist zum Beispiel ein kommerzieller Hersteller eines Applikationsserver wie WebLogic oder eine Open Source Alternative wie JBoss, die vom Rechenzentrum betrieben wird. Als Applikationsentwickler programmiert man lediglich gegen das durch J2EE gegebene Interface und muss sich im Idealfall nicht um komplizierte Dinge wie Transaktionsmanagement, Messaging und Directory Services, Clustering, Persistenz und ähnliches kümmern. Dies wird alles vom Applikationsserver transparent zur Verfügung gestellt. Auch für das sogenannte Deployment stellt die J2EE Architektur geeignete Instrumente zur Verfügung. Die J2EE wiederum stellt Plattform- und Herstellerunabhängigkeit sicher und es zeichnet sich auch hier ab, dass J2EE zu einem weitverbreiteten Industriestandard wird.[S⁺00]

Der wichtigste Teil in der J2EE Architektur wird durch die EJB (Enterprise Java Beans) definiert. Es ist das Komponentenmodell, welches J2EE zugrunde liegt. Es wird unterschieden zwischen Entitybeans, welche die Datenobjekte darstellen, und Sessionbeans, welche die Datenobjekte manipulieren und somit die eigentlichen Prozesse definieren. Der dritte Typ, die sogenannten Message-driven-beans, wird erst mit der Spezifikation 2.0 möglich. Mit diesen Beans können systeminterne Prozesse abgebildet werden. EJB sind nicht zu verwechseln mit JavaBeans, der Komponentenarchitektur welche in das

⁶ <http://www.sun.com>

⁷ <http://java.sun.com>

⁸ Java Database Connectivity. Dies ist eine universelle Datenbankschnittstelle für Javaprogramme, ähnlich zu ODBC von Microsoft.

allgemeine Java API integriert ist. JavaBeans eignen sich für intra-prozessorientierte Komponenten, während EJB für inter-prozessorientierte, serverseitige Komponenten entworfen wurden. Ausser dem Namen haben sie nichts gemeinsam.[MH00]

J2EE, EJB und Java bilden eine ausgezeichnete Grundlage, um ein System wie OLAT 2 zu implementieren. Dies hat auch Implikationen auf die Systemarchitektur, die in diesem Fall komponentenorientiert nach den mit EJB zur Verfügung stehenden Komponententypen entworfen werden soll. Es ist ebenfalls bereits gegeben, dass eine Multi-Tier Architektur mit den Schichten Datenhaltung, Businesslogik und Applikationslogik zur Anwendung kommt. Die Applikationslogik kann als eine herkömmliche Java Applikation oder als eine Kombination aus Servlets/JSP und Webbrowser entwickelt werden. Eine auf J2EE basierende Architektur garantiert Skalierbarkeit und Erweiterbarkeit des ganzen Systems. Bei der Entwicklung der Systemarchitektur von OLAT 2 soll daher darauf geachtet werden, dass eine Implementation gemäss J2EE möglich ist.

2. AUSGANGSLAGE: ERFAHRUNGEN MIT OLAT 1

2.1 Hintergründe für die Entwicklung von OLAT 1

Die Lernplattform OLAT¹ (Online Learning And Testing) wurde 1999 von den drei Studierenden Sabina Jeger, Franziska Schneider und Florian Gnägi in der Software Engineering Gruppe² am IfI³ (Institut für Informatik) entwickelt, um eine Lehrveranstaltung des Informatikgrundstudiums mit Hilfe der neuen ICT-Möglichkeiten (Information and Communication Technology) optimal zu ergänzen. Es handelte sich dabei um eine Übungsveranstaltung, bei welcher die Studierenden in kleinen Klassen praxisorientierte Aufgaben am Computer lösen mussten. Neben einem neuen Übungskonzept ist dabei die webbasierte Lernplattform OLAT entstanden, welche die Studierenden, die betreuenden Personen und die Kursentwickler gleichzeitig miteinbezieht und in den jeweiligen Prozessen unterstützt.[Gnä00]

OLAT bietet ein reichhaltiges Spektrum an Funktionen und Möglichkeiten an. Nennenswert sind die folgenden Elemente:

- Lehrveranstaltungsverwaltung
- Klassenverwaltung
- Gruppenverwaltung
- Skriptenverwaltung
- Testverwaltung
- Evaluationsverwaltung
- Einschreibeprogramm
- Lernumgebung mit Lektionen und Aufgaben
- Testprogramm
- Evaluationsprogramm
- Chat
- Emailintegration

¹ <http://www.olat.unizh.ch>

² <http://www.ifi.unizh.ch/se>

³ <http://www.ifi.unizh.ch>

- Groupware

OLAT integriert all diese Elemente unter einer einheitlichen, webbasierten Oberfläche. Zu betonen ist, dass die Elemente Chat, Email und Groupware nicht eigene Entwicklungen des OLAT-Teams sind, sondern externe, lose an OLAT gekoppelte, eigenständige Applikationen.

Durch ein ausgeklügeltes Rechtesystem ist OLAT für beliebige Lehrveranstaltungen einsetzbar. Diese behindern sich gegenseitig nicht. Für die Arbeit mit OLAT benötigen die Studierenden und Dozierenden lediglich einen Webbrowser, auch die Registrierung für einen Kurs erfolgt konsequent über das Internet. Dabei ist die Lernplattform eng an das Emailsysteem UniAccess⁴ gebunden, welches auch für die Überprüfung der Identität eines Studierenden benutzt wird.

2.2 Einsatz von OLAT 1 am Institut für Informatik

Nach dem ersten erfolgreichen Pilotversuch mit ca. 750 Studierenden weitete sich das Einsatzgebiet von OLAT rasch aus. Zusätzliche Lehrveranstaltungen der Software Engineering Gruppe des Instituts für Informatik wurden mit Hilfe von OLAT unterstützt und andere Gruppen innerhalb des Institutes interessierten sich für die Plattform. In einem zweiten Entwicklungsschub wurde die Administrationsumgebung weiterentwickelt, so dass OLAT auch ohne Kenntnisse der Datenbankstruktur einzig über den Webbrowser bedient werden konnte. Diese Arbeiten mussten jedoch einige Wünsche unbeantwortet lassen, denn das Entwicklungsteam widmete sich ab Mitte des Jahres 2000 voll und ganz seinen Diplomprüfungen. Im Rahmen einer Diplomarbeit wurde der noch recht einfach gestaltete Testbereich von OLAT komplett überarbeitet, so dass nun beliebig komplexe Fragetypen behandelt werden können[Jeg00]. Ebenfalls als Diplomarbeit wurden Guidelines für webbasierte Courseware im universitären Bereich entwickelt.[Sch00]

Im Herbst 2000 gewann das OLAT-Team zusammen mit einem Projekt aus Deutschland den mit 100'000 Euro dotierten mediendidaktischen Hochschulpreis MeDiDa-Prix⁵ für das prozessorientierte Vorgehen beim Erstellen der Lernplattform OLAT. Schon zuvor wurde die ICT-Fachstelle⁶ der Universität Zürich, eine Fachstelle des Prorektors Lehre, welche den Einsatz neuer Medien im Unterricht fördert, auf das Projekt OLAT aufmerksam. Sie beschloss dieses Projekt finanziell zu unterstützen, da sich die finanziellen Ressourcen der Software Engineering Gruppe langsam erschöpften.

2.3 Erweiterung des Einsatzgebietes am OLAT-Zentrum

Es traten nun vermehrt verschiedene Institute der Universität Zürich an die Software Engineering Gruppe mit der Bitte heran, OLAT ebenfalls benutzen zu dürfen. Es lag daher auf der Hand, OLAT längerfristig am ZI⁷ (Zentrum Informatikdienste) anzusiedeln und es als Service der ganzen Universität zur Verfügung zu stellen. Die Software Engineering Gruppe hatte nicht die Ressourcen oder die Aufgabe, für die Universität

⁴ <http://www.access.unizh.ch>

⁵ <http://www.medidaprix.org>

⁶ <http://www.ict.unizh.ch>

⁷ <http://www.zi.unizh.ch>

einen Serverbetrieb aufrecht zu erhalten.

Im Frühjahr 2001 wurde daher das OLAT-Zentrum⁸ geschaffen, welches neben der Wartung und Erweiterung der Lernplattform auch Schulungen und Support für Kursautoren anbietet. Das OLAT-Zentrum hat einen Auftrag der ICT-Fachstelle, ist organisatorisch jedoch beim ZI angesiedelt, welches den Betrieb des Servers sicherstellt. Am OLAT-Zentrum wird OLAT gepflegt und einige fehlende Funktionalitäten werden zu dem jetzigen System hinzuentwickelt, jedoch nur im Rahmen von Wartungsarbeiten. Fundamentale Weiterentwicklungen sind bisher nicht vorgesehen gewesen und sind aus Kapazitätsgründen im Rahmen des OLAT-Zentrum zum derzeitigen Zeitpunkt auch nicht möglich.

Für das kommende Wintersemester WS01/02 sind ca. ein Dutzend Lehrveranstaltungen über OLAT in Vorbereitung, und weitere sind geplant für die kommenden Semester. Die Möglichkeiten des Supports vor Ort und der Schulungskurse des OLAT-Zentrums werden rege benutzt. Die Kunden stammen dabei aus sehr unterschiedlichen Fachbereichen wie Biologie, Informatik, Betriebswirtschaftslehre oder Geschichte, um einige Beispiele zu nennen. Die Lehrveranstaltungen reichen von einer Hand voll Studierenden bis hin zu Veranstaltungen mit über 1000 aktiven Studierenden. Einige Veranstaltungen werden sogar in Kooperation mit der ETH oder anderen Universitäten durchgeführt. Diese Studierenden greifen ebenfalls auf den OLAT-Server der Universität zu. Es besteht Interesse für eigene OLAT-Server an anderen Universitäten um eigene Kurse zu entwickeln.

2.4 Limitationen von OLAT 1

OLAT ist ursprünglich in einem genau definierten Umfeld für einen sehr spezifischen Einsatz entwickelt worden. In diesem Umfeld ist OLAT eine sehr gute und attraktive Alternative zu den kommerziell erhältlichen Lernplattformen. In der Zwischenzeit sind die Anforderungen an die Lernplattform OLAT jedoch in zweierlei Hinsicht massiv gestiegen: erstens wird OLAT in Zukunft in sehr viel mehr Veranstaltungen an der Universität Zürich eingesetzt werden, was zu einem erheblichen Anstieg der gleichzeitig aktiven Benutzer führt. Zweitens finden diese Veranstaltungen in einem anderen Kontext statt und verlangen daher nach erweiterter oder zusätzlicher Funktionalität. Es ist zu befürchten, dass OLAT diesen Forderungen nach Skalierbarkeit und Erweiterbarkeit nicht gerecht werden kann.

2.4.1 Skalierbarkeit

OLAT basiert auf einer typischen Client-Server Architektur, welche bei Webapplikationen zwangsläufig vorgegeben ist: Der Webbrowser fungiert als Client und OLAT als Server. Man könnte nun meinen, dass es sich hier um eine richtige 3-Tier Architektur mit einer Database-, Business- und Clientschicht handelt. Dies ist jedoch nur bedingt der Fall, da die Clientschicht im Falle dieser Webapplikation lediglich zur Darstellung der Daten dient und in dem Sinne nicht viel mehr als ein graphisch erweitertes Terminal darstellt. Abgesehen von wenigen dynamischen, in JavaScript programmierten Elementen hat der Webbrowser nichts mit der Präsentationslogik zu tun, sondern kümmert sich lediglich um das Rendering der bereits formatierten Daten. Daher ist diese

⁸ <http://www.olat-zentrum.unizh.ch>

Architektur zwischen den von Jaccottet genannten Konzepten der *Verteilten Präsentation* (Distributed presentation) und des *Präsentationsspezialisten* (Remote presentation) anzusiedeln.[Jac97] Man könnte die Architektur auch als eine 2.5-Tier Architektur bezeichnen.

Die bei OLAT verwendete Architektur Datenbank - PHP/Apache - Webbrowser ist nicht falsch, aber sie ist problematisch im Hinblick auf die Skalierbarkeit des Systems. Da im Client keinerlei Logik eingebaut ist, muss diese auf der Ebene des Webservers implementiert sein. Der Ablauf einer Interaktion zwischen Client und Server spielt sich folgendermassen ab: ein Client stellt eine Anfrage an den Webserver, dieser führt daraufhin einen OLAT Programmteil aus, greift dafür auf die Datenbank zu, und schickt schliesslich das berechnete Resultat als HTML formatiert an den Client zurück. OLAT ist dabei kein ständig laufender Prozess, welcher konstant im Speicher des Servers läuft, sondern ein in der Programmiersprache PHP geschriebenes Skript, das lediglich zum Zeitpunkt des Aufrufs sequenziell abgearbeitet wird. Der Systemstatus muss somit jedesmal neu initialisiert und die Daten müssen von der Datenbank erneut gelesen werden. Die Tatsache, dass mehrere Leute vermutlich auf identische Datenobjekte wie beispielsweise eine Kapitelseite eines Onlineskriptes zugreifen möchten, kann mit diesem Konzept nicht genutzt werden, da diese Aufrufe vollkommen getrennt voneinander ablaufen und der Systemstatus eines Benutzers nach einer Anfrage sofort in der Datenbank persistiert wird. Cachingmechanismen lassen sich somit auf Ebene OLAT nicht implementieren.

Dies bedeutet, dass die Skalierbarkeit des Systemes nur begrenzt gegeben ist. Die Last des Webservers kann zwar auf mehrere Rechner verteilt werden, doch müssen diese Rechner alle auf dieselbe Datenbank zugreifen. Somit verschiebt sich die Frage der Skalierbarkeit einzig auf die Frage, wie gut das Datenbankmanagementsystem skaliert.

2.4.2 Erweiterbarkeit

Bei der Entwicklung von OLAT wurde die Trennung zwischen Model, View und Controller⁹ zu wenig beachtet. Das Datenmodell wird zwar definiert durch die SQL Tabledenitionen, doch stecken im Programmiercode eine Reihe von verborgenen Teilen des Datenmodells. Eine saubere Kapselung aller Daten durch Objekte gibt es nicht. Überhaupt ist OLAT nicht objektorientiert, sondern grösstenteils sequenziell programmiert, auch wenn dies PHP zulassen würde.

Leider wurde bei der Entwicklung der Trennung zwischen Layout und Logik zu wenig Beachtung geschenkt. So kommt es, dass die Logik nicht gekapselt in einem API (Application Programming Interface) zur Verfügung steht, sondern an den jeweiligen Orten, an denen sie gebraucht wird, direkt programmiert wurde. Dadurch wird der Code zwar relativ effizient, es wird jedoch sehr schwierig, das Programm zu warten. Änderungen an der Datenbank ziehen zum Beispiel Änderungen an diversen Orten im Programmcode nach sich. Eine beliebige Erweiterung des Codes ist nur sehr schwer zu erreichen, da überall mit Seiteneffekten gerechnet werden muss.

⁹ Beim MVC Konzept wird eine Klare Trennung zwischen dem Datenmodell, dem Controller, der die Benutzereingaben behandelt und der Darstellung der Daten unterschieden.[ACM01]

3. SYSTEMIDEE FÜR OLAT 2

3.1 Allgemeine Anforderungen

Die zu entwickelnde Lernplattform soll vollständig webbasiert sein und Veranstaltungen der Form 'Telekooperation', 'Webbased Training' und 'Virtuelle Seminare' unterstützen können. Grossen Wert wird dabei auf die Gruppenprozesse, die Kooperation und Kollaboration und die Administration der Gruppen und Klassen gelegt. Das System soll Komponenten zur Verwaltung und Benutzung von Lehrveranstaltungen und Onlineskripten, ein Testsystem, ein Evaluationssystem und ein System für das Gruppenmanagement beinhalten. Ausserdem sollen diese Dienste beliebig vielen Personen angeboten werden können. Ein spezielles Augenmerk ist dabei auf die Bestimmungen des Datenschutzgesetzes zu legen. An den erforderlichen Stellen sollen die Daten verschlüsselt abgelegt oder transportiert werden.

3.2 Kompatibilität zu OLAT 1

Obwohl die neue Architektur von OLAT vollkommen unabhängig von der jetzigen Version sein soll ist es wichtig, dass die Daten des alten Systemes mit dem neuen System verwaltet werden können, um die getätigten Investitionen der Hersteller von Lerninhalten zu sichern. Dies betrifft hauptsächlich den Bereich Onlineskript und Test, aber auch sämtliche Metadaten der Lehrveranstaltungen sollen in das neue System importiert oder damit verwaltet werden können.

3.3 Universitätsweiter Einsatz

Die Universität hat mit der Institutionalisierung der ICT-Fachstelle den Bereich E-Learning als ein wichtiges Standbein der Zukunft der Universität Zürich definiert. Es ist geplant, bis ins Jahr 2006 10 Prozent aller Lehrveranstaltungen elektronisch zu unterstützen. Dies entspricht rund 250-300 Veranstaltungen. Es ist abzusehen, dass der Trend zu E-Learning bei diesen 10 Prozent nicht Halt machen wird und längerfristig sehr viel mehr Veranstaltungen von asynchronen Kommunikationsmitteln wie Email, Diskussionsforen oder der elektronischen Verbreitung von Materialien Gebrauch machen werden.[Sch01]

Heute werden hauptsächlich Lehrveranstaltungen des Grundstudiums online angeboten, um effizienter mit der Studienmasse umgehen zu können. Es zeichnet sich jedoch ab, dass längerfristig auch Veranstaltungen des Hauptstudiums von den neuen Möglichkeiten profitieren sollen, speziell dann, wenn der doppelte Maturitätsjahrgang in ca. drei Jahren das Hauptstudium erreichen wird. Viele dieser Veranstaltungen werden

in hybrider Form angeboten werden, was keine Verdrängung der alten Form bedeutet, sondern eine angemessene Bereicherung und Unterstützung der bestehenden Form durch die elektronischen Medien darstellt. Auch diese hybriden Formen werden von einer zentral eingesetzten Lernplattform Gebrauch machen, bei welcher man sich nicht um den Rahmen einer Lehrveranstaltung und deren Platzierung im Internet kümmern muss und sich direkt mit dem eigentliche Ziel Mehrwert anzubieten beschäftigen kann.

Die Universität Zürich war schon seit jeher ein sehr heterogenes Umfeld. Ein gemeinsames Erscheinungsbild und gemeinsame Vorgehensweisen sind schwer und nur langsam in allen Instituten durchzusetzen. Dies gilt auch bei der Wahl der Lernplattform. Aus Gründen der Effizienz, des Knowledge Transfers, der Entwicklungs-, Wartungs-, Support-, Schulungs- und Lizenzkosten ist eine nicht zentral gesteuerte und heterogene Softwarelandschaft nicht anzustreben. Auf eine zentral organisierte Plattform, welche offen genug ist um auf alle Anforderungen eingehen zu können, sollte hingearbeitet werden. Nur so kann eine durch elektronische Medien unterstützte, interdisziplinäre, kooperative und kollaborative Arbeitsumgebung geschaffen werden. Eine solche campusübergreifende Lernplattform stellt höchste Ansprüche an die Ausfallsicherheit, Verfügbarkeit, Integrität und Vertraulichkeit des Systems und der Daten.

3.4 Anbindung an universitäre Informationssysteme

Das neue System soll mit den existierenden Verwaltungssystemen der Universität kooperieren können. Eine Schnittstelle, z.B. zu SAP-Campus, ist notwendig um den Administrationsaufwand und die Doppelführung von Datensätzen zu vermeiden. Dadurch wird es beispielsweise möglich, das seit diesem Semester an der Wirtschaftswissenschaftlichen Fakultät eingeführte APS (Anrechnungspunktesystem) sowohl im Verwaltungssystem mit SAP als auch in der für die Studierenden sichtbaren Lernplattform transparent zu behandeln.

3.5 Berücksichtigung von Internationalen Standards

Internationale Standards für E-Learning, im Speziellen die Standards des IMS Global Learning Consortium¹, sind wichtig um Lerninhalte und Personendaten plattformunabhängig austauschen zu können. Damit ist es beispielsweise möglich, dass ein Studierender von Bern nach Zürich kommt um hier sein Studium zu beenden und die persönlichen Daten samt den bisher erreichten APS-Punkten zwischen den Universitäten elektronisch ausgetauscht werden können. Kursinhalte können gekauft, verkauft oder ausgetauscht werden, da ein standardisiertes Format verwendet wird. Diese Standards sollen im neuen System unterstützt werden.

3.6 Interoperabilität

Alle Projekte des SVC² (Swiss Virtual Campus) sind Kooperationsprojekte zwischen mehreren Universitäten. Zur Zeit werden auch auf OLAT SVC Projekte angeboten. Dabei müssen für die externen Studierenden temporäre UniAccess Accounts generiert

¹ <http://www.imsglobal.org>

² <http://www.virtualcampus.ch>

werden, damit diese sich auf dem Server in Zürich einloggen können. Separate Installationen von OLAT an den anderen Universitäten wären zwar möglich, doch würden die Personen dann nicht mehr eine gemeinsame Veranstaltung besuchen, sondern nur eine lokale Kopie davon. Für SVC Projekte und allgemein für die bessere Kooperation und Kollaboration in der Lehre und Forschung zwischen den Universitäten muss eine Lernplattform in Echtzeit transparent mit Installationen an anderen Universitäten kommunizieren können. So wird das Durchführen von gemeinsamen Veranstaltungen mit gemeinsamer Interaktion effektiv möglich und reduziert sich nicht lediglich auf das gemeinsame Erstellen von Lerninhalten. Durch diese direkten Interaktionsmöglichkeiten könnte ein weltweit neuartiges und zukunftsweisendes Wissens- und Lernnetzwerk aufgebaut werden. Die Architektur von OLAT 2 soll so offen gestaltet sein, dass eine solche Interoperabilität später entwickelt werden kann.

4. INTERNATIONALE STANDARDS

E-Learning ist kein neues Thema. Unter dem Name *Computer Based Instruction* wurde schon in den 60-er Jahren versucht, mit Hilfe von Rechenmaschinen den Menschen beim Lernen zu unterstützen. Über die Jahre sind viele verschiedene Systeme entwickelt worden, welche jedoch alle auf ihren eigenen proprietären Datenmodellen aufbauen und daher inkompatibel zueinander sind. Einige Organisationen versuchen diesen Missstand durch die Definition von internationalen Standards aufzuheben. Im folgenden werden drei wichtige Gruppen von Standards besprochen.

4.1 IEEE Learning Technology Standards Committee

Das Institute of Electrical and Electronics Engineers¹ hat 1996 die Arbeitsgruppe Learning Technology Standards Committee², kurz LTSC, ins Leben gerufen. Über die Ziele der Gruppe ist auf ihrer Homepage zu erfahren: „*The mission of IEEE LTSC working groups is to develop technical Standards, recommended Practices, and Guides for software components, tools, technologies and design methods that facilitate the development, deployment, maintenance and interoperation of computer implementations of education and training components and systems.*“ [Rob01]

Die Gruppe trägt die IEEE Nummer 1484 und ist in fünf Bereiche und 15 Untergruppen aufgeteilt. Der Untergruppe ist jeweils eine Unternummer zugeteilt, welche durch einen Punkt von der Hauptnummer getrennt ist:

- **General**

- *1484.1 Architecture and Reference Model working group*
In dieser Gruppe wird eine grundsätzliche Lernarchitektur vorgestellt, welche als Basis für alle anderen Standards dienen soll. Im folgenden Kapitel 4.1.1 wird auf dieses Modell näher eingegangen.
- *1484.3 Glossary working group*
In diesem Standard werden die Begriffe beschrieben, welche in den anderen LTSC-Standards verwendet werden.

- **Learner-Related**

- *1484.2 Learner Model working group*
In dieser Gruppe wird an der Syntax und Semantik eines Lerner-Modells gearbeitet. Das Modell beinhaltet alle Informationen, welche mit einem Lernenden assoziiert werden können. Dies geht von den einfachen Persönlichkeiten über sein Wissen und seine Fähigkeiten bis hin zu Informationen über

¹ <http://www.ieee.org>

² <http://ltsc.ieee.org>

sein Lernverhalten. Das Modell erlaubt auch Ansichten der selben Person aus verschiedenen Perspektiven (Lerner, Coach, Eltern etc.)

– *1484.13 Student Identifiers working group*

Dieser Standard regelt, wie Lernende Zugriff auf ihre Informationen erhalten sollen, zum Beispiel über ein Login und Passwort und wie dieses gestaltet werden soll.

– *1484.20 Competency Definitions working group*

Wenn Personen Kurse absolvieren erarbeiten sie sich damit Kompetenzen. Diese Kompetenzen müssen beschrieben, verglichen und ausgetauscht werden können, damit mehrere Lernsysteme miteinander operieren können. In diesem Standard wird festgehalten, wie diese Definitionen zu gestalten sind.

• **Content-Related**

– *1484.10 CBT Interchange Language working group*

Mit diesem Standard wird definiert, welche Medienelemente und Programmiersprachen in Computer Based Trainingobjekten verwendet werden sollen. Des weiteren wird ein allgemeines Austauschformat ausgearbeitet, mit dessen Hilfe Lerninhalte zwischen verschiedenen Autorensystemen ausgetauscht werden können.

– *1484.6 Course Sequencing working group*

Die Course Sequencing Gruppe arbeitet an einem Modell um die Sitzungen in einem Lernsystem besser handhaben zu können. Dies beinhaltet eine Spezifikationssprache mit Syntax und Semantik, einem Mechanismus für die Übertragung von Kontrollelementen und Daten sowie speziellen Mechanismen, um das Verhalten des Sessionmanagements genau festlegen zu können.

– *1484.17 Content Packaging working group*

Content besteht fast immer aus mehreren Dokumenten. Dazu kommen Metadaten, welche die einzelnen Dokumente oder das ganze Contentpaket beschreiben. Mit diesem Standard wird definiert, wie die Dokumente zusammengepackt werden, so dass diese als ein Objekt von einem zum anderen Punkt übertragen werden können.

• **Data and Metadata**

– *1484.12 Learning Objects Metadata working group*

Um Lerninhalte beschreiben und diese auch austauschen zu können, muss auf ein wohldefiniertes Set von Metadaten zurückgegriffen werden können. Dieser Standard nimmt sich dieses Problems an durch die Definition der Lernobjekt-Metadaten (LOM)

– *1484.9 Localization working group*

Dieser Standard behandelt alle Themen, die mit den sprachlichen und kulturellen Unterschieden zusammenhängen. Dazu gehören nicht nur die Definition der Kodierung von alphanumerischen Zeichen, sondern auch die Standardisierung von Metaphern und Symbolen.

– *1484.14 Semantics and Exchange Bindings working group*

Die bisher aufgezählten Standards beziehen sich meist auf Inhalte, nicht je-

doch wie diese ganz konkret technisch umgesetzt werden. In dieser Gruppe befasst man sich mit der Umsetzung der 1484 Standards durch XML-Bindings.

- *1484.15 Data Interchange Protocols working group*

Um die standardisierten Dokumente austauschen zu können bedarf es eines Übertragungsprotokolls. Ziel dieser Gruppe ist es, ein Übertragungsprotokoll zu definieren, das über eine feinere Granularität als HTTP verfügt und einfach zu implementieren ist.

- **Management Systems and Applications**

- *1484.11 Computer Managed Instruction working group*

Diese Arbeitsgruppe beschäftigt sich mit der inhaltlichen Organisation von Kurseinheiten. Wie werden Lektionen definiert, wie startet ein Lernsystem eine Lektion, wie wird die Reihenfolge festgelegt und wie können Ziele und Aufgaben in diesen Zusammenhang integriert werden? Diese Fragen werden in dieser Gruppe zu beantworten versucht.

- *1484.18 Platform and Media Profiles working group*

Um Lerninhalte portabel gestalten zu können müssen die technischen Voraussetzungen geklärt sein. In diesem Standard wird definiert, was genau unter Namen wie *Webbrowser* verstanden wird, welche Graphiken damit dargestellt werden können, welche HTML Version unterstützt werden muss etc. Für verschiedene Szenarien werden die Anforderungen genau spezifiziert.

- *1484.7 Tool/Agent Communication working group*

In einem speziellen Lernszenario interagiert ein Lerner mit einem normalen Softwaretool wie Microsoft Word um die Funktionen desselben zu erlernen oder eine Aufgabe damit zu erledigen. Eine Lernsoftware, hier Agent genannt, unterstützt den Lernenden dabei. Dieser Standard definiert, wie das Tool mit dem Agenten kommunizieren kann. Das Tool kann dem Agenten mitteilen was der Lernende macht, so dass der Agent entsprechend reagieren kann. Andererseits möchte der Agent auch Objekte in dem Tool direkt manipulieren können um dem Lernenden einen Effekt vor Augen zu führen. Ein weiteres Szenario ist die Kommunikation von Agent zu Agent über diesen Standard.

An all diesen Standards wird intensiv gearbeitet. Sie existieren jedoch erst als Drafts und noch nicht in ihren finalen Versionen. Wie wir im Folgenden sehen werden, hängen die Bestrebungen aller Organisationen Standards zu schaffen eng zusammen und so referenzieren sich fast alle Standards gegenseitig und bauen aufeinander auf. Im nächsten Kapitel wird auf den wichtigsten dieser Standards, das *Architecture and Reference Model* näher eingegangen.

4.1.1 IEEE 1484.1: Architecture and Reference Model

Der IEEE 1484.1 Standard ([T⁺01]) beschreibt ein Referenzmodell einer Lernarchitektur. Dabei handelt es sich nicht direkt um eine Softwaresystemarchitektur, sondern um das Identifizieren von Komponenten eines allgemeinen Lernsystems. Dies ist vollkommen unabhängig davon, ob die Komponenten mit Software implementiert sind oder ob

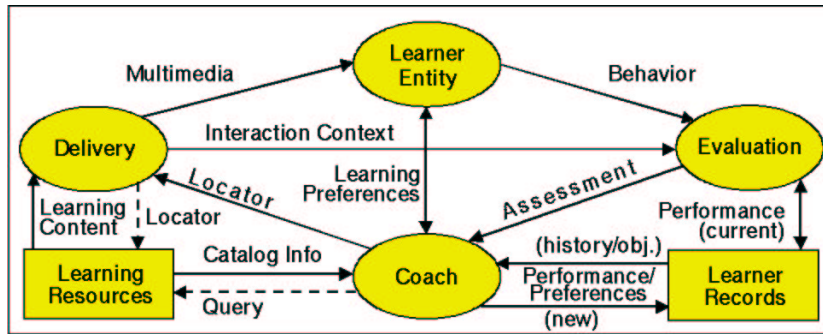


Fig. 4.1: Komponenten der Learning Technology Systems Architecture (Quelle: IEEE 1484.1/D8 [T⁺01])

es sich um andere Technologien handelt. Die LTSC Architektur lässt sich auch auf ein herkömmliches Szenario einer Schule mit Bibliothek und Klassenzimmern anwenden, wie dies im Kapitel 16.3 des Draft 8 des Dokumentes gezeigt wird. Trotzdem ist klar, dass dieses High-Level Design in erster Linie für das technologieunterstützte Lernen und Ausbilden sowie Lernsysteme wie OLAT entwickelt wurde. Das Kürzel LTSA der Architektur steht für *Learning Technology Systems Architecture* und ist nicht zu verwechseln mit dem Kürzel LTSC, welches für alle Gruppen der IEEE 1484 Standards steht.

Der IEEE 1484.1 Standard verhält sich gegenüber pädagogischen Zielen, den Inhalten, den kulturellen Unterschieden und der Wahl von Plattformen neutral. Als wichtigste Ziele diese Standards sind zu nennen:

1. Der Standard soll als High-Level Framework helfen, bestehende und zukünftigen Systeme besser zu verstehen.
2. Die Architektur ist kein Bauplan um ein neues System zu bauen, sondern eine Architektur, welche bei der Analyse und Kommunikation sowie beim Design eines Systems unterstützend wirken sollen.
3. Durch die Identifikation der kritischen Systemschnittstellen und Services sollen Interoperabilität und Portabilität erreicht werden.
4. Die Identifikation von Protokollen und Methoden der Kooperation und Kollaboration.

Der IEEE 1484.1 Standard ist granular aufgebaut. Es gibt nicht nur eine Konformität sondern feine Abstufungen, welche sich auf die Anzahl der implementierten Komponenten beziehen. Ein System gilt als konform, sobald mindestens zwei Komponenten vorhanden sind, wobei die Lerner-Einheit immer ein Teil des Systems sein muss. Höchste Konformität wird mit dem Label *Conforms to IEEE 1484.1:2001 LTSA all components* gekennzeichnet.

Die Abbildung 4.1 zeigt die LTSA Komponenten. Es wird dabei unterschieden zwischen:

- *Prozesse*
 - Lerner Entität
 - Auswertung (Evaluation)
 - Coach Entität
 - Zustellung (Delivery)
- *Datensenken*
 - Akte Lerner (Learner records)
 - Lernmaterialien (Learning resources)
- *Datenflüsse*
 - Lernpräferenzen (Learning Preferences)
 - Verhalten Lerner (Behaviour)
 - Bewertungsinformationen (Assessment information)
 - Leistungs- und Präferenzinformationen (Performance/preference information)
 - Abfrage (Query)
 - Katalog (Catalog info)
 - Zugangsbeschreibung (Locator)
 - Lerninhalt (Learning content)
 - Multimedia (Multimedia)
 - Interaktionskontext (Interaction context)

Die Schwierigkeit des LTSA Modells ist es, die Begriffe so offen zu verstehen wie sie gemeint sind. Unter Coach beispielsweise wird nicht ein personifizierter Coach verstanden, sondern jegliche Coachingfunktion welche dem Lernenden gegenübersteht. Dies kann eine Person sein mit einem Auftrag (Coach), eine beliebige Person mit einer temporären Rolle (Ein Lerner, welcher einem anderen Lernenden in einer Diskussion etwas erklärt), die Vorgaben der Kursbetreuerin diese und jene Kapitel zu bearbeiten oder ein vollautomatisiertes regelbasiertes Computersystem. Die dabei ablaufenden Prozesse sind in ihrer Essenz aber die selben und darauf baut die Architektur auf. Um ein Gefühl für die Bedeutung der Komponenten zu erlangen wird in dem Standard im Detail auf diese Abläufe eingegangen und gezeigt, welche Prozesse, Datensenken und -flüsse jeweils beteiligt sind.

Rückblickend kann gesagt werden, dass OLAT 1 eine sehr hohe LTSA Konformität erreicht hat, wenn nicht sogar alle Elemente der Architektur beinhaltet. Für eine Neuentwicklung sollte eine volle LTSA Konformität angestrebt werden.

4.2 IMS

Das IMS Global Learning Consortium, Inc (IMS)³ besteht aus zahlreichen Mitgliedern aus Wirtschaft und staatlichen Organisationen und hat sich zum Ziel gesetzt, offene Spezifikationen für den Bereich E-Learning zu entwickeln. Hauptsächlichste Themen sind: das Auffinden und Benutzen von Lerninhalten, die Probleme des Festhaltens des Lernfortschritts und der Bewertung von Lernenden sowie der Austausch aller in diesem Zusammenhang möglichen Daten zwischen verschiedenen administrativen Systemen. Neben der reinen Definition von Spezifikationen für die Interoperabilität unterstützt IMS auch die Implementation der Spezifikationen in existierenden Produkten. IMS stellt jedoch selbst keine eigene Software her.

Die IMS Spezifikationen haben sich auf dem Markt bereits grossflächig etabliert und es gibt heute einige Produkte, welche die Spezifikationen implementiert haben. Alle grösseren Hersteller von Software, welche im Bereich E-Learning tätig sind, sowie viele Universitäten sind als *Contributing Members* auf der Homepage von IMS aufgelistet. Darunter findet man bekannte Namen wie ADL, Apple Computer, Blackboard, California State University, Cisco Systems, Click2Learn, IBM, Massachusetts Institute of Technology, Microsoft, Oracle, PeopleSoft, University of Michigan, WebCT. Als *Developer*, welche die IMS Spezifikationen in ihre Produkte integrieren wollen sind über 180 Firmen und Universitäten aufgelistet.⁴

Auch die IMS Spezifikationen sind in ständiger Erweiterung begriffen und machen eine rasante Entwicklung durch. Fast alle Dokumente haben bereits den Grad einer finalen Version 1 erreicht und einige davon nähern sich bereits der Version 2 an. IMS ist in neun Arbeitsgruppen organisiert:

- Competency Team
- Question & Test Team
- Content Management Team
- Profiles Team
- Meta-Data Team
- Learning Design Team
- Accessibility Team
- Simple Sequencing Team
- Digital Repositories Team

Die Arbeitsgruppen publizieren nicht nur Spezifikationen sondern auch Whitepapers, Handbücher und andere nützliche Informationen:

- IMS Learning Resource Meta-Data Specification
- IMS Enterprise Specification

³ <http://www.imsglobal.org>

⁴ Siehe <http://www.imsglobal.org/membership.html>

- IMS Content Packaging Specification
- IMS Question & Test Interoperability Specification
- IMS Learner Information Package Specification
- IMS Reusable Competencies Definition Specification
- IMS Guidelines for Developing Accessible Learning Applications
- XML Bindings and DTDs
- IMS XML Schemas
- IMS Resource Description Framework(RDF) Bindings
- IMS Implementation Handbooks

In den folgenden Kapiteln werden nun die wichtigsten IMS Spezifikationen kurz beschrieben. Sie nehmen Bezug auf die Lernmaterialien einerseits und auf den Lerner als Hauptgegenstand aller Betrachtungen andererseits. Grundsätzlich sind die verschiedenen IMS Spezifikationen unabhängig voneinander, manchmal überlappen sie sogar ein wenig. Dort wo sie überlappen ist auf Kompatibilität geachtet worden, so dass es durchaus Sinn macht verschiedene IMS Spezifikationen zu unterstützen. Es ist aber nicht zwingend, dass man gleich alle Spezifikationen implementiert.

4.2.1 IMS Learning Resource Meta-Data Specification

Die IMS Meta-Data Gruppe befasst sich mit den Metadaten, die Lerninhalte sowie deren Komponenten, die dadurch austauschbar und auffindbar werden. Um besser zu verstehen, welche Arten von Informationen durch diese Spezifikation abgedeckt sind, folgt eine Liste ausgewählter Elemente des Meta-Data Modells.

Elementname	Beschreibung
identifoyer	Global eindeutiges Label für ein Lernobjekt
title	Titel des Lernobjektes
catalog	Katalog, unter dessen Name das Objekt zu finden ist (Z.B. ISBN)
language	Sprache, in welcher das Objekt geschrieben ist
description	Beschreibung des Lernobjekts
keyword	Beschreibung des Lernobjekts mittels Keywords
version	Version des Objektes
typicalagerange	Alter des anvisierten Zielpublikums
difficulty	Schwierigkeit für einen durchschnittlichen Lerner des Zielpublikums
typicallearnertime	Durchschnittliche Zeit, welche zur Bearbeitung benötigt wird

Tab. 4.1: Beispiele der IMS Learning Resource Spezifikation und des IEEE LOM Standards

Neben diversen Beispielen und weiteren Informationen finden sich auf der Homepage ⁵ drei Dokumente, welche die IMS Learning Resource Metadaten spezifizieren:

⁵ <http://www.imsglobal.org/metadata/>

- *IMS Learning Resource Meta-data Information Model [MT01b]*
Im Kapitel 4.1 auf der Seite 16 wird der IEEE 1484.12 *Learning Objects Meta-data (LOM)* Standard angesprochen. Die IMS Learning Resource Meta-Data Spezifikation versteht sich als Arbeitsgrundlage für die IEEE Kommission, unterstützt deren Standard vollumfänglich und ergänzt ihn mit einer Reihe von Erweiterungen. Es wird allgemein erwartet, dass die von IMS vorgeschlagenen Änderungen vom IEEE Standard adaptiert werden. Es besteht eine enge Zusammenarbeit zwischen den beiden Gruppen und einige der Mitglieder arbeiten in beiden Gremien zugleich. In diesem Dokument sind die IEEE Metadaten und die IMS Ergänzungen beschrieben.
- *IMS Learning Resource Meta-data XML Binding [MT01c]*
Während der IEEE Standard die Metadaten grundsätzlich bespricht und keine technischen Hinweise zur Implementation liefert, nimmt sich die IMS Spezifikation genau dieses Problems an. Dabei haben sie sich für XML⁶ (eXtensible Markup Language) entschieden, einem durch das W3C Konsortium⁷ verabschiedeten Standard zur Beschreibung von hierarchischen Daten. XML erfreut sich zunehmender Akzeptanz im Markt und wird zur Zeit von zahlreichen Herstellern in deren Produkte integriert.[SE99] In dem XML Binding Dokument der IMS Spezifikation wird das IMS Meta-Data Modell durch eine XML DTD (Document Type Definition), ein W3C XML und ein Microsoft XML-Data Schema technisch genau spezifiziert. Dokumente, welche diesen Definitionen gerecht werden, sind von System zu System austauschbar.
- *IMS Learning Resource Meta-data Best Practice and Implementation Guide [MT01a]*
Dieses Dokument erklärt, wie die IMS Learning Resource Metadaten in Systemen verwendet werden können. Es ist allerdings keine Anleitung, wie dies implementiert werden soll, also wie eine technische Architektur oder Umsetzung der Spezifikation auf einem System vollzogen werden soll. Neben einer Übersicht über die Metadaten Elemente finden sich auch Angaben über die Anforderungen zur Erreichung der Konformität mit der Spezifikation. Ebenso sind in dem Dokument Angaben zu der Taxonomie ersichtlich sowie eine kurze Erklärung, wie die Dublin Core⁸ (DC) Elemente auf die IMS Elemente abgebildet werden können.

4.2.2 IMS Content Packaging Specification

Ziel der IMS Content Packaging Specification ist es, Interoperabilität von Lerninhalten zwischen Contentauthoring-, Contentmanagement- und Run-Time-Systemen zu erreichen. Die Abbildung 4.2 zeigt eine Übersicht über das IMS Content Framework: Ein Autor erstellt in einem Autorensystem Content. Dieser wird exportiert und in ein Contentmanagementsystem importiert. In diesem System verwaltet ein Administrator oder eine Kursbetreuerin die Daten, fügt diese beispielsweise einem Kurs als Ressource zu. Der Lerner benutzt den Content über ein Run-Time Environment. Wichtig sind dabei

⁶ <http://www.w3c.org/XML>

⁷ <http://www.w3c.org>

⁸ <http://purl.org/dc/>

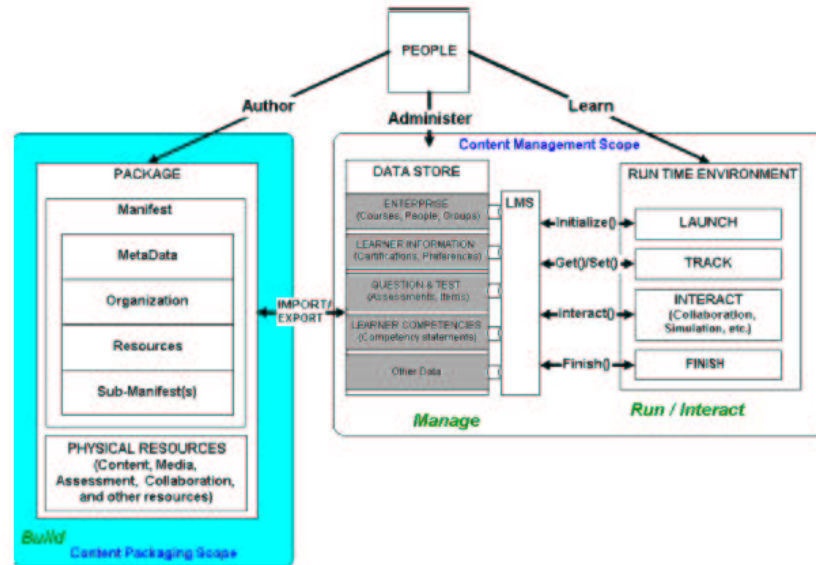


Fig. 4.2: IMS Content Framework (Quelle: IMS Content Packaging Specification Best Practice and Implementation Guide [AM01a])

die Funktionen *Launch*, *Track*, *Interact* und *Finish*. Über diese Funktionen kommuniziert die Run-time Umgebung mit dem Lernmanagementsystem.

Mit Hilfe der im Kapitel 4.2.1 beschriebenen IMS Learning Resource Specification werden die einzelnen Elemente eines Lerninhaltes beschrieben. Die IMS Content Packaging Specification dient nun zur Festlegung, welche Dokumente zu einer Contenteinheit gehören, beispielsweise zu einem Online-Skript, und in welcher Reihenfolge diese zueinander stehen. Diese Informationen werden in einer speziellen XML Datei gespeichert, dem sogenannten *Manifest* mit dem dafür reservierten Namen *imsmanifest.xml*. Das Manifest ist in dem selben Verzeichnis wie der Content selbst gespeichert. Das ganze Verzeichnis wird nun komprimiert mit zip, jar oder cab. Die resultierende Datei ist das *Package Interchange File* gemäss IMS Spezifikation.

Links in der Abbildung 4.2 ist die Struktur eines IMS Content Packages dargestellt: Ein Package besteht aus dem Manifest und Ressourcen. Die Ressourcen stellen den eigentlichen Content dar und sollten gemäss IMS Learning Resource Meta-Data Spezifikation beschrieben sein. Es sind auch Assessment Elemente gemäss der IMS Question & Test Interoperability Specification vorgesehen, dies ist jedoch noch nicht in das Packaging Modell integriert. Ebenso wenig ist das Datenmodell oder das Run-time Environment in dieser Version der Spezifikation integriert. Zur Zeit sagt diese Spezifikation lediglich aus, wie Content für einen Import oder Export verpackt werden muss.

Wie in den meisten IMS Spezifikationen finden wir auch hier drei Dokumente:

- *IMS Content Packaging Information Model*[AM01b]
In diesem Dokument wird das konzeptionelle Modell des IMS Packagings vorgestellt und alle Elemente des Manifestes erklärt.



Fig. 4.3: ASI Datenstruktur der IMS QTI (Quelle: IMS Question & Test Interoperability: An Overview [SSBL01a])

- *Content Packaging XML Binding[AM01c]*
Die hier genau spezifizierte DTD, das W3C XML Schema und das Microsoft XML-Data Schema legen die zulässigen XML Definitionen für die Manifest Elemente fest. In dem Dokument sind auch einige Beispiele enthalten.
- *Content Packaging Best Practice Guide[AM01a]*
Der Best Practice Guide enthält eine Einführung in das IMS Content Framework und viele Beispiele und Hintergrundinformationen, welche für das Verständnis und die Implementation hilfreich sind.

4.2.3 IMS Question & Test Interoperability Specification

Die IMS Question & Test Interoperability Specification, kurz QTI, gibt eine Struktur für Fragen und Tests an und erläutert, wie die Antworten der Lernenden gehandhabt werden sollen. Dies hat wiederum zum Ziel, solche Onlinetests zwischen den verschiedenen Lern- und Autorensystemen austauschen zu können und damit Interoperabilität zu erreichen. Die durch die Spezifikation zur Verfügung gestellten XML Elemente definieren nicht nur ein grosses Set an standardisierten Fragetypen, sondern lassen gezielt auch proprietäre Typen zu, welche dann aber ausserhalb der Spezifikation genauer definiert werden müssen und nicht mehr allgemein austauschbar sind. Zu den standardisierten Fragetypen gehören *True/False*, *Multiple Choice*, *Multiple Response*, *Image hot spot*, *Fill-in-blank*, *Select text*, *Slider*, *Drag object*, *Drag target*, *Order Objects*, *Match item* und *Connect the points*. Neben der Definition von austauschbaren Tests mit standardisierten Attributen nimmt sich die Spezifikation auch des Problems des Packagings der Tests an.

In der IMS Question & Test Interoperability Specification werden die folgenden Elemente verwendet:

- *Item*
Ein Item beinhaltet alle zu einer Frage assoziierten Informationen. Dies ist nicht nur die Frage selbst sondern auch Zusatzinformationen darüber, wie die Frage dargestellt und bewertet werden soll und andere Informationen wie Lösungshilfen, Feedback und Ähnliches. Ein Item besteht aus mindestens einem Paar von Questions und Responses. Der Fragetyp ist dabei nicht abhängig vom Typ der Frage, sondern vom Typ der Antwort (Responses-Type). Auch beliebig zusammengesetzte Typen sind so möglich.
- *Section*
Items werden zusammengefasst zu Sektionen. Sektionen können ihrerseits selbst Sektionen enthalten um so eine beliebig verschachtelte Struktur zu erzeugen.
- *Assessment*
Ein Assessment besteht aus mehreren Sektionen. Ein Test ist eine Instanz eines Assessments. Die Datenstruktur Assessment - Section - Item wird in dem IMS Modell als ASI Struktur bezeichnet. Die Abbildung 4.3 zeigt die möglichen Verschachtelungen mit der ASI Struktur.
- *Object Bank*
Gleichartige Assessments können zu einer Object Bank zusammengefasst werden. Diese Neuerung ist in der soeben erschienenen Version 1.2 hinzugekommen.
- *Participant*
Um Namenskollisionen zu verhindern und so offen wie möglich zu sein wird in dem IMA QTI nicht von Student oder Benutzer sondern von Participant gesprochen.

An der IMS QTI wurde in den vergangenen Monaten viel verändert. So hat es während dieser Arbeit zwei neue Versionen gegeben und die Dokumente wurden neu arrangiert oder neue sind hinzugekommen. Gegenüber den anderen IMS Spezifikationen liefert die IMS QTI nicht nur ein Informationsmodell und die XML Bindings, sondern geht tiefer in die Details der Implementation hinein bis hin zu konkreten Objektmodellen.

Die IMS Question & Test Interoperability Spezifikation besteht in der Version 1.2 aus den folgenden Dokumenten:

- IMS Question & Test Interoperability Overview [SSBL01a]
- IMS Question & Test ASI Best Practice Guide
- IMS Question & Test ASI XML Binding Specification
- IMS Question & Test ASI Outcomes Processing Specification
- IMS Question & Test ASI Selection and Ordering Specification
- IMS Question & Test Results Reporting Best Practice and Implementation Guide
- IMS Question & Test Results Reporting XML Binding Guide
- IMS Question & Test Results Reporting Information Model

4.2.4 IMS Learner Information Package Specification

Die IMS Learner Information Package Specification, kurz LIP genannt, spezifiziert das Datenmodell des Lernenden. Es handelt sich dabei um Daten wie auch um die dazugehörenden Metadaten. Ziel ist wiederum, den Austausch von Daten zwischen verschiedenen Systemen bewerkstelligen zu können.

Die LIP Datenstruktur ist in 11 Hauptkategorien eingeteilt und so aufgebaut, dass die einzelnen Elemente unabhängig vom Ganzen verpackt und versendet oder gespeichert werden können. Bei einem neuen Eintrag muss so nicht der ganze Datensatz übertragen werden. Die Datenstruktur sieht die folgenden Kategorien vor:

- *Identification*
Biographische und demographische Daten.
- *Goal*
Lernziele oder auch Karriereziele in einem Geschäftsumfeld.
- *Qualifications, Certifications and Licenses (qcl)*
Nachweise erbrachter Leistungen.
- *Activity*
Aktivitäten welche im Zusammenhang mit dem Lernen stehen. Darunter werden auch formale und informale Ausbildungen, Trainings, Erfahrung etc. verstanden.
- *Transcript*
Ein spezieller Datensatz mit einer Zusammenfassung der bisherigen akademischen Laufbahn.
- *Interest*
Persönliche Interessen, Hobbies und Freizeitaktivitäten.
- *Competency*
Fähigkeiten und Wissen.
- *Affiliation*
Zugehörigkeit zu Organisationen.
- *Accessibility*
Sprache, Behinderungen, Lernstil etc.
- *Securitykeys*
Passwörter und andere Sicherheitsschlüssel.
- *Relationship*
Wird verwendet, um Beziehungen zwischen den obigen Elementen darzustellen.

Die Metadaten eines Elementes machen Angaben über die Zeit der Erstellung und der Gültigkeitsdauer, über die Identität (z.B. die ID und eine URI der Quelle des Elementes) und über den notwendigen Schutz der Daten wegen Vertraulichkeit und Integrität der Daten. Die IMS LIP können auf Individuen wie auch auf Organisationen angewendet werden. Für Gruppenstrukturen sind sie jedoch nicht vorgesehen. Dies wird in der IMS Enterprise Specification behandelt (Vgl. Kapitel 4.2.5).

Die Spezifikation behandelt zudem nicht, mit welchen Protokollen solche LIP-basierten Datensätze ausgetauscht werden sollen. Das Simple Object Access Protocol (SOAP) wird aber explizit als Messagingprotokoll genannt. Dieses ist für den Austausch von XML basierten Nachrichten optimiert und würde sich daher sehr gut dafür eignen. Ebenso fehlt eine eigene Anfragesprache für diese Datenstruktur.

Die folgenden Dokumente legen die IMS Learner Information Package Spezifikation fest:

- IMS Learner Information Package Information Model [STR01a]
- IMS Learner Information Package XML Binding Specification [STR01c]
- IMS Learner Information Package Best Practices and Implementation Guide [STR01b]

4.2.5 IMS Enterprise Specification

Die IMS Enterprise Specification zielt darauf ab, den Datenaustausch innerhalb einer Organisation zwischen dem Lernmanagementsystem und verschiedenen anderen Systemen zu regeln. Fragen der Sicherheit und der Datenintegrität sind daher nicht Inhalt der Spezifikation. Als andere Systeme kämen beispielsweise das SAP Campus oder ein elektronisches Bibliothekssystem in Frage. Die folgenden Dokumente definieren die Spezifikation. Die Verteilung und Gewichtung der Inhalte ist analog zu den bisher besprochenen IMS Spezifikationen:

- IMS Enterprise Information Model [CVA99b]
- IMS Enterprise XML Binding Specification [CV99]
- IMS Enterprise Best Practices and Implementation Guide [CVA99a]

Die IMS Enterprise Specification dient dem IEEE 1484.8 Enterprise Interface Standard als Grundlage. Andererseits stammen Elemente der Spezifikation ihrerseits wiederum aus dem IEEE 1484.2 Standard der Learning Model Working Group. Vergleichen sie hierzu auch Kapitel 4.1. Die Spezifikation ist mit der IMS Learning Resource Meta-Data (Kapitel 4.2.1) sowie der IMS Learner Information Package Spezifikation (Kapitel 4.2.4) kompatibel. Die IMS Enterprise Spezifikation versteht sich als Subset der IMS Learner Information Package Spezifikation.

Durch das Verwenden dieser Spezifikation soll Interoperabilität zwischen den folgenden vier Prozesskomponenten erreicht werden:

- *Personal Profile Data Maintenance*
Wenn eine Benutzerin im Verwaltungssystem der Universität ihre Adresse ändert, muss diese Änderung an andere Systeme wie zum Beispiel eine Lernumgebung propagiert werden. Der umgekehrte Weg ist ebenso möglich.
- *Group Management*
Das Gruppenmanagement könnte Auswirkungen auf andere Systeme haben. Eine Änderung in einer Gruppe oder Klasse müsste auf anderen Systemen sichtbar gemacht werden können. Ein Institut könnte zum Beispiel eine eigene Groupwarelösung haben, welche jedoch mit der Lernumgebung synchronisiert wird.

- *Enrollment Management*
Es ist vorstellbar, dass Einschreibungen in Kurse zwar über ein Lernsystem vollzogen werden, dann aber mit einem anderen System weitergearbeitet wird. Beispielsweise muss das Einschreiben von APS Veranstaltungen an das SAP System kommuniziert werden können.
- *Final Result Processing*
Hier handelt sich um das Austauschen von Resultaten, welche ganze Gruppen betreffen.

Das Datenmodell der Spezifikation ist einfach gehalten. Es gibt drei Hauptobjekte für die Modellierung der Gruppenbeziehungen:

- *Person*
Dieses Element beschreibt eine einzelne Person.
- *Group*
Dieses Element beschreibt eine Gruppe. Mehrere Arten von Gruppen sind vorstellbar: Ein Kurs, eine Klasse, eine Subgruppe, ein Klub etc. Beziehungen zu anderen Gruppen sind ebenfalls möglich.
- *Group Membership*
Dieses Element beschreibt eine Beziehung einer Person oder Gruppe zu einer Gruppe. Dabei wird eine Rolle festgelegt, welche mit dieser Beziehung assoziiert ist. Subrollen sind erlaubt, beispielsweise um eine Vertretung einer Person darstellen zu können.

Bezüglich der Übermittlung der Daten und des dafür notwendigen Interfaces nennt die Spezifikation zwei Möglichkeiten: Snapshot Interface und Event Driven Interface. Mit der ersten Möglichkeit ist gemeint, dass eine komplette Abbildung der gesamten Struktur übermittelt werden soll. Mit der zweiten Möglichkeit hingegen sollen nur die Änderungen übertragen werden. Beide Schnittstellen haben ihre Vor- und Nachteile. Es dürfte sinnvoll sein, beide Varianten zu implementieren.

4.3 SCORM

Das Sharable Content Object Reference Model, kurz SCORM genannt, ist das Ergebnis der vom Amerikanischen Verteidigungsministerium Department of Defense⁹ (DoD) lancierten *Advanced Distributed Learning initiative*¹⁰ (ADL). ADL erarbeitete high-level Anforderungen an Lerninhalt wie Wiederverwendbarkeit, leichter Zugang mit einfachen Technologien (Webbrowser), lange Haltbarkeit ohne Neukodierung und Interoperabilität und Austauschbarkeit zwischen verschiedenen Plattformen. Das SCORM beschreibt ein Referenzmodell, wie diese Anforderungen befriedigt werden können. Glücklicherweise ist SCORM nicht wieder ein neuer Standard sondern baut auf den bestehenden Standards und Spezifikationen von IEEE, IMS und AICC¹¹ auf. Das SCORM führt die verschiedenen Standards und Spezifikationen zusammen und zeigt, wie diese in einem Gesamtkontext als Paket angewendet werden können,

⁹ <http://www.defenselink.mil>

¹⁰ <http://www.adlnet.org>

¹¹ Die AICC Standards werden in dieser Arbeit nicht weiter besprochen. Mehr Informationen finden sie unter <http://www.aicc.org>

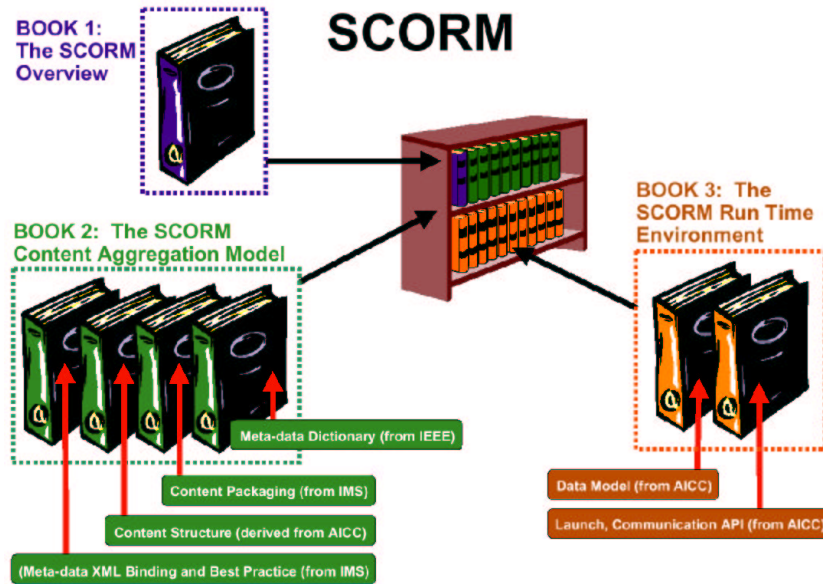


Fig. 4.4: SCORM Overview (Quelle: The SCORM Overview 1.2 [D⁺01b])

damit plattformübergreifende Lerninhalte effektiv möglich werden.

Die Abbildung 4.4 zeigt eine Übersicht über den Inhalt des SCORM, die Aufteilung in die beiden Komponenten *Content Aggregation Model* und *Run-Time Environment* sowie den Zusammenhang zu anderen Standards und Spezifikationen, welche die Grundlage für SCORM bilden. Die folgenden Dokumente definieren das Referenzmodell:

- ADL SCORM: The Overview[D⁺01b]
- ADL SCORM: The SCORM Content Aggregation Model[D⁺01a]
- ADL SCORM: The SCORM Run-Time Environment[D⁺01c]

Neben diesen Dokumenten bietet ADL auch eine Conformance Test Suite um eine Implementation testen zu können. Es handelt sich um eine kleine Implementation von SCORM, so dass Lerninhalte und auch das ganze Referenzmodell getestet werden kann. Um die Software zu benutzen muss man sich bei ADL zuerst registrieren. In einer Conformance Matrix wird zudem genau festgehalten was zur Erreichung der verschiedenen Stufen der SCORM-Konformität geleistet werden muss. Ein weiteres Dokument regelt, wie eine Zertifizierung erreicht werden kann.

4.3.1 SCORM Content Aggregation Model

Während IMS mit der IMS Learning Resource Meta-Data Specification lediglich die Metadaten von beliebigen Lerninhalten definiert und die IMS Content Packaging Specification nur das Packen der Daten für den Transport und den Austausch beschreibt, geht das SCORM mit dem Content Aggregation Model einen Schritt weiter. Mit dem Content Structure Format (CSF) werden die einzelnen Elemente der Lerninhalte in

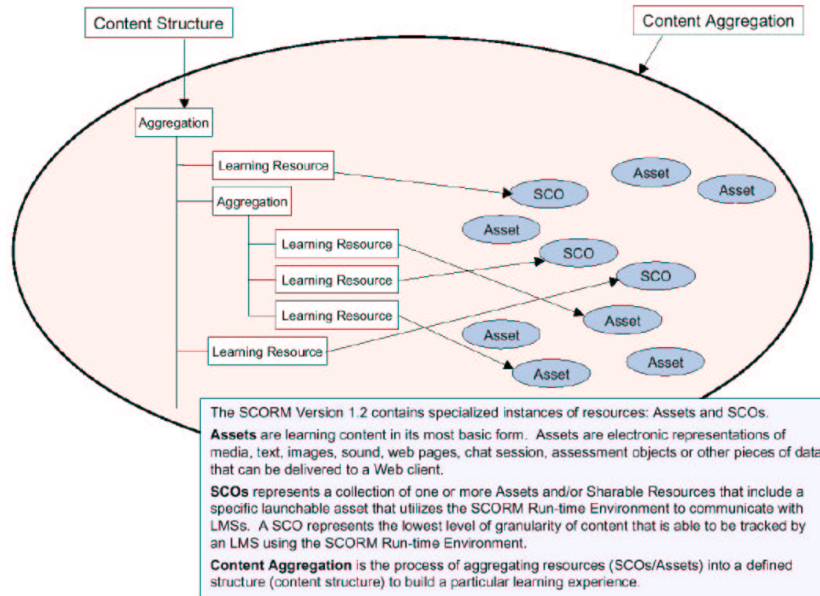


Fig. 4.5: SCORM Content Aggregation (Quelle: SCORM Content Aggregation Model 1.2 [D+01a])

einen Zusammenhang gebracht, oder anders ausgedrückt, sie werden zu einer Einheit assembliert. Man kann sich dies als eine Art Inhaltsverzeichnis der Lerninhalte vorstellen. Dieses CSF basiert auf den *Spezifikationen der Aviation Industry Computer-Based Training Committee (AICC) Computer Managed Instruction (CMI)*. Das SCORM CSF hat das CSF von AICC als Basis übernommen und erweitert dieses durch eigene Elemente. Die XML Definitionen sind ebenfalls Bestandteil des SCORM.

Um dies zu verdeutlichen hier ein Beispiel: Ein Onlineskript, welches auf normalen HTML-Seiten basiert, kann mit den IMS Metadaten beschrieben und gemäss der IMS Packaging Spezifikation verpackt und so ausgetauscht werden. Wie das Skript selbst strukturiert ist, wie navigiert wird etc. ist alles dem Autor überlassen. Die kleinste austauschbare Einheit eines solchen Skripts ist das Skript selbst. Bei SCORM hingegen wird die Struktur des Skriptes mit dem standardisierten CSF festgelegt. Die kleinsten austauschbaren Elemente sind nicht das Skript selbst, sondern die in sich geschlossenen Elemente innerhalb des Skripts, sogenannte *Sharable Content Objects (SCO)*. Somit können wiederverwendbare Lernobjekte erstellt werden, die auf standardisierte Art und Weise zu einem Ganzen assembliert werden können.

Die Abbildung 4.5 zeigt einen Überblick über das SCORM Content Aggregation Model. Lerninhalte gemäss SCORM sind wie folgt aufgebaut:

1. Asset

Das kleinste Element ist ein Asset. Dies kann ein Bild, eine Tondatei, ein XML Fragment, ein HTML Fragment, eine JavaScript Funktion oder Ähnliches sein.

2. *Tagged Asset*

Wird ein Asset mit Asset Meta-Data¹² beschrieben, so wird es ein Tagged Asset genannt.

3. *Sharable Content Object (SCO)*

Dies ist die kleinste selbständige Einheit. Sie kommuniziert über das SCORM Run-Time Environment mit dem Lernsystem und ist unabhängig vom Skript als Ganzes. Sie kennt das Ziel des Skriptes nicht und kann auch in einem anderen Zusammenhang verwendet und somit ausgetauscht werden. Ein minimales Set des API Adapters des Run-Time Environments muss implementiert sein um das SCO überhaupt aufrufen zu können.

4. *Tagged SCO*

Wird ein SCO mit SCO Meta-Data¹³ beschrieben, so wird es Tagged SCO genannt. Die Metadaten erlauben Suchfunktionen und das Auffinden von SCO's.

5. *Content Aggregation*

Das Content Structure Format ist eine hierarchische Anordnung der verschiedenen Content Aggregationen und SCO's um diese zu einer Einheit zu assemblieren. Es definiert die Navigation und die Reihenfolge zwischen den Elementen. Sehr wichtig zu verstehen ist, dass ein SCO nicht ein anderes SCO aufrufen darf, dies muss extern geschehen und wird daher im SCF festgelegt. Die Reihenfolge der SCO ist abhängig vom Kontext. Ein SCO ist eine kontextunabhängige Einheit, sonst wäre sie nicht wiederverwendbar. Über das API kann ein SCO mit dem Lernsystem kommunizieren und beispielsweise mitteilen, dass der Lerner diesen Lernschritt nicht begriffen hat. Es ist dann aber die Aufgabe des Lernsystems, dem Lernenden im nächsten Schritt ein entsprechend leichteres SCO zu präsentieren und nicht diejenige des SCO selbst. Ein SCO kennt nur sich selbst, es kann also nicht auf andere SCO referenzieren. Natürlich kann ein SCO selbst auch Verzweigungen beinhalten, diese sind dann aber auf das SCO begrenzt und dürfen nicht andere SCO's aufrufen.

6. *Tagged Content Aggregation*

Wird eine Content Aggregation mit Content Aggregation Meta-Data¹⁴ beschrieben, so wird sie Tagged Content Aggregation genannt. Die Metadaten erlauben Suchfunktionen und das Auffinden von Content Aggregationen. Dies ist nicht nur notwendig um eine Content Aggregation als Lerner zu verwenden, sondern auch für die Autoren um Content Elemente wiederverwenden zu können.

7. *Content Packaging*

Eine beliebig verschachtelte Content Aggregation wird am Schluss zu einem Packet geschnürt um dann ausgetauscht werden zu können. Die Content Structure¹⁵ definiert die hierarchische Organisation der Teilelemente und enthält Angaben über die Navigation und Reihenfolge der Darstellung der Daten. Das Packaging erfolgt gemäss der IMS Packaging Specification¹⁶. Die genaue Anwendung

¹² in SCORM 1.1 wurde dies noch Raw Media Meta-Data genannt

¹³ in SCORM 1.1 wurde dies noch Content Meta-Data genannt

¹⁴ in SCORM 1.1 wurde dies noch Content Meta-Data und Course Meta-Data genannt. Eine Content Aggregation wurde Block oder Content Aggregation genannt.

¹⁵ In SCORM Version 1.1 wurde hier der Begriff Content Structure Format (CSF) verwendet. Da das SCORM seit Version 1.2 kompatibel mit der IMS Packaging Spezifikation ist, sollte das CSF nicht mehr verwendet werden

¹⁶ vgl. Kapitel 4.2.2

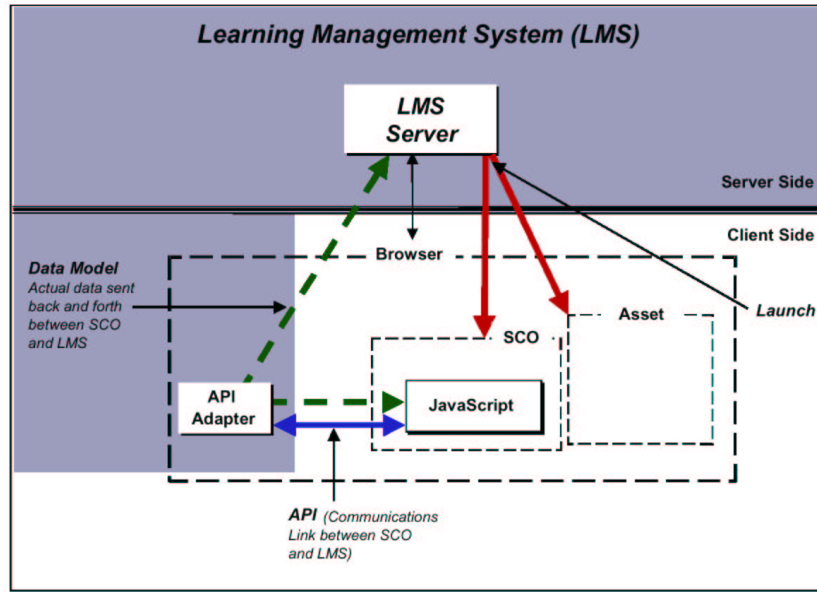


Fig. 4.6: Das SCORM Run-Time Environment (Quelle: The SCORM Run-Time Environment 1.2 [D⁺01c])

der IMS Packaging Spezifikation wird in den SCORM Content Packaging Application Profiles beschrieben.

Die Asset, SCO und Content Aggregation Metadaten basieren alle auf der IMS Learning Resource Specification¹⁷, welche ihrerseits wiederum auf dem IEEE LTSC LOM¹⁸ Standard beruht. Somit ist SCORM IMS-kompatibel, gibt aber gewisse Richtlinie zur einschränkenden Anwendung der Metadaten auf den verschiedenen Stufen des SCORM Content Modells an.

4.3.2 SCORM Run-Time Environment

Das SCORM Run-Time Environment definiert eine Umgebung, in welcher der Benutzer SCORM kompatible Inhalte betrachten kann. Es geht dabei nicht um die Darstellung der Daten, diese wird vielleicht in späteren Versionen ebenfalls in SCORM behandelt, sondern nur um die Kommunikationsmechanismen zwischen den Lerninhalten und der Lernumgebung. Die Abbildung 4.6 zeigt eine Übersicht. Drei Aspekte sind hervorzuheben: der Launch, das API und das Datenmodell. Das SCORM Run-time Environment basiert auf den AICC CMI001 Guidelines for Interoperability.

1. Launch

Unter dem Launch versteht man die Methode des Lernsystems, Lerninhalte zu starten und dem Benutzer zugänglich zu machen. Dies geschieht immer vom Lernsystem her, ein Content Objekt kann kein anderes Content Objekt starten.

¹⁷ vgl. Kapitel 4.2.1

¹⁸ vgl. Kapitel 4.1

Der Launch eines Objektes wird gestartet aufgrund einer in der Content Aggregation definierten Reihenfolge, aufgrund einer direkten Wahl des Lernenden oder durch die adaptive Auswahl des am besten geeigneten Objektes aufgrund des Verhaltens des Lernenden.

2. API

Das SCORM API (Application Programming Interface) stellt die Kommunikationsschnittstelle zwischen Lerninhalten und Lernumgebung dar. Da sich die Lerninhalte nach dem Start auf der Seite des Clients befinden, muss das API auch auf der Seite des Clients implementiert sein, damit die Lerninhalte überhaupt darauf zugreifen können. Das API wird beim Client durch einen API-Adapter zur Verfügung gestellt. Wie der API-Adapter mit dem Lernsystem kommuniziert ist Frage der Implementation des Adapters. Dieser wird vom Lernsystem zur Verfügung gestellt. In den SCORM Unterlagen sind Beispiele enthalten, wie ein solcher Adapter mit einem Java Applet oder mittels JavaScript implementiert werden könnte. Das API muss über ECMAScript (JavaScript) angesprochen werden können. Es erfüllt drei grundlegende Funktionen:

- *Execution State*
LMSInitialize(): Initialisierung einer Verbindung vom Content zum Lernsystem.
LMSFinish(): Beendet die Verbindung. Alle Daten sind gesendet.
- *State Management*
LMSGetValue(): Anfrage eines Parameters vom Lernsystem (z.B. Name Lerner, Sprachpräferenz).
LMSSetValue(): Senden von Daten vom Client an das Lernsystem (z.B. Resultat eines JavaScript-basierten Tests).
LMSCommit(): Alle mit LMSSetValue() gesetzten Daten sollen sofort an das Lernsystem übermittelt werden.
- *Data Transfer*
LMSG GetLastError(): Nachfrage der letzten Fehlermeldung welche durch den letzten API-Aufruf verursacht wurde. Der Fehler wird als Zahlencode ausgegeben.
LMSG GetErrorString(): Übersetzung des Zahlencodes in eine Textnachricht.
LMSG GetDiagnostic(): Nachfrage von herstellerspezifischen Zusatzangaben zu einer Fehlermeldung.

3. Data Model

Das Datenmodell definiert, welche Elemente mit der *Get* und *Set* Funktion abgefragt respektive gespeichert werden können. Verschiedene Gruppen von Informationen können ausgetauscht werden: Daten über das Lernobjekt und den Status, Informationen über den Lernenden (Name, Sprachpräferenzen) und auch Resultate in Form von Punkten.

Problematisch an dem SCORM Run-Time Environment ist die Tatsache, dass der API-Adapter auf der Clientseite implementiert sein muss und mit JavaScript direkt angesprochen werden kann. Bei der Abfrage von Daten spielt das keine Rolle, der Benutzer kann nur Informationen über sich selbst oder über den Lerninhalt erhalten. Beim Speichern hingegen bleiben viele Fragezeichen offen. Da die JavaScript-Funktionen mit

entsprechenden Kenntnissen auch manuell vom Benutzer angesprochen werden können, ist es möglich, manipulierte Daten an das Lernsystem zu senden. Es gibt keine Möglichkeit für das System die Echtheit zu überprüfen. Im Rahmen eines Self Assessments ist dies freilich kein Problem und der Anreiz eines Lernenden zu betrügen ist kaum gegeben. Wenn auf diese Art und Weise jedoch Punkte erreicht werden können, welche zum Erhalt eines Testats beitragen, so muss die Situation anders bewertet werden. Solche Arten von Tests müssen auf der Serverseite ausgewertet werden.

5. GROBANALYSE

5.1 Anforderungsbeitragende

In der Abbildung 5.1 sind die identifizierten Anforderungsbeitragenden der Lernplattform OLAT 2 in einer Übersicht dargestellt. Neben den durch Personen repräsentierten Anforderungsbeitragenden gibt es auch eine Gruppe von abstrakten Anforderungsbeitragenden wie Informationssysteme oder Standards. Im Folgenden werden die einzelnen Anforderungsbeitragenden kurz beschrieben.

5.1.1 Reale Anforderungsbeitragende

Mit realen Anforderungsbeitragenden sind Personen gemeint, welche Anforderungen an das neue System stellen. Sie werden hier unabhängig von der Wichtigkeit aufgelistet und kurz beschrieben:

- *Benutzer:*
Der Benutzer steht für jede Person welche mit dem System interagiert. Er spezifiziert allgemeine Funktionen, welche allen Teilnehmern jederzeit zur Verfügung stehen.
- *Lerner:*
Der Lerner ist diejenige Person, welche einen Kurs als Teilnehmer besucht und damit Wissen erwerben möchte. Es wurde hier absichtlich nicht der Begriff *Student* verwendet, da Student eine organisatorische Bezeichnung für die Stellung einer Person innerhalb der Universität ist. Der Tatbestand des Lernens ist hingegen nicht auf Studierende beschränkt. In den meisten Fällen beziehen sich die aus dieser Perspektive abgeleiteten Anforderungen jedoch auf die Funktionalität, welche einem Studierenden zu Verfügung stehen soll.
- *Coach:*
Als Coach werden Tutoren oder andere Begleitpersonen verstanden, welche im direkten Kontakt zu den Lernenden stehen. Der Begriff Coach ist offener als derjenige des Tutors, welcher fast immer in Verbindung mit Studierenden höheren Semesters in Verbindung gebracht wird. Ein Coach kann aber auch ein Professor oder ein Assistent sein.
- *Professor:*
Der Professor ist für einen Kurs inhaltlich und rechtlich verantwortlich, oft aber nicht der eigentliche Organisator¹.

¹ Ein Kurs wird häufig von einem Assistenten der Professorin organisiert. In diesem Papier wird in dem Zusammenhang von Kursbetreuer gesprochen.

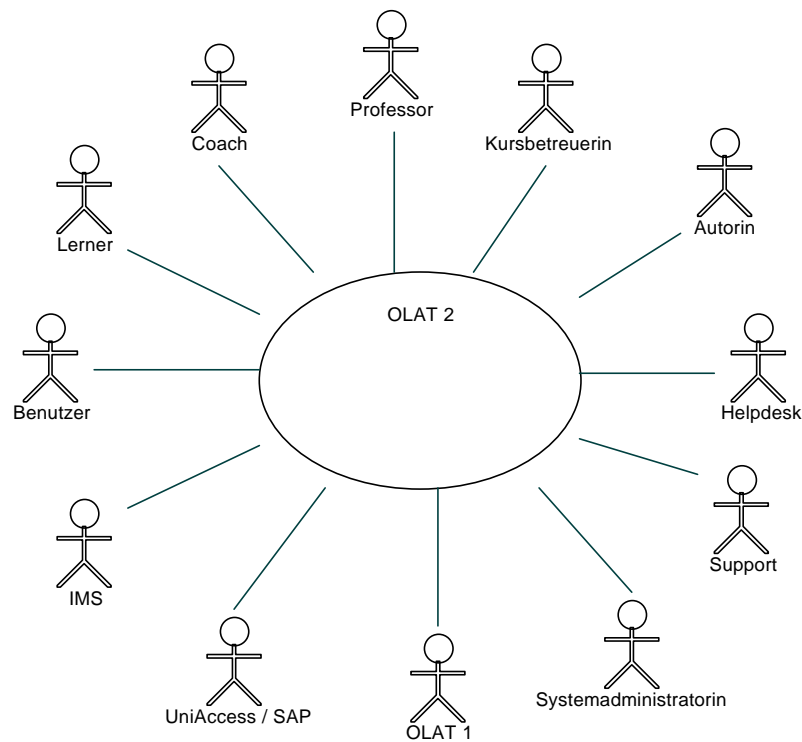


Fig. 5.1: Anforderungsbeitragende des OLAT 2 Systems

- Kursbetreuerin:**
Die Kursbetreuerin ist die eigentliche Organisatorin eines Kurses. Es könnte eine Assistentin oder eine Professorin sein. Die Kursbetreuerin hat nicht primär Kontakt zu den Lernenden sondern ist für die Koordination eines Kurses zuständig.
- Autorin:**
Die Autorin erstellt die eigentlichen Lerninhalte. Unter Lerninhalten verstehen wir die zur Verfügung stehenden Materialien, ein Onlineskript oder ein interaktiver Onlinetest. Die Autorin ist nicht verantwortlich für einen Kurs oder wie die Lerninhalte in einem Kurs eingesetzt werden. Oft ist die Autorin die selbe Person wie der Professor oder die Kursbetreuerin.
- Helpdesk:**
Der Helpdesk, eventuell als Call-Center organisiert, und der Support haben viele Überschneidungen, doch ist der Support nicht für die Endkunden, die Lernenden, gedacht. Bei Problemen beispielsweise mit dem Loginprozess werden die Lernenden voraussichtlich den Helpdesk in Anspruch nehmen. Dort muss man daher über die notwendigen Informationen verfügen.
- Support:**
Die Supportperson muss Anfragen von den oben genannten Personen beantworten können und auf Systemebene Administrationsfunktionalitäten zur Verfügung stellen.

- *Systemadministratorin:*
Während die Personen des Supports auf inhaltlicher Ebene das System administrieren, sorgt die Systemadministratorin für den Betrieb des Servers. Hier kommen Anforderungen bezüglich spezieller Werkzeuge, welche die notwendigen Arbeiten wie Backup optimal unterstützen zutage.

5.1.2 Abstrakte Anforderungsbeitragende

Im Gegensatz zu den durch Personen repräsentierten Stakeholdern gibt es auch abstrakte Anforderungsbeitragende. Damit sind Systeme, Gesetze oder Standards gemeint, welche ebenfalls Anforderungen an die neue Lernplattform definieren. Meistens sind sie selbst nicht vom System betroffen, das heißt, sie müssen ihrerseits nicht auf Anforderungen des neuen Systems Rücksicht nehmen oder ihre Schnittstellen neu definieren.

- *OLAT 1:*
Wie schon erwähnt soll OLAT 2 kompatibel mit dem alten System sein. Die im alten System erstellten Lerninhalte müssen auch im neuen System zu gebrauchen sein. Daraus entstehen Anforderungen an die Importfilter.
- *UniAccess / SAP:*
Das System UniAccess muss im neuen System ebenfalls berücksichtigt werden und stellt somit Anforderungen an OLAT 2. SAP-Campus wird in Zukunft alle Studierenden und ihre Anrechnungspunkte verwalten. Mit diesen Informationen muss in OLAT 2 transparent umgegangen werden können.
- *IMS:*
Die IMS und andere Standards definieren, wie Daten mit anderen Lernsystemen ausgetauscht werden können. Sie beschreiben XML-Spezifikationen für die Daten und Metadaten. Daraus können wichtige Teile der Datenstrukturen des neuen Systems abgeleitet werden.

5.2 Geschäftsprozesse identifizieren

Die Abbildung 5.2 zeigt die wichtigsten Geschäftsprozesse des Lernsystems in einem Überblick. Die Prozesse sind als Use-Cases veranschaulicht und mit dem Stereotyp *workflow* als Geschäftsprozesse gekennzeichnet. Die Geschäftsprozesse widerspiegeln auf einem hohen Abstraktionsniveau die wichtigsten Aktivitäten des zukünftigen Systems. Die Prozesse bestehen dabei aus vielen Teilprozessen, welche hier jedoch vernachlässigt werden. Sie sind in den folgenden Aktivitätsdiagrammen sichtbar. Es wurde versucht, die Prozesse visuell bereits hier in Administrationsprozesse auf der linken Seite und Benutzerprozesse auf der rechten Seite zu unterteilen. Es ist klar, dass sich diese Prozesse gegenseitig beeinflussen. Beispielsweise kann sich ein Student nicht in eine Klasse einschreiben, bevor nicht eine solche Klasse in der Klassenverwaltung für eine Einschreibung freigegeben wurde. Diese wechselseitigen Beziehungen werden später herausgearbeitet.

Um das System herum sind die an den Prozessen beteiligten Benutzergruppen dargestellt. Die zuvor als abstrakt bezeichneten Akteure sind hier zusätzlich mit einem Rechteck als eigene Systeme gekennzeichnet. Es sind dies das bisherige OLAT System,

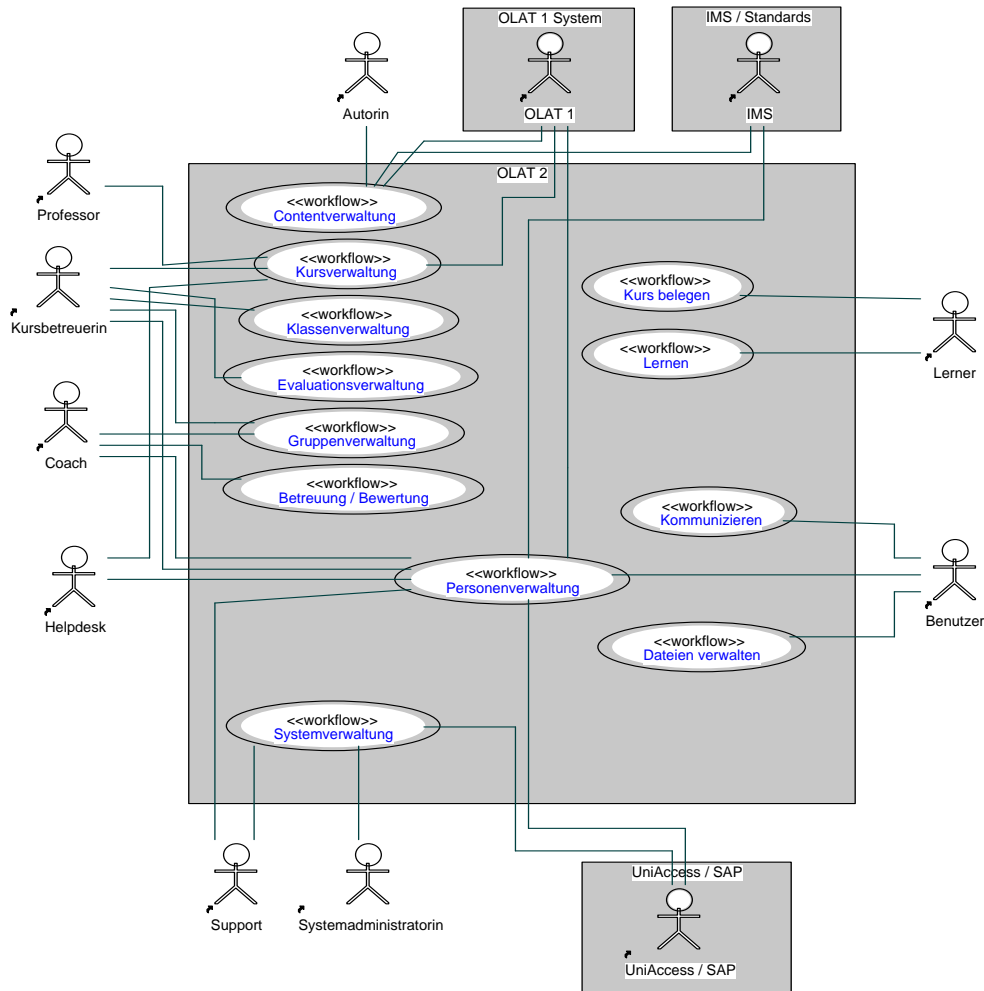


Fig. 5.2: Die wichtigsten Geschäftsprozesse im Überblick

welches dem neuen System auf Bedarf Daten liefern muss, die IMS Standards, welche durch den Import von Daten selbständig Prozesse auslösen können, und die restlichen Informationssysteme der Universität, wie beispielsweise UniAccess oder das SAP Verwaltungssystem, welche hauptsächlich in den Prozessen der Personenverwaltung zu synchronisieren sind.

Im Folgenden werden die einzelnen Geschäftsprozesse als Aktivitätsdiagramme aufgezeichnet und grob beschrieben. Sie bilden die Ausgangslage für die Identifikation der Geschäftsanwendungsfälle.

5.2.1 Contentverwaltung

In der Abbildung 5.3 ist der Geschäftsprozess *Contentverwaltung* abgebildet. *Content* wird im Folgenden als Synonym für *Lerninhalt* oder auch *Online-Skript* verwendet.

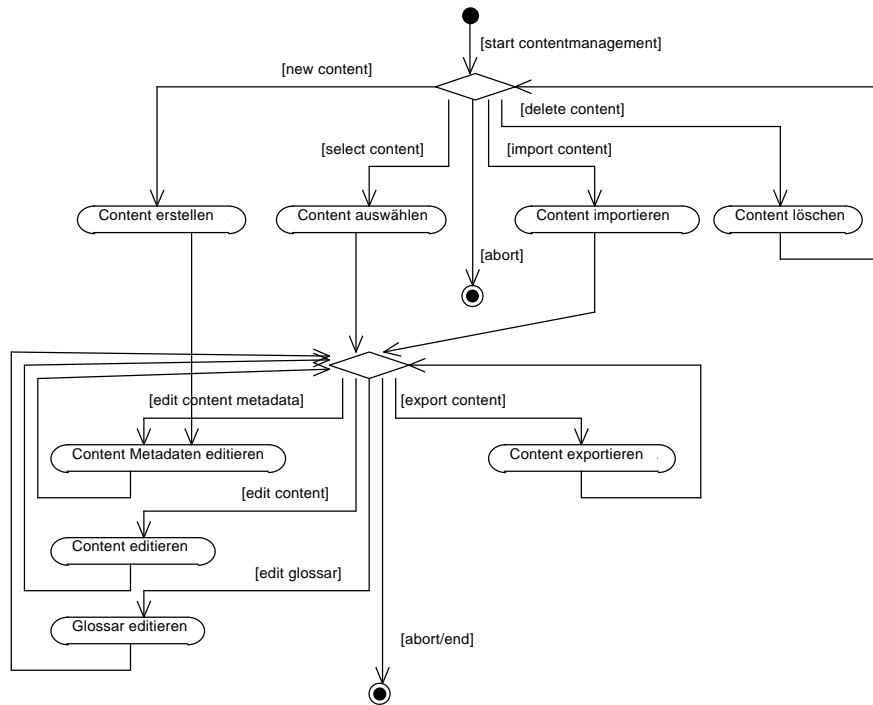


Fig. 5.3: Übersicht über den Geschäftsprozess Contentverwaltung

Durch die IMS und SCORM Standards ist genau definiert, wie ein solcher Inhalt aufgebaut und in XML-Dateien verpackt werden soll. Als Content gilt demnach sowohl normaler Text in elektronischer Form als auch automatisierte Online-Tests, aufgrund derer eine Benotung stattfinden kann. Die Hierarchiestufe von Content ist hier nicht weiter definiert. Das SCORM kennt als kleinste Einheit die SCO's (Sharable Content Object). Dies könnte zum Beispiel ein Bild oder ein Textfragment sein. Mehrere SCO werden zu einem Block² verarbeitet, wobei ein Block auch andere Blöcke beinhalten kann. Mehrere Blöcke können dann zu einem Content zusammengefasst werden und bilden so eine in sich geschlossene Einheit. Ob diese Einheit ein Kapitel oder ein ganzes Buch darstellt ist jedoch nicht weiter vorgegeben.[D⁺01b]

Mit Contentmanagement ist der Prozess des Erstellens und Ändern von Content gemeint. Initiale Aktionen sind daher das Auswählen von konkretem Content aus dem Pool der verfügbaren Contentelemente, für welche man die entsprechende Berechtigung hat. Sonderfälle stellen hier der Import von neuem Content, das manuelle Erstellen von neuem Content sowie das Löschen von bestehendem Content dar.

Nach einem Import kann man den Content bearbeiten wie wenn man diesen aus dem bestehenden Contentpool ausgewählt hätte. Als Import Dateien kommen Dateien gemäss IMS/SCORM in Frage sowie ein durch OLAT 1 definiertes Format. Content kann wiederum im IMS/SCORM Format exportiert werden.

² In SCORM Version 1.2 wird der Begriff Block nicht mehr verwendet. Es wird neu von Content Aggregation gesprochen. Der Begriff Block wird hier aus Gründen der Verständlichkeit trotzdem verwendet.

Wird neuer Content definiert, so müssen als erstes die Metadaten genauer spezifiziert werden, dann kann normal mit dem Content gearbeitet werden.

Content verfügt über Metadaten, welche Informationen wie Autor, Titel, Verwendungszweck, vorgesehene Bearbeitungsdauer und Ähnliches beinhalten, analog zu den Metadaten Elementen des SCORM. Ist ein Content ausgewählt, so können die Metadaten geändert werden. Ebenso kann der Content selbst geändert werden. Dies selbst ist wiederum ein Prozess mit vielen Unterprozessen. Hier werden gemäss der SCORM-Terminologie SCO und Blöcke definiert und in eine Reihenfolge gebracht, einer Art Inhaltsverzeichnis. Dazu müssen die einzelnen Medienelemente auf den Server geladen oder in ein Formular eingetippt werden können.

Die Prozesse der Contentverwaltung werden von einer Autorin initiiert. Diese Rolle kann auch von einer Kursbetreuerin oder einem Professor wahrgenommen werden. Sie ist grundsätzlich jedoch unabhängig davon. Content wird auch nicht für eine spezifische Veranstaltung hergestellt, sondern ist unabhängig davon im System verfügbar, so dass verschiedene Kurse auf den selben Content zugreifen können. Veränderungen im Content werden durch Versionsinformationen in den Metadaten festgehalten.

5.2.2 Kursverwaltung

Der Prozess der Kursverwaltung ist in der Abbildung 5.4 illustriert. Im System OLAT 1 wurde in diesem Zusammenhang von *Lehrveranstaltung* gesprochen. Um das System nicht an die Terminologie der Universität Zürich zu binden wurde hier der weiter verbreitete und offene Begriff *Kurs* gewählt. Analog zur Contentverwaltung muss ein Kurs aus den verfügbaren Kursen ausgewählt werden. Ebenfalls gibt es Unterprozesse um ganze Kurse zu löschen, zu importieren oder um neue Kurse zu erstellen. Das Importieren bezieht sich lediglich auf den Kursaufbau. Es müssen Kurse vom OLAT 1 System sowie Kurse des neuen Systems importiert werden können.

Die Metadaten eines Kurses beinhalten dessen Beginn und Ende, Titel und ähnliches. In der Contentstruktur wird zur Verfügung stehender Content in einer Reihenfolge angeordnet und allenfalls Bearbeitungsdaten zugeordnet. Die in OLAT 1 *Lektionen* genannten Einheiten sind ebenfalls hier zu spezifizieren.

Neben dem Kursaufbau, welcher die Metadaten und die Contentstruktur beinhaltet, können auch die Kursdaten exportiert werden. Damit sind alle während eines Kurses anfallenden Daten gemeint: Personen, Testresultate, Bewertungen und vieles mehr. Der Export dient der Archivierung der Daten ausserhalb des Systems. Diese Kursdaten können nicht mehr importiert werden, sie müssen daher in einem standardisierten Format abgespeichert werden, so dass bei Bedarf mit geeigneten Programmen auf die Informationen zugegriffen werden kann. Ein Import ist nur für den Kursaufbau gedacht. Dies stellt die Wiederverwendbarkeit eines Kurses zu einem späteren Zeitpunkt sicher.

Die bisher genannten Prozesse werden hauptsächlich von der Kursbetreuerin ausgeführt. Der Prozess *Kurs überwachen* ist aber auch für den zuständigen Professor interessant. Hier können das Verhalten und die Resultate der Lernenden beobachtet werden um allenfalls Korrekturen an Content oder der Contentstruktur anbringen zu können.

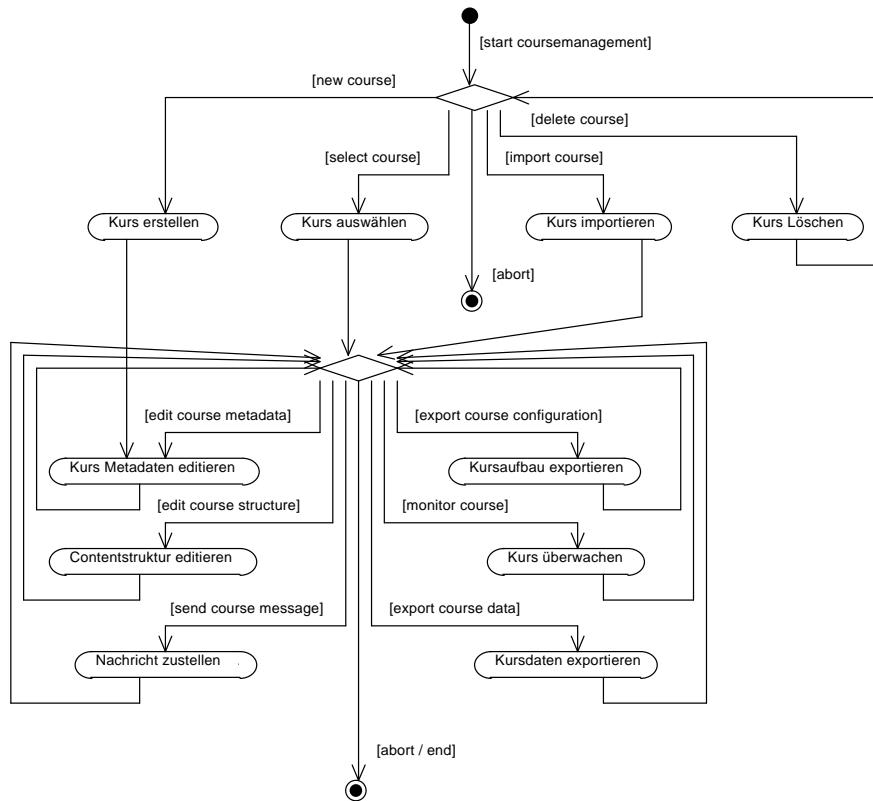


Fig. 5.4: Übersicht über den Geschäftsprozess Kursverwaltung

Mit *Nachricht zustellen* ist gemeint, dass alle Lernenden und Coaches von den Kursbetreuern direkt informiert werden können. Dies soll durch eine Email oder eine News implementiert werden, welche innerhalb der Lernumgebung sichtbar wird.

5.2.3 Klassenverwaltung

Die Abbildung 5.5 zeigt die Prozesse rund um die Klassenverwaltung. Dies ist ein integraler Bestandteil des Lernsystems und kann nicht weggelassen werden, auch nicht bei rein virtuellen Veranstaltungen. In diesem Lernsystem wird Lernen als ein Prozess verstanden, welcher stark mit sozialen Interaktionen verbunden ist, zum Beispiel mit Personen, welche die selben Schritte zur selben Zeit durchlaufen. Daher ist das Vorhandensein einer Klasse immer notwendig.

Klassen sind einem Kurs zugeordnet, daher muss zuerst festgelegt werden für welchen Kurs eine Klasse bearbeitet werden soll. Klassen können gelöscht, erstellt und manipuliert werden. Mit einer speziellen Funktion sollen die aktuellen Belegungen in ein allgemein lesbares Format exportiert werden können. Jede Klasse verfügt über Metadaten wie zum Beispiel die maximale Grösse der Klasse, den Einschreibetermin sowie für wen die Klasse zugänglich ist.

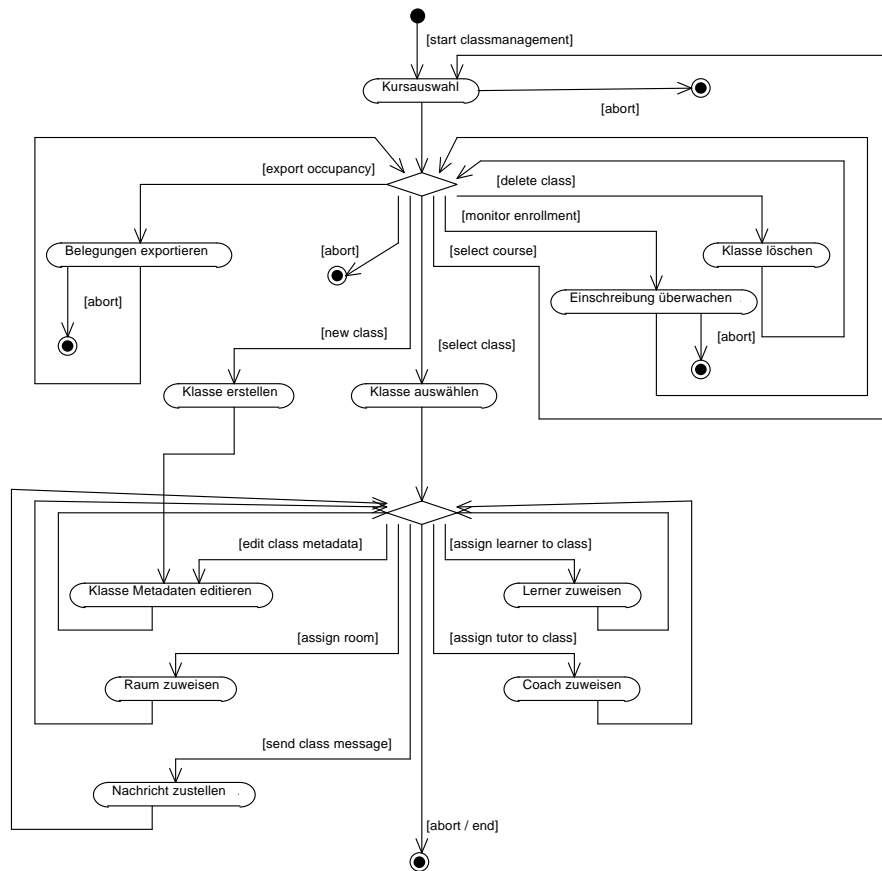


Fig. 5.5: Übersicht über den Geschäftsprozess Klassenverwaltung

Ist die Klasse in den Metadaten als eine Klasse mit physischen Treffen markiert, so kann in einem weiteren Schritt der Klasse ein Raum zugewiesen werden. Dies ist lediglich eine spezielle Metainformation der Klasse. Sie ist jedoch auf zusätzliche Angaben angewiesen und daher als spezieller Prozess extra aufgezeichnet.

Das Zuweisen von Lernenden und Coaches ist ebenfalls ein Teilprozess der Klassenverwaltung, wobei hier zu sagen ist, dass im Normalfall die Lernenden sich selbst in Klassen einschreiben und dies mehr eine Administrationsfunktion ist um später Mutationen durchführen zu können. Trotzdem soll eine Neuzuweisung von Lernenden auch hier möglich sein. Bei Coaches muss zwischen normalen Coaches und Stellvertretern unterschieden werden können, welche im Fall von Krankheiten oder sonstigen Absenzen für einen Coach einspringen können. Diese Rollenzuteilung soll daher zeitlich beschränkt werden können.

Ein wichtiger Teilprozess stellt die Überwachung der Einschreibung dar. Auf einen Blick muss die Kursbetreuerin die Zustände aller Klassen erfassen können um schnell auf Engpässe reagieren und weitere Klassen aufschalten zu können.

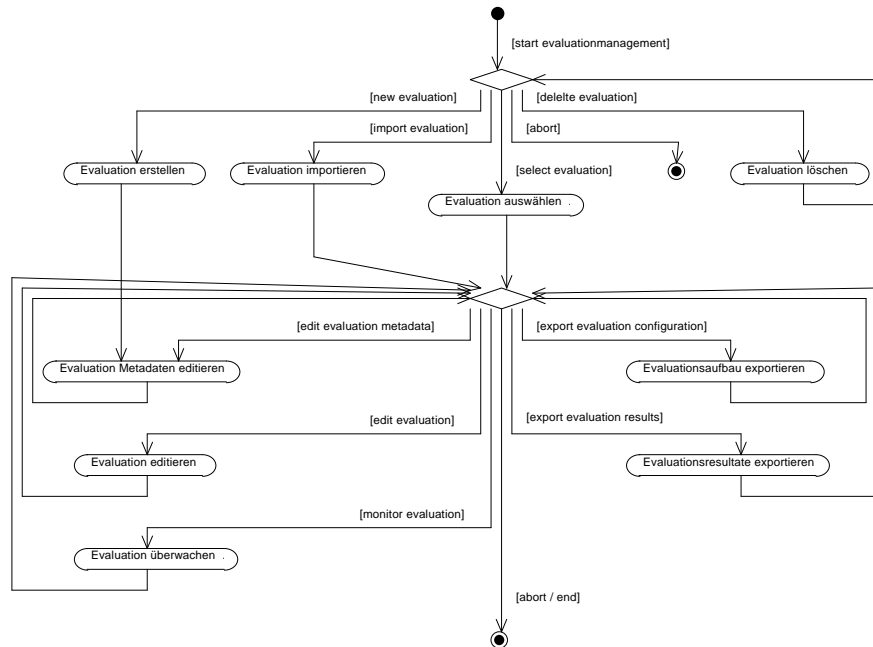


Fig. 5.6: Übersicht über den Geschäftsprozess Evaluationsverwaltung

5.2.4 Evaluationsverwaltung

Unter Evaluation wird bei OLAT ein Fragebogen zur qualitativen Auswertung der Kurse beziehungsweise der Lernplattform selbst verstanden und nicht die Evaluation von Personen im Sinne von Online-Tests. Die Abbildung 5.6 zeigt den Aufbau der Evaluationsverwaltung. Eine Evaluation ist grundsätzlich wie Content unabhängig von einem Kurs und wird von einer Kursbetreuerin wie Content in einen Kurs eingebunden. Mehrere Kurse können die selbe Evaluation verwenden um übergreifende Evaluationen durchzuführen.

Evaluationen können manuell erstellt oder importiert und natürlich auch gelöscht werden. Ist eine Evaluation selektiert, so können wiederum die Metadaten und die Fragen der Evaluation geändert werden. In den Metadaten wird festgelegt, wann eine Evaluation durchgeführt wird und welche Personen die Fragebogen ausfüllen sollen.

Auch hier wird zwischen dem Export des Evaluationsaufbaus und der Evaluationsresultate unterschieden. Exportierte Resultate sind für die Weiterverarbeitung mit Statistikprogrammen wie SPSS gedacht. Der exportierte Aufbau der Evaluation kann für eine Vergleichsevaluation zu einem späteren Zeitpunkt wieder importiert werden. Er enthält daher nur die Fragen und die Metadaten einer Evaluation.

Während eine Evaluation zur Bearbeitung freigegeben ist kann diese überwacht werden, um bei unerwarteten Ergebnissen reagieren oder Tendenzen feststellen zu können. Es versteht sich von selbst, dass die Evaluationsresultate vor dem Export anonymisiert werden.

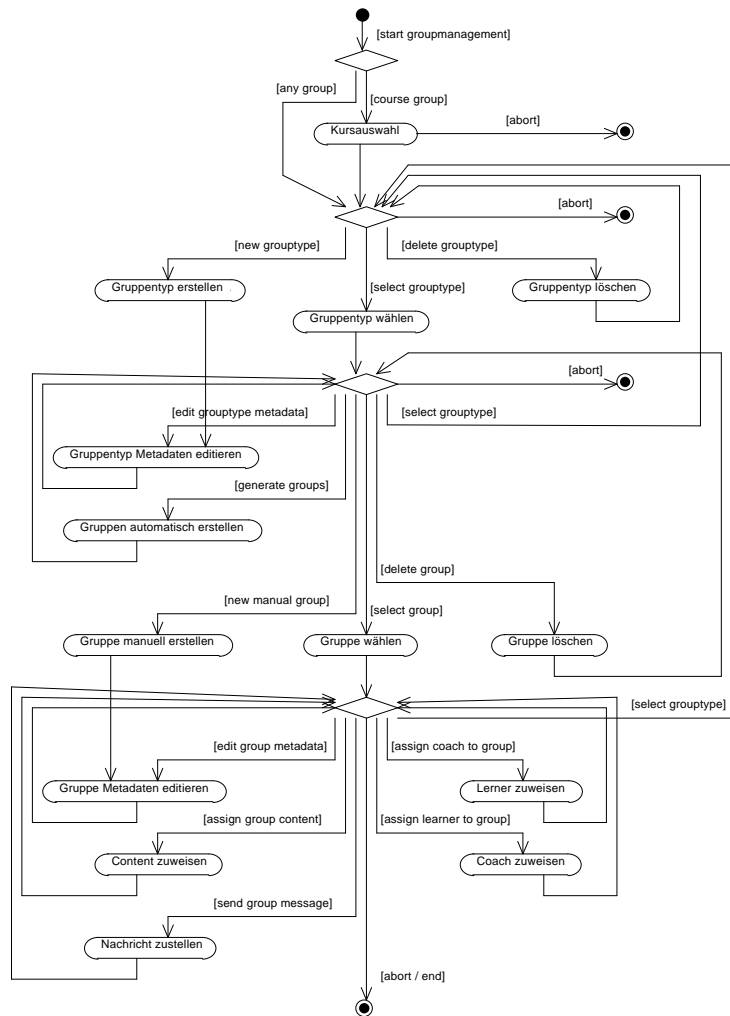


Fig. 5.7: Übersicht über den Geschäftsprozess Gruppenverwaltung

5.2.5 Gruppenverwaltung

Die in der Abbildung 5.7 gezeigten Gruppenverwaltungsprozesse werden vom Coach oder der Kursbetreuerin angestoßen. Gruppen sind nur innerhalb eines Kurses anzutreffen, kursübergreifende Gruppen gibt es nicht. Zuerst wird also der Kurs bestimmt.

In einem zweiten Schritt geht es um die Wahl eines Gruppentyps, beziehungsweise darum, solche zu erstellen oder zu löschen. Ein Gruppentyp ist eine Schablone, nach deren Muster die effektiven Gruppen anschliessend erstellt werden. In den Metadaten eines Gruppentyps werden die folgenden Informationen festgehalten: die anzustrebende Gruppengrösse oder die maximale Anzahl der Gruppen dieses Typs, die Ebene, auf der die Gruppenbildung stattfinden soll (Kurs- oder Klassenebene), Informationen über die Generierung (manuell oder automatisch), ob die Gruppe betreut werden muss, Bezeichnen von Content, welcher den Gruppen zur individuellen Behandlung zugewie-

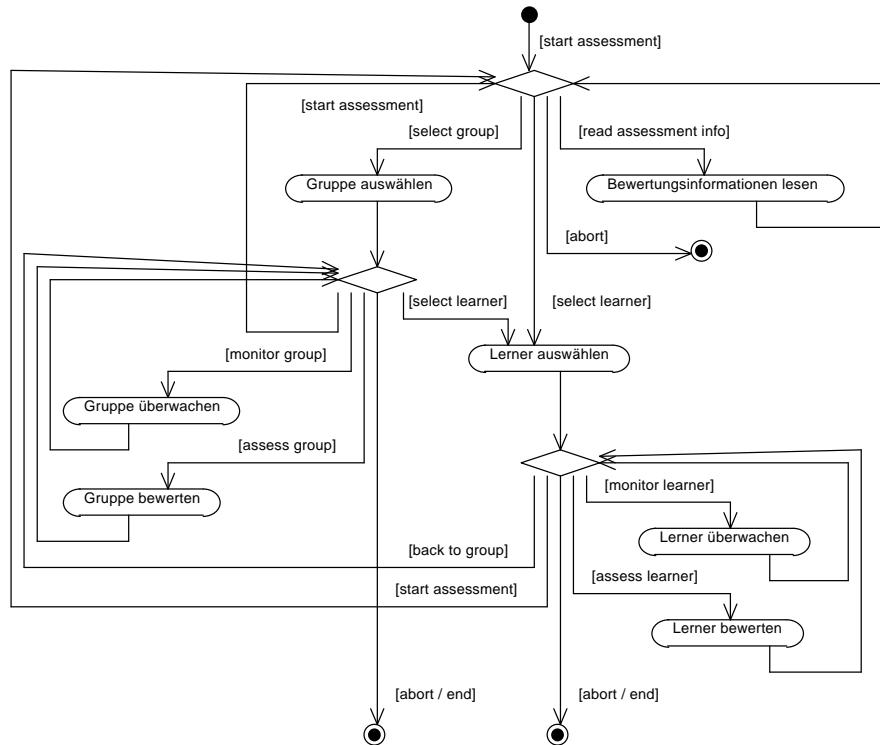


Fig. 5.8: Übersicht über den Geschäftsprozess Betreuung und Bewertung

sen wird und ähnliches. Ein Gruppentyp muss einen innerhalb des Kurses eindeutigen Namen haben und kann in der Kursverwaltung in der Contentstruktur einem oder mehreren Contents zugewiesen werden, für die diese Gruppen ihre Gültigkeit haben sollen.

Ist ein Gruppentyp gewählt, können effektive Gruppen bearbeitet, erstellt und gelöscht werden. Die automatische Gruppenerstellung erfolgt vollautomatisch zum entsprechenden Zeitpunkt oder durch manuelles Auslösen bei Bedarf. Bei der manuellen Gruppenerstellung wird jede einzelne Gruppe von Hand erzeugt. Dann werden die Metadaten, beispielsweise der Namen der Gruppe, ergänzt und falls notwendig gruppenspezifischer Content der Gruppe zugewiesen. Natürlich müssen auch die Lernenden und die Coaches zu den Gruppen hinzugefügt werden können. Über eine spezielle Nachrichtenfunktion soll einer Gruppe direkt eine Nachricht zugestellt werden können.

5.2.6 Betreuung und Bewertung

Der Geschäftsprozess Betreuung und Bewertung ist in der Abbildung 5.8 dargestellt. Dieser Prozess ist synchronisiert mit dem Lernprozess auf der Seite des Lernenden, welcher hier über seine Leistung beurteilt werden soll. Eine Bewertung ist grundsätzlich für jeden einzelnen Content möglich, welcher der Lerner zu bearbeiten hat. Ob und wie eine Bewertung stattfinden soll ist im Content, der Contentstruktur des Kurses und den Metainformationen des Kurses festgehalten.

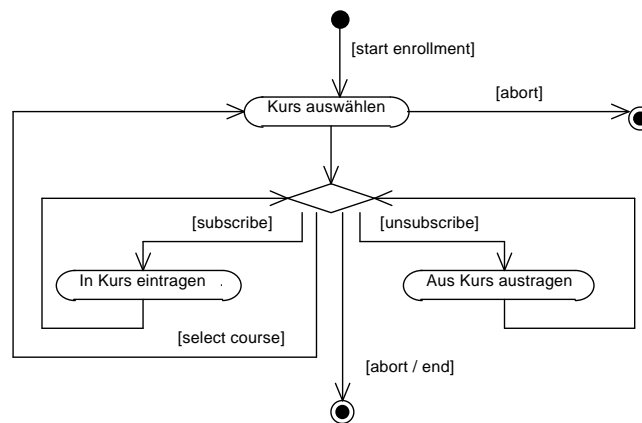


Fig. 5.9: Übersicht über den Geschäftsprozess Kurs belegen

Der Prozess der Beurteilung beginnt normalerweise mit dem Lesen der Bewertungsinformationen. Sie geben dem Coach Aufschluss darüber, welche Dokumente er mit welchen Regeln prüfen muss. Das Bewertungsobjekt kann demnach eine ganze Gruppe oder aber ein individueller Lernender sein.

Das Bewertungsobjekt kann in einem Teilprozess überwacht werden. Dabei können alle relevanten Aktionen, Resultate und abgegebenen Dokumente des Bewertungsobjektes begutachtet werden. Bei Abweichungen des erwünschten Verhaltens kann der Coach eingreifen indem er Hinweise gibt oder direkten Kontakt aufnimmt.

Das Bewertungsobjekt kann dann in einem weiteren Schritt bewertet werden. Die Bewertung wird für die betroffenen Lernenden unmittelbar sichtbar. Ist das Bewertungsobjekt eine Gruppe, so kann neben der Beobachtung und Bewertung der ganzen Gruppe jeder einzelne Lernende individuell abgearbeitet werden. So kann von der Gruppe stark abweichendes Verhalten ebenfalls in eine Beurteilung einfließen.

5.2.7 Kurs belegen

Im Normalfall sind Lerner selbst für das Besuchen eines Kurses zuständig und müssen sich selbstständig für einen solchen anmelden können. Die Abbildung 5.9 verdeutlicht diesen Prozess. Das Einschreiben ist dann möglich, wenn es vom Kursbetreuer in den Metadaten der Klassen dieses Kurses festgelegt wurde. Im Normalfall ist während der Einschreibeperiode auch ein Austragen möglich.

5.2.8 Lernen

Der Geschäftsprozess des Lernens stellt den eigentlichen Kernprozess des gesamten Systems dar. Dieser Prozess macht erst alle anderen vorhandenen Prozesse notwendig. Die Abbildung 5.10 zeigt eine Übersicht.

Coach zur Überprüfung und Bewertung abgegeben wird. Dieser Prozess ist nicht umkehrbar. Das heisst, ein Dokument, das abgegeben wurde kann nicht mehr verändert werden. Es können aber neue Dokumente dem Coach übermittelt werden der dann entscheiden kann, ob er diese neue Lösung akzeptieren will oder nicht.

Der Lernende muss sich jederzeit über seinen Status informieren können. In der LTSC Architektur⁴ der IEEE wird in diesem Zusammenhang von Performance gesprochen. Hier findet der Lerner die Bewertung des Coaches über abgegebene Arbeiten und die Resultate eines Online-Tests.

Während des Lernens muss der Lerner jederzeit Hilfe in Anspruch nehmen können. Es sollen hier automatisierte Hilfestellungen wie auch eine unkomplizierte Kommunikation mit dem Coach angeboten werden.

Die Suche ist eine wichtige Hilfsfunktion. Sie ermöglicht es dem Lernenden in dem Glossar und dem Content nach Stichworten zu suchen. Die Granularität kann dabei fein abgestimmt werden von einer Suche in allen bisher besuchten Kursen bis auf eine Suche, welche sich auf den aktuellen Content beschränkt.

Um die Navigation zu erleichtern muss der persönliche Lernpfad gespeichert und jederzeit abgerufen werden können. Unter Lernpfad wird hier die Abfolge der Bildschirmseiten verstanden, analog zu der History-Funktion in den gängigen Browsern.

5.2.9 Kommunizieren

Der allgemeine Unterstützungsprozess des Kommunizierens ist in der Abbildung 5.11 zu sehen. Er kann losgelöst vom Lernprozess betrachtet werden und soll unabhängig von einem Besuch eines Kurses verwendet werden können. Alle hier dargestellten Funktionen stehen dem Benutzer kontext- und rollensensitiv zur Verfügung. Rollensensitiv heisst, dass sich die Kommunikationsmöglichkeiten nach der aktuellen Benutzerrolle richten. Kontextsensitiv bedeutet, dass je nach aktuellem Kontext die Adressaten der Kommunikation gegeben sind. Eine Kursbetreuerin kann beispielsweise eine Nachricht an alle Lernenden eines Kurses schicken oder für einen ganzen Kurs ein Blackboard eröffnen während ein Lernender lediglich diese Kursnachricht empfangen und am Blackboard teilnehmen kann. Der Lerner kann aber in seiner Gruppe ein lokales Blackboard erstellen und innerhalb der Gruppe allen anderen Mitgliedern direkt eine Nachricht zustellen.

Es wird unterschieden zwischen einer normaler Nachricht und einer Nachricht innerhalb eines Blackboards. Normale Nachrichten sind immer an Personen adressiert und erscheinen als unabhängige Kopie individuell bei jeder angegebenen Empfängerin. Eine Nachricht in einem Blackboard existiert nicht als Kopie sondern ist allen Mitgliedern des Blackboards als Original zugänglich. Der Autor der Nachricht kann entscheiden, ob andere Teilnehmer diese Nachricht abändern können oder nicht. Ein Blackboard wird zum Zweck einer offenen Diskussion verwendet. Ein weiteres Unterscheidungsmerkmal ist, dass Nachrichten in einem Push-Verfahren zum Empfänger gelangen, während Informationen in einem Blackboard grundsätzlich per Pull aktiv

⁴ Die IEEE LTSA wird in Kapitel 4.1.1 besprochen

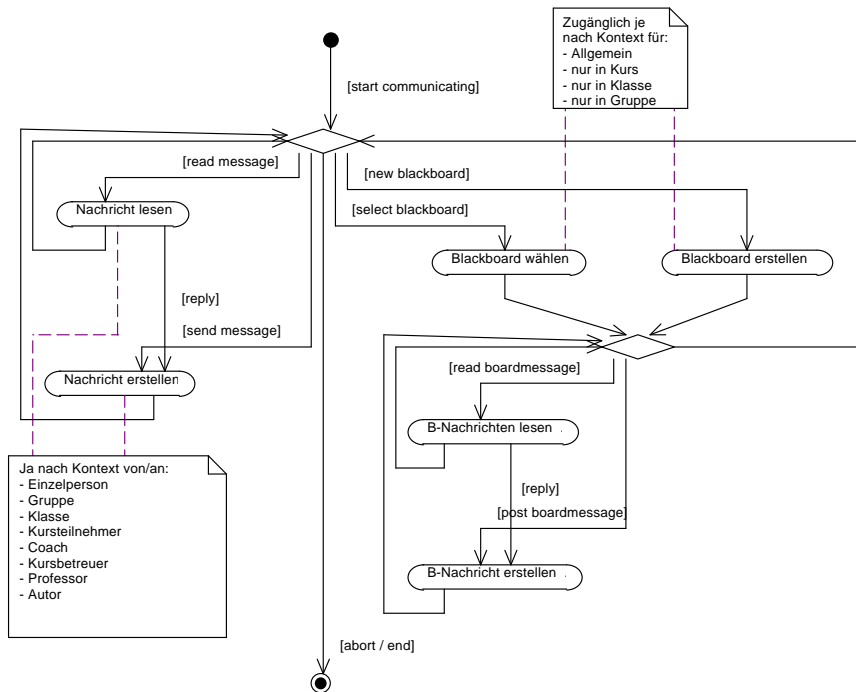


Fig. 5.11: Übersicht über den Geschäftsprozess Kommunizieren

abgerufen werden müssen. Blackboards sind das hauptsächliche Arbeits- und Kommunikationsinstrument für Gruppenarbeiten.

Normale Nachrichten werden als Email verschickt. Das Lesen einer Nachricht wird durch die privaten Emailprogramme der Benutzer erledigt. Eine Nachricht wird jedoch innerhalb des Systems erstellt, damit sie automatisch an die richtigen Personen gelangt und man nicht manuell die Adressaten angeben muss.

In einem Blackboard werden alle vorhandenen Nachrichten untereinander dargestellt, die aktuellsten zuoberst. Antworten auf eine Nachricht werden innerhalb der Ursprungsnachricht als sogenannter Thread aufgelistet.

5.2.10 Dateien verwalten

In der Abbildung 5.12 ist ein weiterer Unterstützungsprozess abgebildet, der ebenfalls rollen- und kontextsensitiv dem Benutzer seine Funktionalität anbietet. Dieser Prozess ist eng mit dem Prozess des Kommunizierens gekoppelt, denn gerade in den Blackboards möchte man oft Dokumente austauschen oder auch archivieren können.

Es wird unterschieden zwischen externen und internen Dateitypen. Die Begriffe intern und extern beziehen sich hier auf das Lernsystem selbst. All jene Dokumente, welche als extern bezeichnet werden, sind nicht vom System generiert und können daher nicht innerhalb des Systems bearbeitet werden (Beispielsweise ein Worddokument oder ein

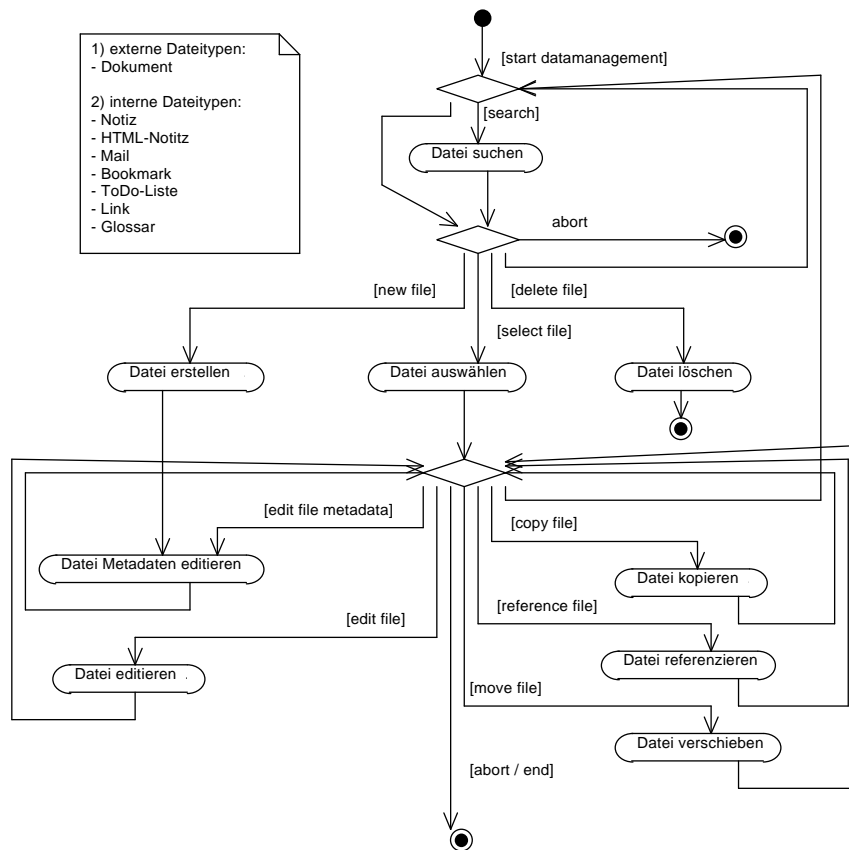


Fig. 5.12: Übersicht über den Geschäftsprozess Dateien verwalten

PDF-File). Bei einer solchen Datei bedeutet editieren das Runterladen des Dokumentes auf den lokalen Rechner, das Ändern mit einem speziellen Programm und das anschließende Speichern der neuen Version auf den Server.

Unter internen Dateitypen sind all jene zu verstehen, welche innerhalb des Systems selbst geändert werden können. Es sind dies Textnotizen, HTML-Notizen, Emails, Bookmarks, ToDo-Listen, kommentierte Links oder private Glossareinträge. Für diese Typen bietet das System spezielle Masken an, um die Daten direkt manipulieren zu können.

Alle Dateien verfügen über Metadaten welche nach dem Erstellen einer Datei sogleich näher spezifiziert werden müssen. Neben dem Typ der Datei werden hier die Keywords definiert, mit deren Hilfe die Datei in einem virtuellen Filesystem abgelegt und später wiedergefunden werden kann. Der virtuelle Pfad des Dokumentes ergibt sich auch durch den aktuellen Kontext des Benutzers im System. Die Metadaten enthalten auch die Informationen bezüglich der Sichtbarkeit und Manipulationsberechtigungen für andere Personen.

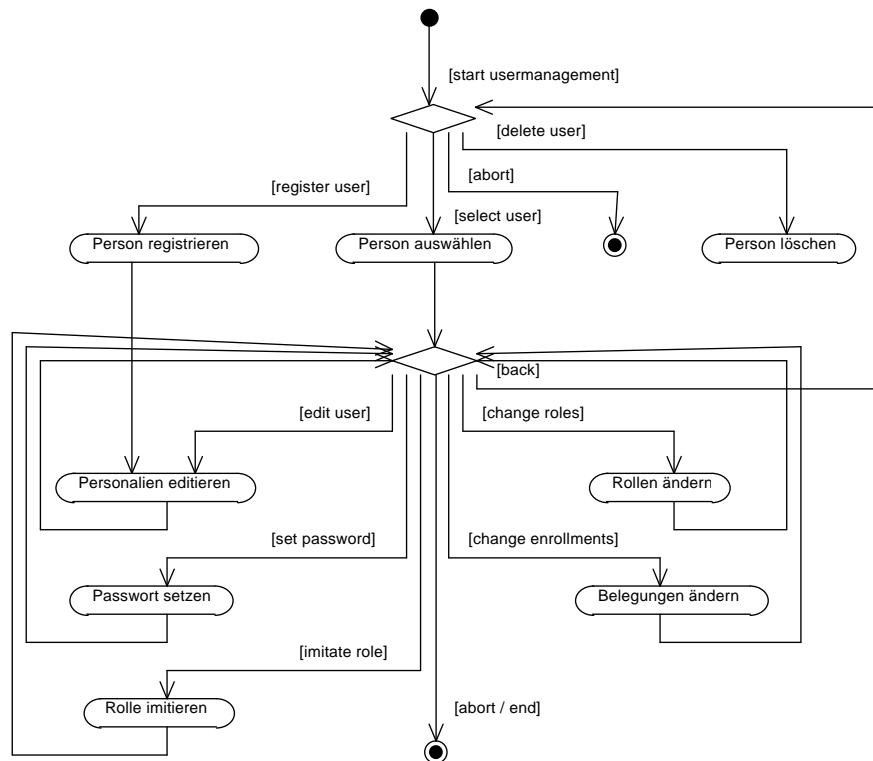


Fig. 5.13: Übersicht über den Geschäftsprozess Personenverwaltung

Neben dem Erstellen der Dateien sind alle gängigen Funktionen wie Löschen, Kopieren, Referenzieren⁵ und verschieben vorhanden. Da in einer webbasierten Umgebung Drag-and-Drop schlecht zu realisieren ist muss das Verschieben über eine Zwischenablage realisiert werden.

Der Prozess des Suchens spielt eine zentrale Rolle. Entweder kann über das virtuelle Dateisystem eine Datei gefunden werden oder es wird direkt nach Keywords gesucht. Die Suche kann nach Datum, Urheber, Keywords und Kontext eingeschränkt werden.

5.2.11 Personenverwaltung

Die Abbildung 5.13 zeigt den Prozess der Personenverwaltung. Personen müssen im System erfasst, verändert und schliesslich wieder gelöscht werden können. Ein Benutzer ohne spezielle, durch sogenannte Benutzerrollen ausgedrückte Zugangsrechte, hat lediglich Zugang zu seinen eigenen Daten und kann sich allenfalls selbständig auf dem System registrieren.

Personen des Helpdesks und des Supports können Benutzer suchen und ihnen ein neues Passwort setzen. Die Kursbetreuerin und der Coach können ebenfalls neue Passwörter

⁵ Hier spricht man auch von Link

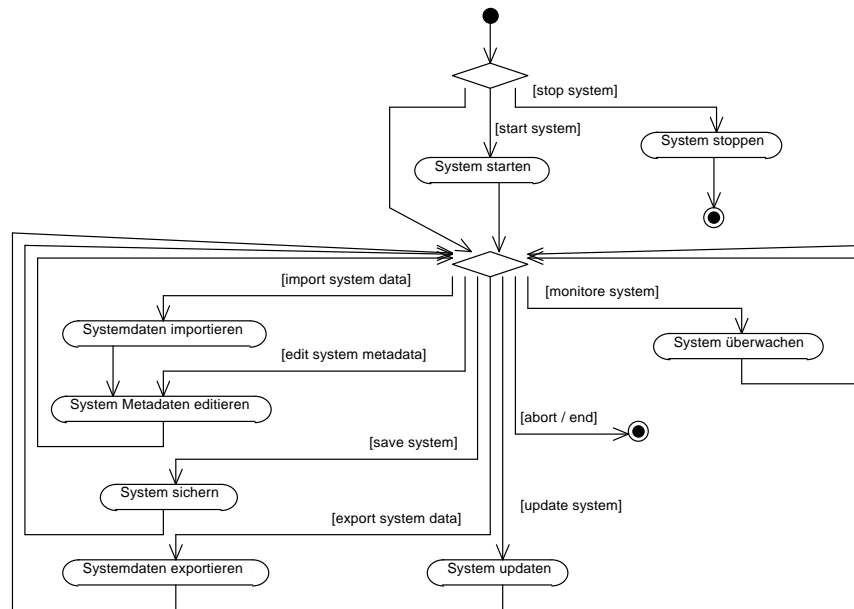


Fig. 5.14: Übersicht über den Geschäftsprozess Systemverwaltung

setzen, allerdings nur für Personen, welche in ihren Klassen eingeschrieben sind.

Rollen von Personen ändern bedeutet, einer Person Zugriffsberechtigungen für Kurse, Klassen, Gruppen oder Ähnliches zuzuteilen. Die Granularität ist sehr fein abgestimmt und bezieht sich immer auf einen Kontext. Es ist kein streng hierarchisches System mit einer einzelnen Wurzel. Eine Person vom Helpdesk kann zum Beispiel beliebige Passwörter ändern, ist also quasi weit oben in der Hierarchie. Trotzdem hat der Helpdesk keinen Einblick in die Bewertungen eines Lernenden. Ein Coach hat dies, obwohl er nicht beliebigen Personen die Passwörter ändern kann.

Belegungen von Kursen können vom Helpdesk und der Kursbetreuerin direkt geändert werden.

Der Prozess *Rolle imitieren* steht Personen des Supportes und Helpdeks, der Kursbetreuerin sowie dem Coach zur Verfügung. Damit können sie innerhalb ihres Kontextes in die Haut einer anderen Person schlüpfen um diese bei Problemen besser beraten zu können. Oft ist eine effiziente Beratung nur möglich, wenn man mit eigenen Augen sieht wie sich ein Problem präsentiert. Dies geht aus Gründen des Datenschutzes und der Sicherheit natürlich nicht automatisch. Es wird lediglich eine Anfrage gestartet, welche von der betroffenen Person angenommen werden muss. Während eine solche Rollenimitation stattfindet, ist dies für den Imitierten deutlich sichtbar und er kann die Imitation jederzeit beenden.

5.2.12 Systemverwaltung

Der Geschäftsprozess der Systemverwaltung ist in der Abbildung 5.14 gezeigt. Ist das System am laufen, so kann es gestoppt werden, andernfalls muss es gestartet werden um es zu verändern oder damit zu arbeiten.

Das System wird konfiguriert über die System-Metadaten. Neben Parametern zur Optimierung des Systems werden hier auch die diversen lokalen Gegebenheiten festgehalten wie der Name der Universität, welche Rollen existieren sollen und die hierarchische Struktur und die Namen der Organisationseinheiten wie Fakultäten und Institute.

Ein System kann als ganzes exportiert und wieder importiert werden. Dies dient der Migration auf neue Hardware oder der Archivierung der Daten als Sicherheitskopie für den Fall eines Totalverlustes der Daten.

Das System kann zur Laufzeit mit neuen Codeversionen erneuert werden, hierfür braucht es Unterstützung um eine Komponente temporär auszuschalten und dann neu zu initialisieren. Für das Austauschen von gewissen Komponenten wird das System allerdings komplett gestoppt werden müssen.

Mit einem Überwachungsprozess können die Aktivitäten auf dem System und dessen Zustand beobachtet werden um Engpässe und Probleme schnell zu identifizieren und darauf reagieren zu können.

6. ANWENDUNGSFÄLLE

In diesem Kapitel werden aufgrund der identifizierten Geschäftsprozesse die vorhandenen Anwendungsfälle aufgelistet. Anwendungsfälle zeichnen sich dadurch aus, dass sie durch einen klar identifizierten Akteur angestoßen werden und als Resultat der Aktivitäten ein Ergebnis von fachlichem Wert zur Folge haben. Die Aktivitäten in einem Anwendungsfall sind dabei zeitlich ununterbrochen und in sich konsistent. Ein fachlicher Wert repräsentiert ein für den Akteur sichtbares und relevantes Ergebnis. Die Tatsache, dass ein Dokument nach einer Suche gefunden wurde ist noch nicht von fachlichem Wert, erst wenn das Dokument für den Akteur effektiv zur Verfügung steht ist ein solcher Wert entstanden.

Die Anwendungsfälle werden in tabellarischer Form gemäss Oestereich dargestellt. Oestereich unterscheidet zwischen den Geschäftsanwendungsfällen, den essentiellen Anwendungsfällen und den Systemanwendungsfällen. In einem ersten Schritt werden alle Geschäftsanwendungsfälle identifiziert und kurz aufgelistet. Es ist eine erste Sammlung der Anwendungsfälle. In einem zweiten Schritt werden die essentiellen Anwendungsfälle herausgeschält. Es wird gefragt nach der eigentlichen Essenz und nach der Langlebigkeit der Anwendungsfälle um die wahrscheinlich stabilen von den sich ändernden Anwendungsfällen zu trennen. Zuletzt werden von diesen essentiellen Anwendungsfällen die effektiven Systemanwendungsfälle abgeleitet, welche später implementiert werden sollen.[Oes01]

Im Folgenden werden die identifizierten Systemanwendungsfälle abgebildet. Für die Implementation des Systems ist eine weitere Verfeinerung der Anwendungsfälle notwendig, zum Beispiel durch Illustrierung mittels eines Aktivitätsdiagrammes. Des weiteren müssen mit Interaktionsdiagrammen die Zusammenhänge zwischen den Anwendungsfällen herausgearbeitet werden.

6.1 Contentverwaltung

Name:	Content erstellen
Nummer:	1
Kurzbeschreibung:	Ein neues Contentelement gemäss SCORM wird erstellt.
Akteure:	Autorin
Auslöser:	Autorin möchte neuen Content erstellen.
Ergebnisse:	Leeres Contentelement ist samt Metadaten erstellt.

Tab. 6.1: Systemanwendungsfall *Content erstellen*

Name:	Content löschen
Nummer:	2
Kurzbeschreibung:	Ein Contentelement wird komplett vom System entfernt.
Akteure:	Autorin
Auslöser:	Autorin möchte bestehendes Contentelement entfernen.
Vorbedingungen:	Das Contentelement wird von keinem Kurs eingebunden.
Eingehende Informationen:	Name, Kategorisierung, Version etc. gemäss IMS/SCORM.
Ergebnisse:	Das Contentelement ist auf dem System gelöscht. Alle darin enthaltenen Elemente sind ebenfalls gelöscht.

Tab. 6.2: Systemanwendungsfall *Content löschen*

Name:	Content importieren
Nummer:	3
Kurzbeschreibung:	Ein Contentelement in Form einer Datei gemäss SCORM oder OLAT 1 Spezifikation wird importiert.
Akteure:	Autorin
Auslöser:	Autorin möchte ein Contentelement importieren.
Eingehende Informationen:	Datei, welche ein Contentelement gemäss SCORM beschreibt.
Ergebnisse:	Bestätigung, dass die Daten importiert sind.
Nachbedingungen:	Ein neues Contentelement ist erstellt. Metadaten und Inhalt sind definiert.

Tab. 6.3: Systemanwendungsfall *Content importieren*

Name:	Content Metadata editieren
Nummer:	4
Kurzbeschreibung:	Die Metadaten eines Contentelementes werden geändert.
Akteure:	Autorin
Auslöser:	Eine Autorin möchte die Metadaten eines Contentelementes ändern.
Vorbedingungen:	Contentelement besteht
Ergebnisse:	Die Metadaten des Contents sind geändert.

Tab. 6.4: Systemanwendungsfall *Content Metadata editieren*

Name:	Content exportieren
Nummer:	5
Kurzbeschreibung:	Ein Contentelement wird in einem SCORM-kompatiblen Format exportiert.
Akteure:	Autorin
Auslöser:	Autorin möchte Contentelement exportieren.
Ergebnisse:	Eine Datei gemäss SCORM Spezifikation ist erstellt und von der Autorin auf den lokalen Computer übertragen worden.

Tab. 6.5: Systemanwendungsfall *Content exportieren*

Name:	Contentstruktur editieren
Nummer:	6
Kurzbeschreibung:	Die Strukturierung der Blockelemente eines Contents wird verändert.
Akteure:	Autorin
Auslöser:	Eine Autorin möchte die Struktur eines Blockelementes verändern.
Vorbedingungen:	Es besteht mindestens ein Block- oder SCO-Element.
Ergebnisse:	Die Anordnung der Blockelemente ist geändert.

Tab. 6.6: Systemanwendungsfall *Contentstruktur editieren*

Name:	Blockelemente erstellen
Nummer:	7
Kurzbeschreibung:	Ein neues Blockelement innerhalb eines Contents wird erstellt.
Akteure:	Autorin
Auslöser:	Autorin möchte ein neues Blockelement erstellen.
Ergebnisse:	Das neue Blockelement ist samt Metadaten erstellt.
Nachbedingungen:	Block- und SCO-Elemente können dem Blockelement zugefügt werden.

Tab. 6.7: Systemanwendungsfall *Blockelemente erstellen*

Name:	Block Metadaten editieren
Nummer:	8
Kurzbeschreibung:	Die Metadaten eines Blocks innerhalb eines Contents werden geändert.
Akteure:	Autorin
Auslöser:	Autorin möchte Metadaten eines Blocks ändern.
Vorbedingungen:	Der Block existiert bereits.
Ergebnisse:	Die Metadaten des Blocks sind geändert.

Tab. 6.8: Systemanwendungsfall *Block Metadaten editieren*

Name:	Element in Block hinzufügen
Nummer:	9
Kurzbeschreibung:	Einem Blockelement wird ein Block- oder ein SCO-Element zugefügt.
Akteure:	Autorin
Auslöser:	Autorin möchte einem Block ein Element zufügen.
Vorbedingungen:	Block und Element existieren.
Ergebnisse:	Die Struktur des Blocks ist geändert und das Element an letzter Stelle hinzugefügt.

Tab. 6.9: Systemanwendungsfall *Element in Block hinzufügen*

Name:	Element von Block entfernen
Nummer:	10
Kurzbeschreibung:	Ein Block- oder SCO-Element innerhalb eines Blocks wird entfernt.
Akteure:	Autorin
Auslöser:	Autorin möchte ein Element löschen.
Vorbedingungen:	Das Element und der Block existieren.
Ergebnisse:	Die Struktur des Blocks ist geändert und das Element gelöscht.

Tab. 6.10: Systemanwendungsfall *Element von Block entfernen*

Name:	Block Metadaten editieren
Nummer:	11
Kurzbeschreibung:	Die Struktur eines Blocks innerhalb eines Contents wird geändert.
Akteure:	Autorin
Auslöser:	Autorin möchte die Struktur eines Blocks ändern.
Vorbedingungen:	Block enthält Block- oder SCO-Elemente.
Ergebnisse:	Die Struktur des Blocks ist geändert.

Tab. 6.11: Systemanwendungsfall *Blockstruktur editieren*

Name:	Block löschen
Nummer:	12
Kurzbeschreibung:	Ein Block innerhalb eines Contents wird gelöscht. Die Blöcke und SCO's, auf welche referenziert wurden, werden nicht gelöscht.
Akteure:	Autorin
Auslöser:	Autorin möchte einen Block löschen, deren Inhalt aber nicht.
Vorbedingungen:	Block existiert.
Ergebnisse:	Der Block ist gelöscht. Die darin referenzierten Blöcke und SCO's bleiben bestehen.
Nachbedingungen:	Die Block- und SCO-Elemente können in einem anderen Block verwendet werden.

Tab. 6.12: Systemanwendungsfall *Block löschen*

Name:	Block vollständig löschen
Nummer:	13
Kurzbeschreibung:	Ein Block wird samt aller darin referenzierten Blöcke und SCO's rekursiv innerhalb eines Contents gelöscht.
Akteure:	Autorin
Auslöser:	Autorin möchte einen Block vollständig löschen.
Vorbedingungen:	Block existiert.
Ergebnisse:	Der Block ist gelöscht. Die darin referenzierten Blöcke und SCO's sind ebenfalls gelöscht.

Tab. 6.13: Systemanwendungsfall *Block vollständig löschen*

Name:	SCO erstellen
Nummer:	14
Kurzbeschreibung:	Ein Sharable Content Object wird erstellt.
Akteure:	Autorin
Auslöser:	Eine Autorin möchte ein neues SCO erstellen.
Eingehende Informationen:	Name, Typ: Metadaten gemäss SCORM.
Ergebnisse:	Ein leeres SCO ist samt Metataden erstellt.
Nachbedingungen:	Assets können dem SCO zugewiesen werden.

Tab. 6.14: Systemanwendungsfall *SCO erstellen*

Name:	SCO Metadaten editieren
Nummer:	15
Kurzbeschreibung:	Die Metadaten eines Sharable Content Objects werden geändert.
Akteure:	Autorin
Auslöser:	Eine Autorin möchte die Metadaten eines SCO's ändern.
Vorbedingungen:	Das SCO existiert.
Ergebnisse:	Die Metadaten sind geändert.

Tab. 6.15: Systemanwendungsfall *SCO Metadaten editieren*

Name:	Asset erstellen
Nummer:	16
Kurzbeschreibung:	Ein neues Asset wird innerhalb eines SCO's erstellt.
Akteure:	Autorin
Auslöser:	Autorin möchte einem SCO ein neues Asset hinzufügen.
Vorbedingungen:	Das SCO existiert.
Ergebnisse:	Ein neues Asset samt Metadaten wurde dem SCO hinzugefügt.
Nachbedingungen:	Das Asset kann innerhalb des SCO's referenziert werden.

Tab. 6.16: Systemanwendungsfall *Asset erstellen*

Name:	Asset Metadaten editieren
Nummer:	17
Kurzbeschreibung:	Die Metadaten eines Assets werden geändert.
Akteure:	Autorin
Auslöser:	Eine Autorin möchte die Metadaten eines Assets ändern.
Vorbedingungen:	Das Asset existiert.
Eingehende Informationen:	Metadaten gemäss SCORM Raw Media Metadata.
Ergebnisse:	Die Metadaten des Assets sind geändert.

Tab. 6.17: Systemanwendungsfall *Asset Metadaten editieren*

Name:	Asset löschen
Nummer:	18
Kurzbeschreibung:	Ein Asset wird gelöscht.
Akteure:	Autorin
Auslöser:	Eine Autorin möchte ein Asset löschen.
Vorbedingungen:	Das Asset existiert.
Ergebnisse:	Das Asset ist gelöscht.

Tab. 6.18: Systemanwendungsfall *Asset löschen*

Name:	Asset editieren
Nummer:	19
Kurzbeschreibung:	Ein Asset wird editiert. Bei Textassets kann dies direkt online geschehen, bei anderen Assets muss eine neue Version übermittelt werden.
Akteure:	Autorin
Auslöser:	Eine Autorin möchte ein Asset ändern.
Vorbedingungen:	Asset existiert.
Ergebnisse:	Das Asset ist in einer neuen Version verfügbar. Die Metadaten sind angepasst.

Tab. 6.19: Systemanwendungsfall *Asset editieren*

Name:	Assessment erstellen
Nummer:	20
Kurzbeschreibung:	Ein neues Assessment-Element wird erstellt.
Akteure:	Autorin
Auslöser:	Autorin möchte ein Assessment erstellen.
Ergebnisse:	Ein neues Assessment ist samt Metadaten erstellt.

Tab. 6.20: Systemanwendungsfall *Assessment erstellen*

Name:	Assessment löschen
Nummer:	21
Kurzbeschreibung:	Ein Assessment-Element wird gelöscht.
Akteure:	Autorin
Auslöser:	Autorin möchte ein Assessment löschen.
Vorbedingungen:	Assessment-Element existiert und ist nicht in Gebrauch (in keinen Kurs eingebunden).
Ergebnisse:	Das Assessment ist vollständig gelöscht.

Tab. 6.21: Systemanwendungsfall *Assessment löschen*

Name:	Assessment importieren
Nummer:	22
Kurzbeschreibung:	Ein Assessment-Element wird importiert von einer Datei gemäss der IMS QTI Spezifikation.
Akteure:	Autorin
Auslöser:	Autorin möchte ein Assessment importieren.
Ergebnisse:	Ein neues Assessment ist samt Metadaten erstellt.

Tab. 6.22: Systemanwendungsfall *Assessment importieren*

Name:	Assessment exportieren
Nummer:	23
Kurzbeschreibung:	Ein Assessment-Element wird gemäss der IMS QTI Spezifikation in eine Datei exportiert.
Akteure:	Autorin
Auslöser:	Autorin möchte ein Assessment exportieren.
Ergebnisse:	Eine Datei wurde erstellt und auf den lokalen Rechner übertragen.

Tab. 6.23: Systemanwendungsfall *Assessment exportieren*

Name:	Assessment Metadaten editieren
Nummer:	24
Kurzbeschreibung:	Die Metadaten eines Assessments werden verändert.
Akteure:	Autorin
Auslöser:	Autorin möchte die Metadaten eines Assessments verändern.
Vorbedingungen:	Assessment existiert.
Ergebnisse:	Die Metadaten sind verändert.

Tab. 6.24: Systemanwendungsfall *Assessment Metadaten editieren*

Name:	Item erstellen
Nummer:	25
Kurzbeschreibung:	Ein Set von Questions und Responses wird erstellt.
Akteure:	Autorin
Auslöser:	Autorin möchte ein neues Item erstellen.
Ergebnisse:	Ein neues Item ist erstellt und kann in Sections verwendet werden.

Tab. 6.25: Systemanwendungsfall *Item erstellen*

Name:	Item löschen
Nummer:	26
Kurzbeschreibung:	Ein Set von Questions und Responses wird gelöscht.
Akteure:	Autorin
Auslöser:	Autorin möchte Item entfernen.
Vorbedingungen:	Das Item existiert und ist nicht in Gebrauch.
Ergebnisse:	Das Item ist gelöscht und aus allen Sections, in welche es eingebunden war, entfernt worden.

Tab. 6.26: Systemanwendungsfall *Item löschen*

Name:	Item editieren
Nummer:	27
Kurzbeschreibung:	Ein Set von Question und Responses wird verändert.
Akteure:	Autorin
Auslöser:	Autorin möchte ein Item verändern.
Vorbedingungen:	Das Item existiert.
Ergebnisse:	Das Item ist verändert. Die Änderungen sind in Tests ab diesem Zeitpunkt aktiv.

Tab. 6.27: Systemanwendungsfall *Item editieren*

Name:	Section erstellen
Nummer:	28
Kurzbeschreibung:	Eine Section wird aus Items und Sections erstellt.
Akteure:	Autorin
Auslöser:	Autorin möchte eine neue Section erstellen.
Vorbedingungen:	Es existieren Items oder Sections, welche Teil dieser neuen Section sein sollen.
Ergebnisse:	Die neue Section ist erstellt und kann verwendet werden.

Tab. 6.28: Systemanwendungsfall *Section erstellen*

Name:	Section löschen
Nummer:	29
Kurzbeschreibung:	Eine Section wird gelöscht.
Akteure:	Autorin
Auslöser:	Autorin möchte eine Section löschen.
Vorbedingungen:	Section existiert und ist nicht in Gebrauch.
Ergebnisse:	Die Section ist gelöscht. Alle darin enthaltenen Sections und Items sind nicht gelöscht und weiterhin verfügbar.

Tab. 6.29: Systemanwendungsfall *Section löschen*

Name:	Section komplett löschen
Nummer:	30
Kurzbeschreibung:	Eine Section wird mit all ihren Items und Sections gelöscht.
Akteure:	Autorin
Auslöser:	Autorin möchte eine Section vollständig löschen.
Vorbedingungen:	Section existiert und ist nicht in Gebrauch. Ihre Items und Sections sind ihrerseits ebenfalls nicht in Gebrauch.
Ergebnisse:	Die Section ist gelöscht. Alle darin enthaltenen Sections und Items sind ebenfalls komplett gelöscht.

Tab. 6.30: Systemanwendungsfall *Section komplett löschen*

Name:	Section editieren
Nummer:	31
Kurzbeschreibung:	Eine Section wird verändert. Beispielsweise werden neue Items oder Sections hinzugefügt.
Akteure:	Autorin
Auslöser:	Autorin möchte Section verändern oder neue Items oder Sections hinzufügen.
Vorbedingungen:	Section existiert.
Ergebnisse:	Die neue Struktur der Section ist ab sofort gültig.

Tab. 6.31: Systemanwendungsfall *Section editieren*

6.2 Kursverwaltung

Name:	Kurs erstellen
Nummer:	32
Kurzbeschreibung:	Ein neuer Kurs wird erstellt.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte einen neuen Kurs erstellen.
Eingehende Informationen:	Name, Zeitdauer, Typ, Schwierigkeitsgrad etc. gemäss SCORM Course Metadata.
Ergebnisse:	Ein neuer Kurs samt Metadaten ist erstellt.
Nachbedingungen:	Dem Kurs können Klassen zugewiesen werden, in welche sich Lerner einschreiben können.

Tab. 6.32: Systemanwendungsfall *Kurs erstellen*

Name:	Kurs löschen
Nummer:	33
Kurzbeschreibung:	Ein bestehender Kurs wird gelöscht.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte einen bestehenden Kurs löschen.
Vorbedingungen:	Kurs existiert.
Ergebnisse:	Der Kurs ist mit allen assoziierten Daten vollständig und unwiderruflich gelöscht.

Tab. 6.33: Systemanwendungsfall *Kurs löschen*

Name:	Kurs importieren
Nummer:	34
Kurzbeschreibung:	Ein Kurs in Form einer Datei gemäss SCORM oder OLAT 1 Spezifikation wird importiert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte einen Kurs importieren.
Eingehende Informationen:	Dateien gemäss SCORM CSF, welche einen Kurs definieren.
Ergebnisse:	Ein neuer Kurs ist erstellt. Die Metadaten und die Kursstruktur sind definiert.
Nachbedingungen:	Es müssen noch Klassen definiert werden damit Lerner diesen Kurs besuchen können.

Tab. 6.34: Systemanwendungsfall *Kurs importieren*

Name:	Kurs Metadaten editieren
Nummer:	35
Kurzbeschreibung:	Die Metadaten eines Kurses werden verändert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte die Metadaten eines Kurses ändern.
Vorbedingungen:	Kurs existiert.
Eingehende Informationen:	Metadaten gemäss SCORM Course Metadata.
Ergebnisse:	Die Metadaten sind geändert.

Tab. 6.35: Systemanwendungsfall *Kurs Metadaten editieren*

Name:	Kursstruktur editieren
Nummer:	36
Kurzbeschreibung:	Die Strukturierung der Contentelemente des Kurses wird verändert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte die Struktur eines Kurses ändern.
Eingehende Informationen:	Informationen bezüglich neu einzubindender Contentelemente oder Evaluationen.
Ergebnisse:	Die Anordnung der Contentelemente des Kurses ist geändert.

Tab. 6.36: Systemanwendungsfall *Kursstruktur editieren*

Name:	Kursnachricht zustellen
Nummer:	37
Spezialisierung von:	Nachricht erstellen.
Kurzbeschreibung:	Eine Nachricht wird allen Coaches oder Teilnehmerinnen eines Kurses zugestellt.
Akteure:	Kursbetreuerin, Professor
Auslöser:	Eine Kursbetreuerin möchte eine Nachricht per Broadcastverfahren an alle Teilnehmerinnen oder Coaches schicken.
Eingehende Informationen:	Nachrichtentext
Ergebnisse:	Die Nachricht ist verschickt.

Tab. 6.37: Systemanwendungsfall *Kursnachricht zustellen*

Name:	Kursstruktur exportieren
Nummer:	38
Kurzbeschreibung:	Die Struktur eines Kurses wird in einem SCORM kompatiblen Format exportiert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte den Aufbau eines Kurses exportieren.
Vorbedingungen:	Kurs existiert.
Ergebnisse:	Eine Datei gemäss SCORM Spezifikation ist erstellt und von der Kursbetreuerin auf den lokalen Computer übertragen worden.

Tab. 6.38: Systemanwendungsfall *Kursstruktur exportieren*

Name:	Kursdaten exportieren
Nummer:	39
Kurzbeschreibung:	Die personenbezogenen Daten eines Kurses werden in einem standardisierten Format exportiert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte Kursdaten exportieren.
Vorbedingungen:	Kurs existiert.
Eingehende Informationen:	Information über die Art der Daten: Resultate eines Contentelementes, alle Resultate, Verweildauer etc.
Ergebnisse:	Eine XML-basierte Datei ist erstellt und von der Kursbetreuerin auf den lokalen Computer übertragen worden.

Tab. 6.39: Systemanwendungsfall *Kursdaten exportieren*

Name:	Kursperformance überwachen
Nummer:	40
Kurzbeschreibung:	Bewertungsstatistiken eines Kurses werden berechnet.
Akteure:	Kursbetreuerin, Professor
Auslöser:	Eine Kursbetreuerin möchte Statistiken bezüglich der Performance der Lernenden erhalten.
Vorbedingungen:	Es existieren Lernende in diesem Kurs.
Eingehende Informationen:	Angabe des Typs: eine Person, eine Gruppe, eine Klasse oder der ganze Kurs sowie Angabe des Contentelements, welches beobachtet werden soll.
Ergebnisse:	Die Zahlen werden angezeigt.

Tab. 6.40: Systemanwendungsfall *Kursperformance überwachen*

Name:	Lernerverhalten beobachten
Nummer:	41
Kurzbeschreibung:	Verhaltensmuster und -statistiken der Lernenden werden berechnet.
Akteure:	Kursbetreuerin, Professor
Auslöser:	Eine Kursbetreuerin möchte Statistiken bezüglich der Benutzung des Contentelemente erhalten.
Vorbedingungen:	Es existieren Lernende in diesem Kurs.
Eingehende Informationen:	Angabe des Typs: eine Person, eine Gruppe, eine Klasse oder der ganze Kurs sowie Angabe des Contentelements, welches beobachtet werden soll.
Ergebnisse:	Die Zahlen werden angezeigt.

Tab. 6.41: Systemanwendungsfall *Lernerverhalten beobachten*

6.3 Klassenverwaltung

Name:	Klasse erstellen
Nummer:	42
Kurzbeschreibung:	Eine neue Klasse wird für einen Kurs erstellt.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine neue Klasse erstellen.
Vorbedingungen:	Es existiert ein Kurs, für den eine Klasse erstellt wird.
Eingehende Informationen:	Typ (virtuell, physisch), Einschreibzeitraum, Bearbeitungszeitraum etc.
Ergebnisse:	Eine leere Klasse wurde samt Metadaten erstellt.
Nachbedingungen:	Benutzer können in diese Klasse eingeteilt werden.

Tab. 6.42: Systemanwendungsfall *Klasse erstellen*

Name:	Klasse löschen
Nummer:	43
Kurzbeschreibung:	Eine bestehende Klasse eines Kurses wird gelöscht.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine Klasse löschen.
Vorbedingungen:	Klasse existiert.
Ergebnisse:	Die Klasse ist gelöscht. Personen und Coaches, welche Teilnehmer beziehungsweise Verantwortliche dieser Klasse waren, wurden zuvor aus der Klasse entfernt. Klasseninterne Gruppen sind ebenfalls aufgelöst worden.

Tab. 6.43: Systemanwendungsfall *Klasse löschen*

Name:	Klassenbelegung exportieren
Nummer:	44
Kurzbeschreibung:	Die Belegungen aller Klassen eines Kurses werden exportiert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte alle Klassenbelegungen exportieren.
Vorbedingungen:	Klasse existiert.
Ergebnisse:	Eine XML-basierte Datei ist erstellt und von der Kursbetreuerin auf den lokalen Computer übertragen.

Tab. 6.44: Systemanwendungsfall *Klassenbelegung exportieren*

Name:	Einschreibung überwachen
Nummer:	45
Kurzbeschreibung:	Die Anzahl der belegten und verfügbaren Plätze aller Klassen wird überwacht.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte die Einschreibung überwachen.
Vorbedingungen:	Klasse existiert
Ergebnisse:	Die Anzahl der besetzten und verfügbaren Plätze wird angezeigt.

Tab. 6.45: Systemanwendungsfall *Einschreibung überwachen*

Name:	Platzangebot verändern
Nummer:	46
Kurzbeschreibung:	Die Anzahl der verfügbaren Plätze in Klassen eines Kurses werden verändert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte die Anzahl der verfügbaren Plätze in einem Kurs verändern.
Vorbedingungen:	Es existieren Klassen für diesen Kurs.
Ergebnisse:	Das neue Platzangebot ist gültig.

Tab. 6.46: Systemanwendungsfall *Platzangebot verändern*

Name:	Klassen Metadaten editieren
Nummer:	47
Kurzbeschreibung:	Die Metadaten einer Klasse werden verändert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte die Metadaten einer Klasse verändern.
Vorbedingungen:	Klasse existiert
Eingehende Informationen:	Typ (virtuell, physisch), Einschreibzeitraum, Bearbeitungszeitraum etc.
Ergebnisse:	Die Metadaten sind geändert.

Tab. 6.47: Systemanwendungsfall *Klassen Metadaten editieren*

Name:	Raum zuweisen
Nummer:	48
Kurzbeschreibung:	Einer Klasse mit physischer Präsenz wird ein Raum zugewiesen. Es können mehrere Räume zugewiesen werden.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte einer Klasse einen Raum zuweisen.
Vorbedingungen:	Raum und Klasse existieren. Ob ein Raum schon belegt ist wird <i>nicht</i> geprüft. Die Verwaltung von Räumen ist nicht Teil dieses Systems. Hier ist ein Raum lediglich ein Informationsobjekt (Bezeichnung, Ort, Anzahl Plätze etc.)
Eingehende Inf.:	Raumbezeichnung, Zeit, Wochentag etc.
Ergebnisse:	Der Raum ist der Klasse zugewiesen. Die Zuweisung wird den Coaches und Lerner dieser Klasse mitgeteilt, falls solche für diese Klasse existieren.

Tab. 6.48: Systemanwendungsfall *Raum zuweisen*

Name:	Raumzuweisung löschen
Nummer:	49
Kurzbeschreibung:	Eine Raumzuweisung wird gelöscht.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine Raumzuweisung rückgängig machen.
Vorbedingungen:	Zuweisung hat existiert.
Ergebnisse:	Die Klasse hat nun keinen Raum zugewiesen. Falls es Coaches oder Lerner in dieser Klasse gibt, werden diese über dieses Ereignis informiert.
Nachbedingungen:	Ist kein einziger Raum dieser Klasse mehr zugewiesen, so ist ihr Typ nun 'virtuell'.

Tab. 6.49: Systemanwendungsfall *Raumzuweisung löschen*

Name:	Klassennachricht zustellen
Nummer:	50
Spezialisierung von:	Nachricht erstellen
Kurzbeschreibung:	Eine Nachricht wird allen Coaches oder Teilnehmerinnen einer Klasse zugestellt.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine Nachricht per Broadcastverfahren an alle Teilnehmerinnen oder Coaches einer Klasse schicken.
Eingehende Informationen:	Nachricht
Ergebnisse:	Die Nachricht ist verschickt.

Tab. 6.50: Systemanwendungsfall *Klassennachricht zustellen*

Name:	Lerner Klasse zuweisen
Nummer:	51
Kurzbeschreibung:	Ein Lerner wird in eine Klasse eingeteilt, unabhängig davon ob die Einschreibefrist schon abgelaufen ist oder nicht.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte einen Lerner in eine Klasse einteilen.
Vorbedingungen:	Klasse existiert, Lerner ist auf System registriert.
Ergebnisse:	Der Lerner ist in der Klasse eingeschrieben. Die Coaches dieser Klasse wurden über den Vorfall informiert falls die Anmeldefrist bereits abgelaufen ist.
Nachbedingungen:	Lerner kann sofort den Kurs besuchen.

Tab. 6.51: Systemanwendungsfall *Lerner Klasse zuweisen*

Name:	Coach Klasse zuweisen
Nummer:	52
Kurzbeschreibung:	Eine Person wird als Coach für die Betreuung einer Klasse verantwortlich gemacht. Diese Rolle ist zeitlich limitiert oder an die Existenz der Klasse gebunden.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte einer Klasse einen Coach zuweisen.
Vorbedingungen:	Klasse existiert, Coach ist auf System registriert.
Ergebnisse:	Die Person ist als Coach für die Klasse für den bestimmten Zeitraum verantwortlich. Die Coaches und die Lerner dieser Klasse wurden über den Vorfall informiert.
Nachbedingungen:	Coach kann sofort den Kurs betreuen.

Tab. 6.52: Systemanwendungsfall *Coach Klasse zuweisen*

Name:	Stellvertretercoach Klasse zuweisen
Nummer:	53
Kurzbeschreibung:	Einem Coach einer Klasse wird für eine bestimmte Zeitdauer eine Stellvertretung zugewiesen. Die Stellvertretung muss bereits eine Coachingrolle in diesem Kurs innehaben.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte einem Coach für eine bestimmte Zeit einen Stellvertreter zur Seite stellen.
Vorbedingungen:	Klasse existiert, Stellvertreter ist bereits Coach in einer anderen Klasse.
Ergebnisse:	Der Stellvertreter hat für diese Zeit die Rechte des Coaches für diese Klasse. Die Stellvertretung gilt auch für die von diesem Coach betreuten Gruppen. Die Coaches und Lerner der Klasse wurden über den Vorfall informiert.

Tab. 6.53: Systemanwendungsfall *Stellvertretercoach Klasse zuweisen*

Name:	Stellvertretercoach entfernen
Nummer:	54
Kurzbeschreibung:	Ein Stellvertreter eines Coaches wird entfernt.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte einen stellvertretenden Coach wieder entfernen.
Vorbedingungen:	Stellvertretung hat bestanden
Ergebnisse:	Der Stellvertreter wurde entfernt. Die Coaches und die Lerner dieser Klasse wurden über den Vorfall informiert.

Tab. 6.54: Systemanwendungsfall *Stellvertretercoach entfernen*

Name:	Coach von Klasse entfernen
Nummer:	55
Kurzbeschreibung:	Ein Coach einer Klasse wird entfernt.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte ein Coach entfernen.
Vorbedingungen:	Coachverhältnis hat bestanden.
Ergebnisse:	Der Coach ist nicht mehr für die Betreuung dieser Klasse verantwortlich. Die Coaches und die Lerner dieser Klasse wurden über den Vorfall informiert.

Tab. 6.55: Systemanwendungsfall *Coach von Klasse entfernen*

Name:	Lerner von Klasse entfernen
Nummer:	56
Kurzbeschreibung:	Ein Lerner wird aus einer Klasse entfernt.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte einen Lerner entfernen.
Vorbedingungen:	Lerner war in Klasse eingeschrieben.
Ergebnisse:	Der Lerner ist aus der Klasse und allen Gruppen dieses Kurses entfernt worden. Die Coaches dieser Klasse und der Lerner selbst wurden über den Vorfall informiert.

Tab. 6.56: Systemanwendungsfall *Lerner von Klasse entfernen*

Name:	Lerner umteilen
Nummer:	57
Kurzbeschreibung:	Ein Lerner wird von einer Klasse in eine andere umgeteilt, auch nach Abschluss der Anmeldefrist des Kurses.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte einen Lerner in eine andere Klasse einteilen.
Vorbedingungen:	Lerner ist in dieser Klasse eingeteilt und die andere Klasse existiert.
Ergebnisse:	Der Lerner ist in eine andere Klasse eingeteilt worden. Dabei wurde er aus den klasseninternen Gruppen der vorherigen Klasse ausgetragen. Klassenübergreifende Gruppen wurden nicht verändert. Die zuvor verantwortlichen Coaches, die neuen Coaches und der Lerner selbst wurden über den Vorfall informiert.

Tab. 6.57: Systemanwendungsfall *Lerner umteilen*

6.4 Evaluationsverwaltung

Name:	Evaluation erstellen
Nummer:	58
Kurzbeschreibung:	Eine neue Evaluation wird erstellt.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine neue Evaluation erstellen.
Vorbedingungen:	Kurs existiert.
Eingehende Informationen:	Metadaten: Name, Zielpublikum, Zeitraum etc.
Ergebnisse:	Eine neue Evaluation samt Metadaten ist erstellt.

Tab. 6.58: Systemanwendungsfall *Evaluation erstellen*

Name:	Evaluation löschen
Nummer:	59
Kurzbeschreibung:	Eine bestehende Evaluation wird komplett gelöscht.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine Evaluation löschen.
Vorbedingungen:	Kurs und Evaluation existiert.
Ergebnisse:	Die Evaluation ist mit samt ihrer Resultate gelöscht.

Tab. 6.59: Systemanwendungsfall *Evaluation löschen*

Name:	Evaluation importieren
Nummer:	60
Kurzbeschreibung:	Eine neue Evaluation wird erstellt durch den Import eines standardisierten Dokumentes.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine Evaluation importieren.
Vorbedingungen:	Kurs existiert
Eingehende Informationen:	Evaluation in Form von XML-Dateien, Metadaten.
Ergebnisse:	Eine neue Evaluation ist samt Metadaten und Inhalt erstellt.
Nachbedingungen:	Lerner können Evaluation ausfüllen.

Tab. 6.60: Systemanwendungsfall *Evaluation importieren*

Name:	Evaluation exportieren
Nummer:	61
Kurzbeschreibung:	Der Aufbau einer Evaluation wird als XML-Datei zusammengestellt und exportiert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte den Aufbau einer Evaluation exportieren.
Vorbedingungen:	Evaluation existiert
Ergebnisse:	Eine XML-basierte Datei ist erstellt und wurde von der Kursbetreuerin auf den lokalen Computer übertragen.
Nachbedingungen:	Diese Datei kann wieder importiert werden, auch in einem anderen Kurs.

Tab. 6.61: Systemanwendungsfall *Evaluation exportieren*

Name:	Evaluationsresultate exportieren
Nummer:	62
Kurzbeschreibung:	Die Resultate einer Evaluation werden berechnet und in einem standardisierten Dateiformat exportiert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte die Evaluationsresultate exportieren.
Vorbedingungen:	Evaluation existiert und ist zur Bearbeitung freigegeben.
Ergebnisse:	Eine Datei welche mit Excel oder SPSS weiterverarbeitet werden kann ist erstellt und von der Kursbetreuerin auf den lokalen Computer übertragen worden.

Tab. 6.62: Systemanwendungsfall *Evaluationsresultate exportieren*

Name:	Evaluation überwachen
Nummer:	63
Kurzbeschreibung:	Die Resultate der bereits ausgefüllten Evaluation und die sich daraus abzeichnenden Trends werden berechnet und angezeigt.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte den aktuellen Stand der Evaluation erfahren.
Vorbedingungen:	Evaluation existiert und befindet sich in der aktiven Phase (Wird zur Zeit von Lernenden beantwortet).
Ergebnisse:	Die aktuellen Resultate werden berechnet und angezeigt.

Tab. 6.63: Systemanwendungsfall *Evaluation überwachen*

Name:	Evaluation Metadaten editieren
Nummer:	64
Kurzbeschreibung:	Die Metadaten einer Evaluation werden verändert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte die Metadaten einer Evaluation ändern.
Vorbedingungen:	Evaluation existiert.
Eingehende Inform.:	Metadaten: Name, Zielpublikum, Zeitraum etc.
Ergebnisse:	Die Metadaten sind geändert.

Tab. 6.64: Systemanwendungsfall *Evaluation Metadaten editieren*

Name:	Evaluationsstruktur editieren
Nummer:	65
Kurzbeschreibung:	Die Strukturierung des Fragebogens wird verändert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte die Struktur einer Evaluation verändern.
Vorbedingungen:	Es existieren mindestens zwei Evaluationsfragen.
Ergebnisse:	Die Evaluation hat eine neue Struktur.

Tab. 6.65: Systemanwendungsfall *Evaluationsstruktur editieren*

Name:	Evaluationsfrage erstellen
Nummer:	66
Kurzbeschreibung:	Eine Frage für einen Fragebogen wird erfasst.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine neue Frage in einem Fragebogen erfassen.
Vorbedingungen:	Evaluation existiert
Eingehende Inform.:	Fragetext, Typeninformation, vordefinierte Antworten.
Ergebnisse:	Eine neue Frage ist erfasst und wurde in der Evaluationsstruktur am Ende angefügt.

Tab. 6.66: Systemanwendungsfall *Evaluationsfrage erstellen*

Name:	Evaluationsfrage löschen
Nummer:	67
Kurzbeschreibung:	Eine Frage in einem Fragebogen wird gelöscht.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine Frage löschen.
Vorbedingungen:	Evaluationsfrage existiert.
Ergebnisse:	Die Frage ist gelöscht und aus der Evaluationsstruktur entfernt worden.

Tab. 6.67: Systemanwendungsfall *Evaluationsfrage löschen*

Name:	Evaluationsfrage editieren
Nummer:	68
Kurzbeschreibung:	Eine Evaluationsfrage wird verändert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine Frage in einem Fragebogen verändern.
Eingehende Informationen:	Fragetext, Typeninformation, ev.
Vorbedingungen:	Evaluationsfrage existiert.
Ergebnisse:	Die Frage ist verändert.

Tab. 6.68: Systemanwendungsfall *Evaluationsfrage editieren*

6.5 Gruppenverwaltung

Name:	Gruppentyp erstellen
Nummer:	69
Kurzbeschreibung:	Für einen Kurs wird ein neuer Gruppentyp erstellt.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte einen neuen Gruppentyp erstellen.
Vorbedingungen:	Der Kurs existiert.
Eingehende Informationen:	Gruppentyp Metadaten: Typ (klasseninterne, klassenübergreifende Gruppen), optimale Gruppengröße, optimale Anzahl Gruppen, Angaben über Erstellung (manuell, automatisch), Gültigkeitsdauer etc.
Ergebnisse:	Ein neuer Gruppentyp ist samt Metadaten erstellt.
Nachbedingungen:	Gruppen dieses Typs können erstellt werden.

Tab. 6.69: Systemanwendungsfall *Gruppentyp erstellen*

Name:	Gruppentyp löschen
Nummer:	70
Kurzbeschreibung:	Ein Gruppentyp wird gelöscht.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte einen Gruppentyp löschen
Vorbedingungen:	Gruppentyp existiert.
Ergebnisse:	Der Gruppentyp ist gelöscht. Alle Gruppen, welche von diesem Typ waren, sind ebenfalls gelöscht.
Nachbedingungen:	Es können keine Gruppen dieses Typs mehr erstellt werden.

Tab. 6.70: Systemanwendungsfall *Gruppentyp löschen*

Name:	Gruppentyp Metadaten editieren
Nummer:	71
Kurzbeschreibung:	Die Metadaten des Gruppentyps werden verändert.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte die Metadaten eines Gruppentyps abändern.
Vorbedingungen:	Gruppentyp existiert. Es existieren noch keine Gruppen dieses Typs.
Ergebnisse:	Die Metadaten sind geändert.

Tab. 6.71: Systemanwendungsfall *Gruppentyp Metadaten editieren*

Name:	Kursgruppen automatisch erstellen
Nummer:	72
Kurzbeschreibung:	Klassenübergreifende Gruppen werden automatisch aufgrund von Angaben bezüglich der optimalen Gruppengröße erstellt.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte die Gruppen eines Typs mit klassenübergreifenden Gruppen automatisch erstellen lassen.
Vorbedingungen:	Es existiert ein Gruppentyp mit dieser Definition.
Eingehende Informationen:	Angabe eines Dokumentes, welches die Namen der zu generierenden Gruppen enthält.
Ergebnisse:	Die Gruppen sind erstellt und Teilnehmer und Coaches in die Gruppen eingeladen worden. Alle Beteiligten wurden informiert.

Tab. 6.72: Systemanwendungsfall *Kursgruppen automatisch erstellen*

Name:	Klassengruppen automatisch erstellen
Nummer:	73
Kurzbeschreibung:	Innerhalb einer Klasse werden Gruppen automatisch erstellt.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte in einer Klasse die Gruppen automatisch erstellen lassen.
Vorbedingungen:	Es existiert ein Gruppentyp mit dieser Definition.
Eingehende Informationen:	Angabe eines Dokumentes, welche die Namen der zu generierenden Gruppen enthält.
Ergebnisse:	Die Gruppen sind erstellt und Teilnehmer und Coaches in die Gruppen eingeladen. Die Coaches entsprechen den Coaches der jeweiligen Klasse. Alle Beteiligten sind informiert worden.

Tab. 6.73: Systemanwendungsfall *Klassengruppen automatisch erstellen*

Name:	Kursgruppe manuell erstellen
Nummer:	74
Kurzbeschreibung:	Einer klassenübergreifende Gruppe wird manuell erstellt.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte eine klassenübergreifende Gruppe erstellen.
Vorbedingungen:	Der Gruppentyp existiert.
Eingehende Informationen:	Name der Gruppe etc.
Ergebnisse:	Die Gruppe ist samt Metadaten erstellt.

Tab. 6.74: Systemanwendungsfall *Kursgruppe manuell erstellen*

Name:	Klassengruppe manuell erstellen
Nummer:	75
Kurzbeschreibung:	Eine klasseninterne Gruppe wird manuell erstellt.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte eine klasseninterne Gruppe erstellen.
Vorbedingungen:	Der Gruppentyp existiert.
Eingehende Informationen:	Name der Gruppe etc.
Ergebnisse:	Die Gruppe ist samt Metadaten erstellt. Die Coaches der Klasse sind als Coaches ebenfalls in die Gruppe eingeladen worden.

Tab. 6.75: Systemanwendungsfall *Klassengruppe manuell erstellen*

Name:	Gruppe Metadaten editieren
Nummer:	76
Kurzbeschreibung:	Die Metadaten einer Gruppe werden verändert.
Akteure:	Kursbetreuerin, Coach, Lerner (Gruppenmitglied)
Auslöser:	Eine Kursbetreuerin möchte die Metadaten einer Gruppe verändern.
Vorbedingungen:	Gruppe existiert.
Eingehende Inform.:	Name der Gruppe etc.
Ergebnisse:	Die Metadaten sind geändert.

Tab. 6.76: Systemanwendungsfall *Gruppe Metadaten editieren*

Name:	Gruppe löschen
Nummer:	77
Kurzbeschreibung:	Eine Gruppe wird gelöscht.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte eine Gruppe löschen.
Vorbedingungen:	Gruppe existiert.
Ergebnisse:	Die Gruppe ist gelöscht. Alle Coaches und Teilnehmerinnen dieser Gruppe wurden benachrichtigt.

Tab. 6.77: Systemanwendungsfall *Gruppe löschen*

Name:	Lerner Gruppe zuweisen
Nummer:	78
Kurzbeschreibung:	Ein Lerner wird einer Gruppe zugewiesen.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte einen Lerner in eine Gruppe eintragen.
Vorbedingungen:	Gruppe und Lerner existieren.
Ergebnisse:	Der Lerner ist in die Gruppe eingeteilt. Alle davon betroffenen Coaches und der Lerner sind informiert worden.

Tab. 6.78: Systemanwendungsfall *Lerner Gruppe zuweisen*

Name:	Coach Gruppe zuweisen
Nummer:	79
Kurzbeschreibung:	Ein Coach wird einer Gruppe zugewiesen.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte einer Gruppe einen Coach zuweisen.
Vorbedingungen:	Gruppe und Coach existieren.
Ergebnisse:	Der Coach ist in die Gruppe eingeteilt. Alle davon betroffenen Coaches und die Lerner sind informiert worden.

Tab. 6.79: Systemanwendungsfall *Coach Gruppe zuweisen*

Name:	Lerner von Gruppe entfernen
Nummer:	80
Kurzbeschreibung:	Ein Lerner wird aus einer Gruppe ausgetragen.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte einen Lerner aus einer Gruppe austragen.
Vorbedingungen:	Gruppe existiert und Lerner ist dieser Gruppe zugeteilt.
Ergebnisse:	Der Lernende ist nicht mehr in dieser Gruppe. Alle betroffenen Coaches und der Lerner wurden informiert.

Tab. 6.80: Systemanwendungsfall *Lerner von Gruppe entfernen*

Name:	Coach von Gruppe entfernen
Nummer:	81
Kurzbeschreibung:	Ein Coach wird aus einer Gruppe ausgetragen.
Akteure:	Kursbetreuerin
Auslöser:	Eine Kursbetreuerin möchte einen Coach aus einer Gruppe austragen.
Vorbedingungen:	Gruppe existiert und Coach ist dieser Gruppe zugeteilt.
Ergebnisse:	Der Coach ist nicht mehr in der Gruppe. Alle betroffenen Coaches und der Lerner wurden informiert.

Tab. 6.81: Systemanwendungsfall *Coach von Gruppe entfernen*

Name:	Gruppencontent zuweisen
Nummer:	82
Kurzbeschreibung:	Einer Gruppe wird ein Contentelement zugewiesen.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte einer Gruppe ein Contentelement zuweisen.
Vorbedingungen:	Gruppe existiert, Contentelement existiert.
Eingehende Informationen:	Contentelementangabe
Ergebnisse:	Die Gruppe weiss, welches Contentelement sie zu bearbeiten hat.
Nachbedingungen:	Das Contentelement ist den Mitgliedern der Gruppe zugänglich.

Tab. 6.82: Systemanwendungsfall *Gruppencontent zuweisen*

Name:	Gruppennachricht zustellen
Nummer:	83
Spezialisierung von:	Nachricht erstellen
Kurzbeschreibung:	Der Gruppe wird eine Nachricht im Broadcastverfahren zugestellt.
Akteure:	Kursbetreuerin, Coach
Auslöser:	Eine Kursbetreuerin möchte einer Gruppe eine Nachricht zukommen lassen.
Eingehende Informationen:	Nachricht
Ergebnisse:	Die Nachricht ist an alle Teilnehmerinnen und Coaches dieser Gruppe verschickt worden.

Tab. 6.83: Systemanwendungsfall *Gruppennachricht zustellen*

6.6 Betreuung und Bewertung

Name:	Bewertungsinformationen lesen
Nummer:	84
Kurzbeschreibung:	Informationen über die zu vollziehende Bewertung werden gelesen.
Akteure:	Coach
Auslöser:	Ein Coach möchte erfahren, wie er einen Lernenden bei einer Aufgabe zu bewerten hat.
Vorbedingungen:	Bewertungsinformation existiert.
Ergebnisse:	Die Bewertungsinformationen sind dem Coach zugänglich gemacht worden.

Tab. 6.84: Systemanwendungsfall *Bewertungsinformationen lesen*

Name:	Lerner überwachen
Nummer:	85
Kurzbeschreibung:	Informationen über die Leistungen eines Lerners werden berechnet und mit seinem Lernpfad angezeigt.
Akteure:	Coach
Auslöser:	Der Coach möchte Informationen über die Performance des Lernenden erhalten.
Vorbedingungen:	Coach ist für diesen Lerner zuständig.
Eingehende Informationen:	Angaben über den Lerner und Einschränkungen bezüglich der zu beachtenden Contentelemente.
Ergebnisse:	Der Coach ist im Bilde über die Aktivitäten des Lerners und seinen erbrachten Leistungen.

Tab. 6.85: Systemanwendungsfall *Lerner überwachen*

Name:	Bewertung erstellen
Nummer:	86
Kurzbeschreibung:	Der Lerner wird mit Punkten und Kommentaren für eine Leistung bewertet. Eine Bewertung kann nicht gelöscht werden.
Akteure:	Coach
Auslöser:	Lerner gibt zu beurteilende Arbeit ab.
Vorbedingungen:	Eine Bewertung ist vorgesehen gemäss dem bearbeiteten Contentelement.
Eingehende Inform.:	Abgegebene Arbeit, erreichte Punktzahl, Kommentar.
Ergebnisse:	Die Arbeit ist mit Note und Kommentar bewertet. Der Lerner ist über die Bewertung informiert.

Tab. 6.86: Systemanwendungsfall *Bewertung erstellen*

Name:	Bewertung editieren
Nummer:	87
Kurzbeschreibung:	Eine Bewertung wird korrigiert. Dies entspricht einer zweiten Bewertung und keiner eigentlichen Korrektur. Die vorherige Bewertung bleibt bestehen, wird aber als ungültig gekennzeichnet.
Akteure:	Coach
Auslöser:	Ein Coach möchte eine Bewertung ändern.
Vorbedingungen:	Bewertung existiert bereits.
Eingehende Informationen:	Punkte, Kommentar
Ergebnisse:	Eine neue Bewertung ist erstellt. Die alte Bewertung ist als ungültig gekennzeichnet. Der Lerner ist über die Änderung informiert.

Tab. 6.87: Systemanwendungsfall *Bewertung editieren*

Name:	Gruppenbewertung erstellen
Nummer:	88
Kurzbeschreibung:	Eine Gruppe wird mit Punkten und Kommentaren für eine Leistung bewertet. Eine Bewertung kann nicht gelöscht werden.
Akteure:	Coach
Auslöser:	Gruppe gibt zu beurteilende Arbeit ab.
Vorbedingungen:	Eine Bewertung ist vorgesehen gemäss dem bearbeiteten Contentelement.
Eingehende Informationen:	Abgegebene Arbeit, Erreichte Punktzahl, Kommentar.
Ergebnisse:	Die Arbeit ist mit Note und Kommentar bewertet. Für jeden Teilnehmer der Gruppe ist eine eigene Bewertung erstellt worden. Die Gruppe ist über die Bewertung informiert worden.

Tab. 6.88: Systemanwendungsfall *Gruppenbewertung erstellen*

Name:	Gruppenbewertung editieren
Nummer:	89
Kurzbeschreibung:	Eine Gruppenbewertung wird korrigiert. Dies entspricht einer zweiten Bewertung und keiner eigentlichen Korrektur. Die vorherige Bewertung bleibt bestehen, wird aber als ungültig gekennzeichnet.
Akteure:	Coach
Auslöser:	Ein Coach möchte eine Gruppenbewertung ändern.
Vorbedingungen:	Gruppenbewertung existiert bereits.
Eingehende Informationen:	Punkte, Kommentar

Fortsetzung auf nächster Seite

Fortsetzung von letzter Seite

Name:	Gruppenbewertung editieren
Ergebnisse:	Eine neue Gruppenbewertung ist erstellt, das heisst, pro Gruppenmitglied wurde eine neue Bewertung erstellt. Die alten Bewertungen sind als ungültig gekennzeichnet. Die Gruppe ist über die Änderung informiert.

Tab. 6.89: Systemanwendungsfall *Gruppenbewertung editieren*

Name:	Gruppe überwachen
Nummer:	90
Kurzbeschreibung:	Statistiken über die Aktivitäten der einzelnen Gruppenmitglieder in dem Blackboard der Gruppe werden berechnet. Die einzelnen Beiträge werden begutachtet.
Akteure:	Coach
Auslöser:	Ein Coach möchte mehr über die Beteiligung der Personen an Gruppenarbeiten und über die Fortschritte der Gruppe erfahren.
Vorbedingungen:	Coach ist für diese Gruppe zuständig
Eingehende Informationen:	Angaben über die Gruppe und Einschränkungen bezüglich der zu beachtenden Contentelemente.
Ergebnisse:	Der Coach weiss, wer einen wie grossen Beitrag an das Endresultat geliefert hat.

Tab. 6.90: Systemanwendungsfall *Gruppe überwachen*

Name:	Anfrage beantworten
Nummer:	91
Spezialisierung von:	Nachricht lesen, Nachricht erfassen.
Kurzbeschreibung:	Eine Anfrage eines Lernalers wird beantwortet. Der Kontext des Lernalers steht dabei dem Coach zur Verfügung um die Frage besser zu verstehen.
Akteure:	Coach
Auslöser:	Ein Lerner nimmt Kontakt mit dem Coach auf.
Vorbedingungen:	Coach ist für diesen Lerner zuständig.
Eingehende Informationen:	Anfrage, Antwort
Ergebnisse:	Der Coach hat die Frage beantwortet und dem Lerner mitgeteilt.

Tab. 6.91: Systemanwendungsfall *Anfrage beantworten*

6.7 Kurs belegen

Name:	Kurs belegen
Nummer:	92
Kurzbeschreibung:	Selbständiges Einschreiben in eine Klasse eines Kurses.
Akteure:	Lerner
Auslöser:	Ein Lerner möchte einen Kurs belegen.
Vorbedingungen:	Kurs existiert, aktuelles Datum ist im Einschreibzeitraum.
Ergebnisse:	Der Lerner ist in der Klasse eingetragen und hat eine Bestätigung mit Informationen über Datum, Zeit, Ort, betreuende Coaches etc. erhalten. Wenn entsprechend in den Kursmetadaten spezifiziert, ist der Coach über die neue Anmeldung informiert worden.
Nachbedingungen:	Lerner kann sofort Kurs besuchen.

Tab. 6.92: Systemanwendungsfall *Kurs belegen*

Name:	Belegung löschen
Nummer:	93
Kurzbeschreibung:	Eine Kursbelegung eines Lerner wird selbständig vom Lerner gelöscht.
Akteure:	Lerner
Auslöser:	Ein Lerner möchte sich aus einer Klasse eines Kurses austragen.
Vorbedingungen:	Belegung existiert.
Eingehende Informationen:	Angabe eines Grundes falls erwünscht.
Ergebnisse:	Der Lerner ist von der Klasse entfernt worden. Alle Daten, welche in der Zwischenzeit über diesen Lerner angefallen sind werden gelöscht. Er ist aus allen Gruppen dieses Kurses ausgetragen. Wenn entsprechend in den Kursmetadaten spezifiziert oder wenn die Anmeldefrist abgelaufen ist, dann ist der Coach über den Schritt des Lerner informiert worden.

Tab. 6.93: Systemanwendungsfall *Belegung löschen*

6.8 Lernen

Name:	Status beobachten
Nummer:	94
Kurzbeschreibung:	Berechnen einer Übersicht über die bisher erreichten Punkte und Bewertungen.
Akteure:	Lerner
Auslöser:	Ein Lerner möchte Auskunft über seine Performance erhalten.
Vorbedingungen:	Lerner ist in Kurs eingeschrieben.
Ergebnisse:	Die Informationen wurden berechnet und dem Lerner mitgeteilt.

Tab. 6.94: Systemanwendungsfall *Status beobachten*

Name:	Coach kontaktieren
Nummer:	95
Kurzbeschreibung:	Senden einer Nachricht an den Coach mit Angaben über die aktuelle Position des Lerners.
Akteure:	Lerner
Auslöser:	Ein Lerner möchte seinen Coach um Rat fragen.
Vorbedingungen:	Ein Coach ist zuständig für den Lerner.
Eingehende Informationen:	Nachricht
Ergebnisse:	Der Coach hat eine Nachricht erhalten zusammen mit dem Kontext, in dem die Nachricht erstellt wurde.

Tab. 6.95: Systemanwendungsfall *Coach kontaktieren*

Name:	Suchen
Nummer:	96
Kurzbeschreibung:	Es wird in den freigeschalteten Contentelementen des Kurses nach einem Stichwort gesucht. Der Lernende kann die Suchresultate betrachten.
Akteure:	Lerner
Auslöser:	Lerner möchte nach einem Stichwort suchen.
Vorbedingungen:	Es existieren freigeschaltete Contentelemente in diesem Kurs.
Eingehende Informationen:	Suchstichwort
Ergebnisse:	Der Lerner hat alle gefundenen Stellen zur Kenntnis genommen.

Tab. 6.96: Systemanwendungsfall *Suchen*

Name:	Lernpfad betrachten
Nummer:	97
Kurzbeschreibung:	Der Lernpfad des Lernalers wird berechnet und angezeigt. Der Lerner hat die Möglichkeit an beliebige Orte des Pfades zu springen und Sequenzen zu wiederholen.
Akteure:	Lerner
Auslöser:	Ein Lerner möchte seinen Lernpfad betrachten.
Eingehende Informationen:	Angabe des Zeitraums.
Ergebnisse:	Der Lerner weiss, wo in dem Kurs er sich in der Vergangenheit aufgehalten hat.

Tab. 6.97: Systemanwendungsfall *Lernpfad betrachten*

Name:	Lernpfad weiterführen
Nummer:	98
Kurzbeschreibung:	Es wird der letzte Schritt des Lernpfades berechnet und geladen.
Akteure:	Lerner
Auslöser:	Ein Lerner möchte mit seinem Lernprozess fortfahren.
Vorbedingungen:	Lerner hat diesen Kurs schon einmal betrachtet.
Ergebnisse:	Der Lerner befindet sich dort, wo er bei der letzten Sitzung aufgehört hat.

Tab. 6.98: Systemanwendungsfall *Lernpfad weiterführen*

Name:	Content lesen
Nummer:	99
Kurzbeschreibung:	Ein Contentelement wird geladen und vom Lerner gelesen beziehungsweise bearbeitet. Dies beinhaltet auch Self Assessment Testfragmente, welche in den normalen Content eingebunden sind.
Akteure:	Lerner
Auslöser:	Ein Lerner möchte ein Contentelement bearbeiten.
Vorbedingungen:	Das Contentelement ist zur Bearbeitung freigegeben.
Eingehende Informationen:	Bei Testfragmenten Angabe von Antworten.
Ergebnisse:	Der Lerner hat das Contentelement bearbeitet. Falls dabei Resultate anfallen, so sind diese gespeichert und der Status entsprechend geändert worden. Das Contentelement ist als bearbeitet gekennzeichnet.

Tab. 6.99: Systemanwendungsfall *Content lesen*

Name:	Test lösen
Nummer:	100
Kurzbeschreibung:	Ein Onlinetest wird, eventuell unter Zeit- und Orts-einschränkung, gelöst. Das erzielte Resultat wird für die weitere Statusberechnung verwendet.
Akteure:	Lerner
Auslöser:	Ein Lerner möchte einen Onlinetest absolvieren
Vorbedingungen:	Der Test ist zur Bearbeitung freigegeben.
Eingehende Informa-tionen:	Antworten zu den Testfragen.
Ergebnisse:	Der Test ist beendet und die Punkte berechnet und dem Lerner mitgeteilt.
Nachbedingungen:	Je nach Testtyp kann der Test nicht mehr wiederholt werden.

Tab. 6.100: Systemanwendungsfall *Test lösen*

Name:	Lösung abgeben
Nummer:	101
Kurzbeschreibung:	Ein Dokument, welches eine Lösung einer Aufgabenstel-lung enthält, wird dem Coach zur Beurteilung abgegeben, beziehungsweise übermittelt. Bei einer Gruppenaufgabe wird dies als Resultat der ganzen Gruppe aufgefasst.
Akteure:	Lerner
Auslöser:	Ein Lerner möchte eine Arbeit dem Coach zur Bewertung abgeben.
Vorbedingungen:	Eine Abgabe von Arbeiten ist in dem Contentelement vorgesehen.
Eingehende Informa-tionen:	Die Arbeit in Form von Dokumenten, Bemerkungen.
Ergebnisse:	Der Coach hat das Dokument erhalten.
Nachbedingungen:	Die übermittelten Dokumente können vom Lerner nicht mehr verändert werden.

Tab. 6.101: Systemanwendungsfall *Lösung abgeben*

6.9 Kommunizieren

Name:	Nachricht lesen
Nummer:	102
Kurzbeschreibung:	Eine persönliche Nachricht wird zur Kenntnis genommen. Bei Bedarf wird auf die Nachricht mit einer Antwort reagiert.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte eine Nachricht lesen.
Vorbedingungen:	Nachricht existiert.
Ergebnisse:	Die Nachricht ist gelesen, eventuell gelöscht oder gesichert und bei Bedarf ist eine Antwort erstellt worden.

Tab. 6.102: Systemanwendungsfall *Nachricht lesen*

Name:	Nachricht erstellen
Nummer:	103
Kurzbeschreibung:	Eine Nachricht an eine dem System bekannte Person wird erstellt. Dazu steht ein persönliches Adressbuch zur Verfügung.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte einem anderen Benutzer eine Nachricht zustellen.
Eingehende Informationen:	Nachricht, Betreff, Empfänger (-Gruppen) etc.
Ergebnisse:	Die Nachricht wurde dem Empfänger zugestellt.

Tab. 6.103: Systemanwendungsfall *Nachricht erstellen*

Name:	Adressbucheintrag einfügen
Nummer:	104
Kurzbeschreibung:	Ein Person wird im Adressbuch erfasst. Der Kontext in dem die Person auftritt wird ebenfalls angegeben.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte eine Person in sein Adressbuch eintragen.
Eingehende Informationen:	Name, Adresse etc. von Person oder Gruppe.
Ergebnisse:	Eine Person wurde dem Adressbuch zugefügt.
Nachbedingungen:	Beim Erstellen einer neuen Nachricht ist diese Person direkt als Empfänger auswählbar.

Tab. 6.104: Systemanwendungsfall *Adressbucheintrag einfügen*

Name:	Adressbucheintrag löschen
Nummer:	105
Kurzbeschreibung:	Eine Person wird aus dem Adressbuch entfernt.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte eine Person aus seinem Adressbuch entfernen.
Vorbedingungen:	Adressbucheintrag existiert.
Ergebnisse:	Die Person wurde aus dem Adressbuch gelöscht.

Tab. 6.105: Systemanwendungsfall *Adressbucheintrag löschen*

Name:	Blackboard erstellen
Nummer:	106
Kurzbeschreibung:	Ein Blackboard wird für einen spezifischen Kontext erstellt. Der Ersteller definiert in den Metadaten die Benutzergruppen des Blackboards.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte ein neues Blackboard erstellen.
Vorbedingungen:	Der Benutzer verfügt über die Berechtigung in diesem Kontext ein Blackboard zu eröffnen.
Eingehende Inform.:	Name, Zugangsberechtigungen, Zeitdauer etc.
Ergebnisse:	Ein neues Blackboard ist mit den entsprechenden Zugriffsberechtigungen erstellt.
Nachbedingungen:	Personen mit Zugangsberechtigung haben Zugriff auf das Blackboard.

Tab. 6.106: Systemanwendungsfall *Blackboard erstellen*

Name:	Blackboard Metadaten editieren
Nummer:	107
Kurzbeschreibung:	Die Metadaten eines Blackboards werden von dem Eigentümer geändert, zum Beispiel der Name, die Zugriffsberechtigungen oder der Kontext.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte die Metadaten eines Blackboards ändern.
Vorbedingungen:	Blackboard existiert.
Eingehende Inform.:	Name, Zugangsberechtigungen, Zeitdauer etc.
Ergebnisse:	Die neuen Metadaten sind erfasst.
Nachbedingungen:	Personen der neu definierten Zugangsberechtigungen haben Zugriff auf das Blackboard, alte Berechtigungen sind gelöscht.

Tab. 6.107: Systemanwendungsfall *Blackboard Metadaten editieren*

Name:	Blackboard löschen
Nummer:	108
Kurzbeschreibung:	Ein Blackboard wird mit all seinen Nachrichten und Dokumenten von seinem Eigentümer gelöscht.
Akteure:	Benutzer (Eigentümer)
Auslöser:	Ein Benutzer möchte ein Blackboard löschen.
Vorbedingungen:	Das Blackboard existiert.
Ergebnisse:	Das Blackboard ist mit all seinen Nachrichten und Dokumenten gelöscht worden.
Nachbedingungen:	Die Mitglieder des Blackboards können dieses nicht mehr betrachten oder bearbeiten.

Tab. 6.108: Systemanwendungsfall *Blackboard löschen*

Name:	Blackboard exportieren
Nummer:	109
Kurzbeschreibung:	Ein Blackboard wird als Verzeichnis mit allen enthaltenen Dokumenten und Notizen als HTML Dokumente erstellt und als komprimiertes ZIP Archiv für die Archivierung oder Offlineverwendung zur Verfügung gestellt.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte ein Blackboard exportieren.
Vorbedingungen:	Das Blackboard existiert.
Ergebnisse:	Ein TAR-Archiv wurde erstellt und auf die lokale Festplatte des Benutzers übertragen.

Tab. 6.109: Systemanwendungsfall *Blackboard exportieren*

Name:	Blackboardnachricht erstellen
Nummer:	110
Kurzbeschreibung:	Eine neue Nachricht in einem Blackboard wird erstellt. Der Nachricht können auch Dokumente beigefügt werden.
Akteure:	Benutzer
Auslöser:	Der Benutzer möchte eine neue Nachricht in einem Blackboard publizieren.
Vorbedingungen:	Blackboard existiert.
Eingehende Informationen:	Nachricht, Betreff etc.
Ergebnisse:	Die neue Nachricht ist im Blackboard platziert.
Nachbedingungen:	Die Nachricht kann von allen Mitgliedern des Blackboards gelesen werden.

Tab. 6.110: Systemanwendungsfall *Blackboardnachricht erstellen*

Name:	Blackboardnachricht lesen
Nummer:	111
Kurzbeschreibung:	Eine Nachricht in einem Blackboard wird gelesen. Bei Bedarf kann darauf mit einer in die ursprüngliche Nachricht integrierten Nachricht reagiert werden.
Akteure:	Benutzer
Auslöser:	Benutzer möchte eine Nachricht lesen.
Vorbedingungen:	Nachricht existiert.
Ergebnisse:	Die Nachricht wurde gelesen. Bei Bedarf wurde eine Antwort zu der Nachricht erfasst und auf dem Blackboard platziert.

Tab. 6.111: Systemanwendungsfall *Blackboardnachricht lesen*

Name:	Blackboardnachricht bewerten
Nummer:	112
Kurzbeschreibung:	Eine Nachricht wird kategorisiert und qualitativ bewertet. Der Mittelwert und die Standardabweichung dieser Bewertungen wird beim Lesen oder Sortieren der Notizen angezeigt.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte eine Nachricht bewerten.
Vorbedingungen:	Nachricht existiert.
Ergebnisse:	Die neuen Mittelwerte und Standardabweichungen sind bestimmt.

Tab. 6.112: Systemanwendungsfall *Blackboardnachricht bewerten*

Name:	Blackboardnachricht löschen
Nummer:	113
Kurzbeschreibung:	Eine Nachricht in einem Blackboard wird gelöscht. Alle darin verschachtelten Nachrichten werden ebenfalls gelöscht.
Akteure:	Benutzer (Eigentümer des Blackboards oder der Nachricht)
Auslöser:	Ein Benutzer möchte eine Nachricht aus dem Blackboard entfernen.
Vorbedingungen:	Nachricht existiert.
Ergebnisse:	Die Nachricht und alle Unternachrichten sind entfernt.

Tab. 6.113: Systemanwendungsfall *Blackboardnachricht löschen*

6.10 Dateien verwalten

Name:	Externe Datei erstellen
Nummer:	114
Kurzbeschreibung:	Ein Dokument wird vom lokalen Rechner auf das System übertragen.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte ein externes Dokument auf dem Server speichern.
Eingehende Informationen:	Dokument, Metadaten wie Name, Dokumenttyp, Version, Keywords, Benutzergruppen etc
Ergebnisse:	Das Dokument ist auf dem Server samt Metadaten gespeichert.

Tab. 6.114: Systemanwendungsfall *Externe Datei erstellen*

Name:	Externe Datei editieren
Nummer:	115
Kurzbeschreibung:	Ein auf dem Server gespeichertes Dokument wird auf den lokalen Rechner übertragen, geändert und in der neuen Version wieder auf dem Server gespeichert.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte ein Dokument verändern.
Vorbedingungen:	Datei existiert.
Ergebnisse:	Das Dokument steht dem Server in einer neuen Version zur Verfügung.
Nachbedingungen:	Die Versionsinformation ist auf den neuesten Stand.

Tab. 6.115: Systemanwendungsfall *Externe Datei editieren*

Name:	Interne Datei erstellen
Nummer:	116
Kurzbeschreibung:	Ein neues, internes Dokument wird erstellt. Als Dokumenttypen kommen eine Textnotiz, eine HTML-Notiz, ein Bookmark, eine Todo-Liste, ein Link oder ein Glossareintrag in Frage.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte ein neues internes Dokument erstellen.
Eingehende Informationen:	Daten für das Dokument, Metadaten wie Name, Dokumenttyp, Version, Keywords, Benutzergruppen etc.
Ergebnisse:	Ein neues internes Dokument ist samt Metadaten erstellt.

Tab. 6.116: Systemanwendungsfall *Interne Datei erstellen*

Name:	Interne Datei editieren
Nummer:	117
Kurzbeschreibung:	Eine interne Datei wird inhaltlich verändert. Je nach Dateityp stehen andere Möglichkeiten zur Verfügung.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte ein internes Dokument verändern.
Vorbedingungen:	Dokument existiert.
Ergebnisse:	Das Dokument ist geändert.
Nachbedingungen:	Die Versionsinformation ist auf den neuesten Stand.

Tab. 6.117: Systemanwendungsfall *Interne Datei editieren*

Name:	Datei Metadaten editieren
Nummer:	118
Kurzbeschreibung:	Die Metadaten eines Dokumentes (intern oder extern) werden verändert. Dies wäre beispielsweise der Dokumentname, Dokumenttyp, die Keywords, Benutzergruppen etc.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte die Metadaten eines Dokumentes verändern.
Vorbedingungen:	Dokument existiert.
Ergebnisse:	Die Metadaten sind geändert.

Tab. 6.118: Systemanwendungsfall *Datei Metadaten editieren*

Name:	Datei löschen
Nummer:	119
Kurzbeschreibung:	Ein internes oder externes Dokument wird vollständig gelöscht.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte ein Dokument löschen.
Vorbedingungen:	Dokument existiert.
Ergebnisse:	Das Dokument ist vollständig gelöscht. Alle Referenzen auf dieses Dokument sind ebenfalls gelöscht.

Tab. 6.119: Systemanwendungsfall *Datei löschen*

Name:	Datei kopieren
Nummer:	120
Kurzbeschreibung:	Von einer internen oder externen Datei wird eine identische Kopie gemacht.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte ein Dokument kopieren.
Vorbedingungen:	Datei existiert.
Ergebnisse:	Das Dokument ist kopiert.

Tab. 6.120: Systemanwendungsfall *Datei kopieren*

Name:	Datei verschieben
Nummer:	121
Kurzbeschreibung:	Eine Datei wird von einem Ort im System an einen anderen Ort verschoben.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte ein Dokument an einen andern Ort hin verschieben.
Vorbedingungen:	Datei existiert.
Eingehende Informationen:	Neue Lokalität des Dokumentes.
Ergebnisse:	Das Dokument ist an seiner neuen Stelle platziert.

Tab. 6.121: Systemanwendungsfall *Datei verschieben*

Name:	Datei referenzieren
Nummer:	122
Kurzbeschreibung:	Eine Datei wird an mehreren Orten sichtbar und zugänglich gemacht, ohne die Datei zu kopieren.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte auf eine Datei referenzieren.
Vorbedingungen:	Dokument existiert.
Ergebnisse:	Eine virtuelle Datei ist erstellt, welche zu der Originaldatei referenziert.
Nachbedingungen:	Das Original kennt seine Referenzen (bidirektional).

Tab. 6.122: Systemanwendungsfall *Datei referenzieren*

6.11 Personenverwaltung

Name:	Person selbständig registrieren
Nummer:	123
Spezialisierung von:	Person registrieren
Kurzbeschreibung:	Eine Person registriert sich selbst auf dem System als neuer Benutzer.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte einen OLAT Benutzeraccount.
Vorbedingungen:	Das System ist so konfiguriert, dass diese Möglichkeit zugelassen ist.
Eingehende Informationen:	Personalien, persönliche Einstellungen.
Ergebnisse:	Ein neuer Account ist generiert und die persönlichen Informationen gespeichert.
Nachbedingungen:	Dem Benutzer kann nun eine beliebige Rolle zugeteilt werden und er kann sich für Kurse anmelden.

Tab. 6.123: Systemanwendungsfall *Person selbständig registrieren*

Name:	Person registrieren
Nummer:	124
Kurzbeschreibung:	Eine Person wird auf dem System von einer anderen Person registriert.
Akteure:	Support, Helpdesk, Kursbetreuerin
Auslöser:	Eine Person des Supports möchte einen neuen Systemaccount für eine Person erstellen.
Eingehende Informationen:	Personalien, persönliche Einstellungen
Ergebnisse:	Ein neuer Account ist generiert und die persönlichen Informationen gespeichert.
Nachbedingungen:	Dem Benutzer kann nun eine beliebige Rolle zugeteilt werden und er kann sich für Kurse anmelden.

Tab. 6.124: Systemanwendungsfall *Person registrieren*

Name:	Person automatisch registrieren
Nummer:	125
Spezialisierung von:	Person registrieren
Kurzbeschreibung:	Eine Person wird auf dem System automatisch über eine Schnittstelle von einem anderen Universitätsinformationssystem erstellt.
Akteure:	UniAccess / SAP
Auslöser:	Ein Informationssystem der Universität möchte einen neuen OLAT-Benutzer generieren.
Eingehende Informationen:	Personalien, persönliche Einstellungen
Ergebnisse:	neuer Account ist generiert und die persönlichen Informationen gespeichert.
Nachbedingungen:	Dem Benutzer kann nun eine beliebige Rolle zugeteilt werden und kann sich für Kurse anmelden.

Tab. 6.125: Systemanwendungsfall *Person automatisch registrieren*

Name:	Person automatisch löschen
Nummer:	126
Spezialisierung von:	Person löschen
Kurzbeschreibung:	Ein Systembenutzer wird auf dem System automatisch über eine Schnittstelle von einem anderen Universitätsinformationssystem gelöscht.
Akteure:	UniAccess / SAP
Auslöser:	Ein Informationssystem der Universität möchte einen Benutzer automatisch löschen.
Vorbedingungen:	Benutzer existiert.
Ergebnisse:	Der Benutzer ist mit all seinen Daten vom System entfernt worden.
Nachbedingungen:	Benutzer kann nicht mehr auf dem System arbeiten. Er ist auf dem System nicht mehr vorhanden.

Tab. 6.126: Systemanwendungsfall *Person automatisch löschen*

Name:	Person löschen
Nummer:	127
Kurzbeschreibung:	Ein Benutzer wird manuell auf dem System gelöscht. Der Benutzer selbst kann dies nicht tun.
Akteure:	Support, Helpdesk, Kursbetreuerin
Auslöser:	Eine Person des Supports möchte eine Person löschen.
Vorbedingungen:	Benutzer existiert.
Ergebnisse:	Die Person ist mit all seinen Daten vom System entfernt worden.
Nachbedingungen:	Benutzer kann nicht mehr auf dem System arbeiten. Er ist auf dem System nicht mehr vorhanden.

Tab. 6.127: Systemanwendungsfall *Person löschen*

Name:	Persönliche Daten editieren
Nummer:	128
Kurzbeschreibung:	Die persönlichen Daten eines Benutzers werden verändert. Darunter sind zu verstehen die Personalien wie die individuellen Einstellungen des Systems. Der Support und Helpdesk können dies für beliebige Personen tun.
Akteure:	Benutzer, Support, Helpdesk
Auslöser:	Ein Benutzer möchte die persönlichen Daten ändern.
Vorbedingungen:	Benutzer existiert.
Eingehende Informationen:	Personalien, persönliche Einstellungen.
Ergebnisse:	Die persönlichen Daten sind geändert.

Tab. 6.128: Systemanwendungsfall *Persönliche Daten editieren*

Name:	Passwort setzen
Nummer:	129
Kurzbeschreibung:	Ein Benutzer setzt sich selbst ein neues Passwort.
Akteure:	Benutzer
Auslöser:	Ein Benutzer möchte sein Passwort ändern.
Vorbedingungen:	Person existiert.
Eingehende Informationen:	Neues Passwort
Ergebnisse:	Das neue Passwort ist aktiviert.
Nachbedingungen:	Altes Passwort ist nicht mehr gültig.

Tab. 6.129: Systemanwendungsfall *Passwort setzen*

Name:	Fremdes Passwort setzen
Nummer:	130
Spezialisierung von:	Passwort setzen
Kurzbeschreibung:	Eine neues Systempasswort wird gesetzt. Support und Helpdesk können dies für beliebige Benutzer tun, eine Kursbetreuerin lediglich für Kursteilnehmer und ein Coach lediglich für Personen welche er betreut. Passwörter von Personen mit speziellen Berechtigungen (Helpdesk, Support, Systemadministration) können nur durch ein zusätzliches Systempasswort geändert werden.
Akteure:	Support, Helpdesk, Kursbetreuerin, Coach
Auslöser:	Eine Person des Supports möchte einem Benutzer ein neues Passwort setzen.
Vorbedingungen:	Person existiert.
Eingehende Informationen:	Neues Passwort
Ergebnisse:	Das neue Passwort ist aktiviert und die Person über die Änderung informiert.
Nachbedingungen:	Altes Passwort ist nicht mehr gültig.

Tab. 6.130: Systemanwendungsfall *Fremdes Passwort setzen*

Name:	Rolle imitieren
Nummer:	131
Kurzbeschreibung:	Die Systemumgebung einer Person wird simuliert. Die betroffene Person muss diesem Prozess zustimmen und kann ihn wieder abbrechen.
Akteure:	Support, Helpdeks, Kursbetreuerin, Coach
Auslöser:	Eine Person des Supports möchte eine Rolle einer Person imitieren.
Vorbedingungen:	Verhältnis der Rollen entspricht Regeln, welche die Simulation bestimmt. Lerner können keine anderen Lerner simulieren, Coaches können Lerner ihrer Klasse simulieren etc.
Eingehende Informationen:	Begründigung, Nachricht für Simulierten.
Ergebnisse:	Die Rollenimitierung ist beendet. Der Benutzer ist über den Vorfall informiert und weiss, dass die Imitierung beendet ist.
Nachbedingungen:	Benutzer hat keinen Zugriff mehr auf die imitierte Person.

Tab. 6.131: Systemanwendungsfall *Rolle imitieren*

Name:	Rollenimitierung akzeptieren
Nummer:	132
Kurzbeschreibung:	Es wird einem anderen Benutzer erlaubt, die private Systemumgebung eines Benutzers zu sehen und diese zu manipulieren. Die Rollenimitation kann jederzeit abgebrochen werden.
Akteure:	Benutzer
Auslöser:	Der Benutzer wird angefragt, ob er einer Rollenimitation zustimmt.
Vorbedingungen:	Verhältnis der Rollen entspricht Regeln, welche die Simulation bestimmt. Lerner können keine anderen Lerner simulieren, Coaches können Lerner ihrer Klasse simulieren etc.
Eingehende Informationen:	Imitationsanfrage
Ergebnisse:	Die Imitation ist abgeschlossen.
Nachbedingungen:	Zugriff ist wieder gesperrt, wie vor der Imitation.

Tab. 6.132: Systemanwendungsfall *Rollenimitierung akzeptieren*

Name:	Rolle zuweisen
Nummer:	133
Kurzbeschreibung:	Einem Benutzer wird eine neue und zeitlich begrenzte Rolle zugewiesen. Rollen können sein: Systemadministratorin, Support, Helpdesk, Kursbetreuerin, Professor oder Autorin oder andere in dem Metadaten des Systems definierten Rollen. Die Rollen eines Coaches oder Lerner entstehen automatisch. Beispielsweise entsteht die Rolle des Coaches durch die Übernahme einer Klasse, ein Coach ohne Klasse gibt es nicht. Hingegen berechtigt die Rolle der Kursbetreuerin überhaupt einen neuen Kurs zu gestalten. Eine Person kann nicht mehr Rechte in Form von Rollen zuweisen als sie selbst besitzt.
Akteure:	Support, Helpdesk, Kursbetreuerin, Professor, Autorin
Auslöser:	Eine Person des Supports möchte einem Benutzer eine neue Rolle zuweisen.
Vorbedingungen:	Rolle existiert.
Eingehende Informationen:	Benutzerwahl, Beschreibung, Zeitraum der Gültigkeit
Ergebnisse:	Der Benutzer hat die Rechte, welche mit dieser Rolle verbunden sind und ist über den Vorfall informiert.
Nachbedingungen:	Rolle wird aktiv sobald Zeitraum gültig ist.

Tab. 6.133: Systemanwendungsfall *Rolle zuweisen*

Name:	Rolle verändern
Nummer:	134
Kurzbeschreibung:	Die Rollendefinition eines Benutzers wird verändert. Eine zugeteilte Rolle kann nicht gelöscht werden, hingegen kann die Zeitbegrenzung so verändert werden, dass die Rolle inaktiv wird. Die Rolle wird nicht gelöscht, um Transparenz bezüglich der Vergangenheit zu schaffen.
Akteure:	Support, Helpdesk, Kursbetreuerin, Professor, Autorin
Auslöser:	Eine Person des Supports möchte die Rollendefinition eines Benutzers ändern.
Vorbedingungen:	Rolle existiert.
Ergebnisse:	Die Rolle des Benutzers ist geändert und dieser ist über die Änderung informiert.

Tab. 6.134: Systemanwendungsfall *Rolle verändern*

6.12 Systemverwaltung

Name:	System starten
Nummer:	135
Kurzbeschreibung:	Das System wird von Grund auf neu gestartet. Alle notwendigen Dienste werden initialisiert.
Akteure:	Systemadministratorin
Auslöser:	Eine Systemadministratorin möchte das System starten.
Eingehende Informationen:	Modus: Wartung, normal
Ergebnisse:	Die Benutzer können auf dem System arbeiten.

Tab. 6.135: Systemanwendungsfall *System starten*

Name:	System stoppen
Nummer:	136
Kurzbeschreibung:	Der Betrieb des laufenden Systems wird unterbrochen. Nachdem das System den Benutzern nicht mehr zugänglich ist werden die Daten so abgelegt, dass nach einem Start wieder der selbe Systemzustand erreicht werden kann.
Akteure:	Systemadministratorin
Auslöser:	Eine Systemadministratorin möchte das System stoppen.
Vorbedingungen:	System läuft.
Eingehende Informationen:	Mitteilung an Benutzer welche zur Zeit online sind.
Ergebnisse:	Das System läuft nicht mehr. Alle Daten sind gespeichert.
Nachbedingungen:	Das System kann wieder gestartet werden.

Tab. 6.136: Systemanwendungsfall *System stoppen*

Name:	Systemdaten importieren
Nummer:	137
Kurzbeschreibung:	Auf ein noch nicht benutztes System werden Daten eines anderen Systems eingespielt. Anschliessend werden die Metadaten des Systems geprüft und angepasst.
Akteure:	Systemadministratorin
Auslöser:	Eine Systemadministratorin möchte Systemdaten auf einem neuen System importieren.
Vorbedingungen:	System läuft, enthält jedoch noch keine Daten.
Eingehende Informationen:	System Metadaten
Ergebnisse:	Das System ist für die Benutzer zugänglich.

Tab. 6.137: Systemanwendungsfall *Systemdaten importieren*

Name:	System Metadaten editieren
Nummer:	138
Kurzbeschreibung:	Die Systemparameter und die lokalen Gegebenheiten werden verändert.
Akteure:	Systemadministratorin
Auslöser:	Eine Systemadministratorin möchte die System Metadaten verändern.
Vorbedingungen:	System läuft.
Eingehende Informationen:	System Metadaten
Ergebnisse:	Die Änderungen sind wirksam.

Tab. 6.138: Systemanwendungsfall *System Metadaten editieren*

Name:	Systemdaten exportieren
Nummer:	139
Kurzbeschreibung:	Der aktuelle Zustand und die Daten des Systemes werden in einer Art und Weise exportiert, dass ein anderes System diese Daten wieder importieren kann.
Akteure:	Systemadministratorin
Auslöser:	Eine Systemadministratorin möchte das System exportieren.
Vorbedingungen:	System läuft und enthält Daten.
Ergebnisse:	Das System ist in Dateien auf der Festplatte abgebildet.
Nachbedingungen:	Diese Daten können in einem anderen System importiert werden.

Tab. 6.139: Systemanwendungsfall *Systemdaten exportieren*

Name:	System updaten
Nummer:	140
Kurzbeschreibung:	Komponenten des Systems werden durch neuere Versionen ersetzt.
Akteure:	Systembetreuerin
Auslöser:	Eine Systembetreuerin möchte das System updaten.
Vorbedingungen:	System läuft.
Eingehende Informationen:	Neue Systemkomponente. Mitteilung an Benutzer.
Ergebnisse:	Die alte Version der Software ist entfernt und das System läuft mit der neuen Version.

Tab. 6.140: Systemanwendungsfall *System updaten*

Name:	System überwachen
Nummer:	141
Kurzbeschreibung:	Statistiken über die Auslastung und die Verhaltensweisen der Systembenutzer werden berechnet. Auf Überlastungen des Systems wird speziell hingewiesen.
Akteure:	Systembetreuerin, Support
Auslöser:	Eine Systembetreuerin möchte den Betrieb des Systems überwachen
Vorbedingungen:	System läuft.
Ergebnisse:	Die Statistiken sind berechnet und die Systembetreuerin informiert.

Tab. 6.141: Systemanwendungsfall *System überwachen*

7. ANFORDERUNGSSPEZIFIKATION

Im vorangehenden Kapitel wurde ein Überblick über die Funktionalität des Systems aus der Perspektive der Benutzerinnen und Benutzer gegeben. Dies genügt jedoch nicht um die Anforderungen an das System vollständig zu beschreiben. Jeder Anwendungsfall ist ein in sich geschlossenes Szenario des Verhaltens eines Benutzers, welches ohne den Gesamtzusammenhang des Systems betrachtet wurde. In diesem Kapitel wird versucht diejenigen Anforderungen aufzulisten, die nicht direkt in einem Anwendungsfall sichtbar werden, sondern von allgemeiner Gültigkeit sind.

Anforderungen werden in dem praxisorientierten Vorgehen von Oestereich eingestuft nach dem Verbindlichkeitsgrad, der Anforderungsart, dem Detaillierungsgrad und der Priorität. Im Verbindlichkeitsgrad nennt Oestereich die vier Stufen Pflicht, Optional, Absichten und Vorschläge. Bei den Anforderungsarten unterscheidet er zwischen Problemen, Zielen, funktionalen Anforderungen, Randbedingungen, Qualitätsanforderungen, Testanforderungen und Abnahmekriterien.[Oes01]

Im Folgenden werden die wichtigsten Anforderungen an das System aufgelistet. Die Liste ist nach dem Verbindlichkeitsgrad geordnet.

7.1 Funktionale Anforderungen

Die funktionalen Anforderungen ergeben sich hauptsächlich aus den im Kapitel 6 aufgelisteten Anwendungsfällen. Zu diesen kommen die folgenden Anforderungen:

Nummer:	Anforderung
142 (Pflicht)	Alle in den <i>Anwendungsfällen</i> beschriebenen Funktionen sind durch einen fein abgestimmten Sicherheitsmechanismus zu schützen. Die Berechtigungen werden durch sogenannte Rollen definiert. Die Rollen beziehen sich immer auf ein Objekt. Ausserhalb des Objektes hat jede Teilnehmerin lediglich die Rechte eines Benutzers. Rollen können auch Subrollen haben, zum Beispiel für den Stellvertreter eines Coaches.

Fortsetzung auf nächster Seite

Fortsetzung von letzter Seite: Funktionale Anforderungen

Nummer:	Anforderung
143 (Pflicht)	Die in der IMS Enterprise Model Spezifikation verwendeten <i>Rollen</i> müssen im System abgebildet sein.[CVA99b] [CV99] [CVA99a]
144 (Pflicht)	<i>Benutzerdaten</i> werden gemäss der IMS Learner Information Package Specification implementiert[STR01a].
145 (Pflicht)	<i>Gruppen und Klassen</i> werden gemäss der IMS Enterprise Specification implementiert[CVA99b]
146 (Pflicht)	Das <i>SCORM</i> dient als <i>Basis</i> für die Implementation von Kursen und Inhalt. Es wird in der Art erweitert, dass neben den normalen <i>Sharable Ressources</i> und <i>Sharable Content Objects</i> auch <i>Assessments</i> enthalten kann. Diese Erweiterung ist in SCORM bereits vorgesehen.[D ⁺ 01b] Die <i>IMS Resource Meta-Data Specification</i> wird voll unterstützt. [MT01b]
147 (Pflicht)	Die im Content eingebetteten Tests basieren auf der <i>IMS Question and Test Interoperability</i> Spezifikation. Im Export und Import kommen die dafür vorgesehenen XML-Bindings zur Anwendung.[SSBL01b]
148 (Pflicht)	Die in Content eingebetteten auf IMS basierenden Tests werden nicht gemäss SCORM auf Klientseite mit JavaScript implementiert, sondern durch ein Testsystem auf <i>Serverseite</i> ausgewertet. Dies garantiert wahrheitsgemässe Resultate. [D ⁺ 01c][SSBL01b]
149 (Pflicht)	Die im <i>SCORM Runtime-Environnement</i> definierten JavaScript Funktionen müssen trotzdem implementiert sein um SCORM Konformität zu erreichen. Für Tests im Sinne von Self-Assessments sind diese Funktionen nützlich.[D ⁺ 01c]
150 (Pflicht)	Eine Autorin muss ihre Contentelemente in einem <i>Content-Preview-Modus</i> betrachten können.
151 (Pflicht)	Eine Kursbetreuerin muss ihren Kurs in einem <i>Kurs-Preview-Modus</i> betrachten können.
152 (Pflicht)	Alle Dateiformate, welche für Export und Import verwendet wird, halten sich an die <i>IMS Content Packaging Specification</i> . [AM01b]
153 (Pflicht)	Die Prozesse der Kommunikation und der Dokumentenverwaltung müssen <i>jederzeit und kontextsensitiv</i> verfügbar sein und direkt angewandt werden können. Ausnahmen stellen lediglich interaktive Seiten wie während eines Testes oder einer Evaluation dar. Konkret bedeutet dies, dass in einer Contentseite eine private Notiz angebracht oder gar ein ganzes Blackboard an eine Seite angehängt werden kann, um über den Inhalt dieser Seite zu diskutieren.

Fortsetzung auf nächster Seite

Fortsetzung von letzter Seite: Funktionale Anforderungen

Nummer:	Anforderung
154 (Pflicht)	Verwaltung von <i>beliebig vielen Kursen und Contentelementen</i> .
155 (Pflicht)	Verwaltung von <i>beliebig vielen Benutzern und Gruppen</i> .
156 (Pflicht)	Ist ein Kurs abgelaufen oder hat der Lerner genügend Punkte erreicht, so generiert das System automatisch ein <i>Zertifikat</i> . Das Zertifikat wird elektronisch signiert.
157 (Absicht)	Der <i>Datenaustausch</i> zu andern Informationssystemen der Universität (SAP / UniAccess) soll nach Möglichkeit gemäss der IMS Enterprise Model Spezifikation implementiert werden. [CVA99b] [CV99] [CVA99a]
158 (Absicht)	<i>Mehrere Installationen</i> sollen miteinander kommunizieren können, um Kurse installationsübergreifend anbieten zu können, ohne dass sich die Benutzer auf mehreren Systemen registrieren müssen. Aus Sicht der Benutzer verhält sich das System vollkommen transparent wie ein einziges System. Dies bezieht sich nicht nur auf die Gruppen und Kursinformationen sondern auch auf die ganzen Kommunikationsmöglichkeiten. Soweit möglich soll dafür auf die IMS Enterprise Model Spezifikation zurückgegriffen werden. [CVA99b] [CV99] [CVA99a]

Tab. 7.1: Funktionale Anforderungen

7.2 Anforderungen an die Benutzbarkeit

Nummer:	Anforderung
159 (Pflicht)	Das System ist <i>webbasiert</i> , das heisst, kann mittels eines Webrowsers bedient werden. Die Darstellung erfolgt über XHTML 1.1, CSS 2.0 und JavaScript 1.2 (ECMA-262). OLAT soll mit allen gängigen Webbrowsern bedient werden können.
160 (Pflicht)	Die Lernumgebung ist <i>personalisiert</i> . Das heisst, jede Seite wird für den Benutzer zusammengestellt, so wie es sich der Benutzer wünscht. Kommentare, Dokumente, Blackboards können eingeblendet oder ausgeblendet werden. Verschiedene Elemente der Darstellung (Menu, Navigation, Dokumentmanagement, Kommunikationsmöglichkeiten) können nach Belieben plziert werden. Zwischen verschiedenen Layouts (Farben, Graphiken) kann ausgewählt werden. Diese Einstellungen werden gespeichert.

Fortsetzung auf nächster Seite

Fortsetzung von letzter Seite: Funktionale Anforderungen

Nummer:	Anforderung
161 (Pflicht)	Es muss möglich sein, <i>mehrere Rollen gleichzeitig</i> wahrzunehmen. Das heisst, es soll möglich sein mehrere Fenster gleichzeitig geöffnet zu haben ohne dass eine Aktion in einem Fenster den Systemstatus des anderen Fensters beeinflusst.
162 (Pflicht)	Sogenannte <i>Themes</i> werden unterstützt, auf System- wie auch auf Kursebene. Themes können in den entsprechenden Metadaten vordefiniert werden, jeder Benutzer kann sich jedoch ein eigenes Theme auswählen.
163 (Pflicht)	Es sind Vorkehrungen zu treffen um das System in <i>mehreren Sprachen</i> benutzen zu können. Die Erweiterung um eine zusätzliche Sprache muss so gestaltet sein, dass hierfür keine Programmierkenntnisse notwendig sind, z.B. durch das Auslagern der Systemnachrichten in ein XML Dokument. Die Mehrsprachigkeit bezieht sich auf alle Dialoge und sonstigen Anzeigen des Systems inklusive der Hilfetexte, nicht aber auf Content oder Kurse. Sollen diese mehrsprachig abgehalten werden, so ist dies Sache der Kursbetreuerin. Eine Unterstützung hierfür ist nicht vorgesehen.
164 (Pflicht)	Es existiert ein <i>Benutzerhandbuch</i> , welches die Funktionalität der verschiedenen Rollen beschreibt. Das Handbuch existiert mindestens auf Deutsch und auf Englisch.
165 (Pflicht)	Der Benutzer kann jederzeit eine <i>Onlinehilfe</i> aufrufen. Die Hilfe ist mit einem geeigneten Symbol gekennzeichnet und zeigt einen in diesem Kontext relevanten Text an.
166 (Absicht)	Soweit praktikabel sollen die IMS Guidelines for Developing Accessible Learning Applications angewendet werden, um OLAT <i>behindertengerecht</i> zu gestalten.[BNRM01]

Tab. 7.2: Anforderungen an die Benutzbarkeit

7.3 Anforderungen an die Zuverlässigkeit, Effizienz und Sicherheit

Nummer:	Anforderung
167 (Pflicht)	Nach dem Stellen einer Anfrage beziehungsweise dem Klicken auf einen Link soll die durchschnittliche <i>Antwortzeit</i> nicht länger als zwei Sekunden dauern.
168 (Pflicht)	Dauert das Bearbeiten einer Anfrage <i>länger als fünf Sekunden</i> , so wird der Benutzer über die Überlastung des Servers oder über die sehr rechenintensive Anfrage informiert. Es ist ein Mechanismus zu entwerfen, welcher die synchrone HTTP-Anfrage in eine asynchrone umwandelt.

Fortsetzung auf nächster Seite

Fortsetzung von letzter Seite: Funktionale Anforderungen

Nummer:	Anforderung
169 (Pflicht)	Alle Downloads werden <i>asynchron</i> umgesetzt. In einem Downloadbereich können beantragte Dokumente dann effektiv heruntergeladen werden, aber erst, wenn sie komplett fertig erstellt sind. Eine Benachrichtigung nach Fertigstellung der Berechnung soll möglich sein.
170 (Pflicht)	Alle Elemente des Systems müssen <i>redundant</i> vorhanden sein, so dass ein Ausfall des gesamten Systems wegen eines Teilausfalls ausgeschlossen werden kann. Dies ist auf die physischen so wie softwaretechnischen Elemente zu beziehen.
171 (Pflicht)	Die einzelnen Teile des Systems sind soweit von einander zu <i>entkoppeln</i> , dass das System auch nach einem Teilausfall weiterhin mit reduzierter Funktionalität funktionieren kann.
172 (Pflicht)	Jede Anfrage an das System wird mit einer <i>Statusmeldung</i> quittiert welche dem Benutzer angezeigt wird. Sie gibt Auskunft darüber, ob die Anfrage korrekt abgearbeitet werden konnte.
173 (Pflicht)	Konnte eine Anfrage nicht korrekt abgearbeitet werden, so ist eine unübersehbare <i>Fehlermeldung</i> auszugeben. Die Fehlermeldung enthält Angaben darüber, was hätte passieren sollen, was alles nicht passiert ist und wie sich der Benutzer nun zu verhalten hat.
174 (Pflicht)	Fehler, welche aufgrund mangelnder Zugriffsrechte oder eines internen Problems (z.B. keine Datenbankverbindung) zustande gekommen sind, werden umgehend an die Systemadministratorin geleitet und in einem <i>Errorlog</i> festgehalten. Andere Fehler werden in das Errorlog geschrieben jedoch nicht speziell gemeldet.
175 (Pflicht)	In allen kritischen Bereichen werden <i>Logaufzeichnungen</i> über die Aktionen der Benutzer gemacht. Diese sind auf einem physikalisch getrennten System zu halten und durch spezielle Sicherheitsvorkehrungen zu schützen. Im Notfall könnte aus diesen Daten die genauen Aktionen von Benutzern rekonstruiert werden. Als kritischer Bereich gilt insbesondere das Test- und Bewertungssystem.
176 (Pflicht)	Skalierung bis zu <i>1000 concurrent Benutzern</i> (zeitgleiches Anfordern von Informationen, z.B. durch anwählen eines Links). Auch wenn zwei Personen gleichzeitig auf einem System aktiv sind fallen Anfragen meist nicht auf den selben Moment hin an. 1000 concurrent Benutzer bedeutet also, dass ein Vielfaches davon an Benutzern gleichzeitig arbeiten kann ohne Performanceeinbußen zu spüren.

Fortsetzung auf nächster Seite

Fortsetzung von letzter Seite: Funktionale Anforderungen

Nummer:	Anforderung
177 (Pflicht)	Alle Netzwerkverbindungen vom Client zum System, über welche personenbezogene Daten verschickt werden, sind zu <i>verschlüsseln</i> . Dies trifft insbesondere zu bei Evaluationen, den Tests und dem Anzeigen und Ändern von persönlichen Daten. Der normale Betrieb (Bearbeiten eines Contentelementes etc.) muss nicht verschlüsselt werden.
178 (Pflicht)	Alle Netzwerkverbindungen vom OLAT-System zu seinen Teilsystemen, zu anderen Informationssystemen der Universität oder zu anderen OLAT-Systemen an anderen Universitäten sind <i>komplett zu verschlüsseln</i> . Hier sollte eine Hardwarelösung ins Auge gefasst werden um den Rechenaufwand bewältigen zu können.

Tab. 7.3: Anforderungen an die Zuverlässigkeit, Effizienz und Sicherheit

7.4 Supportanforderungen

Nummer:	Anforderung
179 (Pflicht)	Ein Plugin-Mechanismus ist vorzusehen, welcher die <i>Erweiterbarkeit</i> der Plattform garantiert. Dies gilt auf Ebene der Businesslogik wie auf der Ebene der Benutzerapplikation. Es soll einfach möglich sein neue Komponenten oder Services in die bestehende Struktur einzubinden.
180 (Pflicht)	Durch Verwendung von Kapselungstechniken und Programmier-API's soll eine <i>Änderbarkeit</i> des Codes erreicht werden. Die Daten müssen von den Prozessen auf Programmebene getrennt werden, damit diese sich unabhängig voneinander weiterentwickeln lassen. Entwickelt wird streng nach Design by Contract.
181 (Pflicht)	Um das System <i>wartbar</i> zu machen ist genügend Dokumentation zu betreiben. Die Dokumentation entsteht immer parallel zur Entwicklung und nicht hinterher. Zu jeder Komponente existiert ein komplettes Klassen- und Ablaufdiagramm sowie weitergehende Erläuterungen. Alle Änderungen werden nachvollziehbar dokumentiert.
182 (Pflicht)	Für die Systemadministratorin wird ein <i>Handbuch</i> mit Kapiteln über die Installation, die Konfiguration und den Betrieb des Systemes geschrieben. Das Dokument ist in Englisch abgefasst.

Tab. 7.4: Supportanforderungen

8. GESCHÄFTSKLASSEN

Die Abbildung 8.1 zeigt die wichtigsten Geschäftsklassen des Lernsystems in einem Überblick. Mit Geschäftsklassen sind nicht Klassen im programmiertechnischen Sinn gemeint, sondern die Beschreibung von Gegenständen, Konzepten, Orten oder Personen des realen Geschäftslebens. Es kann aber durchaus sein, dass im späteren Klassenmodell dieselben Klassennamen wieder auftauchen werden, andere werden verfeinert und weiter unterteilt werden müssen. [Oes01]

Die Geschäftsklassen werden in diesem Schritt ohne tiefen Detaillierungsgrad dargestellt. Attribute und Operationen werden gänzlich weggelassen. Im Vordergrund der Betrachtung liegen die grundsätzlichen strukturellen Zusammenhänge und die Beziehungen zwischen den Klassen auf einem hohen Abstraktionsniveau. Objektorientierte Mechanismen wie Vererbung oder die Assoziationsrichtung werden vernachlässigt. Es sind lediglich die wichtigsten Beziehungen genannt. Neben dem Rollennamen ist auch die jeweilige Multiplizität angegeben.

In dem Klassendiagramm wurden die Benutzerrollen *Helpdesk*, *Support* und *Systemadministratorin* weggelassen. Diese Rollen sind für rein administrative Zwecke vorhanden und tragen nichts zum eigentlichen Geschäft bei.

Um die Schnittstelle gegenüber dem SAP Campus System besser zum Ausdruck zu bringen wurde eine Geschäftsklasse *SAP-Benutzer* geschaffen. Die Synchronisation aller Daten des OLAT Systems mit dem SAP System sollen über diese Klasse laufen. Dies sind hier vor allem personenbezogene Daten wie die Personalien (synchronisiert mit *Benutzer*), Einschreibebogen (synchronisiert mit *Belegung*) und Testate oder APS-Punkte (synchronisiert mit *Zertifikat*). Der Einfachheit halber wurde für die Synchronisation des Lehrangebotes (synchronisiert mit *Kurs*) und der Struktur der Institute und Abteilungen (synchronisiert mit *Org.Einheit*)) ebenfalls diese Geschäftsklasse verwendet.

Im Zentrum des Klassendiagramms befindet sich der Kurs. Um diesen herum sind die verschiedenen Akteure abgebildet: *Lerner*, *Coach*, *Kursbetreuerin*, *Professor* und *Gruppe*. Die Zuordnung von Lerner und Coach zu Klassen und Gruppen erfolgt über *Belegung* und *G.Mitglied*. Der eigentliche Inhalt eines Kurses ist auf der linken (*Content* und *Assessment*) und der rechten Seite (*Evaluation*) zu finden.

Es gibt keine direkte Verbindung zwischen Tutor und Lerner. Diese erfolgt über die gemeinsame Gruppenmitgliedschaft oder Klassenbelegung, natürlich in unterschiedlicher Rollenfunktion. Die beiden Entitäten kommunizieren hauptsächlich über Nachrichten, darum wurde eine spezielle Assoziation zwischen *Lerner*, *Coach* und *Nachricht* in das Diagramm eingezeichnet. Eine zweite Verbindung ist durch das Erarbeiten einer *Lö-*

sung und anschließenden *Bewertung* derselben auszumachen. Eine Lösung basiert auf einer Aufgabenstellung, welche in einem *SCO*¹ beschrieben ist. Eine Lösung ist einem Lerner oder einer Gruppe zugeordnet, je nach Aufgabenstellung.

Die Lernmaterialien sind auf der linken Seite des Diagramms zu finden. Einerseits gibt es den *Content* mit seinen Elementen *Block*, *SCO* und *Asset*, andererseits das *Assessment* mit den Elementen *Section* und *Item*. Die Items bestehen ihrerseits wiederum aus *Question* und *Response*. Ein Lerner löst einen *Test*, welcher auf einem Assessment, der Testschablone, beruht. In diesem Test gibt er *Testantworten* auf Questions. Die Testantwort ist vom Typ einer der zur Verfügung stehenden Responses. Assessments und Contentelemente werden in den Kurs eingebunden und sind dem Lerner somit zugänglich. Die Assoziation von Gruppe zu Content symbolisiert, dass Content auch spezifisch einzelnen Gruppen zugeteilt werden kann.

Unten rechts in der Graphik sind die Klassen *Nachricht*, *Blackboard*, *B.Nachricht* und *Dokument* zu finden. Es wurde darauf verzichtet die speziellen Arten von Nachrichten (Kursnachricht, Klassennachricht, Gruppennachricht) abzubilden. Lediglich die Kommunikation zwischen Tutor und Coach ist separat im Diagramm vermerkt. Auch das Addressbuch wurde für die Vereinfachung weggelassen. Da Diskussionen über beliebige Objekte geführt werden sollen, gibt es Assoziationsbeziehungen von Kurs, Gruppe, Klasse und SCO zu Blackboard. Die Nachrichten in einem Blackboard sind als *B.Nachricht* im Diagramm aufgenommen worden. Sie können ihrerseits B.Nachrichten in Form von Antworten und Dokumenten enthalten. Die Klasse *Dokument* wurde nicht mehr weiter verfeinert in interne und externe Dokumente, wie dies in den Anwendungsfällen beschrieben ist. Dokumente sind einem Benutzer oder einer Gruppe zugeordnet. Sie können in Lösungen oder Nachrichten integriert sein.

Die vier Geschäftsklassen der *Evaluation* sind in sich abgekoppelt und mit einem Kurs und dem Lernenden verbunden. Die Evaluation besteht aus *Eval.fragen*, welche vom Lernenden mit *Eval.resultat* beantwortet werden. Die Antworten sind zusammengefasst als *Eval.bogen*.

¹ Sharable Content Object. Siehe Kapitel 4.3.1

9. ANWENDUNGSARCHITEKTUR

9.1 Anwendungsschichten

Zunächst stellt sich die Frage, wie das neue System in Schichten (Tier) unterteilt werden soll. Diese Schichtung wird auch *virtuelle Maschine Metapher* genannt. Das Schichtenprinzip bietet viele Vorteile gegenüber einem ungeschichteten Modell. Jede Schicht ist in sich geschlossen und ist nur über eine klare Schnittstellen ansprechbar. Somit kann man eine Schicht verstehen ohne die anderen Schichten ebenfalls verstehen zu müssen, eine Schicht ist zudem auch austauschbar. Die Abhängigkeiten zwischen Schichten werden minimiert, wodurch ein besser entkoppeltes System entsteht. Dem gegenüber stehen Performanceeinbußen aufgrund der zusätzlichen Arbeitsschritte. Ausserdem können sich Änderungen in unteren Schichten kaskadierend über alle Schichten hinwegziehen.[Fow]

In den meisten Client-Server Systemen geht man von einem dreischichtigen Modell aus: Präsentationsschicht, Geschäftslogik und Infrastruktur. Wie schon in Kapitel 1.3.4 angesprochen soll das hier entworfene System mit der Java 2 Enterprise Edition Architektur gebaut werden. Eine auf J2EE optimierte Schichtung ist in *Core J2EE Patterns* zu finden. Sie besteht aus fünf Schichten, die in der Abbildung 9.1 dargestellt sind: [ACM01]

1. *Client*

Diese Schicht repräsentiert alle Geräte oder Systeme, welche auf das System zugreifen: Webbrowser, Java-Applikation, WAP-Gerät etc.

2. *Präsentation*

In dieser Schicht wird die Präsentationslogik gekapselt. Sie ist dringend notwendig um ein HTML basiertes System zu realisieren. In einer dreischichtigen Architektur ist diese Funktionalität entweder in der Geschäftslogik oder dem Client zu integrieren. Dies führt jedoch zu einem schlechten Design, denn das Generieren einer HTML Benutzerschnittstelle gehört weder zur Geschäftslogik noch findet sie auf dem Client statt. Der Client ist lediglich für die Darstellung der in HTML definierten Elemente zuständig. Diese Schicht wird auch *Web-Tier* genannt, denn sie ist speziell bei webbasierten Architekturen anzutreffen.

3. *Geschäftslogik*

Hier sind die eigentliche Geschäftslogik und die Geschäftsdaten zu finden. Informationen über die Darstellung und Datenhaltung dürfen in dieser Schicht nicht enthalten sein. J2EE bietet hier Unterstützung durch das Bean Komponentenkonzept.

4. *Integration*

Diese Schicht kapselt die verschiedenen Ressourcen von der Geschäftslogik ab,

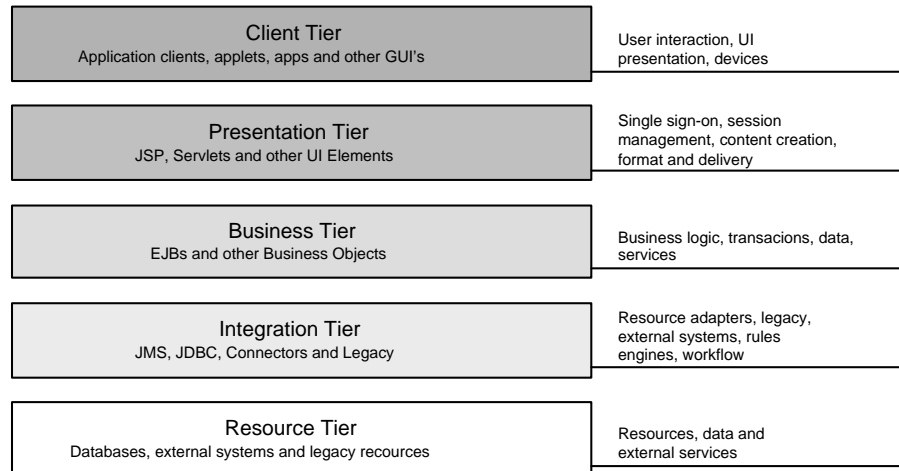


Fig. 9.1: Die J2EE Anwendungsschichten (Quelle: Core J2EE Patterns [ACM01])

so dass in der Geschäftslogik einheitlich darauf zugegriffen werden kann. Dies ist Teil des J2EE Konzepts, welches dem Applikationsentwickler verbieten möchte, dass dieser direkt auf Ressourcen zugreift. Die Anwendung dieses Konzeptes führt zu einer besseren Entkopplung und somit zu besserer Skalierbarkeit und Portabilität des ganzen Systems.

5. Ressourcen

Die unterste Schicht umfasst die eigentliche Datenspeicherung und externe Ressourcen wie Mainframes oder Kreditkarten-Autorisierungssysteme und Ähnliches. Die Integrations- und die Ressourcenschicht können auch zusammengefasst und mit *EIS-Tier* bezeichnet werden (Enterprise Information System).

Im hier beschriebenen System wird auf dieses Schichtungsprinzip zurückgegriffen.

9.2 Komponenten

Die in Kapitel 8 identifizierten Geschäftsklassen werden nun zu fachlichen Komponenten zusammengefasst, um eine bessere Abstraktion zu erlangen. Die Komponenten sind nur über eine klar definierte Schnittstelle zugänglich, der innere Aufbau bleibt verborgen. Die Abhängigkeiten zwischen Komponenten soll dabei minimal sein, um einen hohen Grad an Entkopplung zu erreichen.

Die Abbildung 9.2 zeigt die Komponenten der wichtigsten Entitäten des neuen Systems. Sie wurden direkt in das Geschäftsklassendiagramm eingezeichnet. Die folgenden Komponenten wurden gebildet:

- *Assessment*

Die Komponente Assessment beinhaltet die Geschäftsklassen Question, Response, Item, welche ihrerseits die Komponente *Item* bilden, sowie Section, Assessment und (Assessment-) Autorin. Sie verfügt über zwei Schnittstellen: Eine für die Autorenfunktionen und eine für die Tests der Lerner.

- *Test*
Die Komponente Test besteht aus den Klassen Testantworten und Test. Sie bezieht sich auf die Assessment Komponente und gehört zu einem Lerner.
- *Content*
Die Klassen Asset, SCO und Block, (Content-) Autorin und Content werden zu der Komponente Content zusammengefasst. Intern sind Asset und SCO gekapselt zu einer eigenen Unterkomponente *SCO*. Auch hier existieren zwei Schnittstellen, einerseits für die administrativen Funktionen und andererseits für die Funktionen zur Bearbeitung der Lerninhalte im Sinne von Lernen.
- *Kurs*
Die komplexe Komponente Kurs umfasst die Geschäftsklassen Kursbetreuerin, Professor, Kurs, Gruppentyp, Gruppe, G.Mitglied, Klasse, Belegung und Coach. Innerhalb dieser Komponente gibt es weiter die Unterkomponenten *Gruppe* und *Klasse*. Diese Komponenten existieren aber immer nur im Zusammenhang mit einem Kurs und sind daher innerhalb der Komponente Kurs angesiedelt. Schnittstellen stehen für die Administration und für die Organisation der Klassen und Gruppen zur Verfügung.
- *Lerner*
Die Komponente Lerner widerspiegelt alle Informationen, die über einen Lerner während dem Besuch eines Kurses anfallen. Daher sind die Klassen Lerner, Lernpfad, Lösung, Bewertung und Zertifikat in dieser Komponente enthalten.
- *Evaluation*
Die Komponente Evaluation beinhaltet die Klassen Evaluation, Eval.fragen, Eval.bogen und Eval.resultat. Eine Schnittstelle steht für die administrativen Funktionen zur Verfügung, eine andere um die Evaluation durchzuführen zu können.
- *Nachricht*
Die Komponente Nachricht besteht nur aus der Klasse Nachricht selbst. Eine Zusammenfassung mit anderen Klassen scheint nicht sinnvoll.
- *Blackboard*
Die Komponente Blackboard setzt sich aus der Geschäftsklasse Blackboard und B.Nachricht zusammen.
- *Dokument*
Auch die Komponente Dokument beinhaltet lediglich die Geschäftsklasse Dokument.
- *Organisationseinheit*
Die Hilfsklasse Organ.Einheit bildet eine eigene Komponente.
- *Imitation*
Die Klasse Imitation wird Ebenfalls durch eine eigene Komponente repräsentiert.
- *Benutzer*
Auch der Benutzer als zentrales Element wird in eine eigene Komponente verpackt.

Die Klasse *Raum* wird hier nicht weiter betrachtet. Sie ist nicht in der Komponente Kurs integriert, weil Räume unabhängig von Kursen im System zur Verfügung stehen. Sie ist unkompliziert und wird daher in den weiteren Ausführungen vernachlässigt.

Bei genauerer Betrachtung zeigt sich, dass der *SAP-Benutzer* keine Entität, sondern eine Funktion des Systems ist: eine Abgleichfunktion mit einem externen System. Darum wird daraus keine Entitätskomponente gebildet. In der Architektur finden wir den SAP-Benutzer als *Web Service Controller* und *IMS Enterprise Controller* wieder.

9.3 Architektur

Die Abbildung 9.3 zeigt die Anwendungsarchitektur des Systems OLAT 2. Sie ist gegliedert in die im Kapitel 9.1 vorgestellten fünf Anwendungsschichten Client, Presentation, Business, Integration und Resource. Im Folgenden wird der Aufbau der einzelnen Schichten besprochen.

9.3.1 Client Tier

In dem System sind zwei Typen von Clients auszumachen: Webbrowser und Informationssysteme wie beispielsweise das Verwaltungssystem der Universität SAP Campus. Zwischen der Client- und der Präsentationsschicht liegt das Internet. Bei den anderen Schichten kann dies ebenfalls der Fall sein, dies ist aber nicht zwingenderweise gegeben. Beide nehmen Kontakt mit dem Server über das HTTP Protokoll auf.

9.3.1.1 SAP Campus Client

Externe Informationssysteme wie das SAP Campus System haben einen WebServices Controller implementiert um mit OLAT 2 zu kommunizieren. Der Begriff WebServices steht für den automatisierten Austausch von XML basierten Nachrichten über SOAP (Simple Object Access Protokoll). Von der Idee her gleichen WebServices einem entfernten Methodenaufruf, wie dies mit CORBA, RMI oder RPC gemacht werden kann. Sie ist im Unterschied zu diesen jedoch gänzlich unabhängig von der verwendeten Programmiersprache, dem Programmierparadigma oder dem zugrundeliegenden Datenmodell. Die Nachrichten werden mit XML beschrieben und mit SOAP über HTTP transportiert. Die Spezifikationen rund um SOAP wurden von verschiedenen Herstellern entwickelt, darunter IBM und Microsoft, und wurden vom W3C¹ als internationale Standards verabschiedet.

Für die Integration von externen Systemen wird OLAT 2 also eine WebServices Schnittstelle zur Verfügung stellen müssen. Diese Architektur lässt offen, was für Services effektiv angeboten werden. Zur Zeit ist lediglich ein Datenabgleich gemäss der IMS Enterprise Spezifikation vorgesehen. Die Implementation zusätzlicher Services ist jederzeit problemlos möglich.

9.3.1.2 Webbrowser Client

Die Interaktion der Studierenden, Assistenten oder Professorinnen vollzieht sich über einen Standard-Webbrowser. Dieser verfügt über eine eingebaute Dialogsteuerungs-

¹ <http://www.w3.org/TR/soap12-part1/>

komponente, welche den HTML Code auf dem Bildschirm umsetzt. Damit OLAT 2 SCORM kompatibel ist, muss der SCORM API Adapter² ebenfalls auf dem Client installiert sein. Da die vom API zur Verfügung gestellten Informationen auf dem Server berechnet werden, stellt der SCORM API Adapter lediglich eine entfernte Schnittstelle zu dem richtigen API Controller dar. Diese Komponente könnte als Browser Plugin, als ein JavaScript Programm oder als Java Applet implementiert werden.

Der SCORM API Adapter und die Dialogsteuerung des Webbrowsers kommunizieren über HTTP oder HTTPS mit dem Server. Der Server verhält sich aus Sicht des Webbrowsers wie ein gewöhnlicher Webserver.

9.3.2 Presentation Tier

In der Präsentationsschicht gibt es zwei Controller Komponenten, den Webservice Controller und den Front Controller. Sie stellen die Schnittstellen zu den jeweiligen Clients dar.

9.3.2.1 Webservice Controller

Der Webservice Controller ist ein Front Controller gemäss dem *Front Controller Pattern*. Ein Front Controller nimmt alle Anfragen entgegen und delegiert diese intern weiter (Dispatch). Die Alternative wäre, für jede Funktion des Systems einen eigenen, gegen aussen hin sichtbaren Controller zu implementieren. Dies ist jedoch problematisch, da allgemeine Systemfunktionen wie Déjà-vu Tokens³ oder Authentifizierungs-Filter⁴ redundant implementiert werden müssten. Zur Zeit ist lediglich eine Weiterreichung an den IMS Enterprise Controller vorgesehen.[ACM01]

9.3.2.2 Front Controller

Theoretisch könnte man für die Behandlung aller Client Anfragen nur einen Front Controller einsetzen. Da WebServices Anfragen jedoch auf SOAP basieren, aus Sicherheitsgründen vielleicht sogar auf einem privaten Netzwerk transportiert werden oder aus Gründen der Rechnerbelastung auf einem dedizierten System implementiert sind, ist es durchaus sinnvoll, hier einen eigenen Front Controller für die beiden Client Typen einzusetzen. Der Front Controller stellt Authentizität des Clients sicher und delegiert die Anfragen an den Use Case Controller oder den SCORM API Controller. Als Antwort erhält er rohe Daten. Diese müssen nun als HTML formatiert und mit einem Navigationsmenu versehen werden. Dazu bedient sich der Front Controller der *UseCase View* und der *Module registry* Komponenten.

Die UseCase View Komponenten sind dafür zuständig, dass die rohen Daten in HTML umgewandelt werden. Dies kann beispielsweise mittels XSLT⁵ geschehen. Mit dem

² Ein Adapter wird verwendet, um unterschiedliche Schnittstellen miteinander zu verbinden. In diesem Fall gibt SCORM ein API vor. Dieses muss auf das interne OLAT API gemappt werden.[Coo00]

³ Das Déjà-vu Problem tritt dann auf, wenn ein Benutzer zweimal den selben Request an das System schickt (Reload). Damit das System nicht zwei Mal eine Berechnung durchführt, die zu einem komplett falschen Resultat führen könnte (zweimaliges Abschieken eines Testes), können sogenannte Tokens eingesetzt werden, welche jede Anfrage eindeutig identifizieren und Duplikate aufspüren können.[ACM01]

⁴ Das Filter Pattern wird benutzt um Daten zu filtern. Filter können hintereinander in Ketten angeordnet werden.[ACM01]

⁵ <http://www.w3.org/TR/xslt>

Composite View Pattern werden die UseCase Views mit dem Navigationsmenu ergänzt. Die Informationen des Navigationsmenüs stammen aus dem Module Registry Modul.

9.3.3 Business Tier

Die Geschäftslogikschicht besteht aus den fachlichen Entitätskomponenten und den Controlkomponenten. Dies entspricht dem *Model-View-Controller Pattern*⁶. Das Datenmodell wird durch die Entity-Komponenten abgedeckt, die Controller sind die Control-Komponenten und die View wird durch die UseCase Views implementiert. Die Control Komponenten entsprechen in der J2EE Architektur den sogenannten Session-Beans, die Entity Komponenten den Entity-Beans.[MH00]

9.3.3.1 Control-Komponenten

Der *IMS Enterprise Controller* implementiert das Verarbeiten und Erstellen von Daten gemäss der IMS Enterprise Spezifikation. Es geht dabei vor allem um die Synchronisierung der XML Daten und Metadaten mit den OLAT Datenobjekten.

Der *SCORM API Controller* stellt die Informationen zusammen, welche über den SCORM API Adapter angefragt wurden oder leitet nach Bedarf das Verändern von Werten in Datenobjekten ein.

Die *UseCase Controller* enthalten die Logik, welche in den Anwendungsfällen im Kapitel 6 beschrieben sind. Aus Sicht des Benutzers entsprechen sie jeweils einer atomaren Funktion, also einem Klick oder Submit von einer HTML Seite aus. Die UseCase Controller bekommen ihren Methodenaufruf vom Front Controller und liefern diesem die entsprechenden Daten zurück. Um die Formatierung der Daten kümmern sie sich nicht. Die UseCase Controller sind ohne Status und müssen nicht persistiert werden.[Fow][MH00]

Die *Workflow Controller* haben einen Status und müssen persistiert werden. Für Arbeitsabläufe, welche mehrere Schritte beinhalten und sich über mehrere HTML-Seiten hinweg ziehen, muss eine übergeordnete Kontrolle und Koordination wahrgenommen werden können. Als solche Workflows kommen die im Kapitel 5.2 definierten Geschäftsprozesse in Frage. Das Lesen eines Contentelementes ist zwar ein einzelner Anwendungsfall, dieser spielt sich aber in dem Prozess Lernen ab. Daher muss vor dem Lesen eines Contentelementes zuerst klar sein, in welchem Kurs sich der Lerner befindet etc. Diese Informationen werden vom Workflow Controller dem Use Case Controller zur Verfügung gestellt.

9.3.3.2 Entity-Komponenten

Die Funktionen der Geschäftsklassen und den daraus gebildeten Entity-Komponenten wurden bereits im Kapitel 8 und 9.2 diskutiert. Zusätzlich sind die folgenden zwei Komponenten hinzugekommen:

- Die Komponente *Module registry* wurde bereits angesprochen. Sie dient dazu, dass alle im System vorhandenen Module wie zum Beispiel das Assessment-

⁶ Vgl. hierzu [Fow], [ACM01] und [Coo00]

Content-, Evaluations-, Kommunikations- oder Kursmodul sich im System zur Laufzeit registrieren können. Bei dieser Registrierung wird in der Komponente Module registry festgehalten, über welche Funktionalität das Modul verfügt und in welchem Kontext welche dieser Funktionen zur Verfügung steht. Aus diesen Informationen kann in der Präsentationsschicht ein entsprechendes dynamisches Menü aufgebaut werden. Dies entspricht auch dem geforderten Plugin Mechanismus, um das System einfach erweiterbar zu machen. Möchte man beispielsweise ein synchrones Chatmodul entwickeln, so muss sich dieses neue Modul lediglich registrieren und schon weiss der Front Controller über die neue Funktionalität Bescheid. Er weiss, welche Anfragen er an dieses Modul weiterleiten soll, welche Sicherheitsanforderungen bestehen, wie das Menü neu auszusehen hat und welche UseCase View für die Renderung der Daten verwendet werden soll.

- Die Komponente *Rolle* widerspiegelt die verschiedenen Benutzerrollen des Systems. Zwar sind diese Informationen in den bestehenden Entitäten wie beispielsweise der Contentkomponente zum Teil bereits enthalten, doch für die Praxis ist dies keine gute Lösung. Die Komponente Rolle ist eine Ergänzung zu der Komponente Benutzer. Der Benutzer umfasst lediglich die grundlegenden Informationen über eine Person, alle kontextspezifischen Daten sind einer Rolle zugeordnet. Die Komponente Lerner bleibt hier wegen ihrer zentralen Bedeutung als Spezialfall bestehen.

9.3.4 Integration und Resource Tier

Die Integrations- und Ressourcenschicht wird hier zusammengefasst. Auf diesen Ebenen muss die Speicherung der Datenobjekte und der Dokumente sichergestellt werden. Für diese Problemstellungen gibt es Standardlösungen, darum wird darauf hier nicht weiter eingegangen.

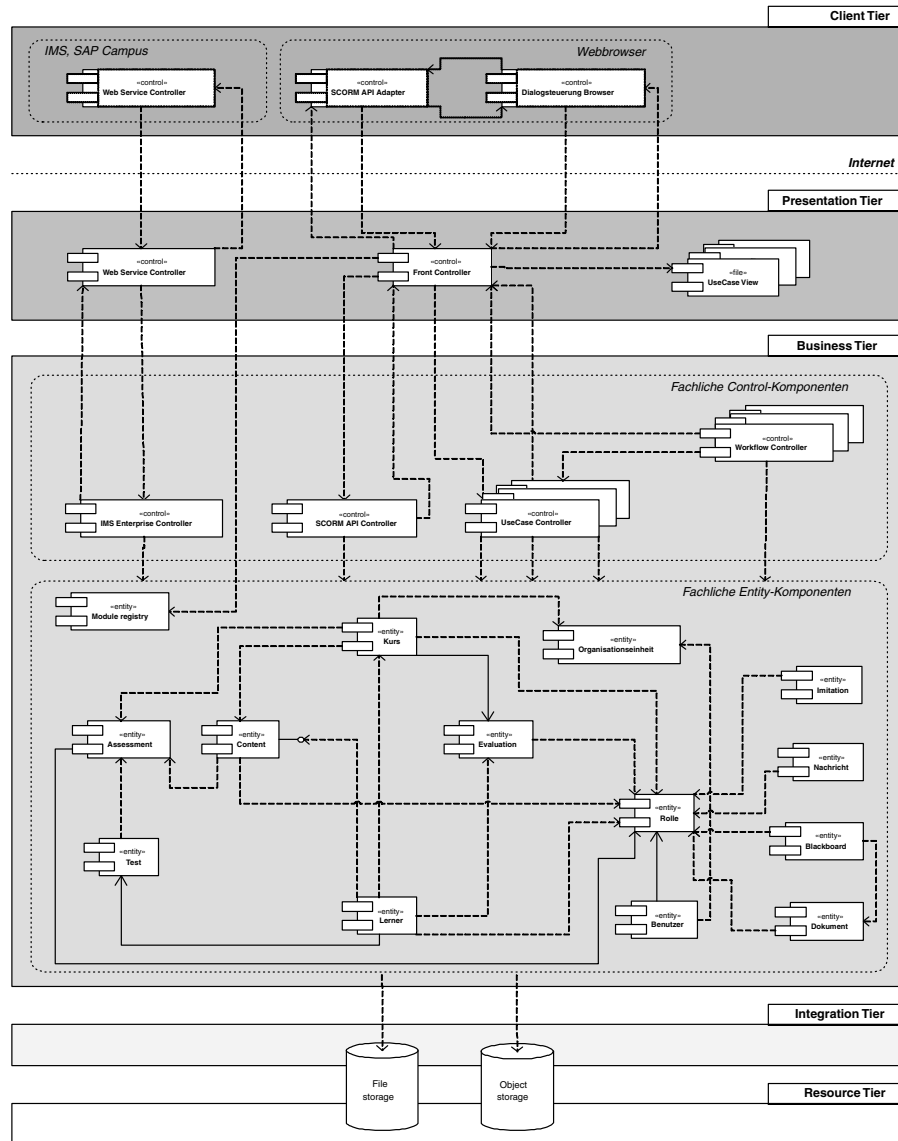


Fig. 9.3: Anwendungsarchitektur von OLAT 2

10. ZUSAMMENFASSUNG

Die an der Universität Zürich entwickelte und eingesetzte Lernplattform OLAT 1 sieht sich längerfristig mit zwei wichtigen Problemen konfrontiert: Erweiterung des Funktionalitätsumfangs und der Skalierung auf mehr Benutzer. Die verwendete Systemarchitektur bietet aber für diese beiden Problemstellungen nur suboptimale Lösungen an. In dieser Arbeit wurde daher eine von Grund auf neue Systemarchitektur für eine solche webbasierte Lernplattform entwickelt.

Das System OLAT 1 besteht im Wesentlichen aus einer Datenbank und einer Hand voll PHP-Skripte. In diesem Konzept gibt es keine konstant verfügbaren Prozesse und keine im Speicher lebenden Objekte, welche bei den nächsten Benutzeranfragen wiederverwendet werden könnten. Ein solches System ist sehr schwer zu verteilen, denn es stehen dafür keine Hilfsmechanismen zur Verfügung. Die neue Architektur basiert daher auf einem Rahmenwerk und einer Laufzeitumgebung für komplexe Geschäftsapplikationen, der Java 2 Enterprise Edition. Dieses Rahmenwerk schafft wichtige Voraussetzungen, um eine skalierbare Applikation entwickeln zu können. Einerseits sind darin diverse Services wie Messaging, Transaktionen oder Directories bereits vom Rahmenwerk zur Verfügung gestellt, andererseits erleichtert ein durchdachtes Komponentenmodell eine optimale Strukturierung der Komponenten im System. Die Frage der Skalierung kann in diesem Modell zu einem grossen Teil an die Laufzeitumgebung delegiert werden und ist nicht mehr ausschliesslich eine Frage des Programmierens.

In der Lernumgebung OLAT 1 wird mit sogenannten transaktionalen Skripten gearbeitet. Das Datenmodell wurde als Entity-Relationship Modell entwickelt. Dies ist die einzige Modellebene. Die Skripte greifen für ihre Funktionen stets direkt auf die Datenbank zu. Ein übergelagertes, objektorientiertes Modell, welches die Daten mit den dazugehörenden Funktionen kapselt, existiert nicht. Dies bedeutet, dass Erweiterungen und Veränderungen nur solange möglich sind, wie sich am Datenmodell selbst oder an dessen Semantik nichts verändert, da das relationale Datenmodell dem eigentlichen API entspricht. Ändert man das API, führt dies zu Änderungen an unterschiedlichen Stellen. Eine solche Architektur ist schlecht erweiterbar. Bei der neuen Architektur wurde daher konsequent nach objektorientierten Methoden vorgegangen und ein sogenanntes Domänenmodell entwickelt. Die einzelnen Anwendungsfälle greifen nun nicht mehr direkt auf die Datenbank zu, sondern interagieren lediglich mit Objekten oder Komponenten des Domänenmodells. Erweiterungen sind in diesem Modell sehr viel besser möglich, weil der Funktionalitätsumfang eines Datenobjekts in dem Objekt selbst gekapselt ist und nicht in dem Code eines einzelnen Anwendungsfalles.

Die Systemarchitektur von OLAT 2 besteht aus fünf Schichten: Client, Präsentation, Business, Integration und Resource Tier. Im Unterschied zu der Architektur des Systems OLAT 1 hat die neue Architektur nun eine klare Trennung zwischen der Präsentations- und der Geschäftslogikschicht. In der Geschäftslogikschicht wird zu-

dem ebenfalls stark unterschieden zwischen dem statischen Datenmodell und den dynamischen Anwendungsfällen. So konnte die in OLAT 1 angestrebte und bei OLAT 1 bemängelte Anwendung des Model-View-Controller Prinzips erreicht werden. Die Anfragen eines Clients werden von einem Front Controller ausgewertet und an den entsprechenden UseCase Controller delegiert. Dieser interagiert mit dem Modulen des Domänenmodells. Die unformatierten Daten werden an den Front Controller zurückgegeben, welche diese an die UseCase View weiterleitet. Die UseCase View sorgt für die graphische aufbereitung gemäss dem vorhandenen Client.

Ein weiteres Schwergewicht wurde bei der Analyse auf die internationalen Standards im Bereich E-Learning gelegt. Im Detail besprochen wurden die Standards des IEEE Learning Technology Standards Committee, im Speziellen die Learning Technology Systems Architecture, die diversen Spezifikationen des IMS Global Learning Consortium und das Sharable Content Object Reference Model (SCORM) von ADL. Nicht weiter eingegangen wurde auf die Standards von AICC, welche ebenfalls eine wichtige Bedeutung haben. Es stellt sich heraus, dass diese Standards und Spezifikationen nicht miteinander konkurrieren sondern sich gegenseitig ergänzen und längerfristig die selben Ziele verfolgt werden: die Lerninhalte sollen so gestaltet und gespeichert werden können, dass sie unabhängig von einer technischen Plattform sind und die Systeme verschiedener Hersteller miteinander interagieren können. SCORM beispielsweise basiert in weiten Teilen auf den IMS Spezifikationen und den IEEE Standards. Ein zweiter wichtiger Punkt der Spezifikationen ist die Definition eines gemeinsamen Datenmodells für den Austausch von Personendaten, um diese zwischen unterschiedlichen Systemen austauschen zu können.

Es scheint sehr sinnvoll zu sein, mit diesen Standards zu arbeiten, auch wenn diese noch lange nicht fertig entwickelt und ständigen Veränderungen unterworfen sind. Nur so ist es möglich, plattformunabhängige Lerninhalte zu entwickeln. Dies hat jedoch auch seinen Preis. Die Standardisierung behindert in einem gewissen Masse die Innovation durch neue Systeme und neue Methoden. In der neuen Architektur wurden einige dieser Standards und Spezifikationen integriert. Die Offenheit des Systems lässt es aber auch zu, proprietäre Komponenten zu entwickeln, welche sich noch nicht durch die bestehenden Standards ausdrücken lassen.

Diese Arbeit zeigt nur die ersten Schritte bei der Entstehung eines neuen Systems. Es wurde versucht, die grundlegenden Problemstellungen einer webbasierten Lernumgebung zu erfassen und dafür eine geeignete Systemarchitektur zu entwickeln. Für eine Implementierung muss diese Architektur noch weiter verfeinert und anschliessend ein Softwareentwurf ausgearbeitet werden.

GLOSSAR

Adressbuch	Sammlung der Adressen von persönlichen Kontakten.
Adressbucheintrag	Erfasste Person im Adressbuch.
Anfrage	Nachricht eines Lernenden an den betreuenden Coach.
Asset	Lernelement. Kleinstes Lernelement, welches in ein SCO integriert werden kann. Zum Beispiel ein Bild.
Assessment	Interaktives Lernelement. Ein Assessment besteht aus Sektionen und Items. Eine Instanz eines Assessments wird Test genannt.
Autorin	Benutzerrolle im System. Eine Autorin ist eine Person, welche Contentelemente erstellt oder verändert. Die Autorin ist in keinen Kurskontext eingebunden und hat keine Berechtigungen an einem Kurs.
Belegen	Vorgang, um einen Benutzer in einer Klasse eines Kurses einzutragen. Auch Einschreiben genannt.
Belegung	Legt fest, dass ein Lerner in einer Klasse Mitglied ist.
Benutzer	Person welche mit der Lernplattform interagiert unabhängig von der Rolle und des Kontexts.
Bewertung	Leistungsausweis. Eine Bewertung kann automatisch oder manuell durch einen Coach zustandekommen.
Blackboard	Diskussionsforum. Jeder Benutzer, welcher Zugang zu dem Blackboard hat, kann in diesem Blackboardnachrichten und Dokumente veröffentlichen.
Blackboardnachricht	Nachricht, welche in einem Blackboard veröffentlicht ist. Sie hat keinen einzelnen Empfänger sondern alle, welche Zugang zu dem Blackboard haben, können diese lesen.
Block	Lernelement. Eine Contentaggregation gemäss SCORM. Ein Block besteht aus Blöcken oder SCO's.
Coach	Benutzerrolle im System. Ein Coach ist eine Person, welche einen Lerner betreut und ihn mit Punkten bewertet. Er ist unmittelbare Ansprechperson für den Lerner. Der Coach steht hierarchisch über dem Lerner.
Coautorin	Benutzerrolle im System. Eine Coautorin hat die selben Rechte an einem Contentelement wie die Autorin, wird aber als Coautorin aufgeführt.

Content	Lernelement. Lerninhalt, welcher als eine Einheit in einen Kurs eingebunden werden kann. Auch Contentelement oder Online-Skript genannt.
Datei, externe	Dokument, welches mit einem externen Programm erstellt wurde, z.B. ein Worddokument.
Datei, interne	Dokument, welches in der Lernumgebung selbst erstellt wurde. Ein internes Dokument ist vom Typ Textnotiz, HTML-Notiz, Bookmark, Todo-Liste, Link oder Glossareintrag.
Daten	Informationstyp. Daten sind Informationen, welche ein Objekt repräsentieren. Ohne die entsprechenden Meta-Daten sind sie wertlos, weil sie nicht sinnvoll interpretiert werden können, bzw. nicht herausgefunden werden kann, um welche Art von Objekt es sich handelt.
Editieren	Vorgang. Verändern von Daten oder Meta-Daten.
Einschreiben	Vorgang, um einen Benutzer in einer Klasse eines Kurses einzutragen. Auch Belegen genannt.
Exportieren	Vorgang. Das Schreiben von Daten in eine Datei gemäss einem Dateiformat. Die Datei kann anschliessend archiviert oder wieder importiert werden.
Evaluation	Fragebogen, welcher in einem Kurs eingebunden werden kann.
Evaluationsfrage	Kleinstes Element innerhalb einer Evaluation.
Evaluationsresultate	Daten, welche durch das Ausfüllen einer Evaluation entstehen.
Evaluationsstruktur	Gliederung und Anordnung der Evaluationsfragen in einer Evaluation.
Gruppe	Zusammenfassung von Benutzern zu einer Einheit mit einem gemeinsamen Ziel und Zweck. Jeder Benutzer hat eine bestimmte Rolle innerhalb der Gruppe.
Gruppenbewertung	Leistungsausweis, vom Coach der Gruppe ausgefüllt.
Gruppencontent	Lernelement, welches nur für eine Gruppe und nicht für alle Lerner eines Kurses zugänglich ist.
Gruppentyp	Schablone für die Erstellung von Gruppen, welche im selben Zusammenhang verwendet werden (z.B. Gruppen, welche für eine spezielle Gruppenarbeit im Zusammenhang mit einer Lerneinheit erstellt wurden).
Helpdesk	Benutzerrolle im System. Personen des Helpdesks haben Zugriff auf Teile der Benutzerdaten, um bei Problemen helfen zu können. Sie haben keinen generellen Zugriff auf persönliche Daten wie Testresultate oder ähnliches.
Importieren	Vorgang. Das Einlesen von Daten aus einer Datei. Das Format der Datei muss dem entsprechenden Importfilter entsprechen.

Item	Element eines Assessments. Ein Item entspricht einem Paar von Question und Respones.
Klasse	Organisationseinheit. Lerner einer Klasse werden von dem selben Coach betreut. Eine Klasse gehört zu einem Kurs und kann räumliche und zeitliche Attribute haben oder rein virtuell sein.
Klassenbelegung	Information darüber, welche Benutzer in einer Klasse eingeschrieben sind.
Klassengruppe	Eine Gruppe mit Mitgliedern, welche alle aus der selben Klasse stammen. Auch klasseninterne oder physische Gruppe genannt.
Kontext	Der Bezug einer Aktion zur aktuellen Umgebung, in welcher sich das Zielobjektion befindet.
Kurs	Lerneinheit. Grösste in sich geschlossene Lerneinheit, welche normalerweise zeitlich befristet ist und mit dem Erwerb eines Zertifikates verbunden ist.
Kursbetreuerin	Benutzerrolle im System. Eine Kursbetreuerin ist eine Person, welche alle Aspekte eines Kurses betreut und verwaltet. Sie hat Zugang zu allen relevanten Daten und kann diese ändern. Sie steht hierarchisch über dem Coach und dem Lerner.
Kursdaten	Daten, welche während eines Kurses von den Lernern generiert werden ohne weitere Verarbeitung.
Kursgruppe	Eine Gruppe mit Mitgliedern, welche aus verschiedenen Klassen stammen. Auch klassenübergreifende oder virtuelle Gruppe genannt.
Kursperformance	Kursdaten, welche in der Art aufbereitet sind, dass statistische Aussagen über die Mittelwerte, Standardabweichungen und Tendenzen der Bewertungen der Lerner gemacht werden können.
Kursstruktur	Beschreibung der Reihenfolge und Anordnung von Elementen eines Kurses.
Lerner	Benutzerrolle im System. Der Lerner ist die Person, welche sich in einen Kurs einschreibt und Lerninhalte konsumiert.
Lernpfad	Aufzeichnung über das historische Verhalten des Lerners bzw. über seine Bewegungen innerhalb der Lernumgebung.
Lernplattform	Benutzerrolle im System. Eine computergenerierte Umgebung, welche vom Benutzer als ein einziges System wahrgenommen wird und auf dem er in seinem Lernprozess optimal unterstützt wird. Dieses Dokument beschreibt eine solche Lernplattform. Im Englischen wird dafür auch der Begriff LMS (Learning Management System) verwendet.

Lernverhalten	Kursdaten, welche in der Art aufbereitet sind, dass Aussagen über das Navigationsverhalten der Lerner gemacht werden können.
Lösung	Dokument, welches ein Lerner dem Coach übermittelt und vom Coach bewertet wird.
Metadaten	Informationstyp. Metadaten sind Daten über Daten. Sie beschreiben die Bedeutung von Informationen. Im Beispiel Name=Florian ist der Begriff Name Metadatum und Florian der eigentliche Wert.
Nachricht	Information, welche einen Absender und einen Empfänger hat. Mehrere Empfänger können angegeben werden, die Nachricht wird aber jedem Empfänger einzeln in einer Kopie zugestellt. Spezielle Nachrichten sind Gruppennachricht, Klassennachricht und Kursnachricht, welche an die Coaches und/oder Mitglieder dieser Organisationseinheiten geleitet wird.
OLAT 1	Externes Informationssystem. Die Lernplattform OLAT 1 ist die Vorgängerversion des Systems OLAT 2.
Organisationseinheit	Strukturierungselement, um die hierarchischen Strukturen auszudrücken. Organisationseinheiten können ihrerseits wiederum Organisationseinheiten enthalten. Typische Beispiele wären Fakultät, Abteilung, Institut und Forschungsgruppe.
Platzangebot	Information darüber, wieviele Plätze in einer Klasse angeboten werden und wieviele davon noch verfügbar sind.
Professor	Benutzerrolle im System. Der Professor ist die rechtlich verantwortliche Person eines Kurses und steht hierarchisch über der Kursbetreuerin, dem Coach und dem Lerner.
Question	Element eines Items eines Assessments. Beinhaltet alle Informationen einer Testfrage. Die möglichen Antworten werden in Responses definiert.
Raum	Ein Raum, in welchem sich die Mitglieder einer Klasse treffen können.
Raumzuweisung	Assoziation eines Raumes zu einer Klasse.
Registrieren	Vorgang, um einen neuen Benutzer auf dem System zu erstellen.
Response	Element eines Items eines Assessments. Eine mögliche Antwort auf eine Frage in einem Test. Der Response enthält auch Angaben über die Darstellung und gibt Aufschluss über den Fragetyp.
Rolle	Eine Rolle definiert die Verantwortlichkeiten und Aufgaben eines Benutzers für einen Kontext. Wichtige Rollen sind beispielsweise der Lerner, der Coach, die Kursbetreuerin und ähnliches.

Rollenimitation	Vorgang, bei welchem ein Benutzer einem anderen Benutzer Zugriff auf sein Benutzerkonto gibt, ohne dafür ein Passwort austauschen zu müssen. Die Imitation kann jederzeit abgebrochen werden und dient zum besseren Verständnis bei Problemen.
SAP	Externes Informationssystem. SAP ist eine Verwaltungssoftware. SAP Campus soll Studentendaten verwalten können, ist zur Zeit aber noch in der Entwicklung begriffen.
SCO	Lernelement. Steht für Sharable Content Objekt und ist Teil des SCORM. Es ist das kleinste austauschbare Lernelement. Ein SCO besteht aus Assets.
Section	Gliederungselement in einem Assessment. Eine Section enthält Items oder andere Sections.
Status (Lerner)	Stand des Lerners in einem Kurs bezüglich möglicher Aktivitäten und vollbrachter Aktivitäten.
Stellvertretercoach	Benutzerrolle im System. Die Stellvertretung eines Coaches ist zeitlich limitiert.
Support	Benutzerrolle im System. Eine Person des Supports hat Zugriff auf alle Daten des Systems und kann alle Daten verändern. Es ist die höchste hierarchische Stufe im System.
System	Jegliche Software, welche für den Betrieb der Lernumgebung auf Seite des Servers erforderlich ist.
Systemdaten	Daten, welche die Konfiguration des Systems beschreiben.
Systemadministratorin	Benutzerrolle im System. Eine Systemadministratorin sorgt für den fehlerfreien Betrieb des Systems. Sie kann das System starten, stoppen und konfigurieren, jedoch keine Daten der Benutzer einsehen oder ändern.
Test	Interaktives Lernelement. Ein Test ist eine Instanz eines Assessments. Er wird vom Lernsystem ausgewertet und normalerweise mit Punkten benotet.
Testantwort	Konkrete Antwort eines Lernenden auf ein Item eines Assessments.
UniAccess	Externes Informationssystem. Unter dem Namen UniAccess werden verschiedene Dienstleistungen des Zentrums für Informatikdienste der Universität Zürich angeboten. Unter anderen bietet UniAccess Emailkonten für alle Angehörigen der Universität und Zugang zum Internet auf dem ganzen Campus an.

Zertifikat	Dokument. Eine Auszeichnung über eine erworbene Fähigkeit.
------------	--

ABKÜRZUNGEN

ADL	Advanced Distributed Learning
AICC	Aviation Industry CBT Committee
APS	Anrechnungspunktesystem
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CBI	Computer-Based Instruction
CBT	Computer-Based Training
CMI	Computer Managed Instruction
CSF	Content Structure Format
CSS	Cascading Style Sheets
DC	Dublin Core
DoD	Department of Defense
DTD	Document Type Definition
EJB	Enterprise Java Beans
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protocol
ICT	Information Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IFI	Institut für Informatik Universität Zürich
IMS	Instructional Management System
ISA	Information Systems Architecture
ISO	International Organization for Standardization
ITS	Intelligent Tutoring Systems
J2EE	Java 2 Enterprise Edition
JDBC	Java Database Connectivity
LMS	Learning Management System
LOM	Learning Object Metadata
LTSA	Learning Technology Systems Architecture
LTSC	Learning Technology Standards Committee
MIME	Multipurpose Internet Mail Extensions
OLAT	Online Learning And Testing
OEP	Object Engineering Process
QCL	Qualifications, Certifications and Licenses
QTI	Question & Text Interoperability
SCO	Sharable Content Object
SCORM	Sharable Content Object Reference Model

SOAP	Simple Object Access Protocol
SE	Software Engineering Gruppe Universität Zürich
SVC	Swiss Virtual Campus
UML	Unified Modeling Language
URI	Universal Resource Identifier
URL	Universal Resource Locator
VLE	Virtual Learning Environment
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	eXtensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations
ZI	Zentrum Informatikdienste Universität Zürich

LITERATURVERZEICHNIS

- [ACM01] Deepak Alur, John Crupi, and Dan Malks. *Core J2EE Patterns*. Sun Microsystems Press, Prentice Hall PTR, first edition, 2001.
In dem Buch werden Patterns besprochen, welche typischerweise bei einer J2EE-Architektur zur Anwendung kommen. Es werden gute und schlechte Design Strategien aufgezeigt. Am Schluss folgen praktische Anwendungsbeispiele der besprochenen Patterns.
- [AM01a] Thor Anderson and Mark McKell. *IMS Content Packaging Best Practice and Implementation Guide*, v. 1.1.2. IMS Global Learning Consortium, 2001.
http://www.msglobal.org/content/packaging/cpv1p1p2/imscp_bestv1p1p2.html
Hier kommt der Leser zu wichtigen Hintergrundinformationen, welche für eine erfolgreiche Implementation des IMS Packaging Modells benötigt werden. Als Einführung in das Thema ist dieses Dokument vor den beiden anderen zu empfehlen.
- [AM01b] Thor Anderson and Mark McKell. *IMS Content Packaging Information Model*, v. 1.1.2. IMS Global Learning Consortium, 2001.
http://www.msglobal.org/content/packaging/cpv1p1p2/imscp_infov1p1p2.html
Beim Packaging geht es darum, mehrere Dokumente zu einem Paket zusammenzubauen und den Inhalt zu beschreiben. Diese Spezifikation beschreibt, wie dies im IMS Modell zu geschehen hat.
- [AM01c] Thor Anderson and Mark McKell. *IMS Content Packaging XML Binding*, v. 1.1.2. IMS Global Learning Consortium, 2001.
http://www.msglobal.org/content/packaging/cpv1p1p2/imscp_bindv1p1p2.html
In diesem Dokument werden die XML Elemente beschrieben, welche für ein Packaging gemäss IMS zur Anwendung kommen.
- [BNRM01] Cathleen Barstow, Liddy Nevile, Madeleine Rothberg, and Mark McKell. *IMS Guidelines for Developing Accessible Learning Applications*, v. 0.6. IMS Global Learning Consortium, 2001.
http://www.msglobal.org/accessibility/accwpv0p6/imsacc_wpv0p6.html
In diesem Papier werden Richtlinien aufgestellt, welche bei der Implementation einer behindertengerechten Umgebung beachtet werden sollen.
- [Coo00] James W. Cooper. *Java Design Patterns, a Tutorial*. Addison-Wesley, first edition, 2000.
In diesem Buch werden die im legendären Buch *Design Patterns* von Gamma, Helm, Johnson und Vlissides beschriebenen Design Patterns sehr praxisorientiert und beispielhaft umgesetzt. Das Buch eignet sich speziell

dann, wenn man mit dem sehr theoretischen Original nicht viel anfangen kann.

- [CV99] Geoff Collier and Wayne Veres. *IMS Enterprise XML Binding Specification*, v. 1.01. IMS Global Learning Consortium, 1999.
<http://www.imsproject.org/enterprise/enbind03.pdf>
In diesem Dokument werden die XML Definitionen zu den IMS Enterprise Elementen erklärt und aufgelistet.
- [CVA99a] Geoff Collier, Wayne Veres, and Thor Anderson. *IMS Enterprise Best Practice and Implementation Guide*, v. 1.01. IMS Global Learning Consortium, 1999.
<http://www.imsproject.org/enterprise/enbest03.pdf>
Dieser Guide erläutert die Hintergründe der IMS Enterprise Spezifikation und gibt Hinweise zur Implementation. Es empfiehlt sich dieses Dokument der Trilogie zuerst zu lesen.
- [CVA99b] Geoff Collier, Wayne Veres, and Thor Anderson. *IMS Enterprise Information Model*, v. 1.01. IMS Global Learning Consortium, 1999.
<http://www.imsproject.org/enterprise/eninfo03.pdf>
Die IMS Enterprise Spezifikation legt fest, wie Gruppenstrukturen standardisiert abgebildet werden können. Es geht um den Austausch von Personeninformationen, Gruppenzugehörigkeiten, das Einschreiben in Kurse und das Austauschen von Grupperesultaten. In diesem Dokument werden die Datenelemente der Spezifikation besprochen.
- [D⁺01a] Philip Dodds et al. *The SCORM Content Aggregation Model*, v. 1.2. ADL: Advanced Distributed Learning, Department of Defense USA, 2001.
Das Content Aggregation Model beschreibt, wie in SCORM die Komponenten Assets, SCOs und Content Aggregations zueinander in Beziehung stehen und welche Metadaten zur Verfügung stehen.
- [D⁺01b] Philip Dodds et al. *The SCORM Overview*, v. 1.2. ADL: Advanced Distributed Learning, Department of Defense USA, 2001.
Das Sharable Content Object Reference Model definiert, wie plattform- und autorensystemunabhängige, austauschbare und wiederverwendbare Lerninhalte mit Metadaten spezifiziert und aufgebaut werden können. Dieses Dokument liefert eine Übersicht über das SCORM.
- [D⁺01c] Philip Dodds et al. *The SCORM Run-Time Environment*, v. 1.2. ADL: Advanced Distributed Learning, Department of Defense USA, 2001.
Das Run-Time Environment definiert das API welches die Contententwicklerinnen benutzen können um ihren Content mit dem Lernsystem kommunizieren zu lassen.
- [Fow] Martin Fowler. Information system architecture.
<http://www.martinfowler.com/isa/>
Dieses Buch werden die folgenden Themen behandelt: Schichtenarchitekturen, Domainlogik versus Transaktionslogik, MVC bei Webservern, Object-Relational Mapping, Verteilungsstrategien und stateless versus stateful Servers. Das Buch ist zur Zeit nur online zu finden, da es noch nicht vollkommen fertiggestellt ist.

- [G⁺00] Gosling et al. *The Java Language Specification*. Addison-Wesley, second edition, 2000.
Dies ist das offizielle Buch der Java 2 Enterprise Edition Spezifikation von Sun Microsystems. Es ist als Nachschlagewerk gedacht.
- [Gli00] Martin Glinz. Software engineering i. Technical report, Institut für Informatik, Universität Zürich, 2000.
Dies ist das Vorlesungsskript der Veranstaltung Software Engineering I an der Universität Zürich. Es enthält eine umfassende Einführung in dieses Fachgebiet.
- [Gnä00] Florian Gnägi. *OLAT - Online Learning And Testing, Projektdokumentation*. Technical report, Juni 2000.
Diese Semesterarbeit gibt einen Überblick über das Projekt OLAT und die in OLAT zur Verfügung stehende Funktionalität. Seit dem Zeitpunkt dieser Dokumentation sind die Funktionen von OLAT stetig erweitert worden, es eignet sich aber dennoch als Einstiegslektüre zum Thema OLAT.
- [Jac97] Beat Jaccottet. *Client/Server-Architekturen: Konzepte und Bedeutung*. PhD thesis, 1997.
In dieser Dissertation werden die verschiedenen Abstufungen der Client/Server-Architekturen vorgestellt. Interessant ist auch eine Analyse der Risiken beziehungsweise der Nutzenpotentiale dieser Architekturen. Ein spezielles Kapitel widmet sich der Integration von heterogenen Datenbanksystemen.
- [Jeg00] Sabina Jeger. *Online Testmethoden - Klassifikation, Implementierung und konzeptionelle Weiterentwicklung*. Master's thesis, Universität Zürich, Institut für Informatik, Juli 2000.
Die Diplomarbeit von Sabina Jeger beschreibt im Detail wie Onlinetests gestaltet werden können. Die im Titel angesprochene Implementation bezieht sich auf das in OLAT implementierte Testsystem.
- [MH00] Richard Monson-Haefel. *Enterprise Java Beans*. Addison-Wesley, second edition, 2000.
Dieses Buch führt in die EJB 1.1 Architektur ein, dem grundlegenden Element der Java 2 Enterprise Edition. In den späteren Kapiteln wird mit Beispielen gearbeitet, die man parallel zum Lesen auf seinem eigenen Rechner ausprobieren kann. Das Buch führt nicht nur in EJB ein sondern erklärt, warum EJB und J2EE überhaupt nützlich sind und warum man damit bessere Applikationen bauen kann.
- [MT01a] Mark McKell and Schawn Thropp. *IMS Learning Resource Meta-Data Best Practice and Implementation Guide*, v. 1.2.1. IMS Global Learning Consortium, 2001.
http://www.imsglobal.org/metadata/imsmdvip2p1/imsmd_bestvip2p1.html
Möchte man die IMS Meta-Data Elemente in einer Software einbauen, so findet man hier unterstützende Informationen und favorisierte Vorgehensweisen.
- [MT01b] Mark McKell and Schawn Thropp. *IMS Learning Resource Meta-Data Information Model*, v. 1.2.1. IMS Global Learning Consortium, 2001.

- http://www.msglobal.org/metadata/imsmdvip2p1/imsmd_infovip2p1.html
Dieses Dokument baut auf den IEEE LTSC LOM auf und beschreibt die in IMS verwendeten Meta-Data Elemente zur Beschreibung von Lerninhalten.
- [MT01c] Mark McKell and Schawn Thropp. *IMS Learning Resource Meta-Data XML Binding*, v. 1.2.1. IMS Global Learning Consortium, 2001.
http://www.msglobal.org/metadata/imsmdvip2p1/imsmd_bindvip2p1.html
Hier sind die genauen XML Definitionen für die IMS Meta-Data Elemente zu finden.
- [Oes01] Bernd Oestereich. *Objektorientierte Softwareentwicklung, Analyse und Design mit der Unified Modeling Language*. Oldenbourg Verlag München Wien, fifth edition, 2001.
Dieses Buch ist gleichzeitig eine Einführung in Objektorientierung, objektorientierte Analyse und Design und in die Unified Modeling Language (UML). Der Autor Bernd Oestereich wird oft als Guru der Objektorientierung bezeichnet. Auf der Innenseite des Buchumschlages ist eine sehr gute Übersicht über die UML-Notation aufgedruckt, welche übrigens auch auf der Homepage von Bernd Oestereich heruntergeladen werden kann.
- [Rob01] Robby Robson. *IEEE Learning Technology Standards Committee*. IEEE LTSC Homepage, 2001.
<http://ltsc.ieee.org>
Dies ist die Homepage der LTSC-Gruppe des IEEE. Von dieser Seite aus findet man den Zugang zu den einzelnen Teilgruppen und zu allen bisher publizierten Dokumenten.
- [S⁺00] Shannon et al. *Java 2 Platform, Enterprise Edition: Platform and Component Specifications*. Addison-Wesley, first edition, 2000.
Dies ist das offizielle Buch der Java Spezifikation von Sun Microsystems.
- [Sch00] Franziska Schneider. *Guidelines für webbasierte Courseware im universitären Bereich*. Master's thesis, Universität Zürich, Institut für Informatik, Juli 2000.
Franziska Schneider beschreibt in ihrer Diplomarbeit alle *do-s* und *don't-s*, die bei webbasierten Onlineveranstaltungen zu beachten sind. Die Arbeit zeigt den heutigen theoretischen Erkenntnisstand über die Fragen, wie online-Lernplattformen zu gestalten sind. Die technischen Aspekte werden dabei nicht berücksichtigt.
- [Sch01] Eva Seiler Schiedt. *Die E-Learning-Strategie der Universität Zürich*. In *Virtueller Campus. Szenarien - Strategien - Studium*. Erwin Wagner, Michael Kindt, 2001.
http://www.ict.unizh.ch/download/010920_Strategie.pdf
In diesem kurzen Artikel beschreibt Frau Seiler, die Vorsteherin der ICT Fachstelle der Universität Zürich, die Situation und weitere Strategie der Universität Zürich im Bereich E-Learning im Sommer 2001.
- [SE99] Stephen Spainhour and Robert Eckstein. *Webmaster In A Nutshell, A Desktop Quick Reference*. O'Reilly & Associates Inc., second edition, June 1999.

In diesem Buch werden die Themen HTML, CSS, XML, JavaScript, CGI, PHP, HTTP und Apache behandelt. Es dient einerseits als Referenznachschlagewerk, beinhaltet andererseits aber auch grundlegende, einführende Kapitel zu den jeweiligen Themen.

- [SSBL01a] Colin Smythe, Eric Shepherd, Lane Brewer, and Steve Lay. *IMS Question & Test Interoperability: An Overview*, v. 1.2. IMS Global Learning Consortium, 2001.
http://www.imsglobal.org/question/quiv1p2pd/imsqti_oviewv1p2.html
Dies ist das Einstiegsdokument um einen Überblick über die IMA QTI Spezifikation zu erlangen. Dieses Dokument ist neu, die Informationen waren zuvor in der QTI Information Model Specification enthalten.
- [SSBL01b] Colin Smythe, Eric Shepherd, Lane Brewer, and Steve Lay. *IMS Question And Test Interoperability: An Overview*, v. 1.2. IMS Global Learning Consortium, 2001.
http://www.imsglobal.org/question/quiv1p2pd/imsqti_oviewv1p2.html
Übersicht über die IMS Test Spezifikation. Insgesamt besteht die Spezifikation aus zehn Dokumenten: ASI Information Model, ASI XML Binding, ASI Best Practice Guide, QTILite Specification, ASI Selection AND Ordering, ASI Outcomes Processing, Results Reporting Information Model, Results Reporting XML Binding, Results Reporting Best Practice Guide und der QTI Overview. Die Question and Test Interoperability beschreibt, wie Test mit XML spezifiziert und so ausgetauscht werden können. Dies einerseits zwischen verschiedenen Lernsystemen, andererseits aber auch zwischen speziellen Authoringtools und einem Lernsystem.
- [STR01a] Colin Smythe, Frank Tansey, and Robby Robson. *IMS Learner Information Package Information Model Specification*, v. 1.0. IMS Global Learning Consortium, 2001.
<http://www.imsglobal.org/profiles/lipinfo01.html>
In diesem Dokument wird die Datenstruktur des Lernalters beschrieben. Dies beinhaltet nicht nur Felder wie Name und Adresse sondern auch alle Informationen über die Kompetenzen und Zugehörigkeiten eines Lernalters.
- [STR01b] Colin Smythe, Frank Tansey, and Robby Robson. *IMS Learner Information Packaging Best Practice Implementation Guide*, v. 1.0. IMS Global Learning Consortium, 2001.
<http://www.imsglobal.org/profiles/lipbest01.html>
Hier finden sich Hintergrundinformationen zu der Spezifikation und Hinweise für deren Implementierung.
- [STR01c] Colin Smythe, Frank Tansey, and Robby Robson. *IMS Learner Information Packaging XML Binding*, v. 1.0. IMS Global Learning Consortium, 2001.
<http://www.imsglobal.org/profiles/lipbind01.html>
Dies sind die XML Definitionen des IMS Lernalters Datenmodells.
- [T⁺01] John Tyler et al. *IEEE 1484.1 Architecture and Reference Model (IEEE P1484.1/D8)*, Draft 8. IEEE, 2001.
<http://ltsc.ieee.org/wg1>

Dieses Dokument beschreibt den IEEE 1484.1 Standard. In Draftversion 8 ist es bereits recht ausgereift und dürfte sich nicht mehr grundlegend verändern. Das Dokument beschreibt nicht eine Systemarchitektur im Sinne des Softwareengineerings sondern eine generische Lernarchitektur und identifiziert so die typischen Komponenten jedes Lernsystems.

INDEX

- 3-Tier Architektur, 7, 10, 112
- 5-Tier Architektur, 112, 116

- ADL, 28
- AICC, 29
- Anforderungen, 4, 12, 103
- Anforderungsbeitragende, 4, 35
 - Abstrakte, 37
 - Reale, 35
- Anrechnungspunktesystem, *siehe* APS
- Anwendungsarchitektur, *siehe* Architektur
- Apache, 11
- API, 11, 33
- Application Programming Interface, *siehe* API
- APS, 13, 109
- Architektur, 3, 5, 112, 116
- ASI, 25
- Assessment, 24, 25, 109, 111
- Asset, 30, 111
- Autorin, 36

- Belegung, 46, 109
- Benutzbarkeit, 105
- Benutzer, 35, 109
- Betreuung, 45
- Betreuung und Bewertung, **81**
 - Anfrage beantworten, 83
 - Bewertung editieren, 81
 - Bewertung erstellen, 81
 - Bewertungsinformationen lesen, 81
 - Gruppe überwachen, 83
 - Gruppenbewertung editieren, 82
 - Gruppenbewertung erstellen, 82
 - Lerner überwachen, 81
- Bewertung, 45
- Blackboard, 48, 111
- Blackboardnachricht, 111
- Block, 111

- Client-Server, 10
- CMI, 29

- Coach, 35, 109
- Composite View, 117
- Content, 22, 38, 109
- Content Aggregation, 31
- Content Packaging, 31
- Contentverwaltung, 38, **54**
 - Assessment erstellen, 60
 - Assessment exportieren, 61
 - Assessment importieren, 60
 - Assessment löschen, 60
 - Assessment Metadaten editieren, 61
 - Asset editieren, 59
 - Asset erstellen, 59
 - Asset löschen, 59
 - Asset Metadaten editieren, 59
 - Block löschen, 58
 - Block Metadaten editieren, 56
 - Block vollständig löschen, 58
 - Blockelemente erstellen, 56
 - Blockstruktur editieren, 57
 - Content erstellen, 54
 - Content exportieren, 56
 - Content importieren, 55
 - Content löschen, 54
 - Content Metadata editieren, 55
 - Contentstruktur editieren, 56
 - Element in Block hinzufügen, 56
 - Element von Block entfernen, 57
 - Item editieren, 62
 - Item erstellen, 61
 - Item löschen, 61
 - SCO erstellen, 58
 - SCO Metadaten editieren, 59
 - Section editieren, 62
 - Section erstellen, 62
 - Section komplett löschen, 62
 - Section löschen, 62
- CSF, 29

- Dateien verwalten, 49, **92**
 - Datei kopieren, 94
 - Datei löschen, 93

- Datei Metadaten editieren, 93
- Datei referenzieren, 94
- Datei verschieben, 94
- Externe Datei editieren, 92
- Externe Datei erstellen, 92
- Interne Datei editieren, 92
- Interne Datei erstellen, 92
- DoD, 28
- Dokument, 111
- Dublin Core Meta-data, 22
- Effizienz, 106
- Einschreibung, 46
- EJB, 6
- Enterprise Java Beans, *siehe* EJB
- Entwurf, 5
- Erweiterbarkeit, 11
- Evaluation, 43, 109, 111
- Evaluationsbogen, 111
- Evaluationsfragen, 111
- Evaluationsverwaltung, 43, **72**, 73
 - Evaluation überwachen, 73
 - Evaluation erstellen, 72
 - Evaluation exportieren, 73
 - Evaluation importieren, 72
 - Evaluation löschen, 72
 - Evaluation Metadaten editieren, 74
 - Evaluationsfrage editieren, 75
 - Evaluationsfrage erstellen, 74
 - Evaluationsfrage löschen, 75
 - Evaluationsstruktur editieren, 74
- Evaluationsresultat, 111
- Fragebogen, 43
- Front Controller, 117
- Funktionale Anforderungen, 103
- Geschäftsanwendungsfälle, 4
- Geschäftsklassen, 4, 109
- Geschäftsprozess, 4
- Geschäftsprozesse, 37
- Glossar, 5
- Gruppe, 44, 109
- Gruppenmitglied, 109
- Gruppenverwaltung, 44, **75**
 - Coach Gruppe zuweisen, 79
 - Coach von Gruppe entfernen, 79
 - Gruppentyp Metadaten editieren, 76
 - Gruppe löschen, 78
 - Gruppe Metadaten editieren, 77
 - Gruppencontent zuweisen, 80
 - Gruppennachricht zustellen, 80
 - Gruppentyp erstellen, 75
 - Gruppentyp löschen, 75
 - Klassengruppe manuell erstellen, 77
 - Klassengruppen automatisch erstellen, 77
 - Kursgruppe manuell erstellen, 77
 - Kursgruppen automatisch erstellen, 76
 - Lerner Gruppe zuweisen, 78
 - Lerner von Gruppe entfernen, 79
- Helpdesk, 36, 109
- HTML, 116
- HTTP, 10, 116
- ICT-Fachstelle, 9, 12
- IEEE, **15**
 - 1484.1, 17
 - 1484.12 (LOM), 16, 22, 32
 - LTSA, 17
 - LTSA Komponenten, 19
 - LTSC, **15**
- IfI, 9
- IMS, 13, **20**, 37
- imsmanifest, 23
- Index, 25
- Institut für Informatik, *siehe* IfI
- Institut of Electrical and Electronics Engineers, *siehe* IEEE
- Instructional Management System, *siehe* IMS
- Interoperabilität, 20
- Item, 111
- J2EE, 6, 112
- Java, 6
- Java 2 Enterprise Edition, *siehe* J2EE
- JavaScript, 116
- Klasse, 41
- Klassenmodell, 109
- Klassenverwaltung, 41, **66**
 - Coach Klasse zuweisen, 69
 - Coach von Klasse entfernen, 70
 - Einschreibung überwachen, 67
 - Klasse erstellen, 66
 - Klasse löschen, 66

- Klassen Metadaten editieren, 68
- Klassenbelegung exportieren, 67
- Lerner Klasse zuweisen, 69
- Lerner umteilen, 71
- Lerner von Klasse entfernen, 71
- Platzangebot verändern, 68
- Raum zuweisen, 68
- Raumzuweisung löschen, 68
- Stellvertretercoach entfernen, 70
- Stellvertretercoach Klasse zuweisen, 70
- Kommunizieren, 48, **88**
 - Adressbucheintrag einfügen, 88
 - Adressbucheintrag löschen, 88
 - Blackboard erstellen, 89
 - Blackboard exportieren, 90
 - Blackboard löschen, 89
 - Blackboard Metadaten editieren, 89
 - Blackboardnachricht bewerten, 91
 - Blackboardnachricht erstellen, 90
 - Blackboardnachricht löschen, 91
 - Blackboardnachricht lesen, 90
 - Nachricht erstellen, 88
 - Nachricht lesen, 88
- Kurs, 40, 109
- Kurs belegen, 46, **84**
 - Belegung löschen, 84
 - Kurs belegen, 84
- Kursbetreuerin, 36, 109
- Kursverwaltung, 40, **63**
 - Klassennachricht zustellen, 69
 - Kurs erstellen, 63
 - Kurs importieren, 64
 - Kurs löschen, 63
 - Kurs Metadaten editieren, 64
 - Kursdaten exportieren, 65
 - Kursnachricht zustellen, 65
 - Kursperformance überwachen, 66
 - Kursstruktur editieren, 64
 - Kursstruktur exportieren, 65
 - Lernerverhalten beobachten, 66
- Lösung, 47, 109
- Launch, 32
- Learning Technology Standards Committee, *siehe* IEEE LTSC
- Lehrveranstaltung, 40
- Lernen, 46, **85**
 - Coach kontaktieren, 85
 - Content lesen, 86
 - Lösung abgeben, 87
 - Lernpfad betrachten, 85
 - Lernpfad weiterführen, 86
 - Status beobachten, 85
 - Suchen, 85
 - Test lösen, 86
- Lerner, 35, 109
- Lerninhalt, 38
- Lernpfad, 48
- Lernsystem, 3
- Limitationen OLAT 1, 10
- LIP, 26
- LOM, 16
- Manifest, 23
- Menu, 117
- Meta-Data, 21, 32
- Model-View-Controller, *siehe* MVC
- Multi-Tier, 7
- MVC, 11, 118
- Nachricht, 48, 109, 111
- Navigation, 117
- Object Bank, 25
- Object Engineering Process, *siehe* OEP
- OEP, 3
- OLAT-1, **8**, 12, 37
- OLAT-2, **12**
- OLAT-Zentrum, 10
- Online-Skript, 38
- Open Source, 6
- Organisationseinheit, 109
- Package Interchange File, 23
- Packaging, 22
- Person, 51
- Personenverwaltung, 51, **95**
 - Fremdes Passwort setzen, 98
 - Passwort setzen, 97
 - Persönliche Daten editieren, 97
 - Person automatisch löschen, 96
 - Person automatisch registrieren, 96
 - Person löschen, 97
 - Person registrieren, 95
 - Person selbständig registrieren, 95
 - Rolle imitieren, 98
 - Rolle verändern, 99
 - Rolle zuweisen, 99
 - Rollenimitierung akzeptieren, 98

- PHP, 11
- Professor, 35, 109
- qcl, 26
- QTI, 24
- Question, 111
- Raum, 42
- Response, 111
- Rolle, 51
- SAP, 13, 27, 37, 109, 116
- SCO, 30, 31, 109, 111
- SCORM, **28**, 38
- SE-Gruppe, 9
- Section, 25, 111
- Sharable Content Object Reference Model, *siehe* SCORM
- Sicherheit, 106
- Skalierbarkeit, 10
- Skript, 30
- SOAP, 27, 116
- Status, 48
- Suchen, 48
- Support, 36, 109
- Supportanforderungen, 108
- SVC, 13
- Swiss Virtual Campus, *siehe* SVC
- Systemadministratorin, 37, 109
- Systementwurf, 5
- Systemidee, 3, 12
- Systemverwaltung, **100**
 - System überwachen, 102
 - System Metadaten editieren, 101
 - System starten, 100
 - System stoppen, 101
 - System updaten, 102
 - Systemdaten exportieren, 101
 - Systemdaten importieren, 101
- telnet, 10
- Test, 24, 111
- Testantwort, 111
- Tier, 112
- UML, 5
- UniAccess, 37
- Web, 3
- webbasiert, 3
- Webbrowser, 116
- WebService, 116, 117
- WebServices, 27
- XML, 22
- Zentrum für Informatikdienste, *siehe* ZI
- Zertifikat, 109
- ZI, 9
- Zuverlässigkeit, 106