



# Individualisierte Ablaufsteuerung in einem Intelligenten Tutoriellen System

Konzeption und Umsetzung am Beispiel des ITS „eTutor“

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Mag.rer.soc.oec.

im Diplomstudium

WIRTSCHAFTSINFORMATIK

**Eingereicht von:**

*Markus Teufl, 0255273*

**Angefertigt am:**

*Institut für Wirtschaftsinformatik – Data & Knowledge Engineering*

**Betreuung:**

*o. Univ.-Prof. Dr. Michael Schrefl*

**Mitbetreuung:**

*Mag. Christian Eichinger*

*Linz, April 2008*

*Für meine Eltern Reinhard und Rosa Teufel*

## **Eidesstattliche Erklärung**

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

April 2008

---

Markus Teufl

## Inhaltsverzeichnis

<b>Einführung und Ansätze zur Ablaufsteuerung .....</b>	<b>8</b>
1 Einführung .....	9
1.1 Problemstellung .....	10
1.2 Beispiel: Kurs Datenmodellierung .....	13
2 Ansätze zur Ablaufsteuerung.....	16
2.1 Ablaufsteuerung von Kursen und Lerninhalten .....	16
2.1.1 Statische Ablaufsteuerung.....	17
2.1.2 Aktive und Passive Ablaufsteuerung .....	18
2.1.3 Adaptive Ablaufsteuerung .....	19
2.1.4 Individualisierte Ablaufsteuerung.....	20
2.2 Auswahlkriterien.....	21
2.3 Fazit .....	22
<b>Stand der Technik und Gegenüberstellung der Ansätze .....</b>	<b>23</b>
3 Stand der Technik .....	24
3.1 SCORM® 2004 3 <sup>rd</sup> Edition .....	27
3.1.1 Book 2 – Content Aggregation Model.....	31
3.1.2 Book 3 – Run-Time Environment .....	35
3.1.3 Book 4 – Sequencing and Navigation.....	36
3.2 Adaptives Hypermedia.....	37
3.2.1 Techniken.....	37
3.2.2 Architektur .....	39
4 Gegenüberstellung der Ansätze .....	41
4.1.1 Gegenüberstellung .....	41
4.1.2 Fazit.....	44
<b>Konzeption einer individualisierten Ablaufsteuerung.....</b>	<b>46</b>
5 Scope einer individualisierten Ablaufsteuerung .....	47
6 Konzepte zur Umsetzung einer individualisierten Ablaufsteuerung .....	51
6.1 Basiselemente .....	51
6.2 Modellierungselemente.....	54
6.3 Regelbasierte Ablaufsteuerung.....	61
6.3.1 Bedingungen .....	62
6.3.2 Aktionen .....	64
6.4 Der Rollup-Prozess der Ablaufsteuerung.....	65
<b>Umsetzung der individualisierten Ablaufsteuerung im eTutor ...</b>	<b>68</b>
7 Die Ablaufsteuerung im eTutor .....	69

7.1	Aktivitätenbaum .....	70
7.2	Administration und Ausführung der Ablaufsteuerung.....	71
7.2.1	Administration eines Aktivitätenbaumes.....	72
7.2.2	Ausführung eines Aktivitätenbaumes .....	74
7.2.3	Präsentation eines Aktivitätenbaumes.....	78
7.3	Bestandteile eines Aktivitätenbaumes .....	79
7.3.1	Aktivitäten.....	79
7.3.2	Lernziele.....	81
7.3.3	Lerninhalte.....	82
7.3.4	Konstrukte .....	83
7.3.5	Regeln.....	84
7.4	Anfragen an die Ablaufsteuerung.....	90
7.5	Beispiel zur Verwendung .....	94
7.6	Fazit.....	96
8	Implementierung.....	98
8.1	Kommunikation der Ablaufsteuerung.....	101
8.2	Externe Komponenten .....	102
	<b>Zusammenfassung und Ausblick .....</b>	<b>104</b>
9	Zusammenfassung.....	105
10	Ausblick.....	108
	<b>Anhang .....</b>	<b>109</b>
	<b>Literaturverzeichnis.....</b>	<b>115</b>
	<b>Abbildungsverzeichnis .....</b>	<b>119</b>
	<b>Tabellenverzeichnis.....</b>	<b>120</b>

## Kurzfassung

E-Learning-Systeme nutzen Benutzermodelle um Informationen über den Lernfortschritt und den Lernerfolg von Studierenden in einzelnen Wissensbereichen zu speichern. Diese Informationen können dazu verwendet werden, die Abfolge der Lerneinheiten, die Lerneinheitssequenz, an die Erfordernisse des Benutzers anzupassen. Je besser sich ein Lernsystem an die Bedürfnisse eines Lernenden anpassen kann, desto effizienter kann Wissen an diesen vermittelt werden. Eine Möglichkeit dies zu erreichen besteht darin, ein Lernsystem um eine so genannte Ablaufsteuerung zu erweitern, welche diese Anpassung übernimmt.

Diese Diplomarbeit zeigt auf, welche Aufgaben eine Ablaufsteuerung in einem Lernsystem übernimmt und welche Ansätze es gibt, die sich mit deren Umsetzung beschäftigen. Ansätze des Adaptiven Hypermedia, sowie Konzepte zur Ablaufsteuerung nach SCORM®, werden gegenübergestellt und es wird untersucht, welche der vorgestellten Konzepte sich am besten im Intelligenten Tutoriellen System „eTutor“, des Instituts für Data & Knowledge Engineering, umsetzen lassen.

Darauf aufbauend zeigt diese Diplomarbeit, wie ein Lehrender eine Lerneinheitssequenz im eTutor modellieren kann, welche Konzepte ihm dazu zur Verfügung stehen und wie die Implementierung der Ablaufsteuerung des eTutors diese wertet und verarbeitet.

## **Abstract**

E-learning-systems are using user models as a basis for capturing information about the learner's progress and success in a specific topic. In particular, this information can be used to adapt the order of learning units - the learning sequence - to the user requirements. The better a learning system can adapt the learning process to different user requirements, the better the learning performance of students will be.

This thesis shows, which tasks a sequencing mechanism has to perform in a learning system and which approaches currently exist to implement them. The approach of adaptive hypermedia is compared with the sequencing & navigation specification introduced by SCORM®. It will be shown, how sequencing mechanisms possibly can be implemented in the Intelligent Tutorial System "eTutor" of the department for Data & Knowledge Engineering.

Furthermore this thesis describes how teachers can model a learning sequence, what concepts are useful for that task, and the way how the implemented sequencing mechanism of the eTutor can use that information and work with it.

## **Teil 1**

### **Einführung und Ansätze zur Ablaufsteuerung**



## 1 Einführung

Bereits seit den 1950er Jahren wird computerunterstütztes Lernen betrieben. Die ersten Lernprogramme waren noch sehr einfach umgesetzt und wurden für spezifische Lernsituationen implementiert. Der Fortschritt im Bereich des Internets stellte die technische Basis für die Entwicklung einer neuen Generation von Lernsystemen dar. Dadurch wurde die Entwicklung einer Vielzahl von webbasierten Lernsystemen ermöglicht.

Dabei wurde jedoch schnell klar, dass neue Techniken alleine nicht ausreichend sind, um ein erfolgreiches Lernsystem zu entwickeln. Schritt für Schritt wurde daher versucht, bestehende Lerntheorien mittels der neuen Technologien umzusetzen und deren neue Möglichkeiten zu nutzen. Die Umsetzung von Lerntheorien wurde vorangetrieben und stetig verbessert [MiPe02]. Hier sind insbesondere die Fortschritte der künstlichen Intelligenz zu erwähnen, die zur Entwicklung von Intelligenten Tutoriellen Systemen (ITS) führten [Kins96]. Diese können dem Kognitivismus zugeordnet werden, bei welcher der Mensch und sein Denkprozess im Vordergrund stehen [Trav95].

Intelligente Tutorielle Systeme vereinen daher neben Expertenwissen auch Informationen zu Benutzern in einer Wissensbasis. Das Wissen über Benutzer wird in sogenannten Benutzermodellen verwaltet. Diese enthalten z.B. aktuelle Informationen über den Hintergrund eines Studenten, seinen aktuellen Wissensstand, individuelle Fehler oder Lernziele. Das Expertenwissen umfasst wiederum Wissen zu Fehlern, zur Domäne oder Wissen über das Verhalten von Lernenden und deren Denkprozess. Durch zu Hilfenahme von Expertenwissen wird ebenfalls versucht, Wissen zu kommunizieren und an die Bedürfnisse des Lernenden anzupassen. Expertenwissen wird auch dazu genutzt, um die Erreichung von Lernzielen abzuschätzen, d.h. darum, ob und in welchem Umfang Lernstoff erlernt wurde.

Insbesondere die Anpassung von Lernsystemen an individuelle Lernsituationen verspricht dabei gute Lernergebnisse. Bloom zeigte in diesem Zusammenhang 1984, dass mit Lernprogrammen, welche sich an den Lernenden anpassen können, das beste Lernergebnis erzielt wird. Demnach ist, individualisiertes

Lernen, das Grundkonzept der ITS, hervorragend zur Wissensvermittlung geeignet [Bloo84].

Der eTutor ist ein solches Intelligentes Tutorielles System welches am Institut für Data & Knowledge Engineering der Johannes Kepler Universität Linz entwickelt und eingesetzt wird [Hofe05]. Ziel dieser Diplomarbeit ist es, das bestehende System um eine Ablaufsteuerung zu erweitern. Diese Ablaufsteuerung soll es dem eTutor ermöglichen, sich während des Lernprozesses an den Studenten anzupassen und ihm je nach seinem Wissensstand unterschiedliche Lerninhalte bereitzustellen. Der Lernprozess soll dadurch individualisiert und somit, wie von Bloom gezeigt, verbessert werden [Bloo84].

Ansätze wie eine solche Ablaufsteuerung in einem webbasierten Lernsystem aussehen kann und wichtige für diese Thematik notwendigen Begriffe werden im Kapitel 2 dieser Diplomarbeit behandelt. Des Weiteren werden im Kapitel 3 zwei Ansätze für die Ablaufsteuerung im eTutor vorgestellt und im Kapitel 4 gegenübergestellt. Hierzu wird die Ablaufsteuerung nach SCORM® mit den Ansätzen des Adaptiven Hypermedia verglichen und die jeweiligen Vor- und Nachteile werden aufgezeigt. Kapitel 5 zeigt auf, welche Modellierungsmöglichkeiten eine individualisierte Ablaufsteuerung im eTutor bieten soll und Kapitel 6 geht detailliert auf die dafür notwendigen Modellierungskonzepte ein. Kapitel 7 beschreibt die im eTutor umgesetzte Ablaufsteuerung und geht auf die Besonderheiten der implementierten Lösung ein. Weiters wird die Architektur der implementierten Ablaufsteuerung im Kapitel 8 beschrieben. Hier werden außerdem wichtige externe Komponenten für die Ablaufsteuerung aufgelistet. Kapitel 9 und 10 fassen schließlich die Ergebnisse der Arbeit zusammen und geben einen Ausblick auf weitere Forschungsfragen.

## **1.1 Problemstellung**

Der eTutor ist ein Intelligentes Tutorielles System welches am Institut für Data & Knowledge Engineering an der Johannes Kepler Universität Linz entwickelt und eingesetzt wird [Hofe05]. Zurzeit können im eTutor nur statische Kurse angelegt und von Studenten absolviert werden. Eine individualisierte Ablaufsteuerung im eTutor würde die Berücksichtigung der Bedürfnisse der Studenten bei ihrem Lernprozess ermöglichen.

Abbildung 1 zeigt die Bestandteile des eTutors und wie diese zusammenwirken. Das Kommunikationsmodell stellt die Schnittstelle zwischen dem eTutor und dem Studenten dar. Es dient sowohl der Darstellung der Problemstellung z.B. eines Übungszettels einer Lehrveranstaltung, als auch der Übermittlung der Antworten eines Studenten an den eTutor (1)(2). Das Unterrichtsmodul ist das Kernmodul des eTutors. Es leitet die Antworten des Studenten an das Studentenmodul (3), sowie an das Expertenmodul (4) weiter. Weiters bestimmt das Unterrichtsmodul, welche Lerninhalte dem Studenten geliefert werden sollen. Das Expertenmodul wertet die Antwort eines Studenten aus und liefert diese wiederum an das Unterrichtsmodul zurück (5). Dieses leitet die Auswertung sowohl an das Studentenmodul (6), als auch an das Kommunikationsmodul (7) weiter, welches dem Studenten die Auswertung präsentiert (8). Das Studentenmodul sammelt Informationen über den Studenten. So werden seine Antworten, wie auch die Auswertungen dieser im Studentenmodul abgespeichert. Das Studentenmodul bietet weiters die Möglichkeit eine Vielzahl von Informationen, z.B. das Vorwissen des Studenten, individuelle Fehler oder Vorlieben, abzubilden. Diese Information stehen dem gesamten eTutor-System zur Verfügung und können von einer Ablaufsteuerung ausgewertet werden (9).

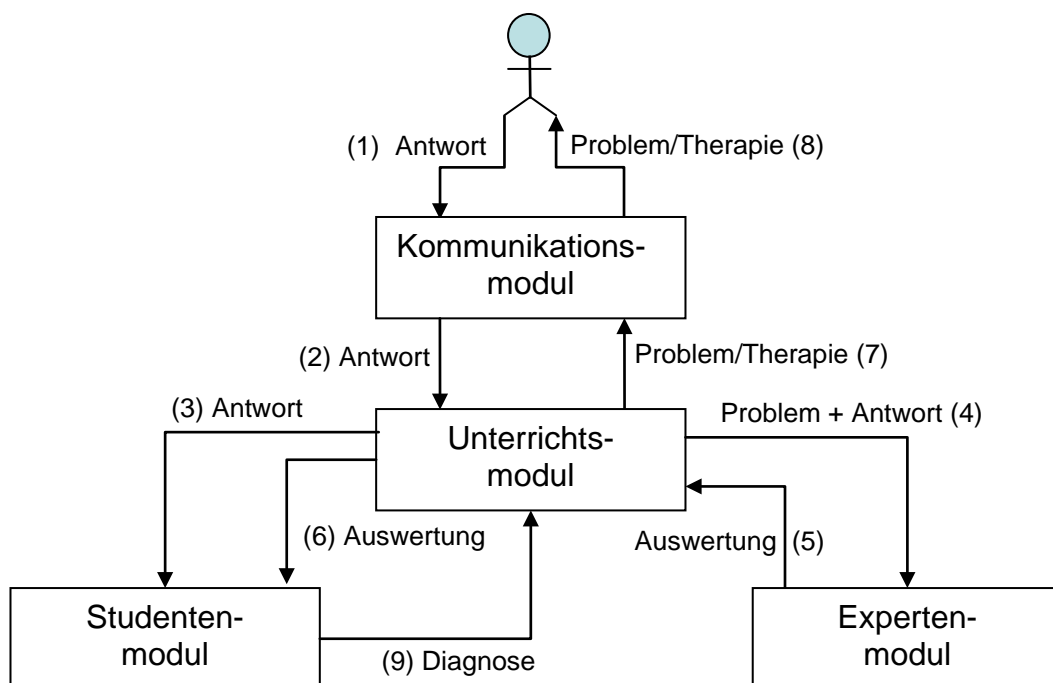


Abbildung 1: Architektur des eTutor-Systems. nach [Hofe05]

Das bestehende System des eTutors setzt die zuvor manuell durchgeführten Tätigkeiten des Lernprozesses um. So ist der eTutor geprägt durch die Metapher des „Übungszettels“:

„Jeder Student einer Lehrveranstaltung soll für unterschiedliche Aufgabenbereiche, wie z.B. das Erlernen der SQL-Datenbanksprache, eine oder mehrere Aufgaben innerhalb eines bestimmten Zeitraumes lösen. Diese Aufgaben werden zuvor vom Assistenten der Lehrveranstaltung in Form eines Übungszettels zur Verfügung gestellt. Nachdem der Student die Lösung abgegeben hat, wird sie vom Tutor korrigiert.“ [Hofer05, S.80]

Der eTutor bildet Übungszettel in Form von allgemeinen Studienleitfäden. Ein allgemeiner Studienleitfaden hat folgende Charakteristika:

- **Lehrveranstaltung (LVA)**

Ein Student erweitert sein Wissen in einer Lehrveranstaltung und das webbasierte Lernsystem soll ihm bei einer konkreten Lehrveranstaltung unterstützen. So könnte eine Lehrveranstaltung z.B. der Kurs Datenmodellierung sein und der eTutor unterstützt den Studenten beim erfolgreichen Abschließen des Kurses.

- **Aufgaben**

Eine Aufgabe dient dazu, dem Studenten beizubringen, wie er eine bestimmte Problemstellung lösen kann. Einer Aufgabe (z.B. SQL Abfragen erlernen) wird eine Übung (eine Lernressource wie z.B. ein Übungszettel mit Musterbeispielen), welcher ein Student lösen soll, zugeordnet. Diese Zuordnung kann entweder fix oder flexibel sein. Eine Aufgabe kann eine bestimmte Anzahl von Punkten haben und somit können diese von Tutoren ausgewertet werden.

- **Aufgabengruppen**

Bestehende Aufgaben können mit Hilfe von Aufgabengruppen gegliedert werden.

- **Übungspool**

Der bestehende eTutor enthält einen Pool von Übungsbeispielen, welche in Form von Aufgaben eines Übungszettels dem Studenten präsentiert werden können.

- **Benutzer**

Das bestehende System kennt drei unterschiedliche Benutzertypen, nämlich Studenten, Tutoren und Assistenten.

Ein Hauptaugenmerk bei der Planung und Implementierung der Ablaufsteuerung ist, das bestehende System zu erweitern und bereits vorhandene Lösungsansätze beizubehalten.

## **1.2 Beispiel: Kurs Datenmodellierung**

Um die Inhalte der Ablaufsteuerung besser veranschaulichen zu können, wird nun zur Erläuterung das Lernen an einer Universität beispielhaft beschrieben. Die Universität ist ein Lernsystem bei welchem z.B. Assistenten die Rolle eines Lehrenden übernehmen und an Studenten (die Lernenden) Wissen vermitteln. Studenten nehmen dafür an Kursen teil. Ein Kurs hat den Zweck bestimmtes Wissen aus einem Bereich, z.B. *Datenmodellierung*, den Studierenden zu lehren. Die Durchführung eines konkreten Kurses ist ein Lernprozess. Um das Wissen über ein bestimmtes Themengebiet in Form eines Kurses lehren zu können, werden die Inhalte des Kurses strukturiert und in einzelne Teile, in so genannte Lerneinheiten, gegliedert. Im Fall des Kurses Datenmodellierung könnten dies die Lerneinheiten „*Konzeptueller Entwurf*“, „*Logischer Entwurf*“ und „*Physischer Entwurf*“ sein. Eine Lerneinheit umfasst wiederum konkrete Lerninhalte, wie z.B. eine Übung oder ein Video zu einem bestimmten Thema. Ein Lerninhalt der Lerneinheit „*Physischer Entwurf*“ könnte daher eine Übung über das Anlegen von Tabellen in einer relationalen Datenbank sein. Der elektronisch abgespeicherte Übungszettel oder das Video zu einem Thema ist eine Lernressource. Zusammenfassend kann man sagen, dass ein Kurs in Lerneinheiten aufgeteilt wird und die Lerneinheit konkrete Lerninhalte unter Verwendung von Lernressourcen an den Studenten vermitteln. Tabelle 1 zeigt diese Bestandteile eines Kurses.

Name	Beschreibung
<b>Lerneinheiten</b>	Dienen zur Strukturierung des Ablaufs eines Kurses
<b>Lerninhalte</b>	Beschreiben die Themengebiete eines Kurses, d.h. was dem Studenten gelernt werden soll.
<b>Lernressourcen</b>	Konkrete Medien welche einem Studenten Lerninhalte präsentieren.
<b>Lernziele</b>	Lernziele bestimmen den Zweck eines Kurses, das Ziel des Lernprozesses. Hat ein Lernender das Lernziel erreicht, ist sichergestellt, dass die Lerninhalte ausreichend beherrscht werden.

**Tabelle 1: Bestandteile eines Kurses**

Absolviert nun ein Student an einer Universität einen Kurs, so muss dieser Kurs natürlich zuerst geplant und zusammengestellt werden. Die Person, die diese Aufgabe übernimmt, wird als der Autor des Kurses bezeichnet. Vom Beginn bis zum Ende eines Kurses arbeitet der Student eine Lerneinheit nach der anderen in einer entweder vorgegebenen (vor Beginn, oder während des Kurses) oder selbst gewählten Reihenfolge ab. Diese Reihenfolge ist die Lerneinheitssequenz. Der gesamte Lernprozess umfasst alle Tätigkeiten des Lernenden und des Lehrenden während des Kurses. Spricht man von einem webbasierten Lernsystem, so ist das Hilfsmittel zur Darstellung und zur Navigation durch die Lerneinheiten eine Webseite.

Im Folgenden wird beispielhaft ein Kurs dargestellt, wie er zurzeit im eTutor enthalten ist. Der konkrete Aufbau des Kurses sowie ein Beispiel zur Durchführung des Kurses befinden sich im Anhang dieser Arbeit.

#### **Kurs Datenmodellierung im eTutor:**

- **Konzeptueller Entwurf – Lerneinheit**
  - Übung 1 – *Lerninhalt/Lernressource*

- Übung 2 – *Lerninhalt/Lernressource*
- **Logischer Entwurf – *Lerneinheit***
  - Übung 1 – *Lerninhalt/Lernressource*
  - Übung 2 – *Lerninhalt/Lernressource*
  - Übung 3 – *Lerninhalt/Lernressource*
- **Physischer Entwurf – *Lerneinheit***
  - Übung 1 – *Lerninhalt/Lernressource*
  - Übung 2 – *Lerninhalt/Lernressource*

Die unterschiedlichen Ansätze und Abläufe durch einen Kurs zu definieren und auszuführen ist Gegenstand des nächsten Kapitels.

## **2 Ansätze zur Ablaufsteuerung**

Das folgende Kapitel 2 behandelt bestehende Ansätze zur Ablaufsteuerung von und innerhalb von Kursen und Lerneinheiten. Der Begriff der Ablaufsteuerung wird erläutert und die Merkmale unterschiedlicher Arten von Ablaufsteuerungen werden beschrieben.

### **2.1 Ablaufsteuerung von Kursen und Lerninhalten**

Generelles Ziel einer Ablaufsteuerung ist es, dem Lernenden die bestmöglichen Lerninhalte zur richtigen Zeit zu vermitteln. Die bestmöglichen Lerninhalte für den Lernenden sind jene Inhalte, welche seinen Wissensstand am effektivsten erweitern. Angenommen, ein Lernender erlernt durch das Absolvieren von Übungen eine bestimmte Thematik, so wird die effektive Wissensvermittlung durch Übungen durch eine Vielzahl von Kriterien, wie z.B. der Schwierigkeitsgrad einer Übung, beeinflusst. Abhängig vom Vorwissen des Lernenden kann dieser z.B. durch zu schwierige Problemstellungen überfordert werden. Andererseits vermitteln z.B. zu einfache Übungen dem Lernenden kein neues Wissen, wodurch die Motivation sinkt. Dieses Problem kann dadurch gelöst werden, dass der Lernende genau jenen Inhalt absolviert, welcher am besten zu seinem derzeitigen Wissensstand passt. Konkret bedeutet dies, dass dem Lernenden die richtige Lerneinheit, sei es eine Aufgabe, eine Frage, ein Beispiel etc. zur richtigen Zeit, abhängig von seinem Wissensstand, präsentiert wird. Ziel ist es, die optimale Abfolge von Lerninhalten aus einem Pool von Lernmaterialien, im Hinblick auf ein gewünschtes Lernziel, zu finden. Diese, an die individuellen Bedürfnisse des Lernenden angepasste Abfolge von Lerneinheiten bezeichnet man als Lerneinheitssequenz [Brus99]. Im Folgenden wird näher auf die Aufgaben und Probleme, mit welcher eine Ablaufsteuerung konfrontiert wird, eingegangen.

Die Ablaufsteuerung in einem webbasierten Lernsystem ermöglicht es dem Autor eines Kurses, diesen zu strukturieren und den Lernprozess durch Gestaltung der Lerneinheitssequenz zu beeinflussen. Die Ablaufsteuerung dient zur Individualisierung und kann nach mehreren Kategorien erfolgen:

- Statische Ablaufsteuerung
- Aktive/Passive Ablaufsteuerung



- Adaptive Ablaufsteuerung
- Individuelle Ablaufsteuerung

Auf die einzelnen Kategorien der Ablaufsteuerung wird nun näher eingegangen.

### **2.1.1 Statische Ablaufsteuerung**

Neben dynamisch adaptiven Systemen gibt es immer noch Systeme, welche nur statische Kurse präsentieren und verwalten können. Ein Kurs ist dann als statisch anzusehen, wenn die Lerneinheitssequenz zum Zeitpunkt des Zugriffs des Lernenden auf den Kurs bereits fest steht. Dieser Abschnitt widmet sich den Techniken, welche eine iterative Verbesserung des Ablaufs statischer Kurse ermöglichen und zeigt Metriken zur Bewertung solcher Kurse auf [Brus00]. Diese Techniken sind aber nicht nur für statische Kurse relevant, sondern können auch zur Beurteilung von dynamischen Kursen herangezogen werden.

**Dynamic Courseware Generation (DCG)** bezeichnet eine Technik, bei welcher ein Kurs an das Wissen eines Lernenden und an ein Lernziel angepasst wird. Entspricht der Lernfortschritt nicht dem erwarteten Lernfortschritt, so wird der gesamte Kurs automatisch neu geplant und angepasst. Wichtig ist hierbei, dass es sich immer um einen statischen Kurs handelt, d.h. die automatische Neugestaltung des Kurses erfolgt bevor ein Benutzer diesen Kurs durchführt. Dies ermöglicht es zum Beispiel einen Kurs zu exportieren und in einem anderen System wieder zu verwenden [BrVa03].

**Concept-based Courseware Analysis (CoCoA)** bezeichnet eine weitere Technik, welche es ermöglicht, die Beschaffenheit und die Qualität von Kursen zu analysieren. Man geht davon aus, dass die Techniken, welche in einem dynamischen Kurs die nächste Lerneinheit in der Lerneinheitssequenz festlegen, ebenso dazu geeignet sind, die Qualität der aufeinander folgenden Lerneinheiten in einem statischen Kurs zu messen. In beiden Fällen müssen Techniken existieren, welche eine Aussage darüber zulassen, welche Lerneinheit dem Lernenden am besten als nächstes präsentiert wird [Brus00]. Bei CoCoA nutzt man dies aus, um statische Kurse zu analysieren. So kann beim Entwurf eines Kurses festgestellt werden, ob die nächste Lerneinheit in der Lerneinheitssequenz des statischen Kurses nicht der, durch die obig genannten Techniken ermittelten,

„besten“ Lerneinheit entspricht und gegebenenfalls darauf reagiert werden [BrVa03].

### **Beispiel**

Der eTutor hat eine Vielzahl von Kursen in Form von Übungszetteln erfasst. Diese Kurse sind als statisch anzusehen. Zurzeit gibt es im eTutor keine Möglichkeit Aussagen über die Qualität dieser Übungszettel zu machen. Eine statische Ablaufsteuerung ermöglicht es, den in Abschnitt 1.2 beschriebenen Kurs Datenmodellierung zu bewerten und gegebenenfalls anzupassen. Zum Beispiel könnte eine statische Ablaufsteuerung aufzeigen, dass zwei Übungen zu den Lerneinheiten „*Konzeptueller Entwurf*“ und „*Physischer Entwurf*“ nicht den gewünschten Lernerfolg bringen. Anhand dieser Aussagen wird der statische Kurs geändert und weitere Übungen werden hinzugefügt.

### **2.1.2 Aktive und Passive Ablaufsteuerung**

Der Unterschied zwischen aktiver und passiver Ablaufsteuerung liegt im Lernziel. Ist dem System das Lernziel explizit bekannt, so spricht man von einer aktiven Ablaufsteuerung. In diesem Fall versucht das System, dieses Lernziel durch Auswahl einer Abfolge von geeigneten Lernmaterialien zu erreichen. Ein Lernziel kann zum Beispiel das Erlangen einer Teilmenge des Domänenwissens, oder aber das erfolgreiche Abschließen eines bestimmten Themas, z.B. Physischer Entwurf, sein. Bei der aktiven Ablaufsteuerung werden weiters Systeme mit einem fixem Lernziel und einem variablen Lernziel unterschieden. Ein variables Lernziel kann sich im Gegensatz zu einem fixen Lernziel während des Lernprozesses ändern. So wird z.B. von der Ablaufsteuerung erkannt, dass dem Lernenden Vorwissen fehlt und deshalb wird das variable Lernziel dahingehend geändert, dass Vorwissen zu erlangen. Die aktive Ablaufsteuerung sucht den besten, individuellen Pfad um das Lernziel zu erreichen [Brus99].

Die passive Ablaufsteuerung ist eine reagierende Technologie. In diesem Fall wird kein festgelegtes Lernziel benötigt. Passive Ablaufsteuerungen können anhand verschiedenster Faktoren, wie die falsche Beantwortung einer Frage, oder ein erfolgloser Versuch ein Problem zu lösen, angestoßen werden. Ist dies geschehen, so stellt das System dem Lernenden Lernmaterialien zur Verfügung, welches ihm das notwendige Wissen vermitteln sollen [Brus99].

In vielen Systemen mit einer Ablaufsteuerung von Kursen kann man des Weiteren eine Mikro- und Makro-Ablaufsteuerung unterscheiden. Wird während des Lernprozesses ein Konzept, ein Thema oder ein Kapitel zur Erreichung des Lernzieles oder eines Teilziels ausgewählt, so spricht man von einer Makro-Ablaufsteuerung oder Wissens-Ablaufsteuerung [Brus99]. Bei einer Mikro-Ablaufsteuerung oder der Aufgaben-Ablaufsteuerung, wird die nächste Aufgabe bestimmt, welche dem Lernenden präsentiert wird. Systeme welche nur eine Ablaufsteuerung von Aufgaben unterstützen, können also nur die Reihenfolge, der dem Lernenden zu präsentierenden Lerninhalte, ändern [BrVa03].

### **Beispiel**

Im eTutor ist es zurzeit nicht möglich Lernziele festzulegen. Angenommen für jede Lerneinheit (Konzeptueller Entwurf, Logischer Entwurf, Physischer Entwurf) des im Abschnitt 1.2 beschriebenen Kurses Datenmodellierung wird ein Lernziel festgelegt. Jedes Lernziel ist erst dann erreicht, wenn der Student eine gewisse Punktzahl bei den Übungen erreicht hat. Eine aktive Ablaufsteuerung liefert einem Studenten solange Übungen, bis alle Ziele erreicht wurden.

### **2.1.3 Adaptive Ablaufsteuerung**

In derzeitigen Lernsystemen beschränkt sich die Individualisierung des Lernprozesses, trotz seiner immer wieder bestätigten Bedeutung, meist auf 2 Aspekte [RFDP06]:

- Anpassung der Kurse vor dem Beginn des Lernprozesses  
(siehe 2.1.1 Statische Ablaufsteuerung)
- Auswahl der nächsten Lerneinheit während des Lernprozesses  
(siehe 2.1.2 Aktive und Passive Ablaufsteuerung)

Ein umfassender Ansatz besteht in der Adaption der Lerninhalte an die Bedürfnisse der Lernenden. Herkömmliche Sequenzmechanismen können dazu verwendet werden die Lerneinheitssequenz zu bestimmen (Mikro-Ablaufsteuerung) und weitere Regeln passen die Lernressourcen an die Vorlieben und Bedürfnisse des Lernenden an. Sequenzmechanismen werden also dahingehend erweitert, dass Lerninhalte an den Lernenden angepasst werden. Zum Beispiel kann je nach Vorwissen von einem Übungszettel zum logischen

Datenbankentwurf jene Information entfernt werden, welche ein Lernender nicht mehr benötigt.

### **Beispiel**

Aufgrund der eTutor-Architektur können eine Vielzahl von unterschiedlichen Übungen dargestellt werden. Es ist jedoch zurzeit nicht möglich eine einzelne Übung an die Bedürfnisse von Studenten anzupassen.

## **2.1.4 Individualisierte Ablaufsteuerung**

Die bereits beschriebenen Ansätze bieten eine gute Möglichkeit zur Anpassung des Lernprozesses an den Lernenden. Sie unterliegen aber auch gewissen Einschränkungen. Grundsätzlich sind dafür zwei Voraussetzungen notwendig:

- Angemessene Abbildung der Stärken und Schwächen des Lernenden im System.
- Auswahl der möglichen Alternativen im Lernprozess bzw. der richtigen Ablaufregeln.

Diese Aufgaben müssen vor Beginn des Lernprozesses von einem Autor identifiziert und wahrgenommen werden. Es handelt sich hierbei um äußerst komplexe und schwierige Aufgaben, von denen oft der Erfolg des Lernprozesses abhängt [DeBr03].

Individuelles Tutoring geht von der Annahme aus, dass es für einen Tutor leichter ist, während des Lernprozesses den Kurs an die Bedürfnisse eines Lernenden anzupassen, als vor Beginn des Lernprozesses. Zum Beispiel kann ein Tutor den Lernenden auf Lernmaterialien im Internet zu diesem Thema verweisen, welche dann automatisch in den Kurs aufgenommen werden [HMSc04].

Bei der individualisierten Ablaufsteuerung werden also so genannte Curriculum Alternativen, das bedeutet individualisierte Kurse für einen Lernenden, erstellt. Dies ist zwar aufwändig, jedoch bietet sich auch die Möglichkeit, bestehende Curriculum Alternativen weiter zu verwenden und außerdem verringern sich dadurch mögliche Fehler beim Erstellen eines Kurses, da die Anpassung während eines Lernprozesses erfolgt ist [HMSc04].

## **Beispiel**

Der im Abschnitt 1.2 beschriebenen Kurs Datenmodellierung muss von jedem Studenten, welcher an dieser Lehrveranstaltung teil nimmt, absolviert werden. Besitzt ein Student bereits Vorwissen, so kann das zurzeit vom eTutor nicht berücksichtigt werden. Da der eTutor jedoch ein Benutzermodell besitzt, welches eine Vielzahl von Informationen über Studenten abspeichert, ist es sinnvoll diese Informationen auch zu nutzen und für jeden Studenten Kurse im eTutor anzubieten, welche auf seine persönlichen Bedürfnisse zugeschnitten sind.

## **2.2 Auswahlkriterien**

Ziel der Ablaufsteuerung ist es, den Lernenden durch den Lernprozess zu führen in dem festgelegt wird, welche Lernressourcen diesem in welcher Abfolge präsentiert wird, um ein bestimmtes Lernziel zu erreichen.

Folgende Kriterien werden zum Vergleich der Lösungsansätze verwendet:

- **Änderungsaufwand**  
Es wird ermittelt, welcher Ansatz im eTutor den geringsten Änderungsaufwand verursacht. Hierzu werden die Architekturen der Ansätze mit der Architektur des eTutors verglichen und die Auswirkungen auf die einzelnen Teile des eTutors werden beschrieben.
- **Umsetzung**  
Dieses Kriterium gibt an, ob ein Ansatz konkrete Vorschläge für die Umsetzung einer Ablaufsteuerung in einem Intelligenten Tutoriellen System enthält. Hierbei spielt es eine Rolle, wie detailliert und konkret diese Vorschläge sind, da sie den Implementierungsaufwand im eTutor verringern.
- **Lernziele**  
Dieses Kriterium gibt an, ob mit einem Ansatz konkrete Lernziele definiert und erreicht werden können. Im eTutor ist eine aktive Ablaufsteuerung sinnvoll.
- **Adaption auf Mikro-Ebene**  
Da der eTutor vor allem zur Durchführungen von Übungen gedacht ist, ist besonders das Anpassen des Lernprozesses durch eine automatische

Auswahl einer Übung wichtig. Es wird untersucht inwieweit dies mit einem der Ansätze möglich ist.

- **Einbindung des Studentenwissens**

Da der eTutor ein Benutzermodell beinhaltet, welches detaillierte Informationen über einen Studenten enthält, ist es sinnvoll diese mit Hilfe einer Ablaufsteuerung auswerten, und zur Modellierung eines individuellen Lernprozesses verwenden zu können. Dieses Kriterium gibt an, ob ein Ansatz Informationen über Studierende berücksichtigt.

- **Modellierungsaufwand**

Dieses Kriterium beschreibt den Modellierungsaufwand für den Autor eines Kurses um einen individuellen Lernprozess zu gestalten.

### **2.3 Fazit**

Die Gegenüberstellung der verschiedenen Möglichkeiten zur Ablaufsteuerung hat gezeigt, dass sich für die Umsetzung im eTutor (siehe Kapitel 7) eine individualisierte Ablaufsteuerung anbietet. Um jedoch auch die Abfolge der Lerneinheiten während der Durchführung eines Kurses beeinflussen zu können und zur Verknüpfung der Ablaufsteuerung mit dem bestehenden Benutzermodell, sind auch Konzepte einer aktiven und passiven Ablaufsteuerung notwendig.

Techniken zur Beurteilung von statischen Kursen (siehe Abschnitt 2.1.1), welche bereits im eTutor enthalten sind, sind nicht von Interesse, da mit der Einführung der Ablaufsteuerung das Konzept der Übungszettel nicht übernommen wird. Weiters werden keine Techniken zur Adaption von Lerninhalten, d.h. zur Adaption von im eTutor gespeicherten Übungen, umgesetzt (siehe Tabelle 7: Kategorien der Ablaufsteuerung).

Aufbauend auf die unterschiedlichen Arten der Ablaufsteuerung eines Lernprozesses, werden im folgenden Kapitel 3 konkrete Ansätze zur Ablaufsteuerung betrachtet und verglichen. Aufgrund dieses Vergleiches wird ein Ansatz gewählt, welcher im eTutor umgesetzt werden kann.

## **Teil 2**

### **Stand der Technik und Gegenüberstellung der Ansätze**

### 3 Stand der Technik

Die stetige Weiterentwicklung webbasierter Lernsysteme hat zur Herausbildung einer Reihe von Standards geführt. Neben den Entwicklungen webbasierter Lernsysteme wurde auch im Bereich des Hypermedia schon früh zu Adaptionmöglichkeiten geforscht. Dieses Kapitel widmet sich der Motivation, welche hinter der Entwicklung von Standards steht und zeigt, im Hinblick auf eine Ablaufsteuerung, auf, in welchen Bereichen diese relevant sind. Weiters werden in diesem Kapitel konkrete Lösungsvorschläge beschrieben, wie man den Lernprozess individualisieren und an die Bedürfnisse eines Studenten anpassen kann. Dazu wird im Abschnitt 3.1 die Spezifikationssammlung SCORM® 2004 3<sup>rd</sup> Edition vorgestellt, welche Konzepte zu einer Ablaufsteuerung von Lernprozessen vorstellt, sowie die Ansätze des Adaptiven Hypermedia. Im letzten Teil dieses Kapitels wird das Intelligente Tutorielle System „eTutor“, welches am Institut für Data & Knowledge Engineering an der Johannes Kepler Universität Linz entwickelt wurde [Hofe05], vorgestellt und eine Entscheidung präsentiert, welche Ansätze in diesem System umsetzbar sind.

Die Bedeutung und die Nachfrage nach Standards im Bereich des webbasierten Lernens wachsen. Ziel ist es, eine rasche und kosteneffiziente Entwicklung von Kursen zu erreichen. Vor allem die Wiederverwendung von bereits erstellten Lerninhalten durch standardisierte Systeme soll gefördert werden [Kazi04]. Wünschenswert wäre ein Repository, welches standardisierte Lernressourcen enthält und dann von einer Vielzahl an webbasierten Lernsystemen genutzt werden kann [Kric05]. Durch die Verwendung von Standards sollen folgende Kriterien verbessert werden [ADLa06, S.6]:

- **Zugänglichkeit**

Der Zugriff und die Suche von entfernt gespeicherten Lernressourcen und deren Weiterverteilung von einem lokalen System zu entfernten Systemen soll ermöglicht werden.

- **Anpassungsfähigkeit**

Lernressourcen sollen schnell und einfach an individuelle und organisationelle Bedürfnisse anpassbar sein.



- **Erschwinglichkeit**

Die Effizienz und Produktivität eines webbasierten Lernsystems soll durch Reduktion der Kosten und der Zeit für die Aufbereitung der Lernressourcen gesteigert werden.

- **Beständigkeit**

Neuen technologischen Entwicklungen soll ohne aufwändige und kostenintensive Neugestaltung eines bestehenden Systems standgehalten werden. Das heißt es soll kein Neudesign, keine Neukonfiguration und keine Neuprogrammierung notwendig werden.

- **Interoperabilität**

Lernressourcen welche an einem bestimmten Ort, mit einem bestimmten Werkzeug, auf einer bestimmten Plattform entwickelt wurden, sollen auch auf einem anderen Ort, mit einem anderen Werkzeug und auf einer anderen Plattform einsetzbar sein.

- **Wiederverwendbarkeit**

Einmal entwickelte Lernressourcen sollen in verschiedenen Systemen, von verschiedenen Anwendungen und in einem anderen Kontext einsetzbar sein.

Im Folgenden werden einige für das webbasierte Lernen relevante Bereiche, in welchen bereits Standards entwickelt wurden, aufgelistet [LTSO06].

- **Metadaten**

Metadaten im Bereich der Lernsysteme sind Informationen über Informationen, welche die Verwaltung, das Suchen und den Zugriff im World Wide Web erleichtern sollen. Metadatenstandards helfen eine Vielzahl von frei und unstrukturiert vorhandenen Dokumenten im Web zu katalogisieren, in dem diese Ressourcen in einer standardisierten Art und Weise beschrieben werden. Im Bereich des webbasierten Lernens werden diese Metadaten um Informationen, welche in diesem Bereich relevant sind z.B. Autor einer Lerneinheit, oder Erstellungsdatum, erweitert. Einige Standards in diesem Bereich sind LOM (Learning Object Metadata) [LTSC06], ARIADNE Metadata [ARIA04], IMS Metadata [SSIB03] und ADL Metadata [ADLb06].

- **Content Aggregation**

Das steigende Bedürfnis Lernressourcen in verschiedenen webbasierten Lernsystemen einzusetzen, motivierte die Entwicklung von Standards im Bereich der *Inhaltsaggregation* (Content Aggregation). Dies führte zur Entwicklung von Formaten und Prozeduren, welche eine Kapselung verschiedenster Lernressourcen, sei es ein Teil eines Kurses, ein ganzer Kurs oder sogar eine Sammlung von Kursen in, in sich geschlossene, Packages ermöglichen. Die Umsetzung unterschiedlicher didaktischer Ansätze und die Einbindung unterschiedlicher Medien soll dabei unterstützt werden.

Einige Standards in diesem Bereich sind SCORM® CAM [ADLb06], IMS Content Packing [SSIB04], AICC Course Structure und AICC Packaging [AICC06].

- **Informationen über Lernende**

Informationen über Lernende sollen einheitlich gespeichert und somit auch austauschbar sein. Standards in diesem Bereich beschreiben Datenmodelle, welche sowohl die Syntax als auch die Semantik zur Beschreibung der charakteristischen Eigenschaften eines Lernenden, seiner Fähigkeiten und seines Wissens ermöglichen.

Einige Standards/Institutionen in diesem Bereich sind IMS Learner Information Package (LIP), LTSC Public and Private Information (PAPI) [LTSC06] und ISO/IEC JTC1 SC36 WG3 [ISO06].

- **Laufzeitumgebung**

Um die Wiederverwendbarkeit von Lernressourcen zu gewährleisten ist es notwendig die Lernressourcen von der Logik, welche zu deren Verarbeitung dient, zu trennen. Diese Verarbeitungslogik ist die *Laufzeitumgebung* eines webbasierten Lernsystems. Die Hauptaufgaben der Laufzeitumgebung sind der Transport der Lernressourcen zu den Lernenden und die Gewährleistung der Kommunikation zwischen Lernressourcen und dem webbasierten Lernsystem.

Einige Standards/Institutionen in diesem Bereich sind AICC CMI/Lesson Communication [AICC06], SCORM® Runtime und IEEE P1484.11 (CMI) [ADLc06].

- **Architektur**

Standards im Bereich der Architekturentwicklung haben das Ziel, ein Framework zu schaffen, welches die Entwicklung einer großen Bandbreite von webbasierten Lernsystemen, sowie einen Vergleich und eine Evaluierung dieser, unterstützt.

Die steigende Bedeutung und der verstärkte Einsatz von webbasierten Lernsystemen hat zu dem Problemen geführt, dass viele webbasierte Lernsysteme für eine bestimmte Zielgruppe z.B. eine bestimmte Firma oder einen bestimmten Kurs entwickelt wurden. Standards im Bereich der Architektur dieser Systeme werden benötigt um die Skalierbarkeit solcher Produkte zu gewährleisten und somit die Kosten für die Integration in bestehende Systeme zu verringern.

Einige Standards/Institutionen in diesem Bereich sind IEEE Learning Technology System Architecture (LTSA) [LTSA06], Schools Interoperability Framework (SIF) und IMS Abstract Framework (IAF) [SSIB03].

- **Ablaufsteuerung**

Die Ablaufsteuerung steuert die Lernsequenz des Lernprozesses. Sie bestimmt also, in welcher Reihenfolge die Lernressourcen dem Lernenden präsentiert werden. Standards in diesem Bereich befassen sich sowohl mit der Modellierung der Lernsequenz, als auch mit der für die Ablaufsteuerung notwendigen Verarbeitungslogik.

In diesem Bereich gibt es drei Standards/Institutionen, SCORM® Sequencing and Navigation [ADLc06], IMS Simple Sequencing (SS) [SSIB03] und IEEE P1484.6 [LTSC06].

Im Folgenden wird auf die Spezifikationssammlung SCORM® näher eingegangen, welche für die Entwicklung einer Ablaufsteuerung in einem Lernsystem eine bedeutende Rolle spielt.

### **3.1 SCORM® 2004 3<sup>rd</sup> Edition**

SCORM® 2004 3<sup>rd</sup> Edition ist eine Sammlung von Spezifikationen und Standards welche von ADL (Advanced Distributed Learning) entwickelt wurde. SCORM® steht für Sharable Content Reference Model. Das SCORM® Referenzmodell beschreibt die Teile, welche zur Lösung eines bestimmten Problems notwendig sind, wie diese zusammengefügt werden müssen und welche Standards zum

Einsatz kommen. [ADLa06, S.5] Des Weiteren benutzt SCORM® den Begriff LMS (Learning Management System) zur generellen Beschreibung eines webbasierten Lernsystems. Der Begriff LMS umfasst Funktionalitäten wie den Transport, die Überwachung, das Reporting und das Management von Lernressourcen, des Lernfortschrittes und der Aktivitäten des Lernenden.

Die folgende Grafik (Abbildung 2) zeigt die Referenzarchitektur eines LMS. Der von SCORM® so bezeichnete Tracking Service verfolgt das Verhalten eines Benutzers während des Lernprozesses mit Hilfe der API Instanz am Client welche die Informationen an den Server schickt. Dieses Verhalten wirkt sich auf die Ablaufsteuerung - Sequencing Service - aus. Die Ablaufsteuerung ermittelt nun aufgrund von Regeln, welche Lerninhalte dem Lernenden präsentiert werden und teilt dies dem Content Management Service mit. Die Aufgabe des Content Management Service ist die Verwaltung der Lerninhalte. Dieser übergibt den Lerninhalt an den Delivery Service, welcher die Darstellung für den Client aufbereitet. Inhalte können nun vom Studenten bearbeitet werden. Die Interaktionen werden wieder an den Tracking Service übergeben und der Bearbeitungszyklus beginnt von Neuem. Der Course Administrative Service (Verwaltung von Kursen), der Learner Profile Service (verwaltet Informationen über Lernende) sowie der Service Testing/Assessment (Erstellen und Testen von Kursen) sind jene Teile eines LMS, auf welche diese Diplomarbeit nicht näher eingeht, da diese bereits im eTutor realisiert sind.

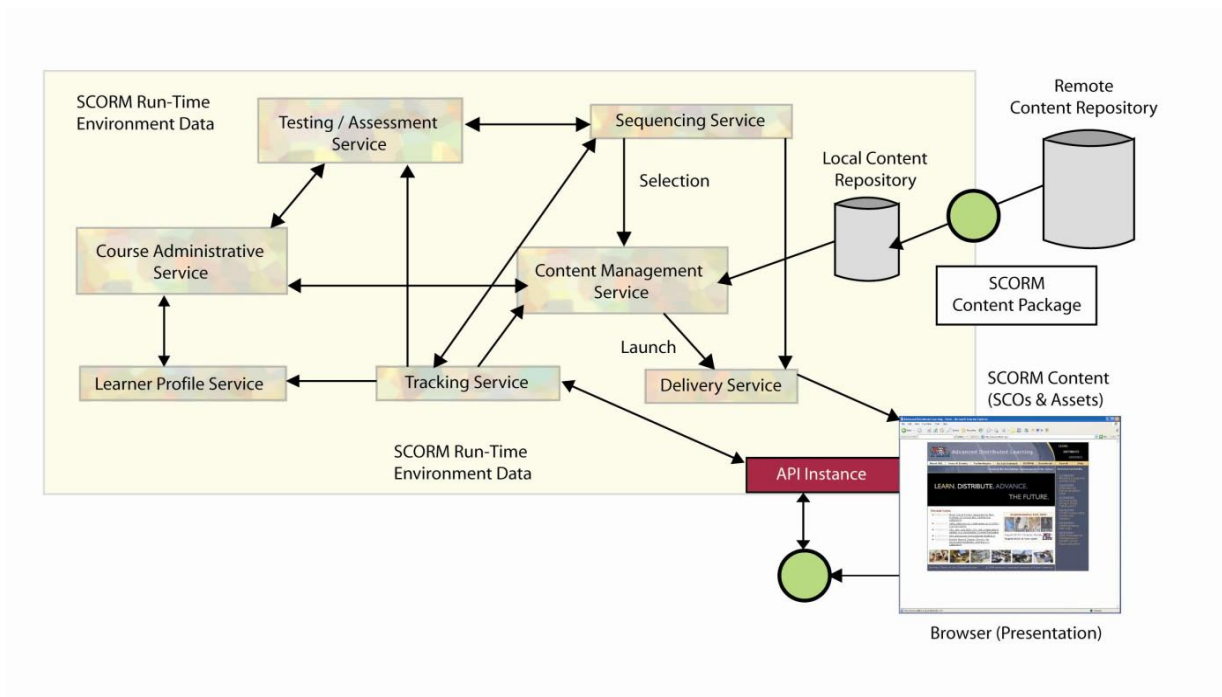


Abbildung 2: Verallgemeinertes Modell eines LMS [ADLa06, S.7]

Neben ADL waren auch die Institutionen ARIADNE, AICC, IEEE LTSC sowie IMS GLC an der Entwicklung von SCORM® 2004<sup>rd</sup> Edition beteiligt. Folgende Spezifikationen und Standards beeinflussten das Referenzmodell von SCORM®

- IEEE Data Model for Content Object Communication
- IEEE ECMAScript Application Programming Interface for Content to Runtime Service Communication
- IEEE Learning Object Metadata (LOM)
- IEEE Extensible Markup Language (XML) Schema Binding for Learning Object Metadata Data Model
- IMS Content Packing
- IMS Simple Sequencing

SCORM® teilt sich in vier, in der SCORM-Terminologie „Book“ (Abbildung 3) genannte, Teile:

- **Book 1 – Overview** gibt einen Überblick über die Books von SCORM® und beschreibt die Ziele und die Entwicklung von SCORM®.

- **Book 2 – Content Aggregation Model (CAM)** beschreibt die Schlüsselkonzepte von SCORM®. Es wird darauf eingegangen, welche Lernobjekte verwendet werden und wie diese für den Austausch zwischen Systemen strukturiert werden müssen. Die Beschreibung der Lernobjekte durch Metadaten für die automatisierte Suche, sowie die Definition von Regeln zur Ablaufsteuerung wird ebenfalls erläutert.
- **Book 3 – Run-Time Environment (RTE)** beschreibt den Aufbau der Laufzeitumgebung zur Sicherstellung der Interoperabilität von Lernsystemen. Zu dieser zählt z.B. die Spezifikation des Aufrufs von Lernobjekten, das Protokoll zur Kommunikation zwischen dem LMS und den Lernobjekten und das Protokoll zur Interaktion des Lernenden mit den Lernobjekten.
- **Book 4 – Sequencing and Navigation (SN)** beschreibt im Detail die Ablaufsteuerung in SCORM®. Es wird beschrieben, welche Konstrukte zur Modellierung einer Ablaufsteuerung notwendig sind und wie diese dann von SCORM® während des Lernprozesses ausgewertet werden. Zu diesem Zweck definiert SCORM® Funktionalitäten welche hierfür vom LMS umgesetzt werden müssen.

Die folgende Grafik (Abbildung 3) zeigt, welche Spezifikationen und Standards welche Teile von SCORM® betreffen.

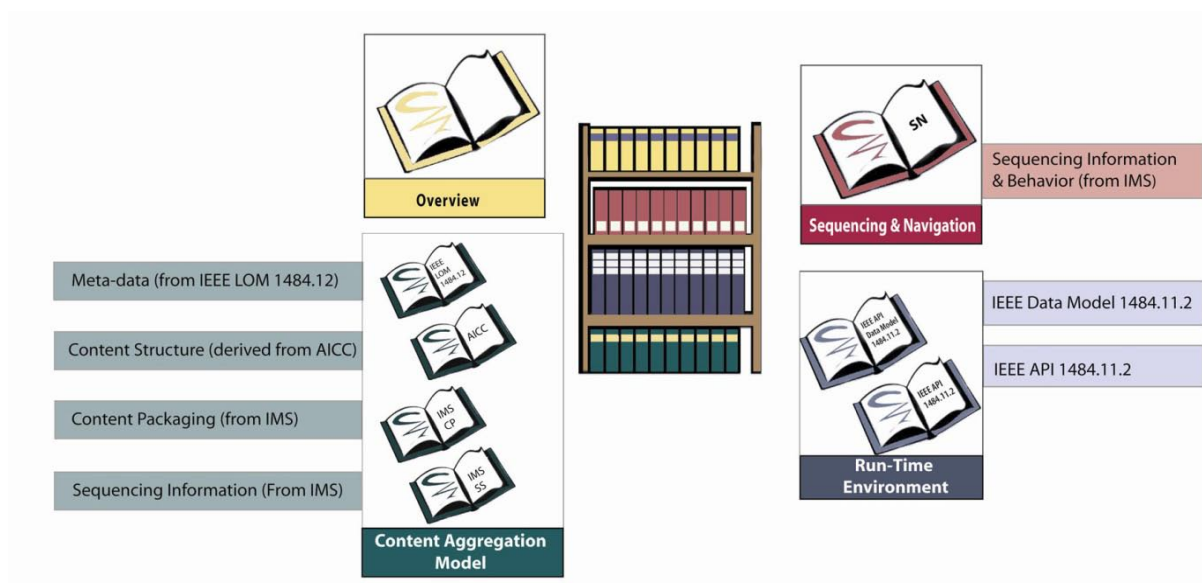


Abbildung 3: SCORM® Books [ADLa06, S.11]

Nachfolgend wird auf das Book2 – Content Aggregation Model näher eingegangen.

### **3.1.1 Book 2 – Content Aggregation Model**

Book 2 gliedert sich in vier Abschnitte, welche im Folgenden näher erläutert werden.

#### **(1) Lerninhaltsmodell**

Das Lerninhaltsmodell definiert die in der Spezifikation verwendeten Begriffe und Komponenten.

#### **(2) Lernpackages**

Diese beschreiben die Voraussetzungen für das Zusammenfassen von Lernressourcen zu Lernpackages.

#### **(3) Metadaten**

Metadaten dienen zur Beschreibung der SCORM®-Komponenten.

#### **(4) Sequencing und Navigation**

Der Abschnitt Sequencing und Navigation beschreibt die Komponenten, welche für die Ablaufsteuerung in einem Lernsystem notwendig sind.

#### **(1) Lerninhaltsmodell**

Das SCORM® Lerninhaltsmodell besteht aus:

- **Assets**

Ein Asset ist der kleinste Bestandteil einer Lernressource (Abbildung 4). Ein Asset ist eine elektronische Repräsentation eines Mediums, einer Lernressource (z.B. ein Text, ein Bild, ein Video usw.), welches in einem Webbrowser angezeigt werden kann. Ein Asset kann wiederum aus anderen Assets bestehen [ADLb06, S.21].

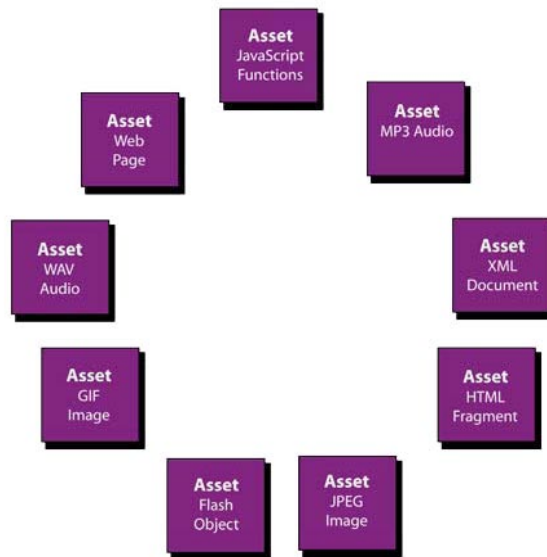


Abbildung 4: Beispiele für Assets [ADLb06, S.21]

- **Shareable Content Object**

Ein Shareable Content Object (SCO) ist eine Sammlung von einem oder mehreren Assets und stellt die kleinste, aufrufbare Lernressource aus der Sicht des LMS dar (Abbildung 5). Der einzige Unterschied zwischen einem Asset und einem SCO ist die Tatsache, dass ein SCO mit der Laufzeitumgebung des LMS, über eine klar definierte Schnittstelle kommuniziert. Ein SCO soll unabhängig vom Lernkontext sein und so die Wiederverwendbarkeit gewährleisten. Unterstützt ein LMS die von SCORM® spezifizierte Laufzeitumgebung, so kann es SCOs beliebiger Autoren aufrufen und überwachen [ADLb06, S.22].



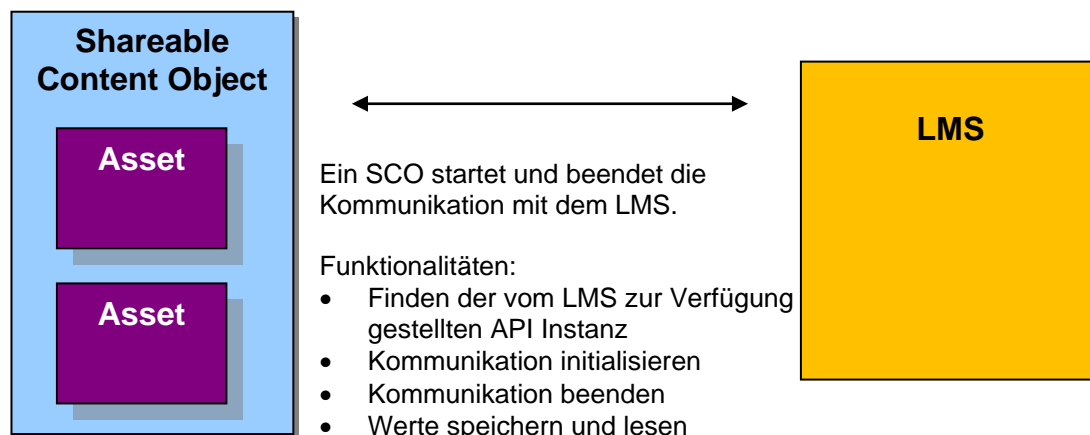


Abbildung 5: SCOs und LMS [ADLb06, S.22]

- **Lernaktivität**

SCORM® definiert eine Lernaktivität als eine bedeutungsvolle Einheit von Instruktionen, also alle Interaktionen die ein Lernender im Rahmen des Lernprozesses ausführt z.B. Auswahl einer bestimmten Lerneinheit wie „konzeptueller Entwurf“. Lernaktivitäten können wiederum aus anderen Lernaktivitäten, Assets oder SCOs bestehen [ADLb06, S.23]

- **Lerninhaltsorganisation**

Die Lerninhaltsorganisation beschreibt den Zustand in welchem Lernressourcen (z.B. Assets oder SCOs) verwendet werden können, d.h. welche Aktivitäten eine Lernressource aktivieren. Diese Inhaltsorganisation kann mit Informationen für die Ablaufsteuerung erweitert werden und bestimmt somit den Ablauf des Lernprozesses eines Lernenden [ADLb06, S.24].

- **Lerninhaltsaggregation**

SCORM® versteht unter der Lerninhaltsaggregation sowohl den Vorgang zusammen gehörende Lernressourcen zu einer Einheit zusammenzufassen, als auch die bei diesem Vorgang entstandene neue Einheit [ADLb06, S.25].

## **(2) Packaging von Lerninhalten**

SCORM® spezifiziert die Zusammenfassung von Lernressourcen zu Packages um Interoperabilität beim Austausch von webbasierten Lernressourcen zu gewährleisten. Ziel ist es, Interoperabilität beim Import, Export sowie beim Zusammenfassen und Zerlegen der Pakete zu erreichen.

Ein Lerninhaltspackage besteht aus einem Manifest und den Lernressourcen. Ein Lerninhaltspackage muss all jene Information beinhalten, welche bei seiner Benutzung benötigt werden. Das Manifest ist eine XML-Datei, welche den Inhalt des Package beschreibt. Es enthält sowohl die Metadaten, als auch die Lerninhaltsorganisation. Weiters ist eine Beschreibung der benötigten Ressourcen, seien es Ressourcen, welche direkt im Package enthalten sind, oder externe Ressourcen, enthalten. Externe Ressourcen werden über eine URI (Universal Resource Indicator) referenziert. Die Lerninhaltspackages werden komprimiert und als Package Interchange File (.pif) abgespeichert [ADLb06, S.30].

## **(3) Metadaten**

Metadaten dienen zur einheitlichen Beschreibung aller im Content Aggregation Model aufgelisteten Komponenten des LMS. Diese einheitliche Beschreibung ermöglicht das Suchen und Auffinden der Komponenten. Weiters dient sie dazu dem Lernenden Informationen über die Inhaltsstruktur (z.B. eines Kurs oder eines Modules) und den Lernressourcen zu vermitteln.

SCORM® verwendet zu diesem Zweck das IEEE LOM Information Model. Dieses gibt vor, welche Elemente zur Beschreibung der Komponenten eingesetzt werden können. Die Elemente sind auf neun Kategorien aufgeteilt [ADLb06, S.93].

- (1) General – Diese Kategorie kann allgemeine Informationen über eine Komponente als Ganzes enthalten.
- (2) Life Cycle – Diese Kategorie kann Informationen über die Geschichte einer Komponente und ihren aktuellen Status enthalten.
- (3) Meta-metadata – Diese Kategorie kann Informationen über den Metadateneintrag selbst enthalten.
- (4) Technical – Diese Kategorie kann Informationen über technische Anforderungen und Besonderheiten einer Komponente enthalten.

- (5) Educational – Diese Kategorie kann Informationen über die pädagogischen Eigenschaften einer Komponente enthalten.
- (6) Rights – Diese Kategorie kann Informationen über rechtliche Bestimmungen (z.B. Urheberrechte oder Nutzungsbestimmungen) einer Komponente enthalten.
- (7) Annotation – Diese Kategorie kann Kommentare und Informationen über den Autor und den Erstellungszeitpunkt des Kommentars enthalten.
- (8) Classification – Diese Kategorie kann angeben, zu welcher Klasse eine Komponente gehört, falls ein Klassifikationsschema verwendet wird.

#### **(4) Sequencing and Navigation**

Im Bereich Sequencing and Navigation (nicht zu verwechseln mit „Book 4 Sequencing & Navigation [ADLd06]“) definiert SCORM® jene XML-Elemente, welche für die Umsetzung der Ablaufsteuerung notwendig sind. Auf die für die Speicherung einer Ablaufsteuerung notwendigen XML-Elemente wird nicht näher eingegangen, da diese für eine Umsetzung in einem bestehenden LMS wie dem eTutor nicht relevant sind.

##### **3.1.2 Book 3 – Run-Time Environment**

Dieser Abschnitt von SCORM® beschreibt die Laufzeitumgebung eines LMS (Abbildung 6). SCORM® führt hier den Begriff Lernobjekt ein. Ein Lernobjekt ist entweder ein Asset oder ein SCO. Folgende Abschnitte werden näher erläutert.

- Mechanismen zum Aufrufen eines Lernobjektes.
- Mechanismen zur Kommunikation zwischen dem LMS und einem SCO.
- Ein Datenmodell um den Lernfortschritt des Lernenden zu verfolgen.

Die Laufzeitumgebung beschreibt ein Modell, welches zum Zeitpunkt des Aufrufes eines Lernobjektes ansetzt. Die Laufzeitumgebung ist nicht für die Auswahl des Lernobjektes zuständig. Hierfür dient die Ablaufsteuerung, welche festlegt, welches Lernobjekt zu einem bestimmten Zeitpunkt an den Lernenden geliefert werden soll. Ziel der Laufzeitumgebung ist es, sowohl die Wiederverwendbarkeit, als auch Interoperabilität bei verschiedensten LMS zu gewährleisten. Die folgende Grafik zeigt die Interaktion zwischen LMS und Lernobjekten. Die drei wichtigsten Aspekte dabei sind (nach [ADLc06]):

- Aufruf eines Lernobjektes
- Kommunikation zwischen dem LMS und dem SCO. Diese wird durch die API-Instanz bereitgestellt. Sie informiert das LMS über den Status eines Lernobjektes (z.B. abgebrochen, wartend oder fehlerhaft) und dient zum Speichern und Lesen von Daten.
- Das Datenmodell beschreibt die Struktur der Daten, welche für ein SCO gespeichert werden. Um die Wiederverwendbarkeit von Lernobjekten zu gewährleisten, dürfen SCOs nur Daten verwenden, welche im Datenmodell definiert wurden.

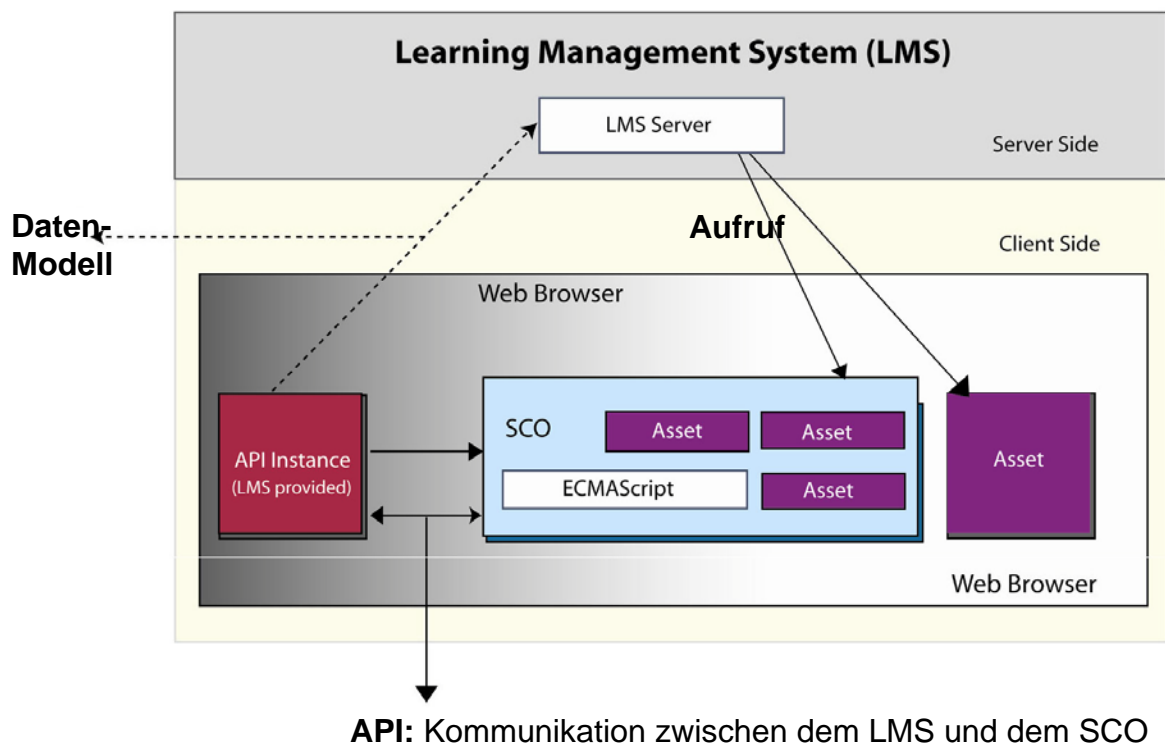


Abbildung 6: SCORM® Modell der Laufzeitumgebung [ADLc06, S.18]

### 3.1.3 Book 4 – Sequencing and Navigation

Dieser Abschnitt von SCORM® beschreibt die Ablaufsteuerung in einer SCORM-Umgebung. Die Ablaufsteuerung basiert auf der IMS Simple Sequencing Spezifikation [SSIB03] und erweitert diese. Es wird spezifiziert, welches Verhalten und welche Funktionen ein SCORM® konformes LMS umsetzen muss, um Informationen über die Ablaufsteuerung während der Laufzeit anzuwenden.

Es beinhaltet folgende Kernkonzepte:

1. Konzepte der Ablaufsteuerung wie z.B. Lernaktivitäten, Aktivitätsbäume, Cluster etc.
2. Das Definitionsmodell der Ablaufsteuerung, welches genau beschreibt und festlegt, welche Informationen für die Ablaufsteuerung mit Lernaktivitäten verbunden werden können.
3. Das Verhaltensmodell der Ablaufsteuerung, welches das Verhalten des LMS in Bezug auf die Ablaufsteuerung und die Lernerfahrung des Lernenden festlegt.
4. Navigationselemente sowie Daten und Anforderung an die Navigation

Kapitel 6 geht detailliert auf die oben erwähnten Konzepte ein. Der folgende Abschnitt 3.2 beschreibt den Ansatz des Adaptiven Hypermedia zur Ablaufsteuerung in einem LMS. Dieser Ansatz wird der von SCORM® beschriebenen Ablaufsteuerung gegenübergestellt.

### **3.2 *Adaptives Hypermedia***

Ein Problem von herkömmlichen Hypermedia-Systemen ist die Tatsache, dass die Webseiten für jeden Benutzer gleich aussehen. Adaptive Hypermedia-Systeme bieten einen Ansatz, Inhalte an die unterschiedlichen Benutzer anzupassen. So ermöglicht dieser Ansatz unter Anderem die Lerninhalte eines webbasierten Lernsystems, sowie auch die Abfolge der Lerneinheiten, je nach Wissen und besonderen Eigenschaften eines Benutzers, zu adaptieren [Brus01]. Adaptives Hypermedia ist also immer dort sinnvoll einsetzbar, wo Benutzer mit verschiedenen Zielen und mit unterschiedlichem Wissen Webseiten benutzen. Ein Adaptives Hypermedia-System verfügt sowohl über ein Benutzermodell, in dem die Eigenschaften des Benutzers gespeichert werden, als auch über ein Domänen-Modell, welches die Konzepte der Lernumgebung enthält. Anhand dieser Modelle können nun die Navigationselemente und die Präsentation einer Webseite angepasst werden. Das Benutzermodell wird dabei immer wieder an das Benutzerverhalten angepasst.

#### **3.2.1 Techniken**

Grundsätzlich werden beim Adaptiven Hypermedia die Techniken Adaptive Präsentation und Adaptive Navigation unterschieden. Die adaptive Präsentation

umfasst alle Techniken zur Anpassung der Darstellung des Contents einer Webseite. Adaptive Navigation umfasst alle Techniken zur Anpassung der Links auf einer Webseite [Brus01, S.87]. Klassifizierung derzeitiger Technologien des Adaptiven Hypermedias [Brus01, S.100]:

- Adaptive Präsentation
  - Adaptive Multimedia-Inhalte
  - Adaptiver Text
    - Anpassung an die natürliche Sprache
    - Adaptive Textblöcke
      - Fragmente einfügen/löschen
      - Ändern von Fragmenten
      - Text sortieren
      - Text ausdehnen
      - Text aufhellen/abdunkeln
  - Adaptive Modalität
- Adaptive Navigation
  - Direkte Lenkung des Benutzers mit Links
  - Adaptive Linksortierung
  - Adaptives Verstecken von Links
    - Verstecken von Links
    - Deaktivieren von Links
    - Entfernen von Links
  - Adaptive Kommentare zu Links
  - Adaptive Linkgenerierung
  - Adaptive Sitemaps

### 3.2.2 Architektur

Eine mögliche Architektur für ein Adaptives Hypermedia System ist AHAM (Adaptive Hypermedia Application Model). Kernelemente von AHAM sind das Domänenmodell, das Benutzermodell und das Adaptionsmodell (siehe Abbildung 7).

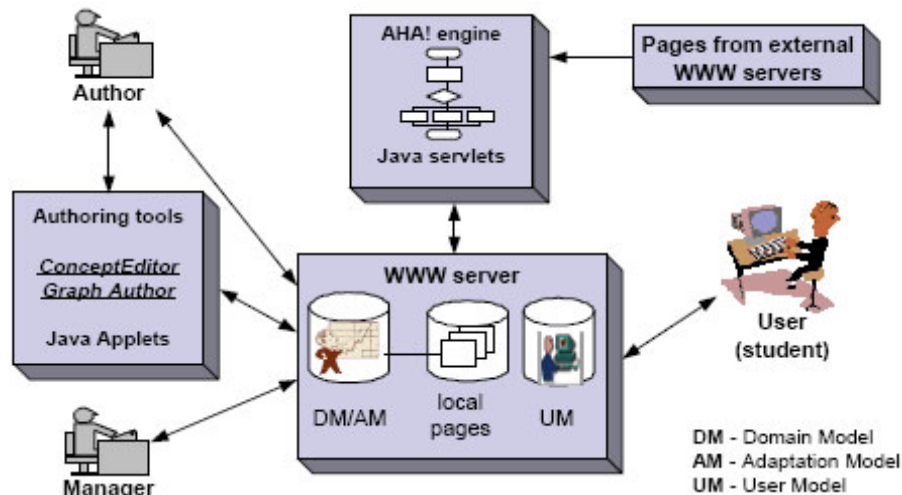


Abbildung 7: Architektur eines Adaptiven Hypermedia Systems [DeBr03]

Abbildung 7 zeigt eine mögliche Architektur eines Adaptiven Hypermedia Systems. Der Benutzer (Student) interagiert mit dem Webserver und fordert von diesem eine Webseite an. Der Webserver reagiert auf diese Anfrage und löst somit eine Reaktion des Adaptionsmodells (AM) aus. Im Adaptionsmodell sind Adaptionsregeln gespeichert, welche die angeforderte Webseite aufgrund des im Domänenmodell (DM) gesammelten Domänenwissens und aufgrund der gespeicherten Informationen im Benutzermodell, adaptiert. So werden zum Beispiel Links von der angeforderten Seite entfernt, falls im Benutzermodell vermerkt ist, dass der Benutzer die verlinkten Seiten bereits mehrfach besucht hat und das Domänenmodell zu diesem Verhalten vorschlägt, die verlinkten Seiten zu verstecken [DeBr03].

#### Domänenmodell

Das Domänenmodell besteht aus einer Vielzahl von Elementen, wobei jedes dieser Elemente ein spezielles Wissensfragment der Domäne darstellt. Diese Elemente werden im Folgenden Konzepte genannt und formen das Domänenmodell [Brus03]. Das Domänenmodell stellt das konzeptuelle Design eines Adaptiven Hypermediasystems dar. Zum Domänenmodell gehört weiters ein

Netzwerk von Webseiten. Es strukturiert also eine Menge von Konzepten, eine Menge von Webseiten und verknüpft diese beiden miteinander [KraGa06].

### **Benutzermodell**

Das Benutzermodell speichert die besonderen Eigenschaften eines Benutzers (z.B. Vorlieben, Wissen, Ziele,...) und anhand dieser wird ein individueller Kurs gestaltet. In diesem Bereich relevante Standards sind LTSC Public and Private Information [LTSC06] und IMS Learner Information Package [IMSG07]. LTSC Public and Private Information spezifiziert sowohl die Syntax, als auch die Semantik für die Erstellung eines Benutzermodells zur Charakterisierung des Wissens und der Vorlieben eines Benutzers. IMS Learner Information Package beschreibt Datenelemente zum Austausch von Informationen über Lernende [KrGa06].

### **Adaptionsmodell**

Das Adaptionsmodell beschreibt wie mit den Konzepten und den Materialien, im Kontext eines speziellen Benutzers, umgegangen wird. Es beinhaltet die Beschreibung der Prozesse, sowie die Adaptionsregeln [KrGa06]. Diese Adaptionsregeln dienen zur Anpassung von Webseiten und somit ist die Umsetzung von Adaptionsregeln ein essentieller Bestandteil eines funktionierenden Adaptiven Hypermedia Systems.



## 4 Gegenüberstellung der Ansätze

Um abschätzen zu können welcher der beiden im Kapitel 3 beschriebenen Ansätze zur Adaption der Lernprozesse für die Implementierung im eTutor-System verwendet werden soll, werden nun die Ansätze von SCORM® und Adaptives Hypermedia gegenübergestellt.

### 4.1.1 Gegenüberstellung

**Adaptives Hypermedia** unterstützt einen Lernenden bei der Navigation durch eine komplexe Vielfalt von Lernressourcen, wobei das Lernziel der Lernende selbst bestimmt. D.h. es werden keine Konzepte zur Erreichung eines konkreten Lernzieles bereitgestellt. Die Ablaufsteuerung wie sie zum Beispiel SCORM® vorstellt, verfolgt zwar den Lernprozess und dabei die Aktivitäten des Lernenden, jedoch nicht dessen Vorlieben und Eigenschaften wie es ein Adaptives Hypermedia-System ermöglichen würde. Die Konzepte von SCORM® berücksichtigen also nicht die Einbindung des Studentenwissens in die Ablaufsteuerung. Weiters gibt es bei der herkömmlichen Ablaufsteuerung kein Domänenmodell, welches die Konzepte der Domäne enthält. Vielmehr wird die Lernstrategie und das Lernziel durch einen Autor, in Form von Regeln der Ablaufsteuerung, festgelegt [AbDa03, S.172].

Es stellt sich nun die Frage, welcher Ansatz im eTutor umgesetzt werden soll und welche Auswirkungen dies hat. Adaptives Hypermedia beschreibt, wie eine ideale Architektur eines Lernsystems aussehen kann. Vergleicht man dieses System mit der bestehenden Architektur des eTutors, so müssten folgende Aspekte angepasst werden um eine Ablaufsteuerung zu ermöglichen.

- Erweiterung des Unterrichtsmoduls des eTutors um ein Adaptionmodell welches Regeln zur Adaption von Lerninhalten des eTutors enthält.
- Erweiterung des Unterrichtsmoduls um ein Domänenmodell welches abbildet, welche Adaptionsregeln aufgrund der gespeicherten Informationen im Studentenmodul angewendet werden müssen.
- Erweiterung des Studentenmoduls dahingehend, dass nicht nur der Wissensstand des Studenten, sondern auch Informationen über das

Verhalten des Studenten z.B. Klickverhalten während des Lernprozesses abgebildet werden.

Möchte man den eTutor in ein Adaptives Hypermedia System umwandeln um dadurch eine Ablaufsteuerung zu ermöglichen, so wäre der Änderungsaufwand sehr groß. Im eTutor macht es wenig Sinn einzelne Lerninhalte mit Hilfe von Adaptionsregeln anzupassen. Diese Lerninhalte stellen Übungen zu einer existierenden Lehrveranstaltung dar und werden jeweils von eigenen Modulen des eTutors dargestellt. Das bedeutet das Kommunikationsmodul abstrahiert die Darstellung von Lerninhalten dahingehen, dass das Unterrichtsmodul zwar weiß welche Lerninhalte dargestellt werden, jedoch nicht in welcher Form. Soll z.B. ein Lerninhalt adaptiert werden, so müssten die einzelnen Module die Adaptionsregeln, welche vom Unterrichtsmodul vorgeschlagen werden, anwenden können. Dies ist im eTutor praktisch nicht umsetzbar.

Sinnvoll wäre die Adaption der Benutzernavigation. Im eTutor werden dem Studenten eine Lehrveranstaltung und die dazugehörigen Übungen in Form eines Baumes präsentiert. Dieser Baum könnte mit Hilfe von Adaptionsregeln angepasst werden, um eine Ablaufsteuerung des Lernprozesses zu ermöglichen. Dies hätte weiter den Vorteil, dass die Adaptionsregeln nur einmal erstellt werden müssten um eine automatische Anpassung von Kursen an Studenten zu ermöglichen, somit wäre der Modellierungsaufwand für einzelne Kurse vernachlässigbar.

Ein weiter Nachteil von Adaptiven Hypermedia Systemen ist, dass es zwar aufzeigt, wie ein solches System idealerweise aussieht und welche Möglichkeiten es zur Adaption gibt, jedoch nicht, wie diese Möglichkeiten im konkreten Fall umgesetzt werden können. Trotz einer Vielzahl von Literatur welche sich mit dem Thema Adaptives Hypermedia beschäftigt, fehlt es noch an Standards zur Umsetzung solcher Systeme [AbDa03].

### **Kriterien für Adaptive Hypermedia Systeme**

- Großer **Änderungsaufwand** für den eTutor
- Keine Standards für eine **Umsetzung**
- Kein Konzepte zur Erreichung von **Lernzielen**
- Ablaufsteuerung auf einer **Mikro-Ebene** ist möglich
- **Studentenwissen** kann verwendet werden

- Der **Modellierungsaufwand** für einzelne individuelle Kurse ist gering

**SCORM® 2004 3<sup>rd</sup> Edition** ist eine Sammlung von Standards und Spezifikationen welche detailliert darauf eingehen, wie ein idealtypisches Lernsystem aussehen könnte, und wie in einem solchen eine Ablaufsteuerung zu realisieren ist. Betrachtet man das idealtypische Lernsystem von SCORM® (siehe Abschnitt 3.1) und vergleicht es mit der bestehenden Architektur des eTutors, so müssten folgende Aspekte angepasst werden um eine Ablaufsteuerung im eTutor zu ermöglichen.

- Erweiterung des Unterrichtsmoduls des eTutors um einen Sequencing Service welcher Regeln zur Ablaufsteuerung enthält.
- Erweiterung des Kommunikationsmoduls um einen Tracking Service welcher das Verhalten des Studenten während des Lernprozesses verfolgt und abspeichert.

Für die Umsetzung der Ablaufsteuerung nach SCORM® müssen weniger Module angepasst werden, als es bei der Umsetzung eines Adaptiven Hypermedia Systems der Fall wäre. SCORM® betrachtet in der Ablaufsteuerung zwar das Verhalten eines Studenten, sieht jedoch nicht vor, dass dessen Wissensstand in Form eines Benutzermodells abgebildet und verwendet wird. Es ist außerdem nicht vorgesehen, dass Lerninhalte adaptiert und an den Studenten angepasst werden. Lediglich die Navigation durch die Lerninhalte wird adaptiert und individuell angepasst. Weiters werden die Regeln zur Ablaufsteuerung, z.B. zur Adaption der Navigation, nicht in Form eines Domänenmodells abgebildet, sondern müssen von einem Autor für jede Lehrveranstaltung an die Erfordernisse der jeweiligen LVA angepasst werden.

#### **Kriterien für die Ablaufsteuerung nach SCORM®**

- Geringerer **Änderungsaufwand** für den eTutor, da nur zwei Module erweitert werden müssen
- SCORM® enthält konkrete Vorschläge für die **Umsetzung**
- **Lernziele** können mit SCORM® festgelegt werden
- Ablaufsteuerung auf einer **Mikro-Ebene** ist möglich
- Keine Einbindung von **Studentenwissen**

- Großer **Modellierungsaufwand**, da jeder Kurs einzeln modelliert werden muss

#### 4.1.2 Fazit

Vergleicht man nun den Ansatz von SCORM® und die Ansätze des Adaptiven Hypermedia zur Ablaufsteuerung eines Lernprozesses, so erscheint eine Kombination der beiden Ansätze für den eTutor sinnvoll. Einerseits kann der eTutor das Wissen des Studenten in Form eines Benutzermodells abbilden und dieses Wissen sollte auch in die Ablaufsteuerung einfließen, andererseits liefert SCORM® detaillierte Auskunft darüber, wie eine Ablaufsteuerung zu implementieren wäre.

Tabelle 2 zeigt die Vor- und Nachteile der Ansätze zur Umsetzung einer Ablaufsteuerung im eTutor und anhand dieser wurde folgende Entscheidung getroffen. Da SCORM® konkret beschreibt, wie eine Ablaufsteuerung umzusetzen ist und welche Konzepte dafür notwendig sind und diese außerdem anhand von bereits bestehenden Standards gewählt wurden, wird im eTutor eine Ablaufsteuerung basierend auf SCORM® umgesetzt. Weiters wird die Ablaufsteuerung im eTutor dahingehend erweitert, dass sie auch auf das Wissen des Studenten, welches in Form eines Benutzermodells im eTutor abgespeichert ist, zugreifen kann. Dieses Wissen fließt in die Ablaufsteuerung ein und kann bei der Modellierung einer Ablaufsteuerung für einen konkreten Kurs eingesetzt werden.

Kriterien	SCORM®	Adaptives Hypermedia
<b>Änderungsaufwand</b>	<b>niedrig</b>	<b>hoch</b>
<b>Konkrete Umsetzung</b>	✓	✗
<b>Lernziele</b>	✓	✗
<b>Mikro-Ebene</b>	✓	✓
<b>Studentenwissen</b>	✗	✓
<b>Modellierungsaufwand</b>	<b>hoch</b>	<b>niedrig</b>

Tabelle 2: Vergleich der Ansätze

Die Modellierungsmöglichkeiten einer individualisierten Ablaufsteuerung, sowie konkrete Konzepte der Ablaufsteuerung werden im Kapitel 5 und 6 detailliert beschrieben. Es wird darauf eingegangen, welche Konzepte zur Ablaufsteuerung dienen und wie diese wirken.

## **Teil 3**

### **Konzeption einer individualisierten Ablaufsteuerung**

## 5 Scope einer individualisierten Ablaufsteuerung

Abbildung 8 zeigt den von IMS definierten Scope der Ablaufsteuerung IMS Simple Sequencing, welche auch die Basis für die Ablaufsteuerung von SCORM® ist. Es wird nun darauf eingegangen, welche Möglichkeiten zur Gestaltung des Lernprozesses die Ablaufsteuerung im eTutor bereitstellen sollte.

### (1) Gelenktes Lernen

Man spricht von gelenktem Lernen, wenn das Lernsystem den Lernprozess des Studenten steuert.

- **Verzweigung**

Betrachtet man den Lernprozess als Pfad durch die Lerninhalte, so stellt die Verzweigung einen Punkt dar, an dem sich dieser Pfad in zwei oder mehrere neue Pfade aufteilt. Entweder der Student wählt explizit einen der neuen Pfade aus, oder das System entscheidet sich am Punkt der Verzweigung selbst für einen neuen Pfad. Dieses Konzept ermöglicht z.B. das Überspringen von Aufgaben aufgrund des Vorwissens des Studenten.

- **Schleife**

Eine Schleife ermöglicht, dass eine Menge von bereits vom Studenten bearbeiteten Lerninhalten, nochmals zur Bearbeitung präsentiert wird. Dies ist insbesondere dann sinnvoll, wenn die Lernziele zu einem bestehenden Stoffgebiet noch nicht erreicht wurden.

- **Linearer Lernprozess**

Es existiert immer eine bestimmte Reihenfolge, in der Lerninhalte abgearbeitet werden. Entweder diese Reihenfolge ist vom Studenten gewählt oder vom System vorgeben. In einem idealtypischen Lernsystem erfolgt die Abarbeitung der Lerninhalte immer sequentiell. Eine parallele Bearbeitung von Lerninhalten ist nicht möglich. Spricht man von einem linearen Lernprozess, so bedeutet dies, dass ein Autor eines Kurses einen Lernprozess modellieren kann, welcher die Reihenfolge der Abarbeitung von Lerninhalten festlegt.

- **Zufälliger Lernprozess**

Es soll möglich sein, den Lernprozess zufällig zu gestalten. Dies wirkt sich auf zwei Aspekte des Lernprozesses aus. Einerseits soll bei einer vom Autor festgelegten Reihenfolge der Abarbeitung von Lerninhalten diese Reihenfolge zufällig gestaltet werden können. Andererseits soll es möglich sein, dem Studenten bei der Auswahl einer Lernaktivität einen zufälligen Lerninhalt zu präsentieren.

## **(2) Selbst gestaltetes Lernen**

Der Student soll in einem modellierten Lernprozess vom Lernsystem die Möglichkeit bekommen, seinen Lernprozess selbst mit zu gestalten (siehe Abbildung 8).

- **Auswahl**

Spricht man von einer Auswahl, so bedeutet dies, dass der Student selbst wählen kann, welche Lerninhalte er abarbeiten will.

- **Eingeschränkte Auswahl**

Hat der Student eine Menge von Lerninhalten zur Auswahl, so soll es möglich sein anhand von Kriterien, wie z.B. dem Vorwissen des Studenten, diese Menge an wählbaren Lerninhalten einzuschränken.

## **(3) Adaption**

Adaption bedeutet die Anpassung von Lerninhalten und der Navigation an die Bedürfnisse und Vorlieben des Studenten.

- **Adaption der Navigation (eingeschränkt)**

Anhand einer Adaption der Navigation ist es möglich die Abfolge der Lerneinheiten in einem Lernprozess zu gestalten.

Ein Lernsystem soll über die Möglichkeit verfügen, die Navigation durch die Lerninhalte dahingehend anzupassen, dass der Student nur jene Lerninhalte sehen kann, welche ihm zurzeit zur Auswahl stehen.

- **Vollständige Adaption**

Spricht man von einer vollständigen Adaption, so kann ein Lernsystem nicht nur die Navigation, sondern auch die Lerninhalte selbst an die Bedürfnisse eines Studenten anpassen.



- **Intelligente Adaption**

Eine intelligente Adaption kann anhand von Informationen über Studenten und dem Domänenwissen automatisch entscheiden wie eine Navigation und Lerninhalte anzupassen sind, um einen individuellen Kurs zu gestalten. Die Ansätze des Adaptiven Hypermedia beschreiben die Möglichkeiten einer intelligenten Adaption.

#### **(4) Gemeinsames Lernen**

Man spricht von gemeinsamem Lernen, wenn mehrere Studenten gemeinsam mit Hilfe eines Lernsystems lernen. Jeder Lernende nimmt Einfluss auf den Lernprozess des anderen.

- **Instruktoren**

Unterstützt ein Lernsystem Instruktoren, so bedeutet dies, dass ein Instruktor den Lernprozess des Studenten im Lernsystem mit verfolgen kann und während des Lernprozesses aktiv darauf Einfluss nimmt.

- **Face to Face**

Beim Face to Face Lernen unterstützt das Lernsystem, dass zwei Lernende von und miteinander lernen.

- **Kollegen**

Das Lernen von und mit Kollegen ist eine Erweiterung des Face to Face Lernens. Hier nimmt eine Menge von Studenten am Lernprozess teil und beeinflusst sich gegenseitig.

Der in Abbildung 8 grau hinterlegt Bereich zeigt, welche Möglichkeiten der Ablaufsteuerung mit Hilfe der von SCORM® beschriebenen Konzepte im eTutor umgesetzt werden können. Das folgende Kapitel 6 zeigt nun wie diese Konzepte bei der Umsetzung einer individualisierten Ablaufsteuerung angewendet werden.

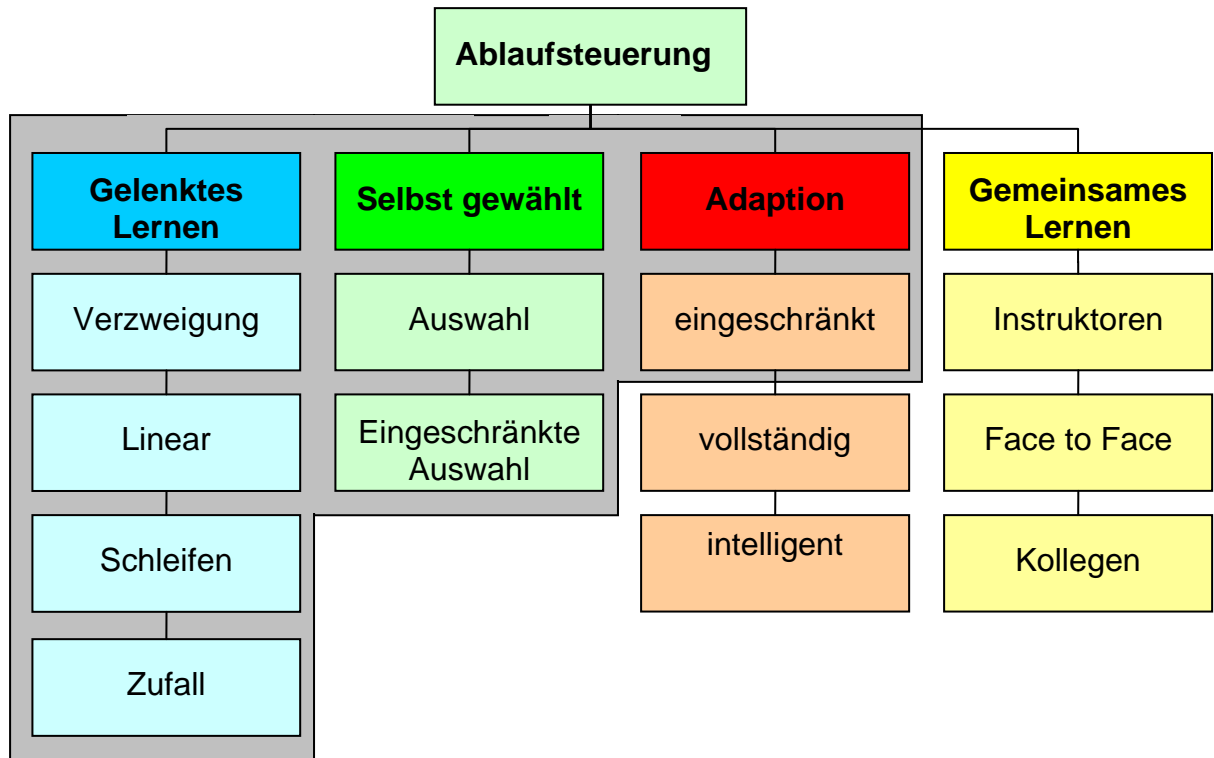


Abbildung 8: Scope der Ablaufsteuerung

## **6 Konzepte zur Umsetzung einer individualisierten Ablaufsteuerung**

Die Ablaufsteuerung legt die Abfolge der Lerneinheiten während des Lernprozesses fest. Ziel ist es, die Lernenden durch eine Sequenz von Lerneinheiten durch den Lernprozess zu führen um ein Lernziel zu erreichen.

Wie im Kapitel 3 aufgezeigt, existieren mehrere Ansätze, wie eine Ablaufsteuerung realisiert werden kann. Die nun vorgestellten Konzepte basieren auf SCORM® und werden gegebenenfalls durch Ansätze des Adaptiven Hypermedia erweitert.

Eine Lerneinheitssequenz wird vor dem Beginn des Lernprozesses modelliert (z.B. vom Autor des Kurses, von einem Assistenten). Die Ablaufsteuerung stellt einen Teil des internen Verhaltens des webbasierten Lernsystems dar und bietet Methoden und Konzepte zur Modellierung der Lerneinheitssequenz. Durch die Konstrukte der Ablaufsteuerung, z.B. dem Auswahl-Konstrukt, legt der Autor die relative Reihenfolge der Lerneinheiten fest. Außerdem bestimmen sie welche Lerneinheit ausgewählt und an den Benutzer geliefert, oder ob diese in einer modellierten sequentiellen Abfolge übersprungen wird. Zur Ablaufsteuerung gehören weiters Konzepte zur Überwachung und Verarbeitung von systeminitiierten oder benutzerinitiierten Navigationsanfragen.

Die folgenden vorgestellten Konzepte kennen nur die Rolle des Lerners im Lernprozess. Also nicht die eines Mentors, eines Instructors usw. Die Ablaufsteuerung kümmert sich nicht um den wirklichen Aufruf einer Lerneinheit und der Kommunikation der Lerneinheiten mit dem System. Hier gibt es z.B. die in SCORM® spezifizierte Laufzeitumgebung, welche diesen Teil umfasst. Weiters kümmert sich die Ablaufsteuerung nicht um die Präsentation der Inhalte, also um das Benutzerinterface. Die Darstellung und Verarbeitung unterschiedlicher Medien, z.B. von Videos, werden von den vorgestellten Konzepten nicht berücksichtigt, da diese bereits von bekannten Web-Techniken bereitgestellt wird.

### **6.1 Basiselemente**

Bevor die konkreten Konzepte zur Ablaufsteuerung beschrieben werden können, sind noch allgemeine Begriffe wie

- Lernaktivität,
- Aktivitätenbaum und
- Cluster

zu erläutern.

**Lernaktivitäten** und deren Bedeutung für die Ablaufsteuerung, sowie die Gestaltung des Lernprozesses durch eine Strukturierung der Lernaktivitäten sind Bestandteil dieses Abschnitts.

Eine Lernaktivität ist all jenes, das der Lernende während des Lernprozesses tut [ADLd06]. Eine Lernaktivität beinhaltet entweder einen Lerninhalt, welcher an den Lernenden geliefert wird, oder andere Lernaktivitäten (Abbildung 9). Enthält eine Lernaktivität A eine andere Lernaktivität B, so bedeutet dies, dass die enthaltene Lernaktivität B ein Teil jenes Wissensgebietes ist, der durch die Lernaktivität A beschrieben wird. Wie in der Abbildung 9 ersichtlich, ist der Einstufungstest ein Teil des Kurses. Mit Hilfe von Lernaktivitäten wird also ein Themengebiet für den Lernprozess strukturiert.

Bearbeitet ein Lernender einen Lerninhalt, so geschieht dies immer im Kontext einer Lernaktivität. Der Lernende interagiert ausschließlich mit jenen Lerninhalten, welche ihm durch eine Lernaktivität zugeordnet wurden. Das bedeutet, dem Studenten werden ausschließlich Lernaktivitäten zugeordnet. An eine Lernaktivität wurde vom System ein fixer Lerninhalt gebunden und dieser wird ihm im Benutzerinterface präsentiert. Die Begriffe Lernaktivität und Aktivität werden im Folgenden als synonym verwendet. Lernaktivitäten haben folgende Eigenschaften [ADLd06, S.28]:

- Eine Lernaktivität hat einen diskreten Start und ein diskretes Ende.
- Eine Lernaktivität hat wohl definierte Bedingungen, wann sie abgeschlossen und gemeistert wurde.
- Eine Lernaktivität kann andere Lernaktivitäten, bis zu einer unbegrenzten Tiefe, enthalten.
- Versuche einer Lernaktivität geschehen im Kontext der Elternaktivitäten.

Der Begriff **Versuch** ist definiert, als die Bemühung eines Lernenden eine Lernaktivität abzuschließen.



Abbildung 9: Beispiel einer Lernaktivität [ADLc06]

Die Strukturierung des Lernprozesses kann mit Hilfe von Bäumen erfolgen. Ein Baum, welcher aus Lernaktivitäten besteht wird **Aktivitätenbaum** genannt und zeigt eine hierarchische Struktur des Lernprozesses. Abbildung 10 zeigt ein Beispiel eines möglichen Aktivitätenbaumes für den Kurs Datenmodellierung mit den Übungen 1-6. Die Bildung eines solchen Aktivitätenbaumes kann, aber muss nicht, anhand der Inhaltsorganisation der Lerninhalte erfolgen. Im Falle des eTutors beschreibt die Inhaltsorganisation bereits im eTutor gespeicherte Kurse. Ein Aktivitätenbaum kann im eTutor nicht anhand einer bereits bestehenden Inhaltsorganisation erstellt werden. Eine Lernaktivität, welche andere Lernaktivitäten enthält, bildet mit diesen einen **Cluster**. Ein Cluster besteht aus einer Eltern-Aktivität und den direkt untergeordneten Kind-Aktivitäten des Baumes ohne wiederum deren Kind-Aktivitäten. Die Kind-Aktivitäten eines Clusters sind entweder Eltern-Aktivitäten eines neuen Clusters oder Blatt-Aktivitäten. Nur die Blätter des Baumes (**Blatt-Aktivitäten**), also Lernaktivitäten, welche keine anderen Lernaktivitäten beinhalten, können Lerninhalte beinhalten [ADLd06, S.26].

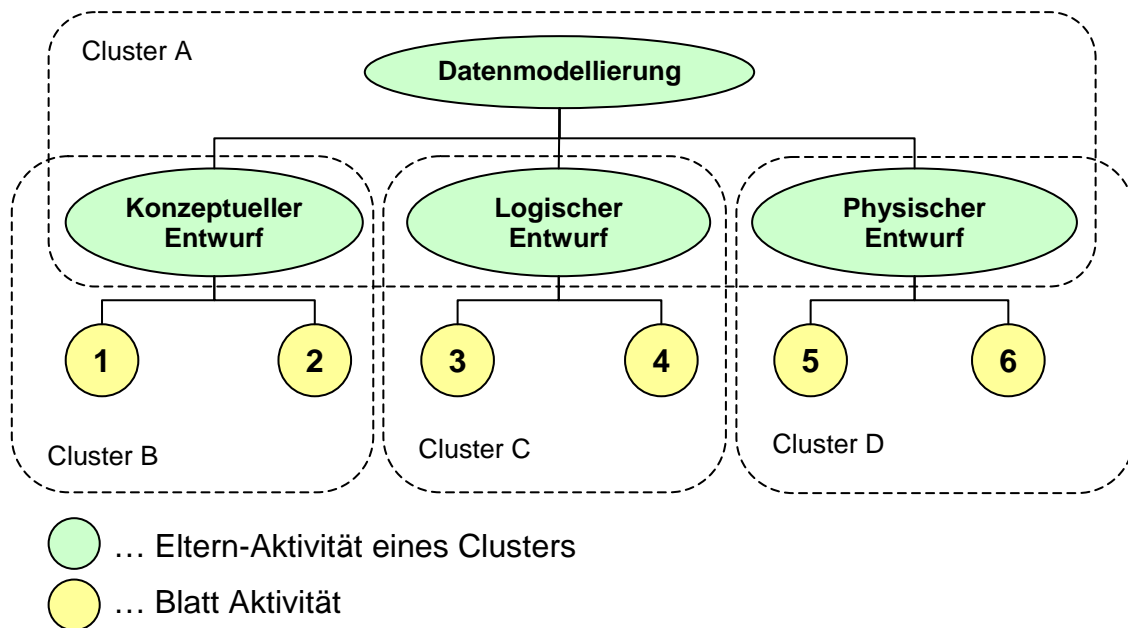


Abbildung 10: Beispiel eines Aktivitätsbaumes

## 6.2 Modellierungselemente

Die Modellierungselemente der Ablaufsteuerung werden im Folgenden als Konstrukte bezeichnet. Diese Konstrukte ermöglichen es einem Autor den Lernprozess eines Lernenden zu gestalten. Zu jedem Konstrukt wird dessen Funktion und etwaige Optionen beschrieben.

### (1) Auswahl-Konstrukt

Das Auswahl-Konstrukt ermöglicht es, eine Verzweigung im Ablauf des Lernprozesses zu modellieren. Dieses Konstrukt kann nur Eltern-Aktivitäten eines Clusters zugeordnet werden. Alle Aktivitäten deren Eltern-Aktivität das Auswahl-Konstrukt zugeordnet wurde, können vom Lernenden bei einer Auswahl-Anfrage (siehe Abschnitt 7.4) ausgewählt werden [ADLd06, S.37]. Eine Aktivität welche von einem Studenten versucht wird, wird als aktive Aktivität bezeichnet. Hat eine Aktivität keine Kind-Aktivitäten die ebenfalls aktiv sind, so ist diese Aktivität die aktuelle Aktivität des Studenten.

### Einschränkungen

Die Menge der auswählbaren Aktivitäten kann einerseits mit Hilfe von Regeln eingeschränkt werden, aber andererseits auch durch Optionen des Auswahl-Konstruktes. Diese Optionen sind:

- **Kein Exit**

Wird diese Option gesetzt, so bedeutet dies, dass nur Nachfolger der gerade aktiven Aktivität zur Auswahl stehen. Ist diese Option nicht gesetzt, so kann eine Auswahl-Anfrage dazu führen, dass die gerade aktuelle Aktivität beendet wird und eine andere Aktivität zur aktuellen Aktivität wird. Ist diese Option nicht gesetzt, so können bei einer Auswahl-Anfrage auch Aktivitäten ausgewählt werden, welche keine Nachfolger der gerade aktuellen Aktivität sind.

- **Keine Aktivierung**

Wird diese Option gesetzt, so muss eine Aktivität bereits aktiv sein, damit ihre direkten Kind-Aktivitäten zur Auswahl stehen. Eine Auswahl-Anfrage kann diese Aktivität nicht aktivieren, falls diese Option gesetzt ist. Diese Option beschränkt die zur Auswahl stehenden Aktivitäten auf die direkten Kind-Aktivitäten und verhindert somit, dass der Lernende zu tief in den Aktivitätenbaum springt ohne zuerst vor geschaltete Aktivitäten erreicht zu haben.

- **Auswahl innerhalb der Lernsequenz**

Wird diese Option gesetzt, so stehen die direkten Kind-Aktivitäten der Aktivität, zu welcher das Auswahl-Konstrukt hinzugefügt wurde, nur dann bei einer Auswahl-Anfrage zur Auswahl, wenn sie sich logisch im Fluss der Lernsequenz vor oder nach der aktuellen Aktivität befinden.

### Beispiel 1: Auswahl-Konstrukt ohne zusätzliche Optionen

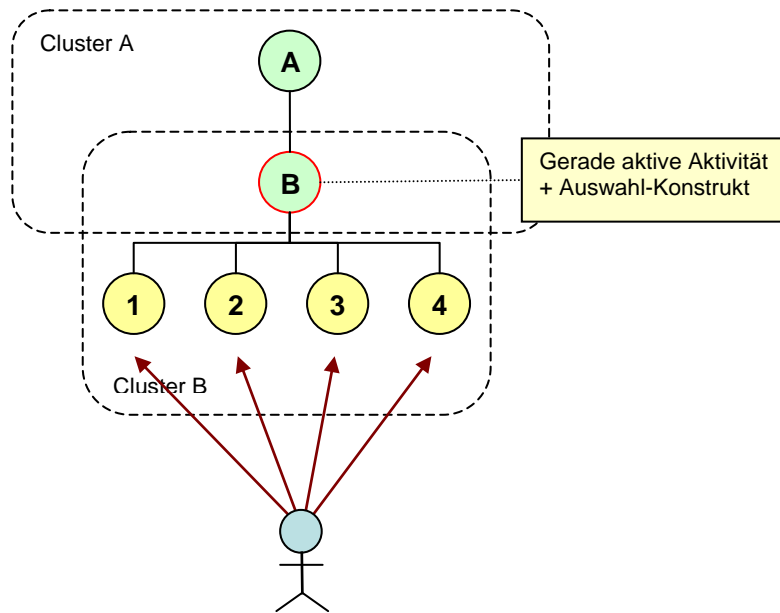
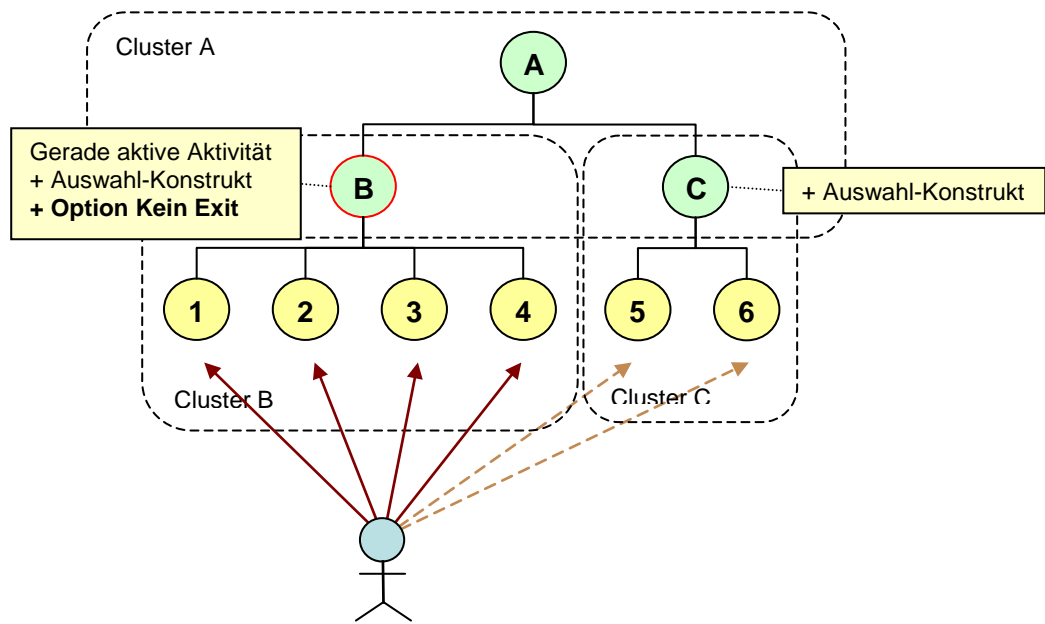


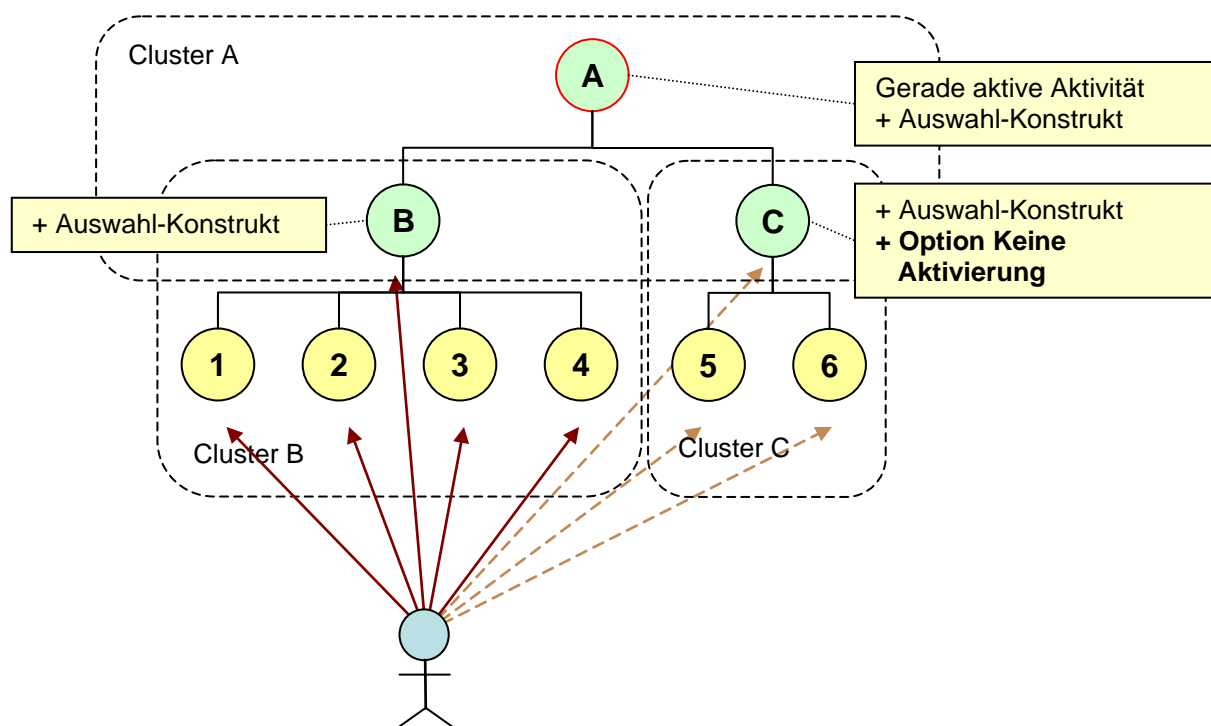
Abbildung 11: Auswahl-Konstrukt ohne Optionen

Der Eltern-Aktivität B des Clusters B ist das Auswahl-Konstrukt zugeordnet (Abbildung 11). Bei einer Auswahl-Navigationsabfrage steht die Menge der Aktivitäten {1, 2, 3, 4} dem Lernenden zur Auswahl.



**Beispiel 2: Auswahl-Konstrukt und die Option *Kein Exit*****Abbildung 12: Auswahl-Konstrukt *Kein Exit***

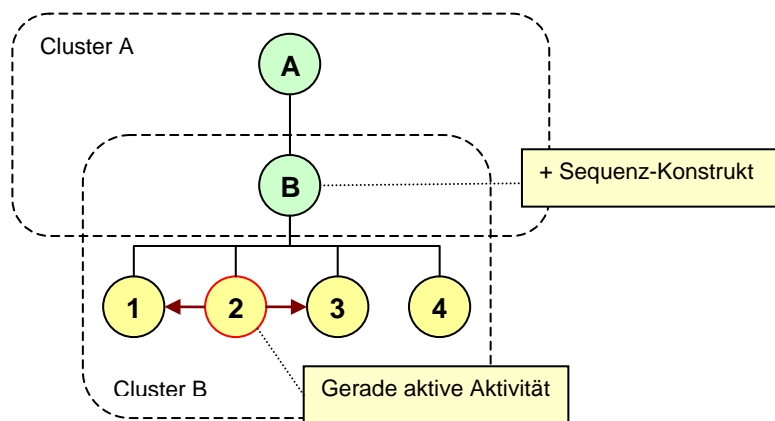
Der Eltern-Aktivität B des Clusters B ist das Auswahl-Konstrukt mit der Option *Kein Exit* zugeordnet (Abbildung 12). Der Eltern-Aktivität C des Clusters C ist das Auswahl-Konstrukt zugeordnet. Bei einer Auswahl-Navigationsabfrage steht die Menge der Aktivitäten {1, 2, 3, 4} zur Auswahl. Würde die Option *Kein Exit* in diesem Beispiel nicht verwendet werden, so würde die Menge der Aktivitäten {1, 2, 3, 4, 5, 6} zur Auswahl stehen. Die Option *Kein Exit* beschränkt also die Menge der auswählbaren Aktivitäten.

**Beispiel 3: Auswahl-Konstrukt und die Option *Keine Aktivierung*****Abbildung 13: Auswahl-Konstrukt *Keine Aktivierung***

Die Eltern-Aktivität A des Clusters A ist die gerade aktive Aktivität, welcher weiters das Auswahl-Konstrukt zugeordnet ist (Abbildung 13). Den Eltern-Aktivitäten B und C der Cluster B und C ist ebenfalls das Auswahl-Konstrukt zugeordnet. Der Eltern-Aktivität C wurde die Option *Keine Aktivierung* zugeordnet. Bei einer Auswahl-Navigationsabfrage steht die Menge der Aktivitäten {B, 1, 2, 3, 4} zur Auswahl. Die Eltern-Aktivität C und alle ihre Nachfolger können nicht ausgewählt werden, da dies zu einer Aktivierung der Eltern-Aktivität C führen würde. Im Gegensatz zur Option *Kein Exit* wird hier der Aktivitätenbaum nicht in seiner Breite, sondern in seiner Tiefe beschränkt, was wiederum zu einer kleineren Menge an Aktivitäten führt, welche dem Lernenden bei einer Auswahl-Anfrage zur Verfügung stehen.

**(2) Sequenz-Konstrukt**

Das Sequenz-Konstrukt ermöglicht es, einen sequenziellen Ablauf des Lernprozesses zu modellieren. Dieses Konstrukt kann nur Eltern-Aktivitäten eines Clusters zugeordnet werden. Wird dieses Konstrukt verwendet, so muss das webbasierte Lernsystem die Anfragen „nächste Aktivität“ und „vorige Aktivität“ bereitstellen und dem Lernenden die in der Lernsequenz folgende Aktivität liefern [ADLd06, S.40].

**Beispiel: Sequenz-Konstrukt****Abbildung 14: Sequenz-Konstrukt**

Der Eltern-Aktivität B des Clusters B ist das Sequenz-Konstrukt zugeordnet (Abbildung 14). Die Aktivität 2 ist die gerade aktuelle Aktivität. Erhält das Lernsystem eine Anfrage „nächste Aktivität“, so wird dem Lernenden die Aktivität 3 geliefert. Erhält das Lernsystem eine Anfrage „vorige Aktivität“, so wird dem Lernenden die Aktivität 1 geliefert.

**(3) Vorwärts-Sequenz-Konstrukt**

Das Vorwärts-Sequenz-Konstrukt bewirkt, dass der sequenzielle Ablauf des Lernprozesses für einen bestimmten Cluster nie rückwärts sondern nur vorwärts durchlaufen werden kann. Dieses Konstrukt kann nur Eltern-Aktivitäten eines Clusters zugeordnet werden. Wird dieses Konstrukt verwendet, so muss das webbasierte Lernsystem die Anfrage „vorige Aktivität“ für die Aktivitäten des Clusters ignorieren. Wird zum Beispiel ein gesamter Cluster mit dem Vorwärts-Sequenz-Konstrukt aufgrund einer Anfrage „vorige Aktivität“ ausgewählt, so wird dem Lernenden die erste Aktivität des Clusters geliefert. Ohne Verwendung des Vorwärts-Sequenz-Konstruktes würde der Lernende die letzte Aktivität des Clusters geliefert bekommen [ADLd06, S.41].

**(4) Zufalls-Konstrukt**

Das Zufalls-Konstrukt ermöglicht die Herstellung einer Zufallsordnung der Aktivitäten eines Clusters. Dieses Konstrukt kann nur Eltern-Aktivitäten eines Clusters zugeordnet werden. Durch eine Option kann festgelegt werden, ob die Neuordnung nur vor dem ersten Versuch eines Clusters hergestellt werden soll,

oder ob dies vor jedem Versuch des Clusters, durch einen Lernenden, erfolgen soll [ADLd06, S.79].

### **(5) Vorauswahl-Konstrukt**

Das Vorauswahl-Konstrukt ermöglicht, es eine Menge von Aktivitäten eines Clusters zu selektieren. Dieses Konstrukt kann nur Eltern-Aktivitäten eines Clusters zugeordnet werden. Es besitzt die Optionen

- Zeitpunkt der Vorauswahl:  
gibt an, ob die Vorauswahl nur vor dem ersten Versuch oder vor jedem Versuch des Clusters durchgeführt wird.
- Anzahl der zu selektierenden Aktivitäten:  
gibt an, wie viele Aktivitäten des Clusters selektiert werden sollen.

Mit Hilfe dieses Konstruktes kann zum Beispiel ein Lernprozess modelliert werden bei welchem beim ersten Versuch einer Eltern-Aktivität eines Clusters nur 5 von 10 Kind-Aktivitäten ausgewählt und dem Lernenden angezeigt werden [ADLd06, S.77].

### **(6) Versuchs-Konstrukt**

Das Versuchs-Konstrukt schränkt die Menge der vom Lernenden wählbaren Aktivitäten im Aktivitätenbaum ein, indem die Anzahl der Versuche, welche ein Lernender bei einer Aktivität hat, eingeschränkt wird. Folgende zwei Möglichkeiten gibt es, die Anzahl der Versuche zu beschränken:

- Anzahl der maximalen Versuche festlegen. Der Lernende darf die Aktivität nicht öfter als gewünscht versuchen.
- Zeitdauer der Versuche festlegen. Wird eine Zeitdauer festgelegt, beschränkt diese den zeitlichen Aufwand, den ein Lernender für seine Versuche an einer Aktivität aufwenden kann. Die Anzahl der Versuche, die der Lernende benötigt, wird hier nicht beschränkt. Um eine zeitliche Einschränkung zu ermöglichen, muss ein webbasiertes Lernsystem die Zeitdauer vom Beginn eines Versuches bis zum Beenden eines Versuches eines Lernenden messen.

## (7) Überwachungs-Konstrukt

Das Überwachungs-Konstrukt ermöglicht es festzulegen, ob der Status einer Aktivität vom webbasierten Lernsystem überwacht werden soll, oder nicht. Dieses Konstrukt kann nur Blatt-Aktivitäten zugeordnet werden. Folgende Optionen sind möglich:

- **Keine Überwachung**

Wird diese Option verwendet, so überwacht das webbasierte Lernsystem weder die Versuche der Aktivität, noch die Zielerreichung, noch ob eine Aktivität abgeschlossen ist oder nicht.

- **Kein Abschließen**

Wird diese Option verwendet, so geht das webbasierte Lernsystem davon aus, dass die Lernressource die Aktivität beendet. Wird kein Überwachungs-Konstrukt mit dieser Option verwendet, so schließt das webbasierte Lernsystem eine Aktivität dann ab, wenn ein Versuch des Lernenden an der Aktivität endet.

- **Keine Zielerreichung**

Wird diese Option verwendet, so bestimmt ein Lerninhalt selbst, ob das zur Aktivität gehörende Lernziel erreicht wurde, oder nicht.

Der folgende Abschnitt 6.3 geht näher auf die regelbasierte Ablaufsteuerung ein. Diese dient vor allem dazu den Aktivitätenbaum während des Lernprozesses zu individualisieren.

### 6.3 Regelbasierte Ablaufsteuerung

Um den Lernprozess an individuelle Bedürfnisse des Lernenden anpassen zu können, werden die Aktivitäten des Aktivitätenbaumes um Regeln erweitert. Zu jeder Aktivität können beliebig viele Regeln der Ablaufsteuerung zugeordnet werden. Jede Regel besteht aus einer Menge an Bedingungen und einer Aktion. Die regelbasierte Ablaufsteuerung, wie in diesem Abschnitt beschrieben, basiert auf den in den Spezifikationen von ADL SCORM® [ADLd06, S. 48ff] und IMS Simple Sequencing [SSBP03] vorgestellten Konzepten.

Eine Regel hat nachfolgende Struktur:

**Regel der Ablaufsteuerung:** If [Bedingung] then [Aktion]

Die Bedingungen und Aktionen einer Regel werden nachfolgend näher erläutert.

### 6.3.1 Bedingungen

Einzelne Bedingungen können zu einer Menge von Bedingungen zusammengefasst werden. Für diese Menge kann man wiederum angeben, wie die einzelnen Bedingungen logisch verknüpft werden sollen. Nur wenn die Auswertung der Bedingungen „wahr“ zurück liefert, wird die Aktion der Regel der Ablaufsteuerung ausgeführt.

- **Verknüpfung mit All**  
Die Menge der Bedingungen liefert „wahr“ zurück, wenn alle in der Menge enthaltenen Bedingungen „wahr“ zurück liefern. Dies entspricht einer logischen UND Verknüpfung.
- **Verknüpfung mit Any**  
Die Menge der Bedingungen liefert „wahr“ zurück, wenn irgendeine der in der Menge enthaltenen Bedingungen „wahr“ zurück liefert. Dies entspricht einer logischen ODER Verknüpfung.

Weiters können Bedingungen mit dem unären logischen Operator NICHT negiert werden. Dieser Operator wird nach der Evaluierung einer Bedingung ausgeführt.

Im Folgenden werden die möglichen Bedingungen, welche in der Ablaufsteuerung verwendet werden können, aufgelistet. SCORM® und IMS Simple Sequencing erlauben nur die nun angeführten Bedingungen. Einige Bedingungen überprüfen die Zielerreichung einer Aktivität. Dies ist nur dann möglich, wenn der Aktivität ein Lernziel zugeordnet wurde.

#### Bedingungen mit Lernzielen

- **Lernziel erreicht**  
Diese Bedingung prüft, ob das der Aktivität zugeordnete Lernziel erreicht wurde und liefert gegebenenfalls „wahr“.
- **Lernziels bekannt**  
Diese Bedingung prüft, ob einer Aktivität ein Lernziel zugeordnet ist und liefert gegebenenfalls „wahr“.

- **Ausmaß der Zielerreichung bekannt**  
Diese Bedingung prüft, ob das Lernziel ein numerischer Wert ist, der das Ausmaß der Zielerreichung angibt und liefert gegebenenfalls „wahr“.
- **Ausmaß der Zielerreichung größer als**  
Diese Bedingung prüft, ob das Ausmaß der Zielerreichung größer als ein bestimmter Wert ist und liefert gegebenenfalls „wahr“. Diese Bedingung ist nur dann zulässig, wenn die Bedingung *Ausmaß der Zielerreichung bekannt* „wahr“ ist.
- **Ausmaß der Zielerreichung kleiner als**  
Diese Bedingung prüft, ob das Ausmaß der Zielerreichung kleiner als ein bestimmter Wert ist und liefert gegebenenfalls „wahr“. Diese Bedingung ist nur dann zulässig, wenn die Bedingung *Ausmaß der Zielerreichung bekannt* „wahr“ ist.

#### **Bedingungen ohne Lernziele**

- **Abgeschlossen**  
Diese Bedingung prüft, ob eine Aktivität abgeschlossen wurde und liefert gegebenenfalls „wahr“.
- **Status der Aktivität bekannt**  
Diese Bedingung prüft, ob eine Aktivität überwacht wird. Hierbei wird geprüft, ob die Versuche des Lernenden an der Aktivität überwacht werden und ob das System überwacht, wann eine Aktivität abgeschlossen ist. Ist der Status einer Aktivität bekannt, so liefert diese Bedingung „wahr“.
- **Versucht**  
Diese Bedingung prüft, ob eine Aktivität bereits vom Lernenden versucht wurde und liefert gegebenenfalls „wahr“.
- **Anzahl der Versuche erreicht**  
Diese Bedingung prüft, ob die Anzahl der möglichen Versuche einer Aktivität von einem Lernenden bereits erreicht wurde und liefert gegebenenfalls „wahr“. Diese Bedingung ist nur in Kombination mit einem Versuchs-Konstrukt zulässig, welches die Anzahl der Versuche einer Aktivität einschränkt.

- **Immer**  
Diese Bedingung liefert immer „wahr“.

### 6.3.2 Aktionen

Ist die Menge von Bedingungen einer Regel „wahr“, so wird die zur Regel gehörende Aktion ausgeführt. Aktionen werden aufgrund ihres Ausführungszeitpunktes gegliedert. Man unterscheidet:

#### **Vor-Auswahl-Aktionen**

Vor-Auswahl-Aktionen sind Aktionen welche ausgeführt werden, wenn das System eine neue Aktivität im Aktivitätenbaum für den Lernenden identifiziert.

- **Auslassen**  
Wurde ein sequenzieller Ablauf modelliert, so bewirkt diese Aktion, dass die Aktivität im Ablauf des Lernprozesses nicht berücksichtigt und ausgelassen wird.
- **Deaktivieren**  
Diese Aktion deaktiviert die Aktivität. Das bedeutet, dass die Aktion weder ausgewählt, noch an den Lernenden geliefert werden kann.
- **Vor der Auswahl verstecken**  
Diese Aktion verhindert, dass die Aktivität Ziel einer Auswahl-Anfrage wird, falls der Eltern-Aktivität des Clusters ein Auswahl-Konstrukt zugeordnet ist.
- **Weiterleitung verhindern**  
Diese Aktion verhindert, dass nachfolgenden Aktivitäten jener Aktivität, welcher die Regel zugeordnet ist, an den Lernenden geliefert werden.

#### **Nach-Auswahl-Aktionen**

Nach-Auswahl-Aktionen sind Aktionen welche ausgeführt werden, nachdem ein Versuch einer Aktivität beendet ist.

- **Eltern-Aktivität beenden**  
Diese Aktion beendet den Versuch der Eltern-Aktivität jener Aktivität, welcher die Regel zugeordnet ist.
- **Alle beenden**  
Diese Aktion beendet alle Versuche im Aktivitätenbaum.



- **Neuer Versuch**  
Diese Aktion startet einen neuen Versuch jener Aktivität, welcher die Regel zugeordnet ist.
- **Alle neu versuchen**  
Diese Aktion beendet zuerst alle Versuche im Aktivitätenbaum und startet dann den Lernprozess erneut.
- **Nächste Aktivität**  
Diese Aktion liefert dem Lernenden die nächste Aktivität im modellierten sequentiellen Ablauf des Lernprozesses.
- **Vorige Aktivität**  
Diese Aktion liefert dem Lernenden die vorige Aktivität im modellierten sequentiellen Ablauf des Lernprozesses.

### **Exit-Aktion**

Exit-Aktionen sind Aktionen welche ausgeführt werden, nachdem ein Versuch einer Kind-Aktivität, oder an irgendeinem der Nachfahren, beendet ist.

- **Beenden**  
Diese Aktion beendet den Versuch jener Aktion, welcher die Regel zugeordnet ist.

## **6.4 Der Rollup-Prozess der Ablaufsteuerung**

Der Rollup-Prozess der Ablaufsteuerung ermöglicht es, den Lernfortschritt des Lernenden mit Eltern-Aktivitäten eines Clusters in Verbindung zu setzen. Da Lerninhalte nur Blatt-Aktivitäten zugeordnet werden können, gibt es keinen direkten Weg den Lernfortschritt eines Clusters zu messen. Abhilfe schaffen hier die Regeln des Rollup-Prozesses, sowie die Konstrukte des Rollup-Prozesses. Eine Regel des Rollup-Prozesses legt fest, wie sich der Lernfortschritt der Kind-Aktivitäten auf die Eltern-Aktivität auswirkt. Die Konstrukte des Rollup-Prozesses legen fest, welche Kind-Aktivitäten von den Regeln des Rollup-Prozesses überwacht werden.

### **Regeln des Rollup-Prozesses**

Einer Eltern-Aktivität eines Clusters können eine oder mehrere Regeln des Rollup-Prozesses zugeordnet werden. Eine Regel beinhaltet immer eine Menge von

Kind-Aktivitäten, welche von der Regel überwacht wird, eine Menge von Bedingungen und eine dazugehörige Aktion. Die Verknüpfung von mehreren Bedingungen sowie die Negation einer Bedingung erfolgt wie bei den Regeln der Ablaufsteuerung (siehe Abschnitt 6.3.1).

**Regel des Rollup-Prozesses:** If [Bedingung] for [Kind-Aktivitäten] then  
[Aktion]

### **Bedingungen**

Die Menge der möglichen Bedingungen einer Regel des Rollup-Prozesses entspricht der Menge der Bedingungen einer Regel der Ablaufsteuerung. Jedoch sind die Bedingungen *Ausmaß der Zielerreichung größer als* und *Ausmaß der Zielerreichung kleiner als* im Rollup-Prozess nicht zulässig.

### **Kind-Aktivitäten**

Grundsätzlich werden von einer Regel des Rollup-Prozesses alle Kind-Aktivitäten des Clusters berücksichtigt. Der Autor einer Regel hat jedoch die Möglichkeit explizit Kind-Aktivitäten aus dem Rollup-Prozess auszuschließen.

1. Mit Hilfe des Überwachungs-Konstruktes kann der Entwickler die Überwachung einer Aktivität deaktivieren. Diese wird dann auch vom Rollup-Prozess nicht mehr berücksichtigt.
2. Mit Hilfe des Rollup-Konstruktes Lernzielerreichung kann der Entwickler bewirken, dass eine Aktivität von einer Rollup-Regel, welche ein Lernziel betrifft, nicht berücksichtigt wird.
3. Mit Hilfe des Rollup-Konstruktes Ausmaß der Zielerreichung kann der Entwickler den Anteil den eine Aktivität an der Zielerreichung eines Lernzieles hat, auf 0 setzen und bewirkt somit, dass diese Aktivität von einer Rollup-Regel, welche ein Lernziel betrifft, nicht berücksichtigt wird.

Die Bedingungen werden für jede Kind-Aktivität, welche nicht durch eine der oben beschriebenen Methoden vom Rollup-Prozess ausgeschlossen wurde, evaluiert. Die folgenden Möglichkeiten bestehen um das Ergebnis dieser Evaluation zu bestimmen.

- **All**  
Die Regel liefert „wahr“, wenn die Bedingungen der Regel für alle Kind-Aktivitäten erfüllt sind.
- **Any**  
Die Regel liefert „wahr“, wenn die Bedingungen der Regel von irgendeiner der Kind-Aktivitäten erfüllt werden.
- **None**  
Die Regel liefert „wahr“, wenn die Bedingungen der Regel von keiner der Kind-Aktivitäten erfüllt werden.
- **Mindestens Anzahl**  
Die Regel liefert „wahr“, wenn mindestens eine definierte Anzahl von Kind-Aktivitäten die Bedingungen der Regel erfüllt.
- **Mindestens Prozent**  
Die Regel liefert „wahr“, wenn mindestens ein zuvor definierter Prozentsatz der Kind-Aktivitäten die Bedingungen der Regel erfüllt.

## **Aktionen**

Folgende vier Aktionen können von einer Regel des Rollup-Prozesses ausgelöst werden.

- **Lernziel erreicht**  
Diese Aktion bewirkt, dass das mit der Aktivität verknüpfte Lernziel den Status „erreicht“ erhält.
- **Lernziel nicht erreicht**  
Diese Aktion bewirkt, dass das mit der Aktivität verknüpfte Lernziel den Status „nicht erreicht“ erhält.
- **Aktivität abgeschlossen**  
Diese Aktion bewirkt, dass die Aktivität als abgeschlossen gilt.
- **Aktivität nicht abgeschlossen**  
Diese Aktion bewirkt, dass die Aktivität als nicht abgeschlossen gilt.

Das folgende Kapitel 7 geht nun darauf ein, wie diese Konzepte im eTutor umgesetzt und erweitert wurden um eine Ablaufsteuerung in einem bestehenden Intelligenten Tutoriellen System zu ermöglichen.

## **Teil 4**

### **Umsetzung der individualisierten Ablaufsteuerung im eTutor**

## 7 Die Ablaufsteuerung im eTutor

Das folgende Kapitel 7 zeigt, wie die im Kapitel 6 vorgestellten Konzepte zur Ablaufsteuerung im eTutor umgesetzt wurden um die im Kapitel 5 aufgezeigten Möglichkeiten einer individualisierten Ablaufsteuerung im eTutor zu erreichen. Es wird beschrieben, welche neuen Prozesse durch das Einführen einer Ablaufsteuerung entstehen und wie diese im Detail aussehen. Konzepte welche im Kapitel 6 beschrieben wurden, werden, falls sie im eTutor umgesetzt wurden, kurz mit ihren Eigenheiten beschrieben. Neben den allgemeinen Konzepten der Ablaufsteuerung wurden jedoch auch spezielle Konzepte für das eTutor-System entwickelt. Auf diese wird in diesem Abschnitt genauer eingegangen. Am Ende dieses Kapitels werden die Bereiche der Ablaufsteuerung im eTutor dargestellt und welche Möglichkeiten sie bieten.

Im Abschnitt 1.1 wurde das derzeitige eTutor-System beschrieben. Im Folgenden wird näher darauf eingegangen wie sich die Implementierung einer Ablaufsteuerung auf das derzeitige System auswirkt.

- **Lehrveranstaltung (LVA)**

Das Konzept einer Lehrveranstaltung wird durch die Einführung einer Ablaufsteuerung nicht verändert. Benutzer werden im eTutor einer Lehrveranstaltung zugeordnet und die Ablaufsteuerung greift dieses bestehende Konzept auf.

- **Aufgaben**

Das Konzept der Aufgaben wird auch mit der Einführung der Ablaufsteuerung erhalten bleiben, denn der eTutor kann definierte Aufgaben dem Benutzer präsentieren und verarbeiten. Diese Aufgaben sind zurzeit die einzigen Lerninhalte, welche die Ablaufsteuerung im eTutor verwenden kann.

- **Aufgabengruppen**

Das Konzept der Aufgabengruppen wird mit der Einführung der Ablaufsteuerung verworfen. Zur Gliederung der Lerninhalte wird der Aktivitätenbaum verwendet.

- **Übungspool**

Das Ziel bei der Konzeption der Ablaufsteuerung war es, die Übungen aus

dem bereits bestehenden Übungspool des eTutors auch in der Ablaufsteuerung verwenden zu können.

- **Benutzer**

Sowohl auf Studenten, als auch auf die Assistenten wirkt sich die neue Ablaufsteuerung aus. Studenten bekommen ihre Aufgaben in einer anderen Form präsentiert und somit soll ihr Lernprozess optimiert werden. Assistenten bekommen die Möglichkeit, Aktivitätenbäume zu erstellen, um auf diese Weise den Lernprozess von Studenten aktiv planen zu können. Die Tutoren sind von der Änderung des eTutors nicht betroffen. Durch die Verwendung des Konzeptes von Aufgaben in der Ablaufsteuerung, können diese weiterhin den Lernfortschritt von Studenten auswerten.

Der folgende Abschnitt 7.1 beschreibt den Aktivitätenbaum der Ablaufsteuerung und geht näher auf dessen Bedeutung und Funktion für die Ablaufsteuerung im eTutor ein.

## 7.1 Aktivitätenbaum

Wie in SCORM® vorgeschlagen, werden die Aktivitäten in Form eines Baumes organisiert und bilden somit den Aktivitätenbaum. Dieser Aktivitätenbaum ist das zentrale Element der Ablaufsteuerung im eTutor. Abbildung 15 zeigt die Teile eines Aktivitätenbaumes. Es wird der Aufbau eines Aktivitätenbaumes mit Aktivitäten und beispielhaft für Aktivität {A} die Elemente welche einer Aktivität zugeordnet werden können, gezeigt.

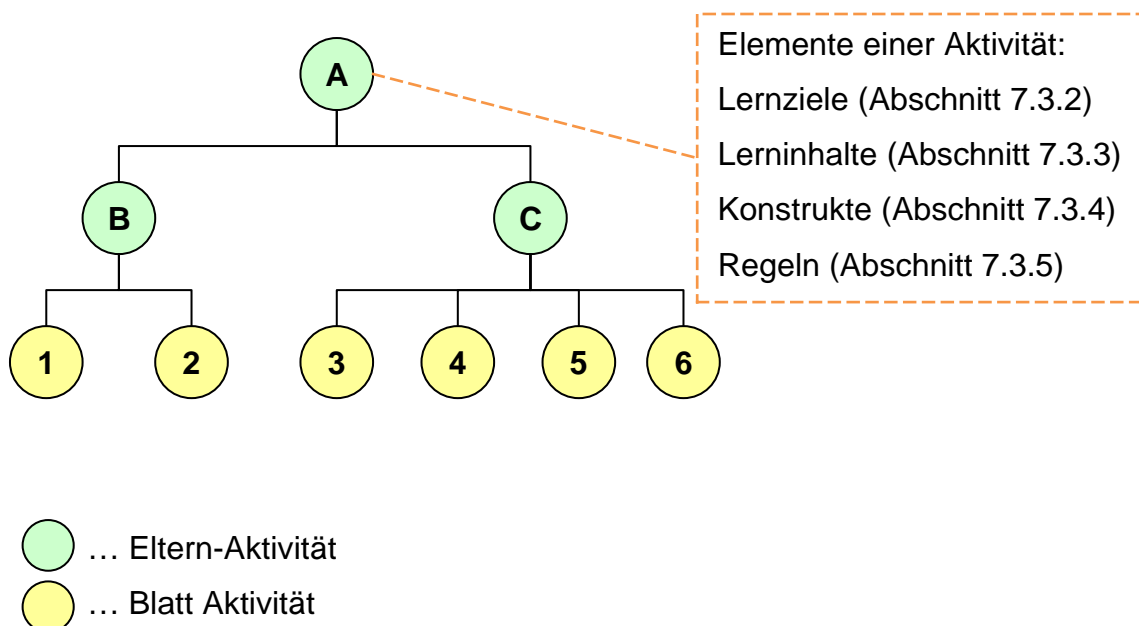


Abbildung 15: Teile eines Aktivitätenbaumes

Das Konzept der Cluster wurde im eTutor nicht übernommen, da die Zuordnung von Konstrukten zu beliebigen Aktivitäten erfolgen kann und hier keine Einschränkung für Blatt-Aktivität besteht. Neben der Wurzel des Baumes verweist der Aktivitätenbaum immer auf eine „aktuelle Aktivität“ und ausgehend von dieser wird durch den Baum navigiert. Jede Anfrage an den Aktivitätenbaum kann die „aktuelle Aktivität“ verändern. Ein Aktivitätenbaum gehört immer zu genau einer Lehrveranstaltung im eTutor. Ist einem Aktivitätenbaum kein Benutzer zugeordnet, so handelt es sich bei diesem Baum um die Vorlage eines Aktivitätenbaumes, welcher bei einer Lehrveranstaltung abgespeichert ist. Für diese Vorlage kann kein Lernprozess gestartet werden.

Das im eTutor umgesetzte Konzept eines Aktivitätenbaumes ist nicht mit anderen Lernsystemen austauschbar. Der Aktivitätenbaum wird zwar auch im eTutor in Form eines XML-Dokumentes gespeichert, dies geschieht jedoch vor allem deshalb, weil dadurch die Einbindung in die bestehende eTutor-Datenbank erleichtert und eine Validierung des Aktivitätenbaumes möglich wird. Im Gegensatz zu SCORM® wurden keine Konzepte umgesetzt, welche es ermöglichen, Packages aus Aktivitätenbäumen oder Teilbäumen zu erstellen.

Im Folgenden wird nun auf die Abläufe der Ablaufsteuerung näher eingegangen. Diese Abläufe gliedern sich in die Bereiche Administration und Ausführung.

## **7.2 Administration und Ausführung der Ablaufsteuerung**

Die Ablaufsteuerung kann aus zwei Gesichtspunkten betrachtet werden. Einerseits dient der Aktivitätenbaum zur Administration eines Kurses durch einen Autor im eTutor-System und andererseits ist er das zentrale Element bei der Ausführung eines Kurses durch einen Studenten. Benutzer, welche im eTutor der Benutzergruppe Assistent angehören, werden im Folgenden nur mehr als Autoren bezeichnet. Benutzer, welche der Benutzergruppe Student angehören, werden im Folgenden als Studenten bezeichnet. Es wird also zwischen

- der Administration und
- der Ausführung

eines Aktivitätenbaumes im eTutor unterschieden.

### **7.2.1 Administration eines Aktivitätenbaumes**

Mit der Ablaufsteuerung ändert sich die bereits vorhandene Organisation der Lerninhalte durch die Einführung von Aktivitätenbäumen (siehe Abschnitt 7.1) im eTutor. Autoren haben die Möglichkeit, nach dem Login am eTutor entweder

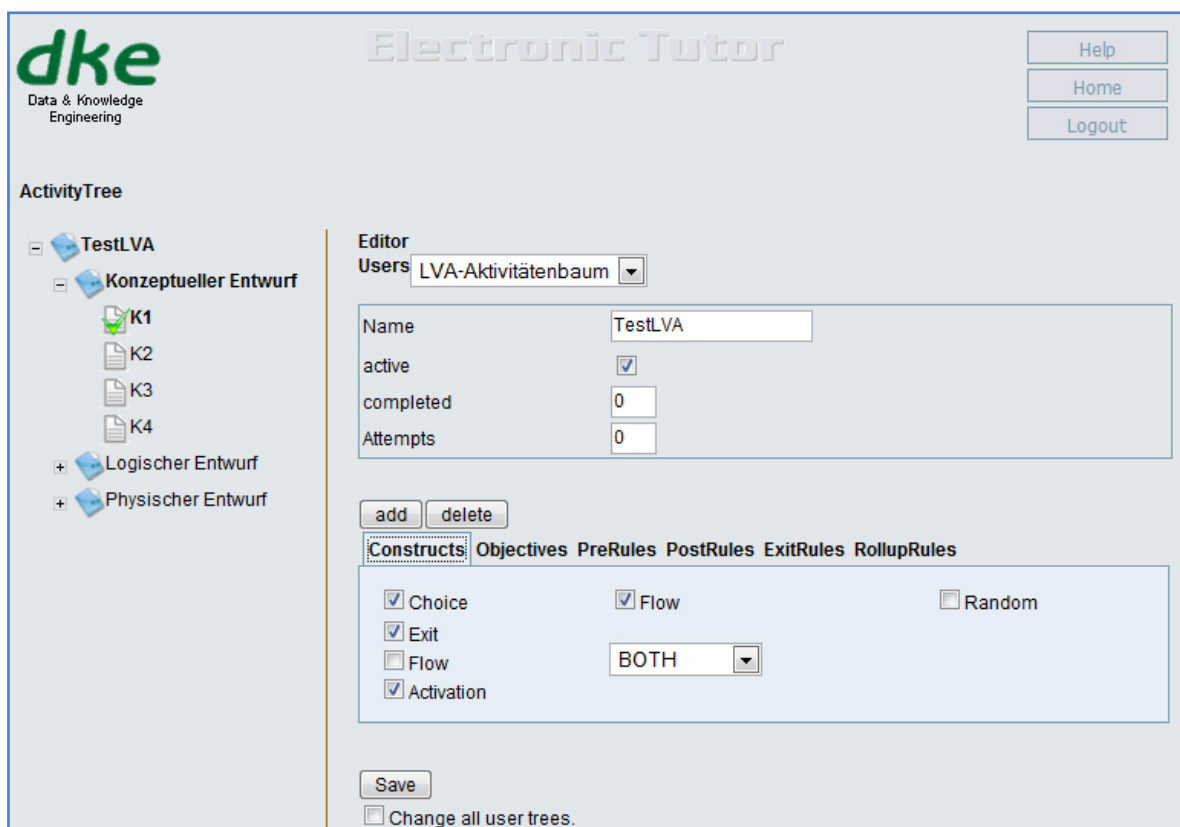
- einen neuen Kurs mit einer Ablaufsteuerung zu erstellen, oder
- einen bereits bestehenden Kurs um eine Ablaufsteuerung zu erweitern.

Zuerst erstellt der Autor im eTutor eine neue Lehrveranstaltung. Die Eigenschaften dieser Lehrveranstaltung und ihrer Bedeutung wie z.B. Name, Beschreibung oder Gültigkeitsdatum ändern sich durch die Einführung der Ablaufsteuerung nicht.

Als nächstes muss der Autor festlegen, welche Aufgaben der Ablaufsteuerung zur Verfügung gestellt werden sollen. Hier ändert sich das Vorgehen des Autors mit der Einführung der Ablaufsteuerung. Ohne die Ablaufsteuerung musste der Autor festlegen, wie die Aufgaben in Form von Kurseinheiten (werden in der Ablaufsteuerung als Lerneinheiten bezeichnet) strukturiert werden, welche Aufgaben eine Kurseinheit enthält und welche Aufgaben welchen Studenten zugeordnet werden sollen. Für die Ablaufsteuerung wird keine Strukturierung der Aufgaben in Kurseinheiten benötigt. Weiters sollten alle Aufgaben jedem Student zugeordnet werden und erst mit Hilfe des Aktivitätenbaumes eine Verfeinerung der Zuordnung vorgenommen werden. Der eTutor kann nur Aufgaben darstellen, welche einem eingeloggten Studenten zugeordnet sind. Der Autor hat in diesem Schritt also einen Pool von Aufgaben festgelegt. Diese Aufgaben können im Aktivitätenbaum an einzelne Aktivitäten gebunden werden.

Sobald eine Lehrveranstaltung existiert, kann der Autor zu dieser Lehrveranstaltung einen Aktivitätenbaum erstellen. Zu diesem Zweck wurde ein zweigeteilter Editor in den eTutor eingebunden, welcher das Erstellen und Ändern von Aktivitätenbäumen ermöglicht.





**Abbildung 16: Aktivitätenbaum Editor**

Der linke Bereich des Editors in Abbildung 16 zeigt die Struktur des Aktivitätenbaumes an. Dieser Baum wird im Folgenden Editor-Baum genannt. Er unterscheidet sich von jenem Baum, welcher der Student präsentiert bekommt. Durch Klicken auf eine Aktivität im Editor-Baum wird diese ausgewählt und deren Informationen (siehe Abschnitt 7.3.1) im rechten Bereich des Editors angezeigt. Der Autor erstellt nun jene Aktivitäten, welche im Lernprozess verwendet werden. Weiters kann er für jede Aktivität folgende Elemente erstellen:

- Konstrukte
- Ziele
- Regeln
- Rollup-Regeln
- Lerninhalte

Jede Lehrveranstaltung, welche die Ablaufsteuerung verwenden möchte, benötigt genau einen Aktivitätenbaum. Jeder Student dieser Lehrveranstaltung erhält automatisch eine Kopie des Aktivitätenbaumes der Lehrveranstaltung. Der Editor bietet dem Autor die Möglichkeit diese individuellen Bäume eines Studenten zu

editieren und gegebenenfalls individuelle Aktivitätenbäume für Studenten zu erstellen.

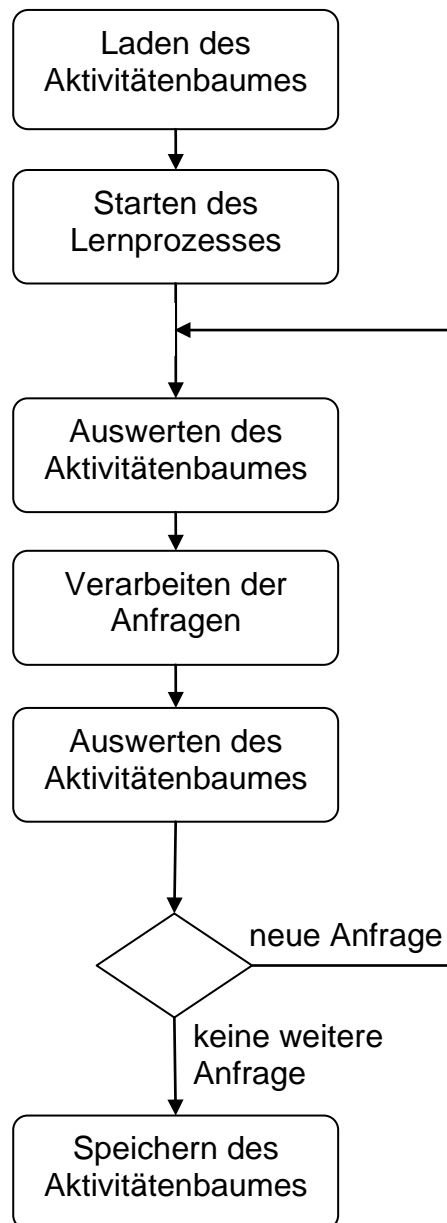
Will ein Autor nachträglich einen Aktivitätenbaum ändern, um z.B. Korrekturen durchzuführen, so kann diese individuelle Anpassung zu Problemen führen. Werden einzelne Aktivitätenbäume von Studenten geändert, so sind diese nicht mehr mit dem globalen Aktivitätenbaum des Kurses konsistent. Um dieses Problem zu lösen, bietet die Ablaufsteuerung im eTutor die Möglichkeit, Änderungen am globalen Aktivitätenbaum auch auf die einzelnen Aktivitätenbäume der Studenten zu übertragen. Diese Übertragung ist jedoch nur dann möglich, wenn keiner der Studenten den zu ändernden Kurs gerade im eTutor absolviert. Bei der Übertragung von Änderungen am Lehrveranstaltungsbaum auf die Studentenbäume gelten folgende Regeln:

- Wird eine Aktivität im Lehrveranstaltungsbaum hinzugefügt, so wird diese auch in den Studentenbäumen hinzugefügt.
- Wird eine Aktivität im Lehrveranstaltungsbaum gelöscht, so wird diese Aktivität auch in den Studentenbäume gelöscht. Alle Informationen dieser Aktivität, betreffend den Lernfortschritt des Studenten, gehen verloren.
- Wird eine Aktivität geändert, so werden diese Änderungen auch in den Studentenbäumen vorgenommen. Ausgenommen sind Änderungen, welche den Lernfortschritt des Studenten betreffen, wie Versuche der Aktivität, Status „abgeschlossen“ der Aktivität, oder Ziele, welche zur Aktivität gehören.

Neben der Administration spielt natürlich auch die Ausführung des Aktivitätenbaumes eine wesentliche Rolle. Die wichtigsten Abläufe dabei werden im folgenden Abschnitt 7.2.2 näher beschrieben.

### **7.2.2 Ausführung eines Aktivitätenbaumes**

Nimmt ein Student an einer im eTutor gespeicherten Lehrveranstaltung teil und ist für diese Lehrveranstaltung eine Ablaufsteuerung vorgesehen, so werden vom SequencingProcessor die in Abbildung 17 gezeigten Aufgaben ausgeführt. Beim SequencingProcessor handelt es sich um eine Erweiterung des eTutor Kern-Modules, welche für die Ablaufsteuerung zuständig ist.



**Abbildung 17: Ausführung eines Aktivitätenbaumes**

Wie in der Abbildung 17 ersichtlich wird, wird die Auswertung und das Verarbeiten von Anfragen während des gesamten Lernprozesses bei jeder neuen Benutzeranfrage wiederholt. Vor und nach jeder Benutzeranfrage wird der Aktivitätenbaum ausgewertet.

### **Laden eines Aktivitätenbaumes**

Wurde ein Student einer Lehrveranstaltung zugeordnet, so erhält er eine Kopie des gespeicherten Aktivitätenbaumes. Der Kopiervorgang erfolgt mit dem ersten Login des Studenten für diese Lehrveranstaltung. Weiters wird nach dem Login

der Aktivitätenbaum aus der Datenbank des eTutors geladen und in der Session des Studenten abgespeichert.

Die Aktivitätenbäume werden in der Datenbank in Form von XML-Dokumenten abgespeichert. Mit Hilfe dieses XML-Dokumentes werden die für die Ablaufsteuerung notwendigen Objekte initialisiert. Weiters werden zu diesem Zeitpunkt die Lerninhalte aus der eTutor-Datenbank geladen (siehe Abschnitt 7.3.3). Die Speicherung des Aktivitätenbaumes entspricht nicht der von SCORM® im Abschnitt 3.1.1 Book 2 – Content Aggregation Model vorgestellten Art und Weise.

### **Starten des Lernprozesses**

Nachdem ein Aktivitätenbaum geladen wurde, muss die Ablaufsteuerung den Lernprozess für diesen Aktivitätenbaum starten. Hier werden zwei Möglichkeiten unterschieden:

1. Die Wurzel des Aktivitätenbaumes ist inaktiv.  
Das bedeutet, dass der Lernprozess des Studenten in diesem Aktivitätenbaum zum ersten Mal beginnt. In diesem Fall wird eine Auswahl-Anfrage an die Wurzel-Aktivität des Baumes gestellt. Diese startet einen Versuch von dieser Aktivität und somit den Lernprozess.
2. Die Wurzel des Aktivitätenbaumes ist aktiv.  
Das bedeutet, dass der Lernprozess des Studenten in diesem Aktivitätenbaum bereits begonnen hat. Es wird nun jene Aktivität gesucht, welche keine aktiven Kind-Aktivitäten hat, denn diese Aktivität ist die letzte aktive Aktivität im Lernprozess des Studenten. Nun wird ein Versuch der gefundenen Aktivität gestartet und der Lernprozess des Studenten fortgesetzt.

Wie man sieht bewirkt das Starten des Lernprozesses auch den Start eines neuen Versuches der gerade aktuellen Aktivität. Wenn der Student also die Webseite des eTutors geöffnet hat und die Session des Studenten beendet wird, so bewirkt das Ende der Session auch das Ende des Versuches der Aktivität. Ein Versuch einer Aktivität dauert also maximal vom Zeitpunkt des Startes bis zum Zeitpunkt der Beendigung einer Session.

### **Auswerten des Aktivitätenbaumes**

Bei der Auswertung des Aktivitätenbaumes werden sowohl die Konstrukte als auch die Vor-Auswahl-Regeln einer Aktivität ausgewertet. Alle Aktivitäten sowie deren Kind-Aktivitäten, welche deaktiviert sind, werden nicht ausgewertet. Die Auswertung des Aktivitätenbaumes erfolgt ausgehend von der Wurzel für jede Aktivität des Baumes. Folgende drei Schritte werden dabei für jede Aktivität durchgeführt:

1. Aktivierung der Aktivität
2. Auswertung der Vor-Auswahl-Regeln
3. Auswertung der Konstrukte

In der Ablaufsteuerung des eTutors ist es also möglich, explizit Aktivitäten zu deaktivieren. Man kann Aktivitäten jedoch nicht explizit aktivieren, da dies automatisch bei jeder Auswertung des Aktivitätenbaumes erfolgt. Es wird angenommen Aktivität A soll mit Hilfe einer Vor-Auswahl-Regel deaktiviert werden. Bei jeder Auswertung des Baumes wird also Aktivität A aktiviert und danach werden die Vor-Auswahl-Regeln dieser Aktivität ausgewertet. Sind die Bedingungen der Vor-Auswahl-Regel erfüllt, deaktiviert diese die Aktivität A und Schritt 3 Auswertung der Konstrukte wird übersprungen.

Weiters sollte man bei der Auswertung des Aktivitätenbaumes beachten, dass sich Konstrukte auf Kind-Aktivitäten einer Aktivität auswirken. Das bedeutet, dass auf eine Aktivität zuerst die Konstrukte ihrer Eltern-Aktivität Wirkung zeigen und erst danach die Vor-Auswahl-Regel der Aktivität.

### **Verarbeitung der Anfragen an die Ablaufsteuerung**

Nachdem der Aktivitätenbaum ausgewertet wurde, können die Anfragen (siehe Abschnitt 7.4) an die Ablaufsteuerung verarbeitet werden. Diese Anfragen steuern den Lernprozess des Studenten. Anfragen können auf zwei Arten zustande kommen:

- Benutzergenerierte Anfragen welche durch ein Navigationsereignis eines Studenten ausgelöst werden.
- Systemgenerierte Anfragen welche durch Regeln der Ablaufsteuerung ausgelöst werden.

Anfragen werden immer sequentiell abgearbeitet. Es kann zu einem bestimmten Zeitpunkt immer nur eine Anfrage an die Ablaufsteuerung geben. Erst wenn diese Anfrage verarbeitet wurde, können neue Anfragen an die Ablaufsteuerung gestellt werden.

### **Auswertung des Aktivitätenbaumes**

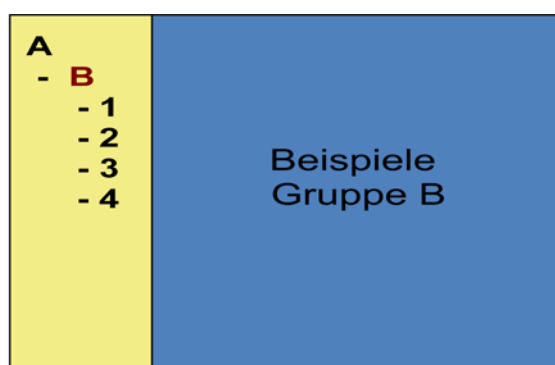
Nach der Verarbeitung der Anfragen an die Ablaufsteuerung muss der Aktivitätenbaum erneut ausgewertet werden.

### **Speichern des Aktivitätenbaumes**

Der Benutzer kann den Aktivitätenbaum nicht explizit speichern. Mit Beendigung einer Session im eTutor wird der Aktivitätenbaum vom System automatisch abgespeichert. Dabei wird zuerst aus dem Aktivitätenbaum ein XML-Dokument erzeugt und dieses Dokument wird danach in der Datenbank des eTutors abgelegt.

### **7.2.3 Präsentation eines Aktivitätenbaumes**

Nach der Auswertung und Anwendung der im Aktivitätenbaum gespeicherten Informationen über den Lernablauf wird das Ergebnis dem Studenten präsentiert. Diese Präsentation enthält eine Baumansicht zur Darstellung der strukturierten Lerninhalte sowie eine Aufgabenansicht, welche die zurzeit dem Studenten zugewiesene Aufgabe darstellt (siehe Abbildung 18).



**Abbildung 18: Schematische Darstellung des Benutzerinterfaces**

Der Baum des Benutzerinterfaces wird aus dem Aktivitätenbaum generiert. Konstrukte bestimmen darüber, welche Aktivitäten dem Benutzer angezeigt werden und welche nicht. Bei der Präsentation der Ergebnisse der

Ablaufsteuerung dient der Baum des Benutzerinterfaces zur Navigation durch das zur Verfügung gestellte Lernmaterial.

### **7.3 Bestandteile eines Aktivitätenbaumes**

Da nun die Abläufe der Ablaufsteuerung unter Verwendung eines Aktivitätenbaumes beschrieben wurden, werden nun die einzelnen Bestandteile des Aktivitätenbaumes näher erklärt.

#### **7.3.1 Aktivitäten**

Im eTutor wurde das von SCORM® vorgesehene Konzept von Lernaktivitäten umgesetzt. Lernaktivitäten werden im Folgenden Aktivitäten genannt. Die Wurzel-Aktivität des Aktivitätenbaumes unterscheidet sich dahingehend von allen anderen Aktivitäten, als dass sie keine Eltern-Aktivität besitzt und immer auswählbar ist. Alle Aktivitäten haben folgende Eigenschaften:

- Eindeutige ID: eine Aktivität kann im Aktivitätenbaum anhand der ID eindeutig identifiziert werden.
- Bezeichnung: eine Aktivität hat eine Bezeichnung, welche der Student im Baum präsentiert bekommt.
- Eltern-Aktivität: eine Aktivität hat genau eine Eltern-Aktivität.
- Aktiv-Status: eine Aktivität kann entweder aktiv oder nicht aktiv sein. Ist eine Aktivität aktiv, so sind auch alle Vorfahren-Aktivitäten aktiv.
- Teilnahme-Status: eine Aktivität weiß, ob sie am Lernprozess teilnimmt, oder nicht. Nimmt eine Aktivität nicht am Lernprozess teil, so gilt dies auch für ihre sämtlichen Nachfahren-Aktivitäten.
- Abgeschlossen-Status: ist eine Aktivität abgeschlossen, so bedeutet dies, dass die Aktivität ihre Aufgabe im Lernprozess erfüllt hat. Eine Aktivität wird im eTutor automatisch abgeschlossen, wenn ein Student das Ergebnis der zur Aktivität zugehörigen Aufgabe an den eTutor übermittelt. Eine Aktivität wird nicht abgeschlossen, wenn der Student sein Ergebnis zum Testen an den eTutor übermittelt.

- Versuche: eine Aktivität weiß, wie oft sie vom Studenten aufgerufen wurde. Jede Aktivierung einer Aktivität startet einen Versuch. Wird eine Aktivität inaktiv, so endet der Versuch.

Zu einer Aktivität können Konstrukte, Regeln und Ziele hinzugefügt werden. Eine Aktivität kann diese auswerten und daraufhin seinen Status oder den Status seiner Kind-Aktivitäten ändern.

### Versuche an einer Aktivität

Das Bemühen des Studenten eine Aktivität abzuschließen wird als Versuch bezeichnet.

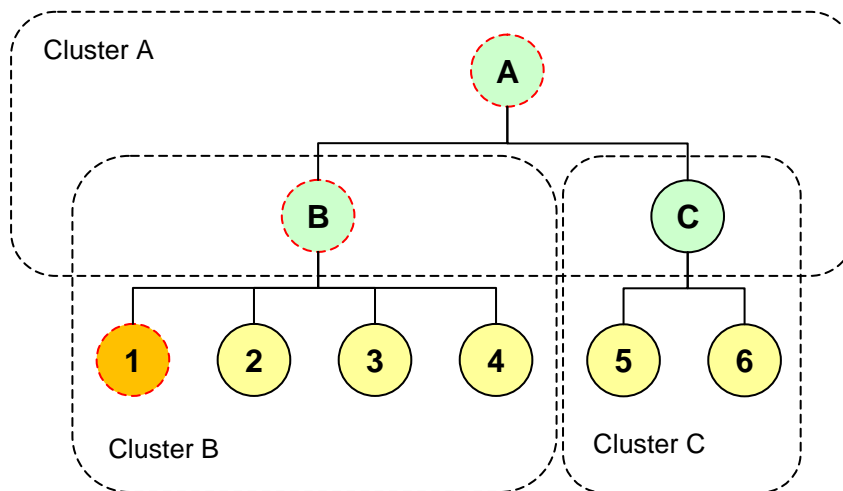


Abbildung 19: Versuch an einer Aktivität

Abbildung 19 zeigt den Status eines Aktivitätsbaumes. Die Aktivitäten {A, B, 1} sind aktiv, d.h. diese Aktivitäten werden vom Studenten zurzeit versucht. Weiters ist die aktuelle Aktivität {1} jene Blatt-Aktivität, welche dem Studenten präsentiert wird. Im folgenden Abschnitt werden die Schritte beim **Starten eines Versuchs** beschrieben unter der Annahme, dass der Student im Aktivitätsbaum der Abbildung 24 die Aktivität {6} auswählt.

1. Überprüfen, ob die gewählte Aktivität {6} bereits aktiv ist. Ist dies der Fall, so wird kein neuer Versuch gestartet.
2. Beenden aller Versuche an der aktuellen Aktivität {1} und deren Eltern-Aktivitäten {B, A} welche keine Vorfahren der gewählten Aktivität {6} sind. D.h. die Versuche an den Aktivitäten {1, B} werden beendet.



3. Wurde durch das Beenden der Aktivitäten eine neue Anfrage an die Ablaufsteuerung ausgelöst, wird der Startvorgang eines Versuchs an Aktivität {6} abgebrochen
4. Aktivität {6} wird zur aktuellen Aktivität und aktiv. Der Versuchszähler wird um 1 erhöht.
5. Alle Eltern-Aktivitäten der Aktivität {6}, welche noch nicht aktiv sind, werden aktiv und ihr Versuchszähler wird um 1 erhöht. Dies betrifft in diesem Beispiel die Aktivität {C}.

Wird der Versuch einer Aktivität beendet, wie hier z.B. der Versuch der Aktivität {B}, so werden folgende Schritte von der Ablaufsteuerung durchgeführt.

1. Überprüfen, ob die Aktivität aktiv ist. Eine inaktive Aktivität wurde vom Studenten nicht versucht.
2. Aktivität auf inaktiv setzen.
3. Rollup-Regeln auswerten.
4. Nach-Auswahl-Regeln auswerten. Löst eine Nach-Auswahl-Regel eine Anfrage an die Ablaufsteuerung aus, so wird diese Anfrage weiterverarbeitet.
5. Falls die Aktivität, deren Versuch beendet werden soll, eine Eltern-Aktivität besitzt, so werden die Exit-Regeln dieser Eltern-Aktivität ausgewertet. Löst eine dieser Exit-Regeln eine Anfrage an die Ablaufsteuerung aus, so wird diese weiterverarbeitet.

### **7.3.2 Lernziele**

Lernziele im eTutor haben die Aufgabe, den Benutzer sowohl über ihren Lernfortschritt zu informieren, als auch Auskunft über Lernziele zu geben. Weiters dienen Lernziele dazu, den Lernprozess zu steuern.

Ein Lernziel hat einen Namen, der im Kontext des Aktivitätenbaumes eindeutig sein sollte. Weiters kann ein Lernziel eine textuelle Beschreibung erhalten. Diese Beschreibung kann dem Studenten im Benutzerinterface angezeigt werden. Somit kann man dem Studenten mitteilen, welche Aufgaben er zu erfüllen hat, um ein Lernziel zu erreichen.

### **Beispiel: Lernziel**

**Name:** Datenmodellierung abschließen

**Anzeigen im Benutzermodell:** Ja

**Beschreibung:** Lösen Sie je eine Aufgabe zum konzeptuellen, logischen und physischen Entwurf.

Lernziele werden immer einer Aktivität zugeordnet. Jede Aktivität kann beliebig viele Lernziele haben. Im Gegensatz zum Status einer Aktivität können Lernziele von jeder Bedingung einer Regel im Aktivitätenbaum abgefragt werden. Der Lernfortschritt, welcher durch ein Lernziel erfasst wird, ist somit für alle Aktivitäten im Aktivitätenbaum sichtbar.

### **7.3.3 Lerninhalte**

Im Gegensatz zu SCORM® kann der eTutor nur Lerninhalte darstellen, welche im eTutor gespeichert sind. Bei diesen Lerninhalten handelt es sich um durch Autoren definierte Aufgaben. Jede Aufgabe enthält wiederum eine Übung aus dem Übungspool des eTutors. Die Lerninhalte kommunizieren direkt mit dem eTutor. Die Ablaufsteuerung bewirkt nur, dass ein gewünschter Lerninhalt angezeigt wird. Wird das Ergebnis einer Aufgabe an den eTutor übermittelt, so teilt dieser der Ablaufsteuerung mit, dass die Aufgabe abgeschlossen wurde. Wie von SCORM® vorgeschlagen, können nur Blatt-Aktivitäten des Aktivitätenbaumes Lerninhalte beinhalten. Die Ablaufsteuerung des eTutors speichert nur die eindeutige ID einer Aufgabe. Wird ein Aktivitätenbaum geladen, so werden die benötigten Aufgaben von der Ablaufsteuerung geladen.

Der eTutor unterscheidet zwischen einer statischen und flexiblen Zuordnung von Übungen zu Aufgaben. Dies spielt auch für die Ablaufsteuerung eine Rolle und wird im Folgenden erläutert.

- **Statische Zuordnung**

Bei einer statischen Zuordnung wird einer Aufgabe genau eine Übung, aus dem im eTutor gespeichert Pool von Übungen, zugeordnet. Wird eine Aufgabe einem Studenten zugeordnet, so wird diese Aufgabe mit der damit verbundenen Übung für den Studenten angelegt.

- Flexible Zuordnung

Bei der flexiblen Zuordnung wird einer Aufgabe ein Muster für eine Übung zugeordnet. Dieses Muster schränkt die Menge der Übungen, welche im eTutor gespeichert sind, ein. Wird eine Aufgabe einem Studenten zugeordnet, so wird aus dieser eingeschränkten Menge eine Übung ausgewählt. Für den Studenten wird die Aufgabe mit der zugeordneten Übung angelegt.

Die Ablaufsteuerung bietet die Möglichkeit bei einer flexiblen Zuordnung von Übungen zu Aufgaben, diese Zuordnung während des Lernprozesses erneut auszuführen. Für jede Aktivität kann angegeben werden, ob die Übung der Aufgabe beim Abschließen einer Aktivität erneut zugeordnet werden sollen.

Wurden einer Aktivität von einem Autor keine Lerninhalte zugeordnet, oder bewirkt die erneute Zuordnung der Übung zur Aufgabe, dass keine neue Übung mehr für die Aufgabe eingetragen ist, so wird die Aktivität von der Ablaufsteuerung automatisch deaktiviert.

### **7.3.4 Konstrukte**

Im eTutor wurden folgende Konstrukte für die Ablaufsteuerung umgesetzt:

- Auswahl-Konstrukt
- Sequenz-Konstrukt
- Zufalls-Konstrukt

Das Auswahl-Konstrukt im eTutor wird einer Aktivität zugeordnet und wirkt auf deren Kind-Aktivitäten. Alle drei von SCORM® vorgeschlagenen Optionen für das Auswahl-Konstrukt (siehe Abschnitt 6.2) wurden im eTutor umgesetzt. Nur wenn der Eltern-Aktivität einer Aktivität ein Auswahl-Konstrukt zugeordnet wurde, wird diese im Baum des Benutzerinterfaces angezeigt.

Wurde einer Eltern-Aktivität einer Aktivität ein Sequenz-Konstrukt zugeordnet, so bedeutet dies, dass der Benutzer im Benutzerinterface die Möglichkeit bekommt, vor und zurück zu navigieren, wenn die Aktivität gerade ausgewählt ist. Ein Sequenz-Konstrukt im eTutor ermöglicht drei Einstellungen:

- Vorwärts Navigation
- Rückwärts Navigation

- Vor- und rückwärts Navigation

Das Zufalls-Konstrukt im eTutor ändert die Reihenfolge der Kind-Aktivitäten einer Aktivität. Dies wirkt sich sowohl auf die Baumdarstellung im Benutzerinterface, als auch auf die logische Abfolge der Aktivitäten in einer Lernsequenz aus.

Im eTutor wurde zurzeit noch kein Vorwärts-Sequenz-Konstrukt umgesetzt. Dieser Unterschied zu SCORM® wird dann ersichtlich, wenn man ein Auswahl-Konstrukt und ein Sequenz-Konstrukt kombiniert. Das Auswahl-Konstrukt bewirkt, dass der Student eine Menge von Aktivitäten im Baum angezeigt bekommt, welche er direkt auswählen kann. Durch das Sequenz-Konstrukt erhält er weiters die Möglichkeit von einer Aktivität zur nächsten zu navigieren. Das Vorwärts-Sequenz-Konstrukt würde all jene Aktivitäten, welche logisch vor der gerade aktiven Aktivität liegen, vor dem Studenten verbergen. Auch jene Aktivitäten, welche der Benutzer direkt auswählen kann, würden ausgeblendet. Diese Art der Ablaufsteuerung kann zurzeit im eTutor mit den bestehenden Konstrukten nicht realisiert werden.

Weiters wurde im eTutor kein Vorauswahl-Konstrukt umgesetzt. Möchte man bei der Auswahl einer Eltern-Aktivität festlegen, dass eine bestimmte Kind-Aktivität als erste dem Studenten präsentiert wird, so muss diese Kind-Aktivität logisch vor allen anderen liegen. Der eTutor springt automatisch von der Eltern-Aktivität zu seiner ersten Kind-Aktivität, falls der Eltern-Aktivität ein Sequenz-Konstrukt zugeordnet wurde. Wurde der Eltern-Aktivität ein Auswahl-Konstrukt zugeordnet, so wählt der eTutor keine Kind-Aktivität automatisch aus, sondern wartet auf die Entscheidung des Benutzers.

### **7.3.5 Regeln**

In Anlehnung an die regelbasierte Ablaufsteuerung von SCORM® unterscheidet der eTutor vier Arten von Regeln, welche sich vor allem dadurch unterscheiden, dass sie zu unterschiedlichen Zeitpunkten ausgewertet werden.

- **Vor-Auswahl-Regeln**

Diese Regeln werden bei jeder Auswertung des Aktivitätenbaumes für jede Aktivität ausgewertet.

- **Nach-Auswahl-Regeln**

Diese Regeln werden ausgewertet, wenn ein Versuch einer Aktivität

beendet wurde. Die Auswertung der Nach-Auswahl-Regeln erfolgt nach der Auswertung der Rollup-Regeln.

- **Exit-Regeln**

Endet ein Versuch einer Aktivität, werden für alle Vorfahren-Aktivitäten dieser Aktivität die Exit-Regeln ausgewertet. Die Auswertung der Exit-Regeln erfolgt nach der Auswertung der Nach-Auswahl-Regeln. Löst z.B. eine Nach-Auswahl-Regel eine Anfrage an die Ablaufsteuerung aus, die dem Benutzer die nächste Aktivität liefern soll, so kann eine Exit-Regel keine Anfrage mehr auslösen.

- **Rollup-Regeln**

Diese Regeln werden ausgewertet, wenn ein Versuch einer Aktivität beendet wurde. Weiters werden alle Rollup-Regeln aller Vorfahren dieser beendeten Aktivität ausgewertet. Mit dem Beenden eines Versuchs werden sozusagen alle Informationen der Kind-Aktivitäten gesammelt und von den Vorfahren ausgewertet.

Der Aufbau einer Regel ist wie folgt:

**Regel der Ablaufsteuerung:** If [Bedingung] then [Aktion]

Jede Regel kann genau eine Aktion ausführen, wenn die Bedingungen der Regel erfüllt sind. SCORM® sieht vor, dass eine Regel mehrere Bedingungen haben kann und diese Bedingungen miteinander verknüpft werden. Im eTutor wurde dies, jedoch vor allem zum Zweck der Gliederung der Bedingungen, ebenfalls umgesetzt.

### **Bedingungen von Regeln**

SCORM® schlägt für Regeln eine klar definierte Menge von Bedingen vor, welche miteinander verknüpft werden. Der eTutor verwendet für Bedingungen von Regeln das Konzept der dynamischen Codegenerierung zur Laufzeit. Eine Bedingung der Ablaufsteuerung des eTutors besteht aus folgenden Bestandteilen:

- Bedingungstext
- SQL-Statement (optional)

Der Bedingungstext enthält Java-Ausdrücke. Diese Ausdrücke werden zur Laufzeit ausgewertet und müssen einen booleschen Wert zurück liefern. Die

Ausdrücke können mit den logischen Operatoren UND und ODER verknüpft und mit einem logischen NICHT negiert werden. Die Auswertung einer Bedingung wird verkürzt, wenn ein Ergebnis bereits fest steht.

**Beispiel:**

*ATTEMPTS > 5 || LERNZIEL\_1 == TRUE*

Diese Bedingung prüft, ob eine Aktivität öfter als fünf mal versucht und ob ein Lernziel mit dem Namen „LERNZIEL\_1“ bereits erreicht wurde. Ist der erste Teilausdruck dieser Bedingung erfüllt, so liefert die Bedingung wahr und das Lernziel wird nicht mehr überprüft. Angenommen, das Lernziel wurde im Benutzermodell gespeichert, so erspart man sich hier den Zugriff auf das Benutzermodell und die Verarbeitung der Bedingung wird beschleunigt.

Wie die folgende Tabelle zeigt, können im Bedingungstext mehrere Schlüsselwörter verwendet werden um auf die Daten der Ablaufsteuerung zuzugreifen.

Schlüsselwort	Bedeutung
<b>ACTIVE</b>	Boolescher Wert der angibt, ob die Aktivität gerade aktiv ist, oder nicht.
<b>ATTEMPTS</b>	Numerischer Wert welcher die Anzahl der Versuche an dieser Aktivität enthält.
<b>COMPLETED</b>	Numerischer Wert der angibt, wie oft die Aktivität vom Studenten bereits abgeschlossen, d.h. das Ergebnis der Aufgabe an den eTutor übermittelt, wurde.
<b>ACTIVITY</b>	Dieses Schlüsselwort referenziert jene Aktivität zu welcher die Regel mit der enthaltenen Bedingung gehört. Mit Hilfe dieses Schlüsselwortes kann auf jedes in der Aktivität enthaltene Objekt zugegriffen werden. <b>Beispiel: Zugriff auf die TaskID der Aufgabe</b> ACTIVITY.getContent().getTaskID().intValue()
<b>ACTIVITYTREE</b>	Dieses Schlüsselwort referenziert den aktuellen Aktivitätenbaum. Mit Hilfe dieses Schlüsselwortes kann eine Bedingung auf jedes beliebige Objekt im Aktivitätenbaum zugreifen. <b>Beispiel: Ermitteln der UserID</b> ACTIVITYTREE.getUserID().intValue()

**Tabelle 3: Schlüsselwörter in Bedingungen**

Weiters kann im Bedingungstext auf drei verschiedene Informationen zugegriffen werden:

- Lernziele
- Ergebnisse der SQL-Abfrage auf die eTutor-Datenbank
- Daten des Benutzermodells

All diese Informationen werden im Bedingungstext über einen Namen referenziert. Da dieser Name auch Leerzeichen enthalten kann, muss er im Bedingungstext mit eckigen Klammern umgeben sein.

### **Beispiel: Referenzieren eines Lernzieles im Bedingungstext**

[Datenmodellierung abgeschlossen] == true

Zusätzlich zu einem Bedingungstext kann eine Bedingung ein SQL-Statement enthalten und ermöglicht so den Zugriff auf die eTutor-Datenbank. Zur Laufzeit wird dieses SQL-Statement vor der Auswertung des Bedingungstextes ausgeführt und liefert eine Ergebnismenge zurück. Betrachtet man die Ergebnismenge als Tabelle, so wird von dieser Tabelle genau die erste Zeile weiterverwendet. Der Spaltenname bildet zusammen mit dem ersten Wert der Spalte ein neues Schlüsselwort, welches im Bedingungstext verwendet werden kann.

### **Beispiel: Zugriff auf die eTutor-Datenbank**

SQL-Statement: *Select count(\*) as ANZAHL\_LVAS from Iva\_group;*

Ergebnismenge:

<b>ANZAHL_LVAS</b>
<b>110</b>

Nach Ausführung dieses Statements gibt es das neue Schlüsselwort ANZAHL\_LVAS mit dem Wert 110. Das Mapping der zwischen dem SQL-Typ der Spalte und dem Java-Typ des Schlüsselwortes erfolgt automatisch anhand der JDBC-Spezifikation des verwendeten Datenbanktreibers.

Die in Tabelle 3: Schlüsselwörter in Bedingungen angegebenen Schlüsselwörter können auch im SQL-Statement verwendet werden. Um zu erreichen, dass das SQL-Statement als Java-Ausdruck erkannt und somit ausgewertet wird, muss dieses mit einem doppelten Anführungszeichen (") beginnen.

### Beispiel: Evaluiertes SQL-Statement

SQL-Statement:

```
"Select SUGGESTPOINTS from submission where USER_ID = " +
ACTIVITYTREE.getUserID().intValue() + " and task_id = " +
ACTIVITY.getContent().getTaskID().intValue()
```

Nach Ausführung dieses Statements gibt es das Schlüsselwort SUGGESTPOINTS welches angibt, wie viele Punkte der Student bei einer übermittelten Aufgabe erhält. Mit Hilfe dieses Schlüsselwortes könnte man z.B. eine Regel „IF [SUGGESTPOINTS] > 10 THEN RETRY“ erstellen.

### Aktionen von Regeln

Sind die Bedingungen einer Regel erfüllt, so wird die der Regel zugeordnete Aktion ausgeführt. Im eTutor gibt es eine, wie von SCORM® vorgeschlagen, feste Menge von Aktionen. Je nach Art der Regel gibt es unterschiedliche Aktionen.

Vor-Auswahl-Regeln	
Aktion	Beschreibung
<b>HIDE</b>	Diese Aktion verbirgt die Aktivität im Benutzerinterface sowohl vor einer direkten Auswahl, als auch in einem sequenziellen Ablauf.
<b>NOCHOICE</b>	Diese Aktion verbirgt die Aktivität im Benutzerinterface vor einer direkten Auswahl. Die Aktion wird nicht im Baum des Benutzerinterfaces angezeigt.
<b>NOFLOW</b>	Diese Aktion verbirgt die Aktivität in einem sequentiellen Ablauf. Die Aktivität kann nicht mehr durch „weiter“ und „zurück“ im Benutzerinterface ausgewählt werden.
<b>DISABLE</b>	Diese Aktion deaktiviert die Aktivität. Das bedeutet, dass die Aktion weder ausgewählt, noch an den Lernenden geliefert werden kann. Ist eine Aktivität deaktiviert, so wird sie von der Ablaufsteuerung nicht mehr berücksichtigt.



Nach-Auswahl-Regeln	
Aktion	Beschreibung
<b>EXITPARENT</b>	Diese Aktion beendet den Versuch an der Eltern-Aktivität der Aktivität, zu welcher die Regel gehört. Diese Aktion löst eine Exit-Anfrage für die Eltern-Aktivität aus.
<b>END</b>	Diese Aktion beendet den Lernprozess. Es wird eine Exit-Anfrage für die Wurzel-Aktivität des Aktivitätenbaumes ausgelöst.
<b>RETRY</b>	Diese Aktion startet einen neuen Versuch der aktuellen Aktivität.
<b>RETRYALL</b>	Diese Aktion beendet den Lernprozess, also den Versuch der Wurzel-Aktivität und startet danach einen neuen Versuch an der Wurzel-Aktivität.
<b>NEXT</b>	Diese Aktion liefert dem Benutzer die nächste Aktivität im Lernprozess, falls ein sequentieller Ablauf modelliert wurde. Wurde eine nächste Aktivität gefunden, startet ein neuer Versuch dieser.
<b>PREVIOUS</b>	Diese Aktion liefert dem Benutzer die vorige Aktivität im Lernprozess, falls ein sequentieller Ablauf modelliert wurde. Wurde eine vorige Aktivität gefunden, startet ein neuer Versuch dieser.
Exit-Regeln	
Aktion	Beschreibung
<b>EXIT</b>	Diese Aktion beendet den Versuch an der Aktivität zu welcher die Exit-Regel gehört. Mit Hilfe von Exit-Regeln kann also das Ende eines Versuchs einer Aktivität im Aktivitätenbaum nach oben weitergereicht werden.
Rollup-Regeln	
Aktion	Beschreibung
<b>COMPLETE</b>	Diese Aktion schließt eine Aktivität ab. Dies ist für alle Aktivitäten von Bedeutung, welche keine Blätter im Aktivitätenbaum sind. Im Gegensatz zu Blatt-Aktivitäten können diese nicht durch das Abschicken einer Aufgabe an den eTutor abgeschlossen werden.
<b>INCOMPLETE</b>	Diese Aktion bewirkt, dass eine Aktivität als nicht abgeschlossen gekennzeichnet wird.

<b>ACHIEVED</b>	Diese Aktion bewirkt, dass ein Lernziel als erreicht gilt. Als zusätzlichen Parameter erwartet diese Aktion den Namen eines Lernzieles. Wurde kein Name eines Lernzieles angegeben, so werden alle Lernziele, welche der Aktivität der Regel zugeordnet sind, als erreicht markiert.
<b>NOTACHIEVED</b>	Diese Aktion bewirkt, dass ein Lernziel als nicht erreicht gilt. Als Parameter wird die bei der Aktion ACHIEVED der Name eines Lernzieles erwartet.

**Tabelle 4: Aktionen von Regeln**

Regeln im eTutor unterscheiden sich von SCORM-Regeln vor allem durch die möglichen Bedingungen im eTutor. Weiters können auch noch andere, als in SCORM® aufgelistete, Aktionen in den eTutor eingebaut werden. So wäre es z.B. denkbar eine Aktion für Nach-Auswahl-Regeln zu implementieren, welche zu einer neuen Auswahl-Anfrage führt und es somit ermöglicht eine beliebige Aktivität im eTutor abhängig von einer Regel auszuwählen.

#### **7.4 Anfragen an die Ablaufsteuerung**

Zur Steuerung der Ablaufsteuerung dienen die so genannten Anfragen an die Ablaufsteuerung. Abschnitt 7.2 beschreibt wie diese Anfragen zustande kommen und wie sie verarbeitet werden. Dieser Abschnitt geht nun näher auf die einzelnen Anfragen an die Ablaufsteuerung ein und beschreibt deren Auswirkung auf die Ablaufsteuerung. Abbildung 20 zeigt ein Zustandsdiagramm welche zeigt, wie sich eine Anfrage auf den Zustand einer Aktivität auswirkt.

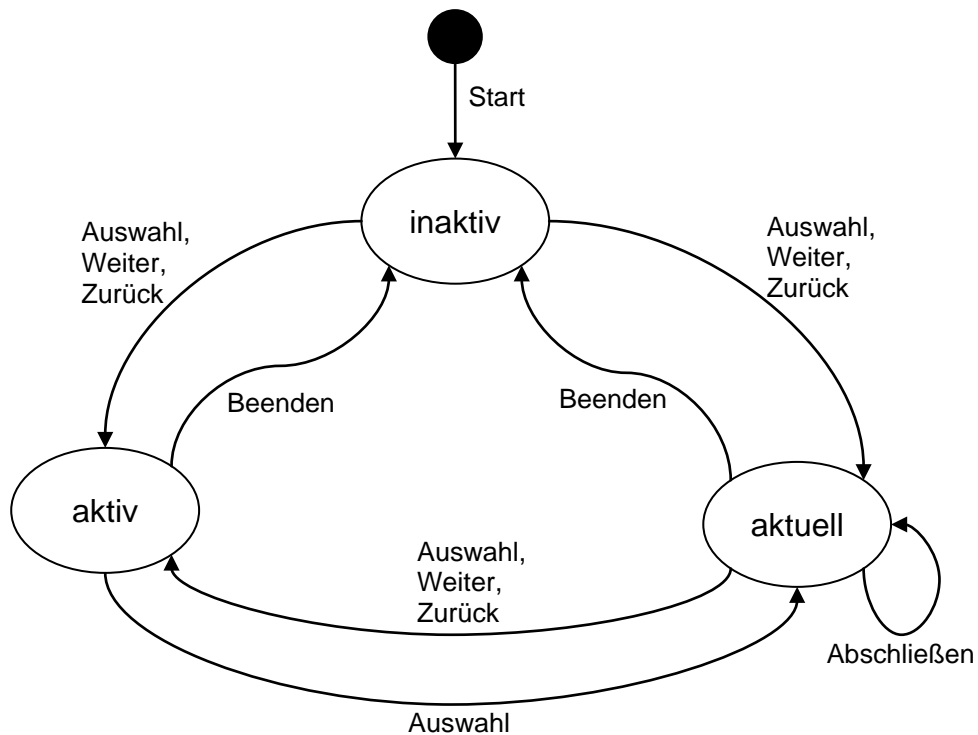


Abbildung 20: Zustandsdiagramm Anfragen/Aktivitäten

### (1) Start-Anfrage

Diese Anfrage startet den Lernprozess des Studenten und kennzeichnet den Beginn der Ablaufsteuerung (siehe Abschnitt 7.2.2).

### (2) Auswahl-Anfrage

Eine Auswahl-Anfrage kommt zustande, wenn der Student im Baum des Benutzerinterfaces eine Aktivität direkt auswählt. Eine Auswahl-Anfrage hat als Parameter immer eine Ziel-Aktivität, also jene vom Studenten ausgewählte Aktivität. Folgende Bedingungen müssen erfüllt sein, damit es sich um eine gültige Auswahl-Anfrage handelt:

- Die Ziel-Aktivität muss zur Auswahl zur Verfügung stehen. D.h. der Eltern-Aktivität der Ziel-Aktivität muss ein Auswahl-Konstrukt zugeordnet sein und die Ziel-Aktivität darf nicht aufgrund einer Regel von dieser Auswahl ausgeschlossen sein.
- Die Ziel-Aktivität muss aktiviert sein. Eine deaktivierte Aktivität kann nie das Ziel einer Anfrage an die Ablaufsteuerung sein.

Sind die oben genannten Bedingungen erfüllt, startet ein Versuch an der Ziel-Aktivität. Nach dem Start des Versuches wird überprüft, ob es sich bei der Ziel-Aktivität um eine Blatt-Aktivität handelt, oder nicht. Handelt es sich nicht um eine Blatt-Aktivität, bedeutet dies, dass die Ziel-Aktivität Kind-Aktivitäten besitzt und die Ablaufsteuerung wertet nun das Sequenz-Konstrukt aus. Wurde ein sequenzieller Ablauf des Lernprozesses in beiden Richtungen, also vorwärts und rückwärts, modelliert, so startet die Ablaufsteuerung einen Versuch der ersten Kind-Aktivität. Wurde ein sequentieller Ablauf modelliert, welcher nur die Navigation zurück erlaubt, so startet ein Versuch der letzten Kind-Aktivität. Dies bedeutet, dass bei einer Auswahl-Anfrage auf eine Aktivität mit Kind-Aktivitäten automatisch eine Weiterleitung auf eine Kind-Aktivität erfolgt, falls der Ziel-Aktivität der Auswahl-Anfrage ein Sequenz-Konstrukt zugeordnet ist.

### **(3) Weiter-Anfrage**

Eine Weiter-Anfrage kommt zustande, wenn der Student im Benutzerinterface das Navigationselement „weiter zur nächsten Aktivität“ auswählt. Eine Weiter-Anfrage liefert dem Studenten ausgehend von der aktuellen Aktivität des Aktivitätenbaumes die nächste Aktivität im sequentiellen Ablauf des Lernprozesses. Folgende Bedingungen müssen erfüllt sein, damit es sich um eine gültige Weiter-Anfrage handelt:

- Für die aktuelle Aktivität muss ein sequentieller Fluss modelliert worden sein, welcher eine Vorwärts-Navigation zulässt. D.h. der Eltern-Aktivität der aktuellen Aktivität muss ein Sequenz-Konstrukt zugeordnet sein und die aktuelle Aktivität darf nicht aufgrund einer Regel vom sequentiellen Ablauf ausgeschlossen sein.
- Die aktuelle Aktivität muss aktiviert sein.

Gibt es keine weitere Aktivität im sequentiellen Ablauf, so handelt es sich zwar um eine gültige Anfrage an die Ablaufsteuerung, es wird dem Studenten jedoch keine neue Aktivität geliefert und die aktuelle Aktivität bleibt aktiv.

### **(4) Zurück-Anfrage**

Eine Weiter-Anfrage kommt zustande, wenn der Student im Benutzerinterface das Navigationselement „zurück zur vorigen Aktivität“ auswählt. Eine Zurück-Anfrage liefert dem Studenten jene Aktivität welche im sequentiellen Ablauf des

Lernprozesses vor der aktuellen Aktivität liegt. . Folgende Bedingungen müssen erfüllt sein, damit es sich um eine gültige Zurück-Anfrage handelt:

- Für die aktuelle Aktivität muss ein sequentieller Fluss modelliert worden sein, welcher eine Rückwärts-Navigation zulässt. D.h. der Eltern-Aktivität der aktuellen Aktivität muss ein Sequenz-Konstrukt zugeordnet sein und die aktuelle Aktivität darf nicht aufgrund einer Regel vom sequentiellen Ablauf ausgeschlossen sein.
- Die aktuelle Aktivität muss aktiviert sein.

Gibt es keine Aktivität im sequentiellen Ablauf welche vor der aktuellen Aktivität liegt, so handelt es sich zwar um eine gültige Anfrage an die Ablaufsteuerung, es wird dem Studenten jedoch keine neue Aktivität geliefert und die aktuelle Aktivität bleibt aktiv.

#### **(5) Wiederholen-Anfrage**

Diese Anfrage startet einen neuen Versuch der aktuellen Aktivität.

#### **(6) Neustart-Anfrage**

Diese Anfrage startet den Lernprozess neu. Es wird eine Ende-Anfrage an die Ablaufsteuerung gestellt und danach startet ein neuer Versuch des Wurzel-Knotens des Aktivitätenbaumes.

#### **(7) Beenden-Anfrage**

Diese Anfrage beendet den Versuch an einer Aktivität. Wird an diese Anfrage eine Aktivität als Parameter übergeben, so wird der Versuch dieser übergebenen Aktivität und alle Versuche von Kind-Aktivitäten der übergebenen Aktivität beendet. Wird an die Anfrage keine Aktivität übergeben, so wird der Versuch der gerade aktuellen Aktivität beendet.

#### **(8) Ende-Anfrage**

Diese Anfrage beendet den Lernprozess. Es wird eine Beenden-Anfrage an die Wurzel-Aktivität des Aktivitätenbaumes gestellt.

#### **(9) Abschließen-Anfrage**

Diese Anfrage schließt die aktuelle Aktivität ab. Das bedeutet, der Student hat seinen Versuch an der Aktivität beendet und das Ergebnis der ihm gestellten

Aufgabe an den eTutor übermittelt. Der Zähler der Aktivität, welcher misst, wie oft die Aktivität abgeschlossen wurde, wird um 1 erhöht.

Weiters prüft diese Anfrage, ob die aktuelle Aktivität Teil eines modellierten sequentiellen Ablaufs ist. Ist dies der Fall, so wird eine Weiter-Anfrage an die Ablaufsteuerung gestellt und der Lernprozess des Studenten wird fortgesetzt. Ist die Aktivität nicht Teil eines sequentiellen Ablaufs, so wird eine Beenden-Anfrage für die aktuelle Aktivität erstellt.

### **Neuzuordnung von Übungen**

Wurde einer Aktivität ein Lerninhalt zugeordnet, so wird beim Abschließen einer Aktivität geprüft, ob es sich um eine dynamische Zuordnung der Übung zur Aufgabe der Aktivität handelt. Ist dies der Fall, so erfolgt eine erneute Zuordnung einer Übung. Übungen, welche bereits als Lerninhalte der Aktivität zugeordnet waren, werden nicht mehr berücksichtigt. Findet die Ablaufsteuerung keine neuen Übungen mehr, so wird die Aktivität deaktiviert. Das bedeutet der Student hat die Übung abgeschlossen und da es keine neue Übung gibt, kann die Aktivität im Lernprozess keine Rolle mehr spielen.

### **7.5 Beispiel zur Verwendung**

Im Kapitel 7 wurde beschrieben, wie die Konzepte der Ablaufsteuerung im eTutor umgesetzt wurden. Dieser Abschnitt widmet sich nun der Frage wie diese Konzepte angewendet werden müssen um den im Kapitel 5 definierten Scope der Ablaufsteuerung bei der Modellierung einer Lehrveranstaltung zu erreichen. Es wird angenommen, dass ein Benutzer eine Aktivität solange wiederholen soll, bis er ein gewünschtes Lernziel erreicht hat. Abbildung 21 zeigt einen Aktivitätenbaum bei dem die Aktivität {B} solange wiederholt werden soll, bis der Student alle Teilaktivitäten {1, 2, 3, 4} einmal versucht hat. Erst dann soll es ihm möglich sein die Aktivität {C} auszuwählen und mit dem Lernprozess fortzufahren.

Einige der definierten Anforderungen wie ein zufälliger Lernprozess, eine Verzweigung oder ein linearer Lernprozess lassen sich einfach durch die Anwendung eines im eTutor implementierten Konstruktes umsetzen. Das Modellieren einer Schleife erfordert jedoch die Kombination mehrerer vorgestellter Konzepte und wird hier nun anhand des Beispiels umgesetzt.

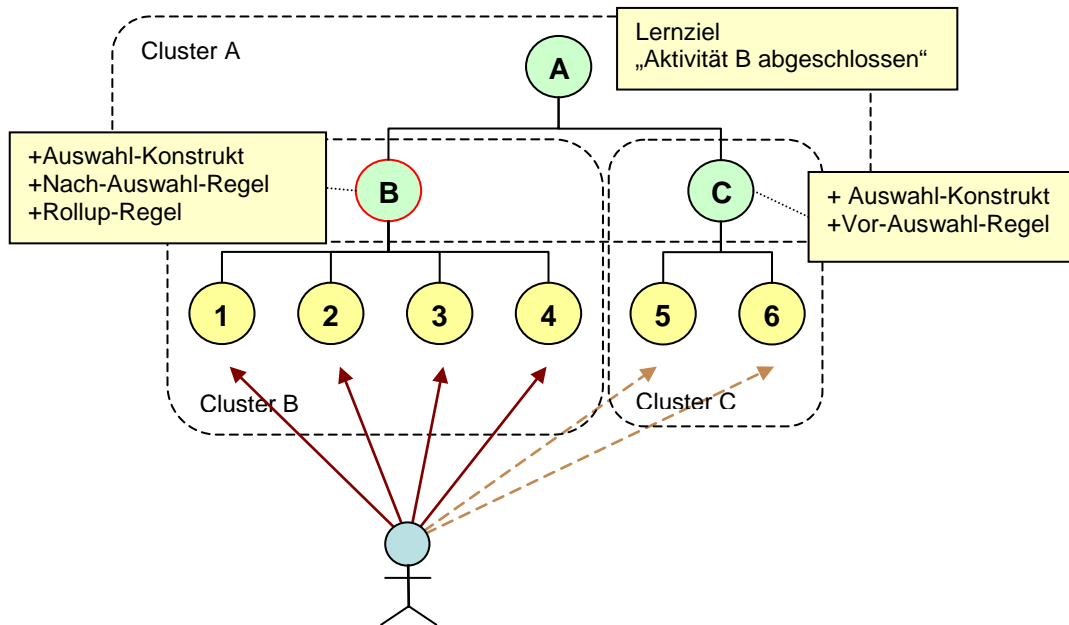


Abbildung 21: Beispiel einer Schleife

Um eine Schleife zu modellieren, sind einige Konstrukte, Regeln und Lernziele notwendig.

- Lernziel „Aktivität B abgeschlossen“

Die Schleife benötigt eine Bedingung um festzustellen wie lange der Student die Aktivität {B} wiederholen soll. Diese Bedingung prüft ob das Lernziel „Aktivität B abgeschlossen“ erreicht wurde. Solange dies nicht der Fall ist, wird die Aktivität {B} wiederholt. Am Namen des Lernzieles kann man schon erkennen, dass die Information, ob die Aktivität B abgeschlossen ist, oder nicht, ja auch bei der Aktivität selbst abgespeichert ist und deshalb für die Schleife kein eigenes Lernziel notwendig wäre. Zusätzlich zur Schleife ist in diesem Beispiel auch gefordert, dass die Aktivität {C} erst dann zur Auswahl steht, wenn die Aktivität {B} abgeschlossen wurde und diese Information steht nicht zur Verfügung, falls nicht explizit ein Lernziel festgelegt wird.
- Nach-Auswahl-Regel der Aktivität {B}

Die Nach-Auswahl-Regel der Aktivität {B} prüft ob das definierte Lernziel erreicht wurde und wenn dies nicht der Fall ist, so wird der Versuch an der Aktivität wiederholt.
- Rollup-Regel der Aktivität {B}

Die Rollup-Regel der Aktivität {B} prüft für jede Kind-Aktivität der Aktivität

{B} ob diese bereits einmal versucht wurde. Ist diese Bedingung erfüllt, so markiert die Rollup-Regel das definierte Lernziel als erfüllt. Die Rollup-Regel wird immer dann ausgelöst, wenn der Versuch einer Kind-Aktivität beendet wird.

- Vor-Auswahl-Regel der Aktivität {C}

Die Vor-Auswahl-Regel der Aktivität {C} versteckt die Aktivität solange bis das definierte Lernziel erreicht wurde. Erst dann steht dem Studenten die Aktivität zur Auswahl zur Verfügung.

Das beschriebene Beispiel zeigt, wie man die Konzepte der Ablaufsteuerung im eTutor kombinieren kann um einen komplexen Lernprozess zu gestalten.

## 7.6 Fazit

In diesem Kapitel wurde die Ablaufsteuerung im eTutor beschrieben. Tabelle 5 zeigt welche Aspekte zurzeit umgesetzt werden können.

Ablaufsteuerung	eTutor-System
Verzweigung	✓
Schleife	✓
Linearer Lernprozess	✓
Zufälliger Lernprozess	✓
Auswahl	✓
Eingeschränkte Auswahl	✓
Eingeschränkte Adaption	✓
Vollständige Adaption	✗
Intelligente Adaption	✗
Instruktoren	✗
Face to Face	✗



<b>Kollegen</b>	
-----------------	---



**Tabelle 5: Ablaufsteuerung im eTutor**

Das eTutor-System bietet nun die Möglichkeit die Sequenz der Lerninhalte auf vielfältige Weise an die Bedürfnisse der Benutzer anzupassen. Aspekte von gelenktem Lernen (siehe Kapitel 5) wie Verzweigungen, Schleifen, ein linearer Lernprozess oder ein zufälliger Lernprozess können nun im eTutor modelliert werden. Weiters hat ein Student die Möglichkeit den Lernprozess selbst mit zu gestalten. Dennoch konnten nicht alle Aspekte individualisierten Lernens abgedeckt werden. Der eTutor unterstützt keine vollständige oder intelligente Adaption. Zurzeit ist es also nicht möglich die Lerninhalte selbst an die Bedürfnisse eines Studenten anzupassen. Des Weiteren wurden keine Ansätze für gemeinschaftliches Lernen und für virtuelle Klassenzimmer implementiert. Es ist nicht vorgesehen, dass ein Instruktor einen Studenten durch den Lernprozess führt, oder dass Studenten gemeinsam einen im eTutor modellierten Kurs absolvieren.

## 8 Implementierung

Nachfolgend wird auf die Implementierung der Ablaufsteuerung eingegangen. Die dafür neu hinzugekommenen Packages und wichtige Klassen (Tabelle 6: Packages der Ablaufsteuerung) im eTutor und externe Komponenten werden aufgelistet.

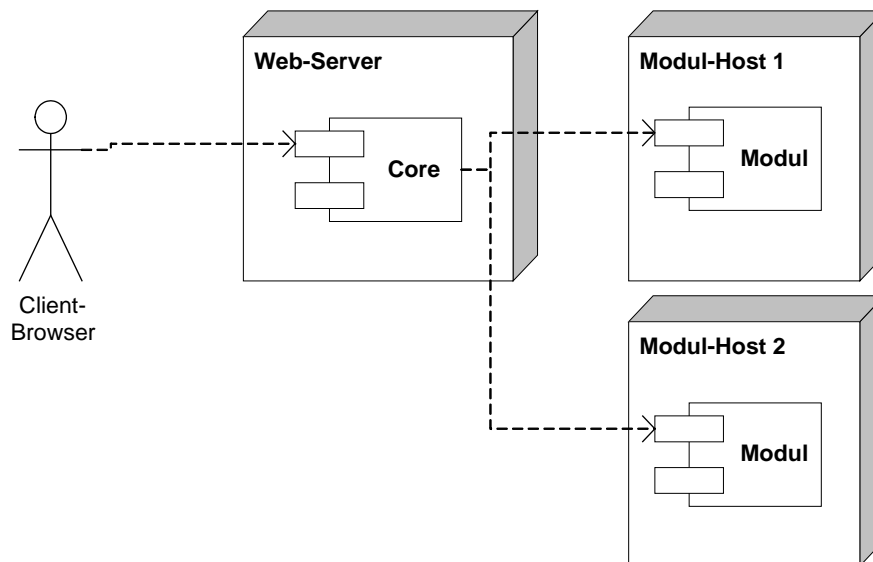


Abbildung 22: eTutor Systemarchitektur [EKNi06]

Abbildung 22 zeigt die drei Schichten der eTutor Systemarchitektur. Beim Client handelt es sich um einen Web-Browser zur Interaktion mit dem eTutor. Der WebServer enthält das Kern-Modul des eTutors welcher mit den einzelnen Modulen des eTutors kommuniziert (siehe Abschnitt 1.1). Die Auswertungsmodule wiederum stellen aufgabenspezifische Auswertungsfunktionalitäten zur Verfügung.

Abbildung 23 zeigt, dass die Ablaufsteuerung als Teil des Kernmoduls des eTutors auf dem Server implementiert wurde. Ein weiterer Teil der Ablaufsteuerung wurde auf dem Client (im Falle des eTutors dem Webbrowser) in Form von JavaScript-Funktionalitäten umgesetzt. Der clientseitige Teil der Ablaufsteuerung wurde wie die serverseitige Ablaufsteuerung in Java implementiert und mit Hilfe eines Compilers in JavaScript übersetzt. Die Teile der Ablaufsteuerung kommunizieren unabhängig vom eTutor Kernmodul in Form einer asynchronen Kommunikation miteinander (siehe Abschnitt 8.1). Für diese Kommunikation tauschen die clientseitige und die serverseitige Ablaufsteuerung so genannte Sequencing Beans miteinander aus.

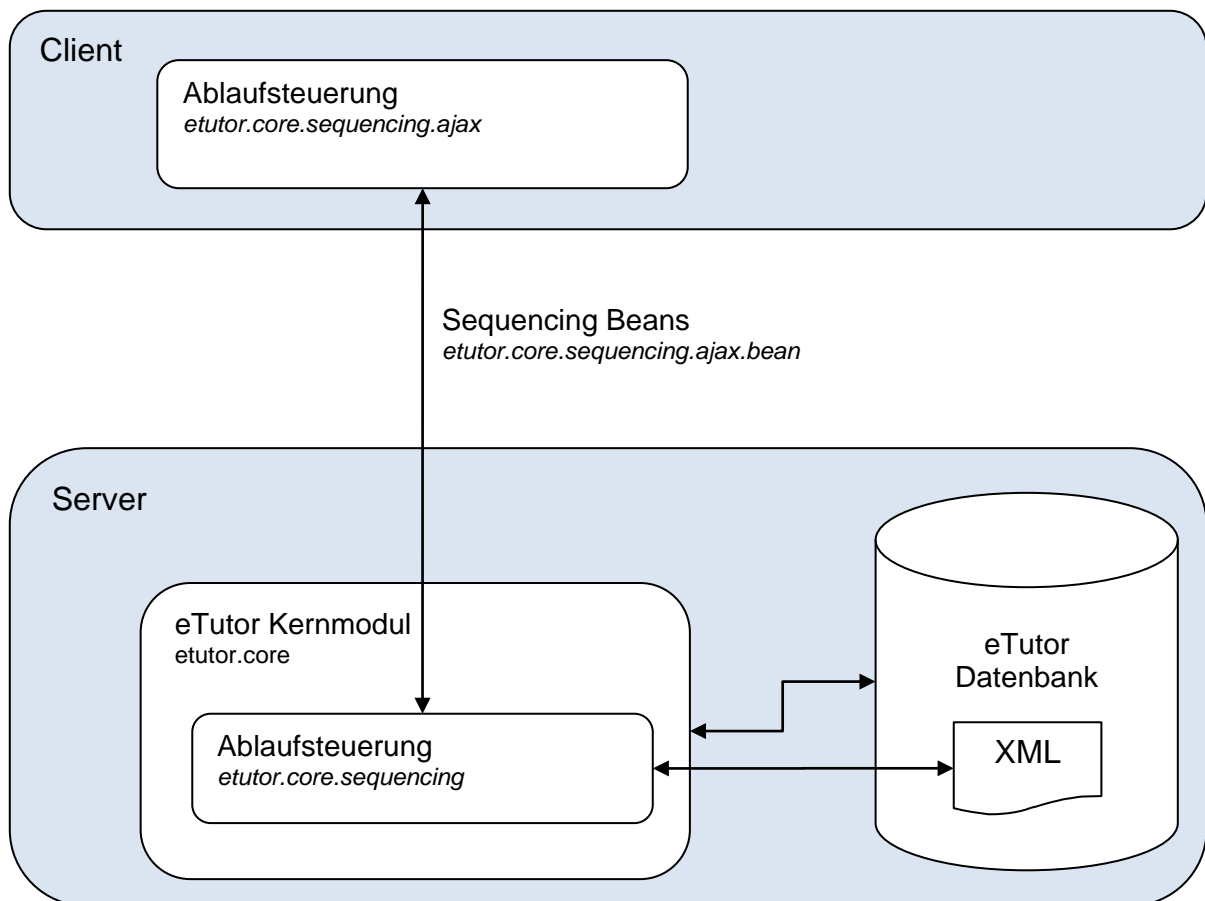


Abbildung 23: Architektur der Ablaufsteuerung

Des Weiteren greift die serverseitige Ablaufsteuerung direkt auf die eTutor-Datenbank zu um Aktivitätenbäume zu laden und zu speichern. Die Aktivitätenbäume werden in der Datenbank in Form von XML-Dokumenten gespeichert. Diese Dokumente werden in Form eines zusätzlichen CLOB-Feldes in den bereits bestehenden Tabellen abgelegt. Die Implementierung der Ablaufsteuerung hat keine Veränderungen an anderen bestehenden Modulen und Teilen des eTutors vorgenommen.

Zusätzlich zum Ablaufsteuerungsmodul wurde das Kernmodul des eTutors um die Servlets `etutor.core.SequencingServlet` und `etutor.core.SequencingAssistentServlet` erweitert. Diese Servlets dienen der asynchronen Kommunikation der Ablaufsteuerung.

Packages/Klassen	Beschreibung
<b>etutor.core. SequencingServlet</b>	Dienst zur asynchronen Kommunikation der Ablaufsteuerung
<b>etutor.core. SequencingAssistantServlet</b>	Dienst zur asynchronen Kommunikation des Editors zur Erstellung von Aktivitätenbäumen
<b>etutor.core. sequencing</b>	Serverseitige Implementierung der Ablaufsteuerung. Dieses Package enthält alle Klassen zur Realisierung der Ablaufsteuerung im eTutor. Dieses Package enthält keine Klassen zur Visualisierung eines Aktivitätenbaumes oder zur Erstellen eines solchen.
<b>etutor.core. sequencing.construct</b>	Dieses Package enthält alle Klassen welche die Konstrukte der Ablaufsteuerung abbilden
<b>etutor.core. sequencing.rule</b>	Dieses Package enthält alle Klassen welche die Regeln der Ablaufsteuerung abbilden
<b>etutor.core. sequencing.ActivityTree</b>	Diese Klasse implementiert einen Aktivitätenbaum
<b>etutor.core. sequencing.ajax</b>	Clientseitige Implementierung der Ablaufsteuerung. Alle Klassen dieses Packages werden mit Hilfe eines Compilers in JavaScript übersetzt.
<b>etutor.core.sequencing. ajax.SequencingTree</b>	Diese Klasse visualisiert einen Aktivitätenbaum auf dem Client. Die Klasse wird in JavaScript übersetzt und erzeugt dynamisch einen Baum zur Navigation durch den Lernprozess.
<b>etutor.core. sequencing.ajax.beans</b>	Dieses Package enthält alle Beans, welche zwischen der clientseitigen und der serverseitigen Ablaufsteuerung ausgetauscht werden können.
<b>etutor.core. sequencing.editor</b>	Dieses Package enthält Klassen für einen Editor zum Erstellen und Ändern von Aktivitätenbäumen.

Tabelle 6: Packages der Ablaufsteuerung

## 8.1 Kommunikation der Ablaufsteuerung

Die Kommunikation zwischen dem Studenten und dem eTutor erfolgt für die Ablaufsteuerung asynchron.

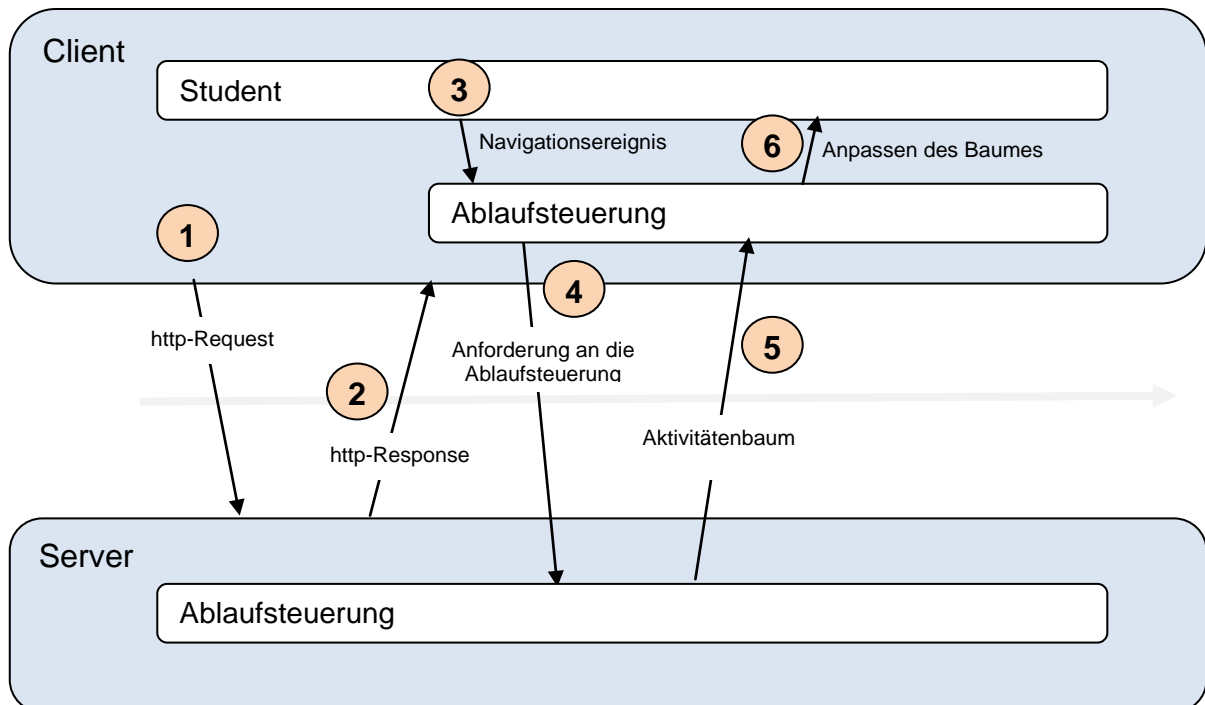


Abbildung 24: Kommunikation der Ablaufsteuerung

Als erstes fällt bei der Abbildung 24 ins Auge, dass die Ablaufsteuerung keine http-Request verwendet um Informationen zwischen dem eTutor und dem Studenten auszutauschen. Lediglich das Laden der Ablaufsteuerung erfolgt auf die klassische, im Web übliche, Art und Weise.

1. Beim Zugriff auf die Webseite, welche die Lerninhalte eines Kurses anzeigt, generiert der Browser aufgrund der Benutzeraktion einen http-Request an den Server.
2. Nachdem der Server den Request bearbeitet und das Benutzerinterface für den Benutzer generiert hat, wird dieses in Form eines http-Response an den Client zurück geschickt und als Webseite dargestellt. Diese Webseite enthält Javascript-Programmcode, welcher die Ablaufsteuerung auf dem Client übernimmt.
3. Nachdem die Ablaufsteuerung auf dem Client initialisiert wurde, wird ein Navigationsereignis des Studenten nicht mehr direkt als http-Request an den Server geschickt, sondern zuerst von der Ablaufsteuerung bearbeitet.

4. Die Ablaufsteuerung erstellt aus dem Navigationsereignis eine asynchrone Anfrage und schickt diese direkt an die Ablaufsteuerung des Servers. Die Webseite auf dem Client wird hierbei nicht erneut geladen und der Student kann inzwischen weiterarbeiten.
5. Nachdem die Ablaufsteuerung auf dem Server die Anfrage verarbeitet hat, wird der durch die Anfrage veränderte Aktivitätenbaum zurück an die Ablaufsteuerung des Clients geschickt.
6. Die Ablaufsteuerung des Clients erkennt, dass eine zuvor abgeschickte Anfrage eine Antwort vom Server erhalten hat. Der übermittelte Aktivitätenbaum wird nun vom Client ausgewertet und die Änderungen werden in das bestehende Benutzerinterface übertragen.

Der folgende Abschnitt 8.2 beschreibt die bei der Implementierung der Ablaufsteuerung im eTutor verwendeten externen Komponenten.

## **8.2 Externe Komponenten**

Mit der Einführung der Ablaufsteuerung wurde der eTutor um folgende externe Komponenten erweitert.

- **Bean Shell** (Version 2.0b2)  
Bean Shell ist eine Erweiterung, welche es dem eTutor ermöglicht in textueller Form vorliegenden Java-Code zur Laufzeit auszuwerten und auszuführen. Diese Erweiterung wird für die Implementierung der Bedingungen in der Ablaufsteuerung benötigt. Die Bedingungen können Java-Code enthalten welcher zur Laufzeit von der Ablaufsteuerung zur Evaluierung einer Bedingung ausgeführt werden muss [BeSh07].
- **Google Web Toolkit** (Version 1.4)  
Das Google Web Toolkit ermöglicht die Programmierung von Anwendungen in Java, welche von einem Compiler in Java-Script umgewandelt und somit von einem Webbrowser ausführbar sind. Es stellt Funktionalitäten zur Programmierung eines Benutzerinterfaces einer Webanwendung in Java zur Verfügung. Elemente des Benutzerinterfaces wie z.B. Bäume oder Tabellen werden mit Hilfe von Java-Script dynamisch in einer Webseite erzeugt. Die Ablaufsteuerung wendet diese Funktionalitäten zur Implementierung eines Baumes an, welcher im eTutor

die Darstellung eines Aktivitätenbaumes und die Kommunikation zwischen der clientseitigen und der serverseitigen Ablaufsteuerung übernimmt. Das Google Web Toolkit stellt Mechanismen für eine asynchrone Kommunikation einer Webanwendung zur Verfügung. Des Weiteren wird es zur Implementierung eines Editors, welcher das Erstellen und Ändern eines Aktivitätenbaumes ermöglicht, herangezogen [GWT07].

## **Teil 5**

### **Zusammenfassung und Ausblick**



## 9 Zusammenfassung

Diese Arbeit hat gezeigt, wie eine Ablaufsteuerung im Intelligenten Tutoriellen System „eTutor“, des Instituts für Data & Knowledge Engineering, realisiert werden kann. Dabei wurden bestehende Ansätze zur Ablaufsteuerungen betrachtet und in Kategorien eingeteilt. Statische Ablaufsteuerungen zur Beurteilung von bereits bestehenden Systemen, sowie aktive und passive Ablaufsteuerungen, welche das Lernziel des Lernenden berücksichtigen wurden beschrieben. Da ein individualisierter Lernprozess zu guten Lernergebnissen führt, wurde auch die Umsetzung dieser Art von Ablaufsteuerung analysiert. Die Bedeutung von der SCORM®-Spezifikation zur Umsetzung einer Ablaufsteuerung wurde aufgezeigt. Des Weiteren wurden die Ansätze des Adaptiven Hypermedia beschrieben, welche ebenfalls zur Ablaufsteuerung eines Kurses eingesetzt werden können. Um eine Entscheidung treffen zu können, welcher Ansatz bei der Umsetzung im eTutor am sinnvollsten ist, wurde der Ist-Zustand des eTutor-System untersucht und Evaluierungskriterien erfasst. SCORM® und Adaptives Hypermedia wurden an Hand dieser gegenübergestellt. Aufgrund dieser Gegenüberstellung erfolgte die Auswahl von Konzepten zur Realisierung einer Ablaufsteuerung. Die Konzepte bieten die Möglichkeit den Lernprozess des Studenten zu lenken, erlauben es ihm jedoch auch, selbst Einfluss zu nehmen und den Lernprozess zu gestalten. In Abschnitt 2.1 wurden unterschiedliche Kategorien von Ablaufsteuerungen vorgestellt. Im Folgenden wird darauf eingegangen, welchen dieser Kategorien die Ablaufsteuerung im eTutor zugeordnet werden kann.

- **Statische Ablaufsteuerung**

Im eTutor ist es weiterhin möglich statische Kurse anzulegen und zu absolvieren. Die Ablaufsteuerung im eTutor greift jedoch nicht auf diese statischen Kurse zu und kann die Qualität dieser Kurse nicht beurteilen.

- **Aktive und Passive Ablaufsteuerung**

Im eTutor ist es nun möglich, Lernziele für einen Benutzer festzulegen. Durch das Modellieren eines Aktivitätenbaumes kann ein Autor im eTutor nun festlegen, wie ein Student dieses Lernziel erreichen soll. Mit Hilfe der Regeln der Ablaufsteuerung ist der eTutor nun auch in der Lage, auf Veränderungen im Wissensstand des Studenten zu reagieren. Das

Benutzermodell speichert dieses Wissen und je nach Bedarf verändert sich dementsprechend der Lernprozess des Studenten. Er bekommt andere Aufgabenstellungen präsentiert oder er kann Teile, welche sein Wissen nicht mehr erweitern würden, überspringen. Der eTutor umfasst nun sowohl eine aktive, als auch eine passive Ablaufsteuerung.

- **Adaptive Ablaufsteuerung**

Der eTutor ist nicht in der Lage Lerninhalte an die Bedürfnisse des Studenten anzupassen. Dies resultiert vor allem daraus, dass der eTutor zur Darstellung der Lerninhalte unterschiedliche Module verwendet und diese Module wurden mit der Einführung der Ablaufsteuerung nicht erweitert. Mit der Einführung der Ablaufsteuerung ist es nun möglich die Navigation durch die Lerneinheiten entsprechend den Vorgaben der Ablaufsteuerung anzupassen.

- **Individualisierte Ablaufsteuerung**

Im eTutor ist es möglich, für Kurse Aktivitätenebäume zu erstellen. Diese Aktivitätenebäume werden den einzelnen Studenten, welche an einem Kurs teilnehmen, zugeordnet. Es ist möglich den Aktivitätenbaum eines Studenten individuell zu verändern und an seine speziellen Bedürfnisse anzupassen. Das Wissen des Studenten wird in Form eines Benutzermodells im eTutor abgespeichert und kann von der Ablaufsteuerung mit Hilfe von Bedingungen von Regeln abgefragt werden. Diese Verknüpfung der Ablaufsteuerung mit dem Benutzermodell des eTutors ermöglicht individuelles Tutoring.

Dimensionen der Ablaufsteuerung	Ablaufsteuerung im eTutor
<b>Statische Ablaufsteuerung</b>	
<b>Aktive und Passive Ablaufsteuerung</b>	
<b>Adaptive Ablaufsteuerung</b>	
<b>Individualisierte Ablaufsteuerung</b>	

Tabelle 7: Kategorien der Ablaufsteuerung

Die im Kapitel 6 vorgeschlagenen Konzepte wurden für eine Ablaufsteuerung im eTutor herangezogen. Besonderheiten des eTutors wurden, z.B. durch den direkten Zugriff von Bedingungen auf die eTutor-Datenbank, berücksichtigt. Der Wissensstand von Studenten kann nun mit Hilfe der Regeln der Ablaufsteuerung, welche auf das Benutzermodell des eTutors zugreifen können, abgefragt werden. Des Weiteren verfügen Lehrende jetzt über die Möglichkeit die Lerneinheitssequenz von Studenten im eTutor zu modellieren und gegebenenfalls an die individuellen Bedürfnisse der Studenten anzupassen.

Weiters Bedeutung einer regelbasierten Ablaufsteuerung wurde aufgezeigt. Diese ermöglicht es, das Benutzermodell des eTutors mit der Ablaufsteuerung zu verknüpfen und hat somit eine große Bedeutung für die Individualisierung des Lernprozesses. Die Umsetzung der Ablaufsteuerung im eTutor wurde erklärt. Hierbei wurden wichtige Elemente wie Aktivitäten, Lernziele, Lerninhalte näher erläutert und die einzelnen Abläufe der Ablaufsteuerung beschrieben. Bei der Implementierung der Ablaufsteuerung wurde gezeigt, wie diese in das bestehende System integriert werden kann, und welche Auswirkungen dies hat. Der eTutor bietet nun die Möglichkeit die Lernsequenz von Kursen zu modellieren und an die Bedürfnisse und Kenntnisse von Studenten anzupassen.

## 10 Ausblick

Die offen gehaltene Architektur des eTutor-Systems und die hohe Flexibilität der Ablaufsteuerung bieten weit reichende Möglichkeiten zur Erweiterung des Systems. So ist z.B. eine Unterstützung der in dieser Arbeit nicht näher betrachteten Metriken zur Qualitätsbestimmung von Kursen denkbar, wodurch das bereits bestehende Kursangebot für die Ablaufsteuerung nutzbar machbar wäre. Außerdem wäre eine Erweiterung der Ablaufsteuerung hin zu einer vollständig adaptiven Ablaufsteuerung möglich. Dadurch wird auch die Anpassung von Lernressourcen selbst denkbar. Der eTutor kann zwar eine Vielzahl von unterschiedlichen Übungen aus dem Themengebiet Data & Knowledge Engineering darstellen, diese Übungen können jedoch nicht an die individuellen Bedürfnisse der Studenten angepasst werden.

Andere Aspekte, wie gemeinsames Lernen, sind ebenfalls zurzeit im eTutor nicht abgebildet und erfordern eine weiter reichende Anpassung des Systems. So wäre es denkbar den eTutor dahingehend zu erweitern, dass Studenten gemeinsam einen Kurs absolvieren können, um voneinander zu lernen.

Am Beispiel des eTutors wurde gezeigt, wie eine individualisierte Ablaufsteuerung in einem Intelligenten Tutoriellen System konzipiert und umgesetzt werden kann. Die gesteckten Ziele konnten alle mit der gewählten Umsetzung der Ablaufsteuerung im eTutor erreicht werden und bieten weiters noch die Möglichkeit das System leicht für zukünftige Anforderungen anzupassen.

## Anhang

Im folgenden Abschnitt wird näher auf den Kurs Datenmodellierung eingegangen. Wichtige Punkte beim Anlegen des Kurses im eTutor, sowie bei der Erstellung des Aktivitätenbaumes werden aufgelistet. Abschließend wird beispielhaft die Durchführung des Kurses durch einen Studenten gezeigt.

### (1) Kurs Datenmodellierung im eTutor

#### 1. Anlegen des Kurses

Name und Gültigkeitsbereich werden festgelegt.

Course Information	
Course name:	<input type="text" value="Datenmodellierung"/>
Course description:	<input type="text"/>
Course Code:	<input type="text"/>
Show from:	<input type="text" value="11-04-2008"/> (dd-MM-yyyy)
Show until:	<input type="text" value="30-03-2009"/> (dd-MM-yyyy)
<input type="button" value="Save"/>	

Abbildung 25: Einstellungen für den Kurs Datenmodellierung

#### 2. Aufgabenpool erstellen

Lerninhalte für die Ablaufsteuerung müssen zuvor in Form von Tasks eines Kurses angelegt werden, da der eTutor zurzeit nur Tasks darstellen kann.

	Begin	End	Status	Description	#Tasks	Max points	Sorting	
▶				<default unit>	1	0.0		
▶				Aufgabenpool	10	0.0	△ ▽	<a href="#">Edit</a>   <a href="#">Delete</a>

Abbildung 26: Aufgabenpool Kurs Datenmodellierung

## (2) Aktivitätenbaum des Kurses Datenmodellierung

Die nachfolgende Abbildung 2 zeigt den bereits in Abschnitt 1.2 beschriebenen Aktivitätenbaum. Des Weiteren werden nun jene Aktivitäten beschrieben (Tabelle 8), welchen Konstrukte oder Regeln zugeordnet wurden.

Aktivität Datenmodellierung	
<b>Konstrukt</b>	Auswahl-Konstrukt (Exit und Aktivierung sind möglich)
<b>Lernziele</b>	<i>[Information]</i> Status: erreicht / sichtbar Text: Willkommen beim Kurs Datenmodellierung.
Aktivität Konzeptueller Entwurf	
<b>Konstrukt</b>	Auswahl-Konstrukt (Exit und Aktivierung sind möglich)
<b>Lernziel</b>	<i>[kEntwurf]</i> Status: nicht erreicht / sichtbar Text: Lösen Sie die beiden Aufgaben.
<b>Rollup-Regel</b>	Bedingung: COMPLETED == 1 Verknüpfung: ALL Ziel: Konzeptueller Entwurf Aktion: abschließen  Bedingung: COMPLETED == 1 Verknüpfung: ALL Ziel: kEntwurf Aktion: erreicht

Aktivität Logischer Entwurf	
<b>Konstrukt</b>	Auswahl-Konstrukt (Exit und Aktivierung sind möglich) Zufalls-Konstrukt
<b>Lernziel</b>	[!Entwurf] Status: nicht erreicht / sichtbar Text: Sehen Sie sich jede Aufgabe an.
<b>Rollup-Regel</b>	Bedingung: ATTEMPTS >= 1 Verknüpfung: ALL Ziel: Logischer Entwurf Aktion: abschließen Bedingung: ATTEMPTS >= 1 Verknüpfung: ALL Ziel: !Entwurf Aktion: erreicht
Aktivität Physischer Entwurf	
<b>Konstrukt</b>	Sequenz-Konstrukt (Navigation in beide Richtungen)
<b>Vor-Auswahl-Regel</b>	Bedingung: ![kEntwurf] && ![!Entwurf] Aktion: Verstecken
<b>Rollup-Regel</b>	Bedingung: ACTIVE Verknüpfung: ANY Ziel: Physischer Entwurf Aktion: abschließen
Aktivität Physischer Entwurf – Aufgabe 1	
<b>Nach-Auswahl-Regel</b>	Bedingung: ATTEMPTS < 3 Aktion: wiederholen

Tabelle 8: Konstrukte und Regeln des Kurses Datenmodellierung

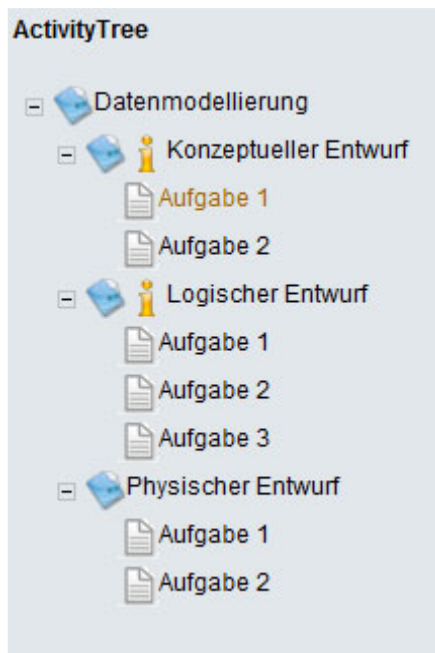


Abbildung 27: Aktivitätenbaum des Kurses Datenmodellierung

### (3) Durchführung des Kurses

Im folgenden Abschnitt erfolgt die Durchführung des Kurses Datenmodellierung durch einen Studenten. Es werden die Anfragen aufgelistet, welche das System, oder der Student an die Ablaufsteuerung stellt. Weiters werden für jede Anfrage wichtige Regeln oder Konstrukte aufgelistet, welche den Status des Aktivitätenbaumes verändern.

#### 1. Start-Anfrage (System)

- a. Auswertung der Konstrukte
- b. Auswertung der Vor-Auswahl-Regel der Aktivität *Physikalischer Entwurf*

**Der Lernprozess wurde gestartet →**

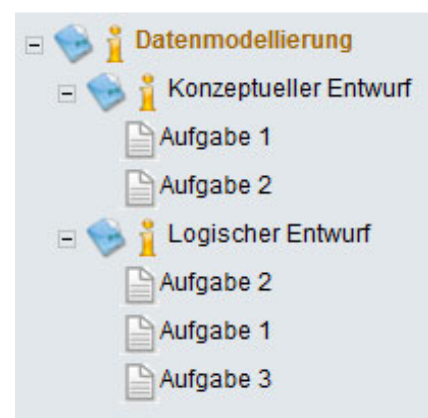


Abbildung 28: Kursstatus 1



2. Auswahl-Anfrage (Benutzer wählt die Aktivität *Konzeptueller Entwurf - Aufgabe 1*)
3. Abschließen-Anfrage (Benutzer schließt die Aktivität *Konzeptueller Entwurf - Aufgabe 1 ab*)

**Aufgabe 1 abgeschlossen →**

4. Auswahl-Anfrage (Benutzer wählt die Aktivität *Konzeptueller Entwurf - Aufgabe 2*)
5. Abschließen-Anfrage (Benutzer schließt die Aktivität *Konzeptueller Entwurf - Aufgabe 2 ab*)

- a. Ausführen der Rollup-Regel der Aktivität *Konzeptueller Entwurf*

**Aufgabe 2 abgeschlossen**

**Konzeptueller Entwurf abgeschlossen**

**Lernziel kEntwurf erreicht →**

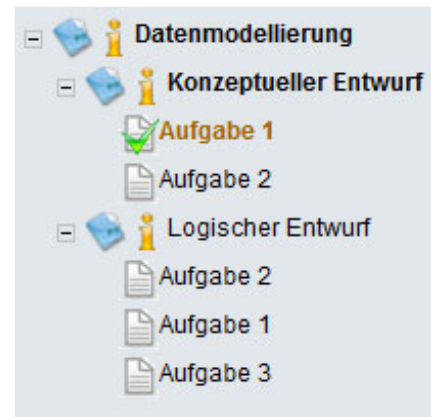
6. Auswahl-Anfrage (Benutzer wählt die Aktivität *Logischer Entwurf - Aufgabe 2*)
7. Auswahl-Anfrage (Benutzer wählt die Aktivität *Logischer Entwurf - Aufgabe 1*)
8. Auswahl-Anfrage (Benutzer wählt die Aktivität *Logischer Entwurf - Aufgabe 3*)

- a. Ausführen der Rollup-Regel der Aktivität *Logischer Entwurf*
- b. Ausführen der Vor-Auswahl-Regel der Aktivität *Physischer Entwurf*

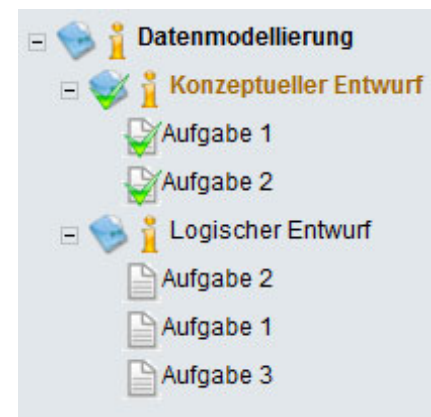
**Logischer Entwurf abgeschlossen**

**Lernziel lEntwurf erreicht**

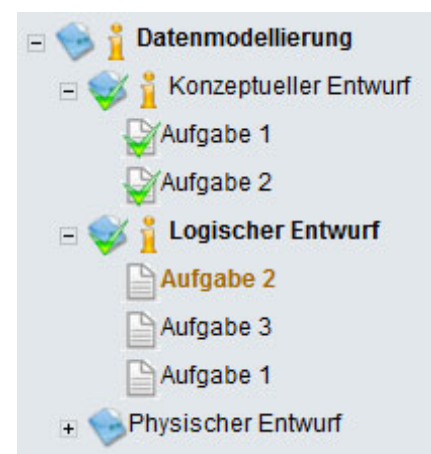
**Physischer Entwurf wird angezeigt →**



**Abbildung 29: Kursstatus 2**



**Abbildung 30: Kursstatus 3**



**Abbildung 31: Kursstatus 4**

9. Auswahl-Anfrage (Benutzer wählt die Aktivität *Physischer Entwurf*)

10. Weiter-Anfrage (Benutzer)

a. Ausführen der Rollup-Regel der Aktivität *Physischer Entwurf*

b. Ausführen der Rollup-Regel der Aktivität *Datenmodellierung*

**Physischer Entwurf abgeschlossen  
Datenmodellierung abgeschlossen→**

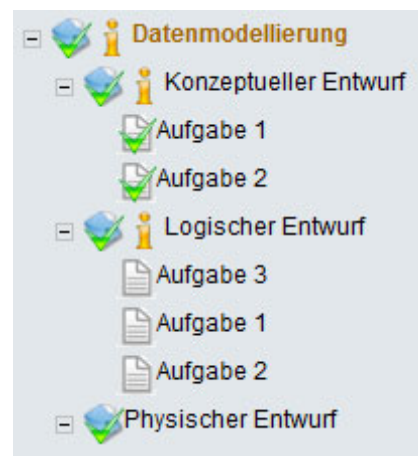


Abbildung 32: Kursstatus 5

## Literaturverzeichnis

- [AbDa03] Abdulah, N. A., Davis, H.: Is simple sequencing simple adaptive hypermedia?, In Conference on Hypertext and Hypermedia, Proceedings of the fourteenth ACM conference on Hypertext and hypermedia, 2003, S.172-173.
- [ADLa06] Advanced Distributed Learning (ADL), Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Overview, 2006.
- [ADLb06] Advanced Distributed Learning (ADL), Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Content Aggregation Model Version 1.0, 2006.
- [ADLc06] Advanced Distributed Learning (ADL), Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Run-Time Environment Version 1.0, 2006.
- [ADLd06] Advanced Distributed Learning (ADL), Sharable Content Object Reference Model (SCORM®) 2004 3rd Edition Sequencing and Navigation Version 1.0, 2006.
- [Bloo84] Bloom, B.: The 2 Sigma Problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher 13, 1984, S. 4-15.
- [Brus00] Brusilovsky, P.: Course Sequencing for Static Courses? Applying ITS Techniques in Large-Scale Web-based Education, 2000.
- [Brus01] Brusilovsky, P.: Adaptive Hypermedia, In User Modeling and User-Adapted Interaction 11, 2001, S.87-110.
- [Brus99] Brusilovsky, P.: Adaptive and Intelligent Technologies for Web-based Education. Künstliche Intelligenz , 4 , 1999, S.19-25.
- [BrVa03] Brusilovsky, P., and Vassileva, J.: Course Sequencing Techniques for Large-Scale Web-Based Education, In Int. J of Continuing Education and Lifelong Learning 13(1/2), 2003, S.75-94.
- [DeBr03] DeBra, P.: Pros and cons of adaptive hypermedia, In web-based education, Journal on CyberPsychology and Behavior, 3(1), 2003, S.71-77.

- [EKNi06]** Eichinger, C., Karlinger, M., Nitsche, G.: eTutor Modularchitektur, Institut für Wirtschaftsinformatik Data & Knowledge Engineering, Version 2, 2006.
- [HMSc04]** Helic, D., Maurer, H., Scerbakov, N.: Combining individual tutoring with automatic course sequencing in WBT systems, In International World Wide Web Conference, Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters, ACM Press, 2004, S. 456-457.
- [Hofe05]** Hofer, A.: E-Exercises in Data & Knowledge Engineering, Konzeption und Entwicklung eines intelligenten tutoriellen Systems, 2005.
- [Kazi04]** Kazi, S. A.: A Conceptual Framework for Web-based Intelligent Learning Environments using SCORM-2004, In Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04), 2004.
- [Kins96]** Kinshuk: "Computer Aided Learning for Entry Level Accountancy Students", PhD Thesis, De Montfort University, Leicester, England, 1996.
- [KrGa06]** Kravčik, M., Gašević, D.: Adaptive Hypermedia for the Semantic Web, In Conference on Hypertext and Hypermedia, Proceedings of the joint international workshop on Adaptivity, personalization & the semantic web, ACM Press, 2006, S. 3-10.
- [Kric05]** Krichen, J. P.: Dynamically adjusting to learner's competencies and styles in an online technology course, In Conference On Information Technology Education, Proceedings of the 6th conference on Information technology education, ACM Press, 2005, S. 149-154.
- [MePe02]** Millan, E., Pérez-De-La-Cruz, J. L.: A Bayesian Diagnostic Algorithm for Student Modeling and its Evaluation, In User Modeling and User-Adapted Interaction 12, 2002, S. 281-330.
- [RFDP06]** Rey-López, M., Fernández-Vilas, A., Díaz-Redendo, R., Pazos-Arias, J.: Providing SCORM with Adaptivity, In International World Wide Web Conference, Proceedings of the 15th international conference on World Wide Web, 2006, S.981-982.

- [SSIB03] IMS Global Learning Consortium, Inc., IMS Simple Sequencing Information and Behaviour Model Version 1.0, 2003.
- [SSXM03] IMS Global Learning Consortium, Inc., IMS Simple Sequencing XML Binding Version 1.0, 2003.
- [SSBP03] IMS Global Learning Consortium, Inc., IMS Simple Sequencing Best Practice and Implementation Guide Version 1.0, 2003.
- [Trav95] Travers, R. M. W.: Essentials of learning, Grundlagen des Lernens, Übers. von Ludwig Bittlinger, 1. Aufl., Oldenbourg, München [u.a.], 1975.

### Internetquellen

- [BeSh07] Bean Shell, 2007, <http://www.beanshell.org>, [27.09.2007]
- [GWT07] Google Web Toolkit, 2007, <http://groups.google.com/group/Google-Web-Toolkit>, [28.12.2007]
- [LTSO06] CEN - Learning Technologies Standards Observatory, 2006, [www.cen-ltso.net](http://www.cen-ltso.net), [04.01.2007]
- [ISO06] ISO/IEC JTC1 SC36 Home Page, 14.2.2006, [jtc1sc36.org](http://jtc1sc36.org), [04.01.2007]
- [LTSC06] Learning Technologies Standardization Committee (LTSC), [ieeeltsc.org](http://ieeeltsc.org), [04.01.2007]
- [CEN06] Europäisches Komitee für Normung, 22.12.2006, [www.cenorm.be](http://www.cenorm.be), [04.01.2007]
- [ADL07] Advanced Distributed Learning, 03.01.2007, [www.adlnet.org](http://www.adlnet.org), [04.01.2007]
- [AICC06] AICC - Aviation Industry CBT Committee, 06.09.2006, [www.aicc.org](http://www.aicc.org), [04.01.2007]
- [IMSG07] IMS Global Learning Consortium, Inc., 03.01.2007, [www.imsglobal.org](http://www.imsglobal.org), [04.01.2007]
- [EUN06] eun.org Portal - The gateway to education in Europe, [www.eun.org](http://www.eun.org), [04.01.2007]

- [**ARIA04**] ARIADNE - Foundation for the European Knowledge Pool, 2004, [www.ariadne-eu.org](http://www.ariadne-eu.org), [04.01.2007]
- [**EDNA07**] Education Network Australia - edna.edu.au, 02.01.2007, [www.edna.edu.au](http://www.edna.edu.au), [04.01.2007]
- [**DCMI07**] Dublin Core Metadata Initiative (DCMI), 05.01.2007, [dublincore.org](http://dublincore.org), [05.01.2007]

## Abbildungsverzeichnis

Abbildung 1: Architektur des eTutor-Systems. nach [Hofe05] .....	11
Abbildung 2: Verallgemeinertes Modell eines LMS [ADLa06, S.7] .....	29
Abbildung 3: SCORM® Books [ADLa06, S.11] .....	30
Abbildung 4: Beispiele für Assets [ADLb06, S.21].....	32
Abbildung 5: SCOs und LMS [ADLb06, S.22] .....	33
Abbildung 6: SCORM® Modell der Laufzeitumgebung [ADLc06, S.18] .....	36
Abbildung 7: Architektur eines Adaptiven Hypermedia Systems [DeBr03].....	39
Abbildung 8: Scope der Ablaufsteuerung .....	50
Abbildung 9: Beispiel einer Lernaktivität [ADLc06].....	53
Abbildung 10: Beispiel eines Aktivitätenbaumes .....	54
Abbildung 11: Auswahl-Konstrukt ohne Optionen .....	56
Abbildung 12: Auswahl-Konstrukt <i>Kein Exit</i> .....	57
Abbildung 13: Auswahl-Konstrukt <i>Keine Aktivierung</i> .....	58
Abbildung 14: Sequenz-Konstrukt .....	59
Abbildung 15: Teile eines Aktivitätenbaumes .....	70
Abbildung 16: Aktivitätenbaum Editor.....	73
Abbildung 17: Ausführung eines Aktivitätenbaumes .....	75
Abbildung 18: Schematische Darstellung des Benutzerinterfaces .....	78
Abbildung 19: Versuch an einer Aktivität.....	80
Abbildung 20: Zustandsdiagramm Anfragen/Aktivitäten.....	91
Abbildung 21: Beispiel einer Schleife .....	95
Abbildung 22: eTutor Systemarchitektur [EKNi06] .....	98
Abbildung 23: Architektur der Ablaufsteuerung .....	99
Abbildung 24: Kommunikation der Ablaufsteuerung.....	101
Abbildung 25: Einstellungen für den Kurs Datenmodellierung.....	109
Abbildung 26: Aufgabenpool Kurs Datenmodellierung .....	109
Abbildung 27: Aktivitätenbaum des Kurses Datenmodellierung .....	112
Abbildung 28: Kursstatus 1 .....	112
Abbildung 29: Kursstatus 2 .....	113
Abbildung 30: Kursstatus 3 .....	113
Abbildung 31: Kursstatus 4 .....	113
Abbildung 32: Kursstatus 5 .....	114

## Tabellenverzeichnis

Tabelle 1: Bestandteile eines Kurses .....	14
Tabelle 2: Vergleich der Ansätze.....	44
Tabelle 3: Schlüsselwörter in Bedingungen .....	86
Tabelle 4: Aktionen von Regeln.....	90
Tabelle 5: Ablaufsteuerung im eTutor .....	97
Tabelle 6: Packages der Ablaufsteuerung.....	100
Tabelle 7: Kategorien der Ablaufsteuerung .....	106
Tabelle 8: Konstrukte und Regeln des Kurses Datenmodellierung .....	111