

Feingranulare Module in PlaNet-ET

Andreas Bollin, Martin Hitz, Roland Mittermeir

Institut für Informatik-Systeme

Universität Klagenfurt

Abstract

Gute Medienunterstützung von Lehre ist eine beachtliche Investition. Darin besteht jedoch auch eine gewisse Gefahr, denn der Erstaufwand einer gut aufbereiteten, mediengestützten Lehrveranstaltung ist in der Regel so hoch, dass man sich scheut, diesen durch radikale Adaptierungen aufs Spiel zu setzen. Konsequenterweise nimmt man Aktualisierungen allenfalls in den Details vor. Verschiebungen in der Grundstruktur, die allenfalls nötig wären, um veränderte Zielgruppen besser zu erreichen, werden nur bei hinreichend großem Veränderungsdruck vorgenommen.

Das Projekt PlaNet-ET¹ ist eine Initiative zur Förderung des Einsatzes neuer Medien auf organisatorischer, mediendidaktischer und informationstechnischer Ebene. Es wendet sich mithin sehr stark an „Einsteiger“ in Mediendidaktik. Doch gerade bei Entwicklern mediendidaktischer Erstprodukte muss man davon ausgehen, dass ihre Produkte nach erstem Einsatz in der Lehre noch entsprechender Überarbeitung bedürfen, bevor Autoren wie Auditoren zufrieden sind. Daher ist es erforderlich, im Rahmen von PlaNet-ET der Aktualisierungsproblematik und dem Content- und Versionsmanagement elektronisch gestützter Lehrveranstaltungen besonderes Augenmerk zu widmen.

Durch ein mehrstufiges Gliederungskonzept, bei dem von Lehrzielen getragene explizite Vor- und Nachbedingungen angegeben werden, sollte es Autoren möglich sein, mit Hilfe von feingranularen Modulen Konfigurations- und Versionsmanagement ihrer Lehreinheiten systematisch zu betreiben. Darüber hinaus ermöglicht die konsequente Realisierung dieses Konzepts auch Lernenden, Stoff adressatenkonform (etwa auf der Basis eines Einstiegs-Assessments) abzurufen, ohne dabei in einer personalisierten Konserve eingengt zu werden.

1. Motivation

Ein hohes Ausmaß der Medienunterstützung von Lehre nimmt ihr in der Regel etwas an Spontanität und gefährdet unter Umständen ihre Aktualität. Dies erscheint bedauerlich und, relativ zur Zielsetzung

¹ PlaNet-ET (*Platform and Network for Educational Technology*) ist ein Kooperationsprojekt zwischen den Universitäten Graz, Innsbruck und Klagenfurt sowie dem Management Center Innsbruck und der Österreichischen Gesellschaft für Hochschuldidaktik. Das Projekt wird vom Bundesministerium für Bildung, Wissenschaft und Kultur im Rahmen der Initiative „Neue Medien für die Lehre an Universitäten und Fachhochschulen“ unter Projektnummer P 88 gefördert.

einer Verbesserung der Lehre durch Medienunterstützung, widersinnig. Berücksichtigt man jedoch den hohen Aufwand, der in die Vorbereitung einer mediengestützten Lehrveranstaltung investiert wurde, werden diese Negativ-Aspekte verständlich. Ziel einer Lehr-/Lernumgebung sollte daher sein, den in Medienunterstützung investierten Grundaufwand so zu sichern, dass Lehrende unterstützt werden, ihre Materialien umzustellen, ohne dabei zu viel des in eine frühere Version der Materialien investierten Aufwands zu verlieren.

Diese Argumentation gilt insbesondere dann, wenn man, wie im Projekt PlaNet-ET, Lehrende motivieren möchte, ihre Lehre durch gezielten Einsatz neuer Medien zu verbessern. Gerade bei Angehörigen der Zielgruppe *Mediendidaktik-Einsteiger* ist davon auszugehen, dass noch sehr viel Experimentierfreudigkeit und damit auch eine Fülle von Veränderungswünschen zu einer neu entwickelten Lehreinheit vorliegen. Darüber hinaus wird gerade die „Erstaufgabe“ einer mediengestützten Lehrveranstaltung auch objektiv gesehen noch hohes Veränderungs- und Verbesserungspotential aufweisen. Deshalb widmet sich dieses Projekt unter anderem neben der Entwicklung mediendidaktischer und organisatorischer Begleitmaßnahmen auch der Unterstützung von Adaptierbarkeit und Versionsmanagement von Lehreinheiten.

Dennoch darf die Wartungsproblematik von Lehreinheiten nicht lediglich unter der Perspektive von Anfängerproblemen gesehen werden. Die Frage des Versionsmanagements ergibt sich ebenso, wenn man berücksichtigt, dass man ein Stoffgebiet unterschiedlichen Zielgruppen näher bringen möchte und dabei aus einem gemeinsamen Repositorium von Grundmaterialien Lehreinheiten unterschiedlichen Tiefgangs bzw. Lehreinheiten mit unterschiedlichen Schwerpunktsetzungen entwickeln möchte. Da selbstverständlich auch diese Lehreinheiten für jede der betroffenen Zielgruppen, möglicherweise zu unterschiedlichen Zeiten, weiterentwickelt werden müssen, wird die Problematik der Aktualisierung mediengestützter Lehre bei Berücksichtigung unterschiedlicher Zielgruppen zu einem mehrdimensionalen Versionsverwaltungsproblem. Dies legt es nahe, die Wartungsproblematik von Lehreinheiten aus der Perspektive von Software-Konfigurationsmanagement zu betrachten.

Allerdings sind die Konzepte des Software-Konfigurationsmanagements nur als Zwischenziel anzusehen. Dies deshalb, weil Software, vergleichbar zu intramural vorgetragener Präsenzlehre, produzentenkonfiguriert ist. Bei computergestützter Lehre muss dies nicht zutreffen. Hier wäre es denkbar, dass Lernende – beispielsweise durch Selbsteinstufung oder durch freiwillige Einstiegs-Tests – dem System ihren spezifischen Lernbedarf bekannt geben und das System auf dieser Grundlage eine personenspezifische Zusammenstellung von Lehrmaterialien vorschlägt. Im Großen finden wir dieses Konzept unter Schlagworten wie zielgruppenkonforme Lehre. Als individuelle Zusammenstellung des Lehrprogramms ist solches jedoch derzeit nur im persönlichen Einzelunterricht möglich.

Ausgehend von diesen Zielsetzungen wollen wir in der Folge ein Modell vorstellen, das die eben gesteckten Ziele zumindest weitgehend erreicht. Wir diskutieren dies vorerst am Beispiel der Weiterentwicklung klassischer Präsenzlehrveranstaltungen. In Kapitel 3 zeigen wir, wie die für die Aktualisierungsproblematik entwickelten Konzepte für computergestütztes Selbststudium angewendet werden können, ohne das Grundproblem, dem sich dieser Aufsatz widmet, zu erschweren oder den Lösungsansatz zu kippen. Kapitel 4 zeigt, wie diese Überlegungen auf der Basis der gewählten E-Learning Plattform umgesetzt werden können.

2. Evolution klassischer mediengestützter Lehrveranstaltungen

2.1. Architektur einer Lehrveranstaltung

In diesem Kapitel widmen wir uns der Frage, wie eine Aktualisierung mediengestützten Lehrmaterials für unterschiedliche Zielgruppen mit erträglichem Aufwand realisiert werden kann, sodass die daraus resultierenden Einzelkurse auch über Wartungszyklen hinweg in sich konsistent bleiben. Dabei ist Wartung auf zwei Ebenen zu unterscheiden:

- Auf der *Mikroebene* sind in der Regel vor Abhaltung einer neuen Instanz einer Lehrveranstaltung Aktualisierungen von Details vorzunehmen. Beispiele für Aktualisierungen dieses Typs sind der Einbau neuer wissenschaftlicher Erkenntnisse ebenso wie die simple Korrektur von Fehlern, die während der Abhaltung des Kurses (oder in der Vorbereitung der Neuauflage) zum Vorschein kamen.
- Auf der *Makroebene* geht es nicht so sehr um eine Veränderung der Inhalte, als vielmehr um eine Veränderung des Aufbaus des Kurses. Änderungen dieser Art werden in der Regel auf didaktischen Überlegungen fußen. Sie können jedoch auch daraus entstehen, dass man eine bestehende, für gemischtes Auditorium angebotene Lehrveranstaltung nunmehr fokussiert für unterschiedliche Zielgruppen anbieten möchte.

Während Änderungen auf Mikroebene durchaus allgemein geübt, guter Standard sind, stellen sich Änderungen auf Makroebene sehr oft die in der Einleitung erwähnten Investitionssicherungs-Überlegungen entgegen. Allerdings ist zu beachten, dass durch eine Vielzahl von Mikroänderungen im Regelfall die Makrostruktur erodiert oder an Zweckmäßigkeit leidet. Ebenso werden Makroänderungen in der Regel auch einige Änderungen im Mikrobereich nach sich ziehen. Mikro- und Makroebene sind daher nicht vollständig voneinander unabhängig. Durch guten Entwurf einer Lehrveranstaltung lässt sich allerdings ein gewisses Maß an Unabhängigkeit erzielen.

In monolithisch konzipierten Kursmaterialien lassen sich Umstrukturierungen größeren Stils ebenso schwer durchführen wie in einem großen Software-System. Um das Problem dennoch ein wenig in den Griff zu bekommen, nehmen wir bei unseren Überlegungen Anleihe bei Lösungsvorschlägen, die im Bereich der systematischen Pflege von Software-Systemen entstanden sind^{2,3,4,5}. Schließlich sind Kursmaterialien und Computer-Software ja aufgrund der selben zu vermittelnden Grundsubstanz, der Übertragung von Information, verwandt. Lediglich unter teleologischen und notationellen Aspekten bestehen wesentliche Unterschiede.

² Berlack R.H.: *Configuration Management*; in: Marciniak J.J. (ed.): *Encyclopedia of Software Engineering*; Vol. I, John Wiley & Sons, Inc., 1994, pp. 179 – 207.

³ Tichy W.F. (ed.): *Configuration Management*; John Wiley & Sons, 1994.

⁴ Conradi R., Westfechtel B.: *Version Models for Software Configuration Management*; ACM Computing Surveys, Vol. 30 (2), June 1998, pp. 232 – 282.

⁵ Estublier J.: *Software Configuration Management: A Roadmap*; in: Finkelstein A. (ed.): *The Future of Software Engineering 2000*; ACM press, 2000, pp. 279 – 289.

2.2. Veränderungsproblematik bei Präsenzlehre

Beginnen wir unsere Überlegungen vorerst bei einem Kurs für eine in ihren Ausgangsvoraussetzungen stabile, jedoch in sich durchaus nicht besonders homogene Zielgruppe. Eine typische universitäre Vorlesung im Grundstudium mag als Prototyp für eine derartige Lehrveranstaltung dienen. Diese wird im Jahres- oder Semesterrhythmus angeboten und (hoffentlich) laufend auf Mikroebene aktualisiert. Daraus ergibt sich im Zeitablauf eine Folge von Lehrveranstaltungsinstanzen, die sich jeweils in Details unterscheiden. Dies bedeutet, dass eine neue Instanz nicht einfach die Kopie der jeweils vorausgehenden Instanz ist. Sie unterscheidet sich somit von dieser und ist mithin eine neue *Version* einer Lehrveranstaltung, die jedoch unter gleichem Titel in Studienplänen oder Vorlesungsverzeichnissen aufscheint.

Nehmen wir der Einfachheit halber an, die einzelnen Detailänderungen seien so gering, dass sich selbst aus der zeitlichen Folge dieser Änderungen noch kein nennenswerter Änderungsbedarf auf Makroebene ergibt. Die einzelnen Kursinstanzen können mithin als eine Folge von aufeinander aufbauenden *Revisionen* oder *Releases* aufgefasst werden. Korrekturbedarf, der jeweils beim Abhalten der Lehrveranstaltung festgestellt wird, kann in dieser Situation gemeinsam mit fachlichen Aktualisierungen in die jeweils nächste Release eingearbeitet werden. Das Entstehen von Zwischenrevisionen wäre denkbar, soll hier jedoch nicht weiter verfolgt werden.

Sobald jedoch, etwa durch Einladung, die Vorlesung auch an einer Partneruniversität anzubieten, zwei unterschiedliche Zielgruppen bedient werden sollen, werden nicht bloß Korrekturen und Aktualisierungen am Stoff vorzunehmen sein. Die Änderungen, die nötig sind, um die Lehrveranstaltung angemessen in das Gesamtangebot der anderen Universität einbinden zu können, wird sich auch strukturell auswirken. So werden neue Kapitel einzubauen sein, etwa um fehlende Vorkenntnisse der anderen Studentenspopulation auszugleichen. Andererseits werden einige Teile zu kürzen oder überhaupt auszulassen sein, da der in diesen vermittelte Stoff der neuen Zielgruppe aus anderen Lehrveranstaltungen bereits bekannt ist (oder ggf. für diese Zielgruppe nicht relevant ist). Obzwar also die Grundthematik der Lehrveranstaltungen erhalten bleibt, sind die Änderungen so gravierend, dass sie sich wenigstens auf Detailebene in der Definition der Lehrziele niederschlagen. Es entsteht mithin eine neue *Variante* des Kurses, die, wenn aus der Einladung eine Einladungsserie wird – oder wenn es sich um eine Aufspaltung der Zielgruppe an der Mutteruniversität handelt – künftig ebenso zu aktualisieren ist wie die Stammversion..

Denken wir darüber hinaus auch noch an die Verwendung der Kursmaterialien in außeruniversitärer Erwachsenenbildung, für Überblickslehreinheiten vor Schülern oder als Auffrischungseinheit in einer auf besagter Grundvorlesung aufbauenden Lehrveranstaltung im Hauptstudium, so sehen wir weitere Chancen für die Mehrfachverwendung der Grundinvestition in eine gut aufbereitete Lehrveranstaltung. Wir erkennen jedoch auch, dass hier ein zweidimensionales Gestaltungsproblem vorliegt (siehe Abb. 1). Der Stoff selbst entwickelt sich in der Zeitdimension weiter (Revisionen). Er entwickelt sich allerdings auch in der Zielgruppendifferenzdimension (Varianten) weiter. Die Änderungen, die zu einer neuen Revision führen, sind dabei von den Änderungen, die zu einer neuen Variante führen im wesentlichen unabhängig. Allerdings sollen/müssen die jeweils neuen Releases einer Variante inhaltlich mit den neuen Revisionen der Basisvariante konsistent gehalten werden. Dies sollte möglich sein, ohne dass Aktualisierungsschritte in jeder Version explizit (erneut) eingebaut werden müssen.

Insbesondere dann, wenn die neue Variante nicht der Entwicklung einer (einmaligen) Gastvorlesung diene, sondern einer speziellen Zielgruppe der eigenen Universität gewidmet ist, werden sich die

beiden Kursversionen über die Zeit zusehends auseinander entwickeln. Der in Abb. 1 gezeigte zweidimensionale Raum birgt somit die Gefahr einer zusehenden Verästelung in sich. Er sollte daher nicht dem Wildwuchs preisgegeben werden, sondern die darin entstehenden Verzweigungen sollten gezielt entworfen, bewusst gepflegt, und letztlich an geeigneten Stellen wieder zu einer gemeinsamen *Baseline* integriert werden. Aus einer solchen Baseline wird die Grundkonzeption der hier betrachteten Lehrveranstaltung in aktualisierter Form ersichtlich. Sie dient damit der Konfigurationsbereinigung und als Basis für neuerliche Variantenbildung.

Zur effizienten Weiterentwicklung von Lehrmaterialien innerhalb eines solchen sich zusehends stärker verzweigenden Gesamttraums wird letztlich eine Form des Content-Managements erforderlich sei, die den Konzepten klassischen Software-Konfigurationsmanagements ähnelt. Dies bedingt allerdings auch, dass Lehrveranstaltungen, die sich durch sukzessive Weiterentwicklung immer weiter verästelten, nicht auf Lehrveranstaltungs-niveau gewartet werden sollten, sondern dass diese Aktualisierungen besser auf kleineren, relativ unabhängigen Einheiten vorgenommen werden sollten. Effizientes Konfigurationsmanagement ruht also auf Prinzipien der Modularisierung. Diese Überlegungen werden im folgenden Abschnitt näher ausgeführt.

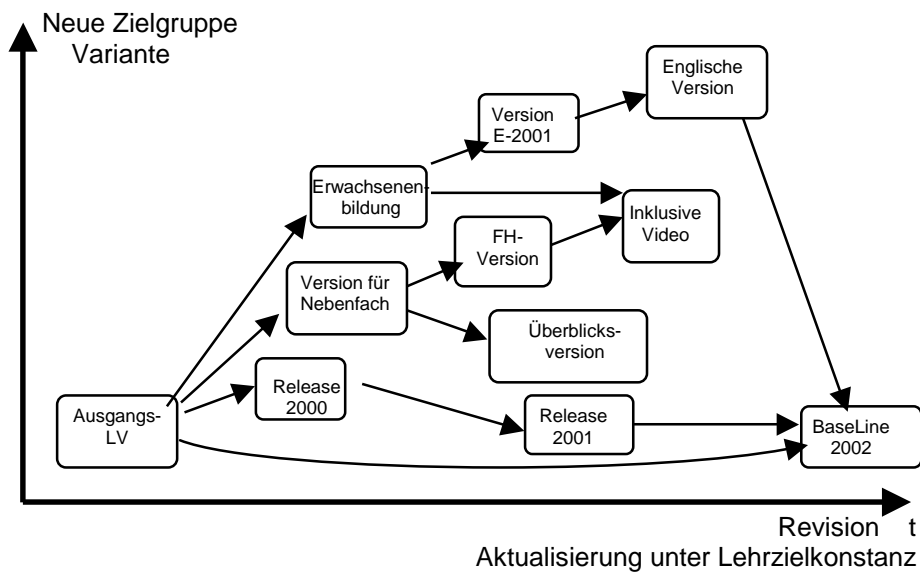


Abb. 1
Strukturierung des Modifikations- bzw. Versionierungsraums

2.3. Software-Engineering Analogie – Architektur des Lösungskonzepts

Greifen wir zum Management dieses zweidimensionalen Modifikationsraums auf Überlegungen modernen Software-Engineerings zurück, hilft neben Konfigurationsmanagement wohl auch das Konzept des *Product-Line-Engineerings*^{6,7}. Wir können die für eine spezifische Zielgruppe entwickelte Lehrveranstaltung als Instanz eines Produkts aus einer Produktlinie auffassen. Die Produktlinie selbst wird aus einzelnen wiederverwendbaren Bausteinen so aufgebaut, dass sowohl die Bedürfnisse unterschiedlicher Zielgruppen berücksichtigt werden, als auch der Produktionsprozess selbst effizient auf der Basis der Entwicklung wiederverwendbarer, bzw. leicht adaptierbarer Komponenten erfolgen kann. Dies ermöglicht, Wartung auf Baustein-Ebene, also Aktualisierung auf der Mikroebene der Lehrveranstaltung, so auszuführen, dass der jeweils aktualisierte Baustein unmittelbar in all jenen Produkten (Lehrveranstaltungen) aktualisiert wird, in denen er eingebaut ist.

Selbstverständlich kann es sein, dass eine bestimmte Änderung, etwa eine Vertiefung des Materials, nur in ein Produkt oder nur in einige ausgewählte Produkte eingebaut werden soll. Konsequenterweise sind die im letzten Abschnitt angeführten Konzepte von Revisionen und Varianten daher nicht nur auf Produktebene sondern auch auf Komponentenebene anzuwenden. Dies führt dazu, dass wir die einzelnen Kurse vorerst auf zwei unterschiedlichen Modellebenen betrachten wollen:

- *Kursebene*: Legt die Struktur des Kurses bzw. der Lehrveranstaltung fest. Hier finden mithin, ausgehend von den primären Lehrzielen des Kurses die wesentlichsten Architekturentscheidungen statt. Vergleichbar zur Architektur von Software-Systemen wird im Rahmen einer top-down-Entwicklung einer Lehrveranstaltung durch Gliederung des zu vermittelnden Stoffs in Module auf dieser Ebene festgelegt, welche Teillehrziele von den einzelnen Modulen zu erfüllen sind, und in welcher Abfolge die einzelnen Modulen erarbeitet werden sollen, damit die Lehrveranstaltung ihr Gesamtziel erreicht.
- *Modulebene*: Als Modul sehen wir jene Lehreinheit an, die aus Entwicklersicht eine in sich abgeschlossene Einheit von Lehrstoff (innerhalb einer Lehrveranstaltung) behandelt. Daher ist davon auszugehen, dass inhaltliche Aktualisierungen eines Moduls unabhängig vom Rest des Gesamtkurses durchgeführt werden können.

Analog zum Architekturbegriff im Software-Engineering⁸ ist das Architekturkonzept medienunterstützter Lehrveranstaltungen nicht nur als konzeptives Element zu verstehen. Es enthält vielmehr neben der Festlegung der „tragenden Wände“ vor allem klare Schnittstellenbeschreibungen für die einzelnen Lehrmodule, durch die diese Architektur im Sinne einer Fertighausarchitektur aufgefüllt werden muss. Weiters sollte die Kursarchitektur auch jene Begründungszusammenhänge enthalten, die bei Perry und Wolf „rationale“ genannt werden. Sie dienen dazu, Querverbindungen zwischen Inhalten auf Modulebene zu erfassen, die über die reine Festlegung von Lehrzielen einzelner Module hinausgehen und die vor Kombination entsprechender Inhalte zu berücksichtigen sind.

⁶ DeBaud J.M., Schmid K.: *A systematic approach to derive the scope of software product lines*: in : Proceedings 21st International Conference on Software Engineering, May, 1999, IEEE-CS press, pp. 34 – 44.

⁷ Northrop L.M.: *Reuse that pays*; in Proceedings 23rd International Conference on Software Engineering, May, 2001, IEEE-CS press, pp. 667.

⁸ Perry D.E., Wolf A.L.: *Foundations for the Study of Software Architectures*; ACM SigSoft Software Engineering Notes, Vol. 17 (4), Oct. 1992, pp. 40 – 52.

Allerdings erkennt man bei dieser Trennung sehr bald, dass die Gliederung in Modul- und Kursebene noch unzureichend ist. Wie sieht es etwa mit Übungs- oder Prüfungsbeispielen aus, die bestimmten Stoffeinheiten zuzuordnen sind, damit diese als abgeschlossen gelten dürfen, die jedoch einerseits periodischen Variationen unterliegen, andererseits innerhalb einer Lehrveranstaltungsinstanz konsistent gehalten werden sollten? Die Modulebene ist mithin nicht wirklich elementar, sondern Module bestehen aus feingranularen Einheiten, wir wollen sie *Lernelemente* oder *Chunks* nennen. Diese Chunks sind nun Inhaltselemente, die zu Modulen zusammengeschlossen werden. In Vorgriff auf Realisierungsvorschläge in einer E-Learning-Umgebung wollen wir diese Verbindungen von Chunks zu Modulen daher auch als *Learnstrings* bezeichnen.

Die dreistufige Gliederung mag anfänglich komplex wirken. Tatsächlich ist sie jedoch Autoren etwa in Form der Gliederung eines Textes in Kapitel und Unterkapitel durchaus geläufig.

2.4. Trennung in Inhalts- und Beschreibungsebene

Durch die Gliederung des Lehrstoffs in Module und weitere Gliederung in Chunks wird klar, dass die Beschreibung der einzelnen feingranularen Module in unterschiedlichen konzeptionellen Dimensionen zu erfolgen hat. Dies ist nötig, um im Wartungsfall zu erkennen, auf welche Vorbedingungen ein bestimmtes Lehrelement aufbaut, beziehungsweise welche Nachbedingungen durch Erarbeitung des entsprechenden Lehrelements definiert werden. Dies rechtfertigt eine Trennung in Inhalts- und Beschreibungsebene, wie sie schematisch in Abb. 2 dargestellt ist.

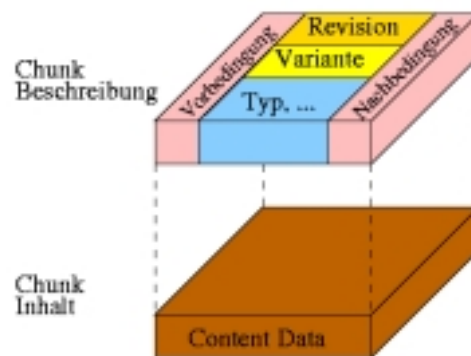


Abb. 2
Lehrelement (Chunk) mit zugehöriger Beschreibung

Durch die Beschreibungsebene sollte der Inhalt einer feingranularen Lehreinheit vollständig umschrieben werden. Dabei wollen wir uns grundsätzlich an entsprechenden Standardisierungsvorschlägen⁹ orientieren. Diese sind allerdings um spezifische, inhaltsbezogene Informationen zu erweitern. Zur externen Sichtbarmachung von Lehrmaterialien wird auf Kursebene die Verwendung eines auf breiter Basis eingesetzten Klassifikationssystems, auf Modulebene demgegenüber der Einsatz eines feingranulareren, fachlich abgestimmten Klassifikationssystems und auf Elementarebene schließlich eine innerhalb der jeweiligen Autorengemeinschaft konsistent gehaltene Beschlagwortung vorgesehen¹⁰.

Zusätzlich zu diesen formalisierten bzw. formatisierten Metadaten empfehlen wir:

- Auf der *Inhaltsebene*: Eine freisprachliche Umschreibung des zu vermittelnden Stoffs.
- Auf der *Zielebene*: Explizite Definition der Lehrziele für dieses Modul bzw. Chunk und Angabe der Funktion, wie diese Lehrziele erreicht werden sollen (analytisch, konstruktiv, ...).
- Auf der *Kontextebene*: Beschreibung des zielgruppenspezifischen Kontexts, innerhalb dessen die Lehrziele des Moduls zu interpretieren sind.
- Explizite *Annotation von Verwandtschaftsbeziehungen* zwischen Modulen bzw. zwischen Chunks. Hierher zählen am bisherigen Diskussionsstand vor allem Alternativen-Beziehungen zwischen Demonstrations- oder Prüfungsbeispielen zu einer bestimmten Stoffauswahl. Zwischen diesen bestehen einerseits „Alternativenbeziehungen“, andererseits lassen sich, wenn die Lehrveranstaltung für unterschiedliche Zielgruppen konzipiert wurde – oder wenn die Beispiele von einer innerhalb einer Lehrveranstaltungsinstanz gemeinsamen, aber über Instanzen variierenden Grundthematik getragen sind – Beispiele auch „longitudinal“ etwa zu gemeinsamen Anwendungs-Clustern verbinden (siehe Abb. 5).

Die inhaltsbezogenen Informationen sind noch durch „buchhalterische“ Informationen auf Modulebene, wie etwa Revisionsidentifikation und Variantenkennung des Moduls, Aktualisierungsdatum zu ergänzen. Allerdings führt das Aufbrechen einer Lehrveranstaltung in Module dazu, dass die Unterscheidung in Revisionen und Varianten an Bedeutung verliert. Auf feingranularer Ebene wird man letztlich entweder eine bestehende Version überarbeiten und als neue zusätzliche Variante ablegen oder als Revision so ablegen, dass die vorhergehende Version dadurch überdeckt wird.

Freilich muss zu dieser Trennung in Inhalts- und Metaebene angemerkt werden, dass das Verfassen solcher Beschreibungen für die Ersteller mit Aufwand verbunden ist und dass ein gerüttelt Maß an Disziplin erforderlich ist, dies auch umzusetzen. Der Nutzen aus diesen Investitionen mag Anfängern nicht unmittelbar ersichtlich sein. Dieser Nutzen stellt sich jedoch im Zuge der Weiterentwicklung der Lehrveranstaltung ein und findet insbesondere durch Übertragung des Lehrmaterials in eine Selbstlernumgebung deutliche Rechtfertigung.

⁹ IEEE Learning Technology Standards Committee (LTSC): *Learning Object Metadata*; Draft Document v3.6, 5. Sept. 1999, URL: //ltsc.ieee.org/doc/wg12/LOM3.6.html

¹⁰ Steinacher M.: *Mehrstufige Indizierung von Lehrmaterialien*; PlaNet-ET interner Bericht, Klagenfurt 2002.

2.5. Wiederverwendbare Bausteine

Durch die Trennung in Inhalts- und Beschreibungsebene lässt sich die Beschreibungsebene im Sinne einer top-down Entwicklung eines Kurskonzepts als Spezifikation für nötige Kursbausteine, im Sinne einer bottom-up Sichtweise aber auch als Zusicherung über vermittelte Inhalte einer Lehreinheit oder eines Moduls auffassen. Abb. 3 zeigt diese Dualität schematisch.

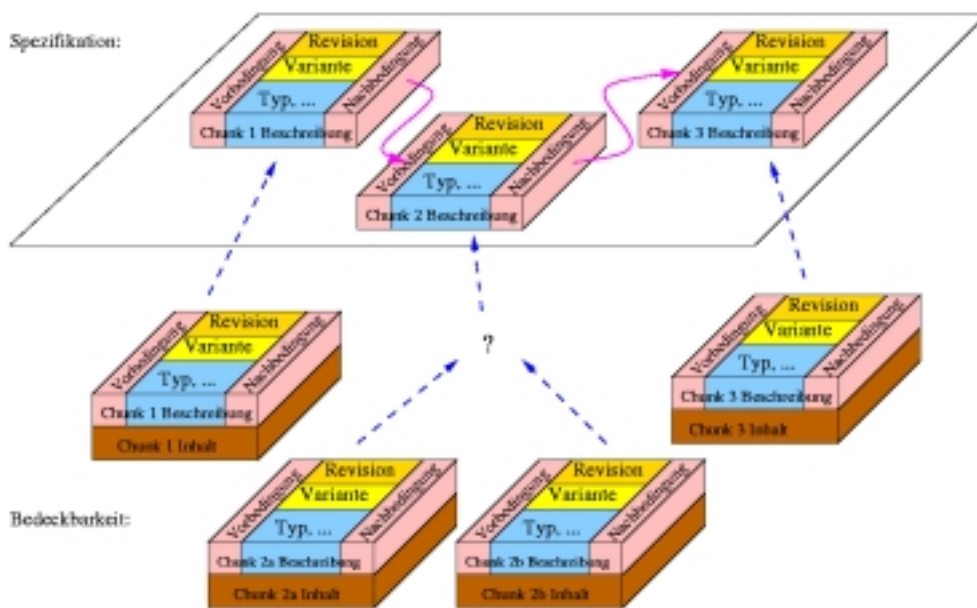


Abb. 3
Trennung in Spezifikation und Zusicherung

Dies ermöglicht es, wie in Abb. 3 dargestellt, nicht nur Chunks zu Lernstrings und damit zu Lernmodulen zu verbinden, sondern Chunks auch als abgeschlossene, wiederverwendbare Einheiten aufzufassen. Durch die auf Metadatenebene beschriebenen Inhalte und Querbeziehungen werden Kursautoren jedoch unterstützt, nur konsistente Einheiten aufzubauen.

Aus obigen Überlegungen ergibt sich eine Gesamtstruktur für Lehrveranstaltungen, die aus einer Kursarchitektur besteht, die auf Architekturebene unterschiedliche Versionen unterstützt und damit eine zielgruppenspezifische Makrostruktur bietet. Die konkreten Inhalte selbst werden jedoch in feingranularen Modulen dargelegt, die aufgrund ihrer Beschreibungen in die auf Architekturebene vorgesehenen Schnittstellen passen.

Die Verbindung zwischen Modul- und Architekturebene ist innerhalb dieses Konzepts nicht fix vorgegeben, sondern kann über die Schnittstellen und über „rationale“-Beschreibungen auf Architekturebene einerseits, sowie durch die strukturierte Beschreibung der einzelnen Module andererseits, flexibel gestaltet werden. Dabei gehen wir in diesem Abschnitt noch davon aus, dass die Verbindung

zwischen Architekturebene und den in diese Architektur einzufügenden Modulen durch die die Lehrveranstaltung entwickelnde Person geleistet wird. Dadurch kann der Formalisierungsgrad der Schnittstellen- und Modulbeschreibungen relativ gering gehalten werden. Im nächsten Abschnitt wollen wir die Voraussetzung der Autorenintervention fallen lassen. Damit erhöht sich allerdings der von Metadaten geforderte Formalisierungsgrad.

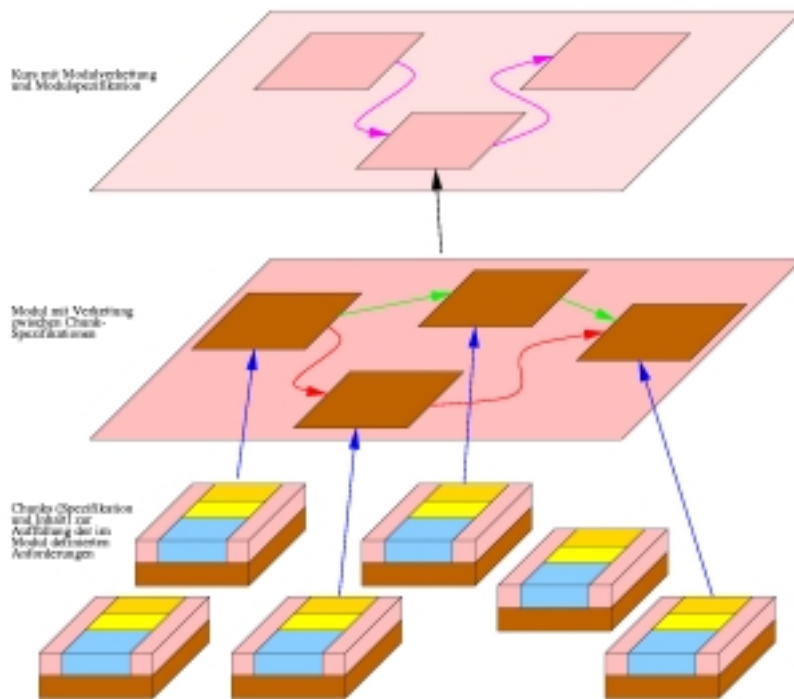


Abb. 4
Modul- und Architekturebene, verbunden durch Beschreibungsstruktur

Bevor wir uns dieser Thematik zuwenden, soll jedoch gezeigt werden, dass innerhalb des eben vorgestellten Konzepts Module auf Kursebene durch eine vom Kursautor vorgegebene Verbindungsstruktur festgelegt sind (*hard links*) und auch die Chunks mittels solcher *hard links* innerhalb des Moduls verbunden sind. Zwischen den Chunks zeigt Abb. 4 jedoch noch keine weiteren Verbindungen. Dies ist aus Sicht einer einzelnen Kursinstanz auch nicht erforderlich. Betrachtet man jedoch die mit der Evolutionsproblematik verbundenen Aspekte eines gezielten Versions- (und insbesondere Varianten-) Managements, zeigt sich die Notwendigkeit, dass auch innerhalb des Pools von Chunks Strukturierungsmöglichkeiten vorgesehen werden. Dies wird in Abb. 5 gezeigt.

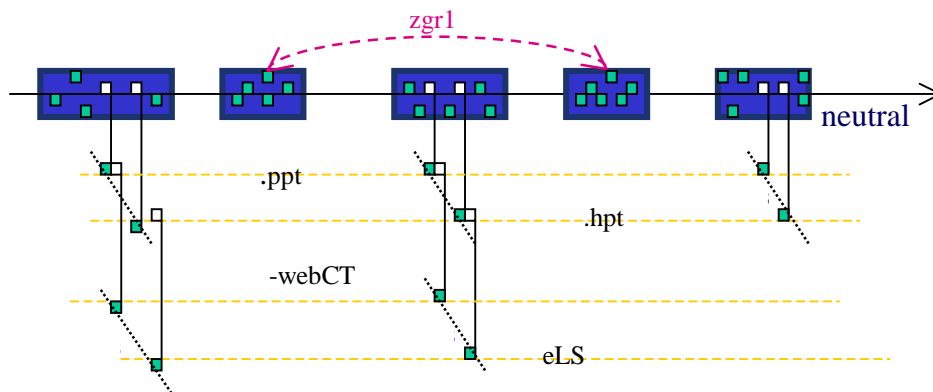


Abb. 5
Variantenstruktur innerhalb einer Kursfamilie

Man könnte etwa davon ausgehen, dass die beiden in Abb. 4 nicht verwendeten Chunks Alternativen zu tatsächlich verwendeten Chunks darstellen. Diese könnten etwa thematische Alternativen darstellen, wie sie in Abb. 5 näher dargestellt sind. Abb. 5 lässt sich etwa so interpretieren, dass eine Lehreinheit zum Thema Content-Management grundsätzlich aus einer Folge von fünf „präsentationsneutralen“ Modulen besteht. Diese bestehen jeweils aus einer gewissen Anzahl von „präsentationsunabhängigen“ und jeweils einigen „präsentationsspezifischen“ Chunks. Letztere gliedern sich in Chunks die sich mit dem Inhaltsmanagement für Folienpräsentationen und solche, die sich mit der Verwaltung von Hypertext auf E-Learning Plattformen, beispielsweise auf webCT oder auf eLS beschäftigen. Um hier entsprechende Autorenunterstützung zu haben, sind diese thematisch verwandten Chunks durch sogenannte *R-Links* verbunden. Ebenso könnte man R-Links zur Verwaltung von zielgruppenspezifischen Inhalten verwenden.

3. Ausweitung des Konzepts für das Selbststudium

In der bisherigen Darstellung sind wir davon ausgegangen, dass die betrachtete Lehrveranstaltung in klassischer Weise vorgetragen wird. Dies bedeutet, dass die Architekturebene im Wesentlichen eine lineare Sequenz von Modulen vorsieht. Bei Selbststudium erscheint dies zu einschränkend.

Einzelne Studierende bevorzugen wohl eine systematisch von den Grundlagen des Faches ausgehende Stoffvermittlung, andere wollen eher eine beispielegetragene Vermittlungsstrategie. Einige haben hohen Übungsbedarf, während andere aufgrund unterschiedlicher Vorbedingungen oder unterschiedlicher Auffassungsgabe einige Einheiten eher auf der „Überholspur“ durcharbeiten mögen. Möglicherweise wollen diese aber an bestimmten, gerade für sie besonders interessanten Stellen noch Vertiefungsmaterial konsultieren. Klassische Präsenzlehre kann solche Inhomogenitäten nur bedingt berücksichtigen. Lehrveranstaltungsleiter können allenfalls durch spezifische didaktische Tricks einiges an Inhomogenität auszugleichen.

Bei computergestütztem Selbststudium scheint dies nicht mehr nötig. Da jede einzelne Person die Chance des Dialogs hat, kann sie auch Möglichkeiten des Mediums zur Individualisierung von Lehrinhalten nutzen, sofern diese Individualisierungsmöglichkeiten durch die Kurskonzeption unterstützt

werden. Selbst bei primitiven Lernumgebungen besteht Individualisierungsmöglichkeit in der Zeitdimension. Lernende können das Material in dem der Person gemäßen Tempo durcharbeiten. Durch Variation von Beispielen lassen sich auch unterschiedliche Vorkenntnisse entsprechend berücksichtigen. Letztlich könnte Kursmaterial mit unterschiedlichem Tiefgang angeboten werden. Das Chunk-Konzept könnte auch dazu genutzt werden, Vertiefungseinheiten einzubauen bzw. Abkürzungswege für jene, die meinen, bestimmte Aspekte nur mehr „auffrischen“ zu müssen, vorzusehen. Ebenso ermöglicht dieses Konzept die Unterstützung unterschiedlicher Lerntypen (z.B. Text versus Graphik, formale Systematik versus beispielgetrieben, etc.). Weiters sind Variationen der Anwendungsdomäne denkbar.

Dies bedingt, dass die Kursarchitektur bei Lehrveranstaltungen, die zum Selbststudium vorgesehen sind, nicht linear sein sollte. Vielmehr ist es sinnvoll, ein Netzwerk festzulegen, in das die einzelnen Lehrmodule einzufügen sind. Die Architektur der Lehrveranstaltung bleibt mithin nicht wie im vorstehenden Abschnitt vielleicht implizit erschließbar, ein Rahmen linear aufgefadelter Modulspezifikationen, der, sobald er gefüllt ist, in der jeweiligen Version des Gesamtkurses aufgeht. Sie tritt vielmehr als Netzwerk von Pfaden durch den Kurs in den Vordergrund. Lernende können an „Kreuzungen“ dieser Pfade anhand von „Wegweisern“, die Elemente von Ziel- und Kontext-Metadaten zeigen, auswählen, welchen Weg durch die aus Modulen einer Produktlinie aufgebauten Lernlandschaft sie wählen wollen.

In der eben beschriebenen Konzeption ist die „Wegeplanung“ durch die Lernlandschaft noch von den Autoren festgelegt. In einer Weiterentwicklung des Konzepts muss dies nicht notwendigerweise der Fall sein. Als software-technische Analogie greifen wir dabei etwa auf logikbasierte Programmierung zurück. So besteht etwa ein Prolog-Programm^{11, 12} aus einer Menge von Fakten und einer Menge von Regeln. Fakten sind passive Elemente, wir können sie mit Daten vergleichen. Regeln sind Elemente mit Aktivierungspotential. Wenn das Inferenzsystem, ein Element der Programmierumgebung aber nicht des Programms und einziges wirklich aktives Element innerhalb des Gesamtsystems, feststellt, dass eine bestimmte Regel aufgrund der Faktenlage erfüllbar ist, versucht es, diese anzuwenden und dadurch neue Fakten abzuleiten. Welche Regel zu welchem Zeitpunkt ausgeführt wird, wird also nicht vom Programmierer (Autor) festgelegt, sondern ergibt sich dynamisch aus der Ausgangssituation und Faktenlage des Gesamtsystems.

Diese Gedanken lassen sich unmittelbar auf die Aspekte von lernergesteuertem, computergestütztem Lernen übertragen. Interpretieren wir die einzelnen Lehrmodule als Regeln und den Wissensstand der lernwilligen Person als Faktenmenge, so lässt sich durch Abgleich von Fakten mit den Vorbedingungen, die vor Studium einer Lerneinheit gegeben sein sollten, eine Situation beschreiben, die der Suche nach einer auswertbaren Prolog-Regel vergleichbar ist. Durch Auswertung der Regel ergeben sich im Fall der Ausführung eines Prolog-Programms positive Veränderungen in der Faktenbasis. Damit wird eine neue Grundlage geschaffen, um zu prüfen, welche Regeln nun anzuwenden sind. In einem Lehr-/Lernsystem findet durch Studium des entsprechenden Moduls eine Ergänzung des Wissensstandes der die Vorbedingungen erfüllte durch das aus den Lehrzielen des Moduls geschaffene Wissen statt. Die so implizit angenommene Erfüllung von Nachbedingungen kann gegebenenfalls noch durch Selbstevaluation verifiziert werden. Dadurch wird abgesichert, dass die Voraussetzungen für weitere Lernschritte geschaffen sind. Die Lehrumgebung kann mithin in Analogie zum

¹¹ Clocksin F.W., Mellish C.S.: *Programming in Prolog*; Springer-Verlag, 1981.

¹² Bratko I.: *PROLOG – Programming for Artificial Intelligence*; 2nd ed., Addison-Wesley, 1990.

Prolog-Inferenzsystem im Anschluss an das Studium des eben gewählten Moduls neue Wahlmöglichkeiten zur Auswahl weiterer vorschlagen.

Die Definition der Ausgangs-/Vorbedingungen kann entweder durch einen Einstiegstest oder durch Selbsteinschätzung festgelegt werden. Eine detaillierte Behandlung dieses Themas wie auch die Behandlung der diffizilen Problematik des dynamischen Aufstellens aussagekräftiger „Wegweiser“ würde den Rahmen dieser Arbeit allerdings sprengen. Im folgenden Kapitel werden die durch Vor- und Nachbedingungen gelegten impliziten Verbindungen zwischen Lehrmaterialien als *soft links* bezeichnet.

4. Realisierungsmöglichkeiten im Rahmen existierender Plattformen

4.1. Vorbemerkungen

Um auf der einen Seite eine vollständige Metadatenunterstützung und auf der anderen Seite wirksame Selektionsmechanismen bieten zu können müssen in jedem Fall Erweiterungen an bereits existierenden Systemen vorgenommen werden. Neben der reinen Erzeugung, Verwaltung und Darstellung von Lernmodulen (in Abbildung 2 als Chunk bezeichnet) sind es vorwiegend folgende Bereiche, die zusätzlich unterstützt werden müssen:

- Die Erzeugung/Speicherung/Verwaltung der Metadaten.
- Das Anbieten von Selektionsmechanismen (um aus relevanten Lerneinheiten Kurse zu erzeugen).
- Eine Unterstützung des Kursautors bei der Erstellung von Lerneinheiten (in Form von Konsistenzprüfungen basierend auf Vor- und Nachbedingungen.)
- Die dynamische Darstellung von Lerneinheiten und somit eine (auf per-User Basis) geeignete Repräsentation als Kurseinheit.

Um sich um die Präsentation des Kurses und um das Framework einer Lehrveranstaltung nicht weiter kümmern zu müssen, baut unser System auf dem Hyperwave Information Server (HIS)¹³ und auf dessen Add-on, die E-Learning Suite (eLS)¹⁴ auf. HIS und eLS sind im Rahmen des HAUP Programmes frei verfügbar, auch der Programmcode ist verfügbar, was das Erweitern der Serverfunktionalität und das Adaptieren der eLS erst möglich macht.

¹³ Maurer H.: *HyperWave – The Next Generation Web Solution*. Addison Wesley Longman, 1996. ISBN 0-201-40346-3.

¹⁴ eLS Homepage (Stand Februar 2002). http://www.hyperwave.com/d/products/els_elearning.html.

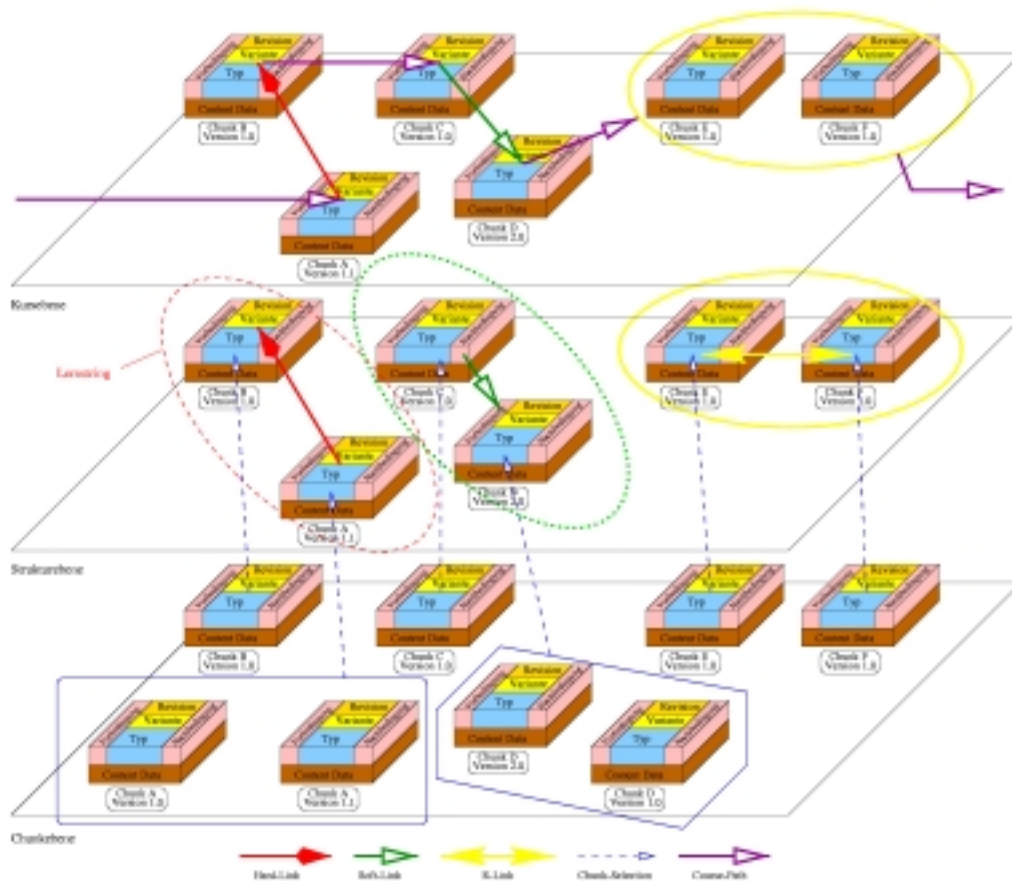


Abb. 6
Modul- und Architekturebene, verbunden durch Beziehungsstruktur

4.2. Das Planet-ET Konzept und die Verbindung zur eLS

Abbildung 6 zeigt die drei im Planet-ET Konzept enthaltenen Ebenen. Gezeigt wird auch die Verwendung der vorkommenden Hard- und Soft-Links (basierend auf den Metadatenbeschreibungen wie Vor- und Nachbedingungen), wobei es eine spezielle Art von Soft-Links gibt: die R-Links. Damit lassen sich Ähnlichkeitsbeziehungen beschreiben, wie z.B. zwei (inhaltlich gleiche) Chunks, die jedoch auf Englisch bzw. auf Französisch ausformuliert sind. Eine weitere Art der Beziehung wäre die Verwandtschaft zwischen zwei Chunks (wieder derselbe Inhalt), wobei der eine Chunk einen eher textuellen Schwerpunkt hat, der andere einen audio-visuellen. Somit ist es möglich, auf verschiedene Benutzerprofile und Lerntypen zur Laufzeit zu reagieren.

Auf der Modulebene bietet der Hyperwave Information Server nun einige wesentlichen Vorteile: Link-Konsistenz (was bedeutet, dass Links zwischen Objekten bidirektional sind, und somit nicht ins Leere zeigen können), Versionierung (was bedeutet, dass von einem Objekt mehrere Versionen existieren können) und eine umfangreiche Indizierung aller Objekte. Weiters können auf dieser Ebene sehr einfach XML Metadatenbeschreibungen einzelnen Objekten in Form von Attributen zugeordnet werden. Somit wird es auch möglich sein, einzelne Lehreinheiten zu exportieren und auf andere Systeme zu übertragen.

Der HIS bietet aber auch alle Grundlagen, um die Konzepte der Strukturebene und der Kursebene umzusetzen. Auf der Strukturebene reichen die den Objekten hinzufügbaren Attribute und typisierte Links aus, um die vom Benutzer vorgegebene Verbindungen zwischen Lehreinheiten herzustellen. Auf der Kursebene kann auf weitere zwei Konzepte der HIS zurückgegriffen werden: das Konzept der Collection-Types und das der Presentation-Hints.

Die HIS unterstützt mehrere Arten von Collections (Verzeichnisse), von denen zwei wiederum die Implementierung dynamischer Lehreinheiten unterstützen. Der erste Collection Typ trägt die Bezeichnung Multiclustern. Dieser bietet die Möglichkeit, mehrere Dokumente in einen Ordner zu legen und diese als ein einzelnes Dokument anzeigen zu lassen. Der zweite Collection-Typ nennt sich Alternative-Cluster und bietet die Möglichkeit, ein Dokument zur Laufzeit und abhängig von Attributen aus einer Reihe von weiteren Dokumenten auszuwählen. Mit dem zweiten Konzept, den Presentation-Hints (diese legen Sichtbarkeitseigenschaften von Dokumenten fest), ist es möglich, die Anzahl der für Benutzer sichtbaren Lehreinheiten zu verringern und somit dafür zu sorgen, dass nur für benutzerrelevante Information angezeigt wird.

4.2. Technische Grundüberlegungen

4.2.1. Vorbemerkung und Hintergründe

Die HIS Version 6 wird explizit XML unterstützen, sogar einen XML-Parser integriert haben. Im Rahmen von Planet-ET ist bereits daran gedacht, Lehreinheiten mittels LOM und XML zu beschreiben, aufgrund der Änderungen mit der kommenden Release wird jedoch darauf verzichtet, einen eigenen XML Editor zu implementieren. Die Beschreibung der Lehrinhalte inklusive der Pre-, und Postconditions erfolgt daher im Augenblick durch Attribute, die zu jedem HIS-Objekt gehängt werden können. Eine spätere Überführung in eine XML Struktur ist ohne großen Aufwand über Scripts und über HIS Commandline Tools möglich.

4.2.2. Die Idee: Zweistufiges Verfahren

Unser Ansatz fußt auf einem zweistufigem Ablauf – der jedoch nicht HIS bezogen ist sondern eine reine Folge des in Kapitel zwei vorgestellten Konzeptes ist.

- Jeder Autor hat die Möglichkeit sich seine Lehreinheiten = Chunks (außerhalb der HIS) zusammenzustellen. Mittels des ersten Werkzeugs (das über dynamic Objects gänzlich in die HIS Oberfläche integrierbar ist) kann dann ein Chunk in das Planet-ET Repository gestellt werden. Dieses „Hineinstellen“ bedingt gleichzeitig die Metadatenbeschreibung durch den Autor. Man hat somit ein sich füllendes Repository (Dokumente, die Metadatenattribute tragen). Diese können über die HIS Versionskontrolle editiert und modifiziert werden, die Konsistenz der Beziehung zwischen LOM-Attribut und Objekt ist somit gewährleistet.

- Der zweite Schritt besteht im Zusammenstellen des Kurses durch einen Kursautor – dazu wird das zweite Werkzeug benötigt. Geplant ist hier, diese Funktionalität in die HIS Oberfläche zu integrieren (soweit man mittels der PLACE-Language über die passenden API Funktionen verfügt). Sollte das nicht funktionieren, ist daran gedacht, die Basisfunktionalität über serverseitige Scripts, die von einem Java Applet angestoßen werden, durchzuführen. Der Kursmanager bietet die Möglichkeit, Kurseinheiten aus dem Repository zusammenzuhängen und gibt, basierend auf der Metadatenbeschreibung und entsprechender Pre- und Postconditions, dem Autor Hinweise beim Erstellen des Kurses. Wichtig dabei ist, dass das Tool in der eLS nur LINKS auf das Repository Objekt ablegt. Das bedeutet, dass eine Änderung im Modul auch gleichzeitig die Änderung des Objekts in allen Kursen bedingt. Hier ist angedacht, die Versionsverwaltung dahingehend zu erweitern, dass der Kursautor über einen möglichen Konflikt mit Vor/Nachfolgermodulen gewarnt wird. Das bedingt ebenso einen Eingriff in den Wavemaster-Layer.

Nach der Überführung in die eLS hat man ein statisches Abbild des Kurses. In einer ersten Stufe wird jedoch schon daran gedacht, dass zur Laufzeit vom Autor vorbereitete ALTERNATIV-Pfade begangen werden können. In diesem Fall wird auf das Multicluser- und Alternativ-Konzept zurückgegriffen. Durch den Multicluster können vor das eigentliche Objekt dynamische HTML Seiten vorgeschaltet werden, die sich aus einer Metadatenbeschreibung und aus dem Userprofil Informationen holen und dementsprechend eine Seite mit Hinweisen und Links zu weiteren HTML-Seiten heranziehen.

In einer zweiten Stufen (die jedoch technisch noch nicht ganz durchdacht wurde), ist es denkbar, dass von dieser dynamischen Seite ein Script angestoßen wird, das an geeigneter Stelle ein weiteres Modul aus dem Repository zum Kurs linkt – und die Rechte in der HIS so setzt, dass nur der bestimmte User die Rechte hat, dieses Dokument zu sehen, d.h. auch in seiner Kursstruktur wiederfindet. Hier gibt es aber einen Pferdefuss: die Rechte auf Links werden in zukünftigen Versionen nicht mehr unterstützt. Das bedeutet, dass man den HIS-Wavemaster Layer erweitern müsste (somit eine Java-Script und PLACE Angelegenheit) um nicht über das Rechte-Attribut sondern über Metadatenattribute eine Einschränkung in der Anzeige realisieren zu können.

4.2.3. Erweiterungen der HIS/eLS

Die vom Hyperwave Information Server zur Verfügung gestellten Konzepte und Funktionen werden im Rahmen des Projekts nun noch um zwei Werkzeuge und um einen Aufsatz auf die eLS erweitert. Abbildung 7 zeigt die zusätzlich zur HIS/eLS hinzugefügten Teile (in der Grafik Rot eingefärbt).

Das erste Werkzeug, der Repository Manager, kümmert sich um die Metadatenbeschreibung der Lerneinheiten und legt diese gemeinsam in der HIS Datenbank ab. Die erste Anforderung zur Unterstützung der Metadatenverwaltung kann dadurch erfüllt werden.

Selektionsmechanismen und Konsistenzprüfung werden durch das zweite Werkzeug, dem Course Manager, zur Verfügung gestellt. In unserem Ansatz werden die Lehrinhalte damit zunächst außerhalb der eLS zusammengestellt und danach in die eLS Umgebung portiert. Bei der Zusammenstellung der Einheiten wird auf die Versioning und Clustering Konzepte der HIS zurückgegriffen.

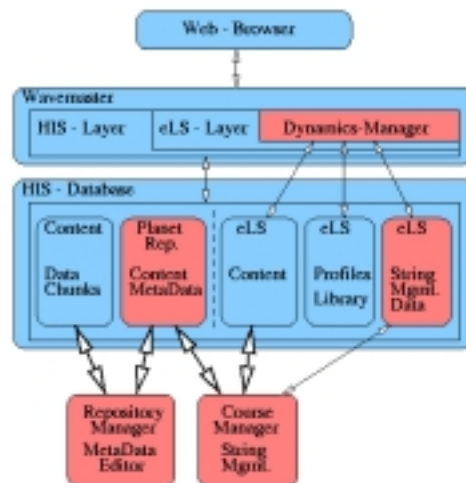


Abb. 7: Planet-ET Erweiterungen am HIS/eLS System

Um dynamische Wegweiser bzw. die dynamische Auswahl an Kursinhalten in der eLS zu gewährleisten, muss der eLS Layer (der sich um die Struktur und die Anzeige des Lerninhalts kümmert) jedoch um einen Aufsatz, dem Dynamics Manager, erweitert werden. Dieser in JavaScript gehaltene Teil greift dabei auf die Metadatenbeschreibung der Lehreinheit und auf das Profil des Benutzer zu, um sich so den Inhalt eines eLS-Kurses zur Laufzeit zusammenzustellen bzw. um hilfreiche Links zur Laufzeit einem Dokument hinzuzufügen.

5. Zusammenfassung

Ausgehend von einer vielschichtigen Lehrveranstaltungs-Evolutionsproblematik wurden Konzepte für die Individualisierung von Lehrveranstaltungen vorgestellt. Durch Trennung in Inhalts- und Beschreibungsanteile einerseits sowie durch ein Mehrebenenkonzept andererseits lassen sich diese Überlegungen mit vertretbarem Implementierungsaufwand in einer modernen open source e-Learning Software wie eLS implementieren.

Anhang: Verwendete Schlüsselbegriffe:

Baseline: Grundform einer medienunterstützten Lehrveranstaltung, die sich in Form von Revisionen und Varianten weiterentwickelt. Wenn durch Evolution der Lehrveranstaltung eine zu starke Verästelung in Varianten und entsprechend gleichzeitig zu aktualisierenden Versionen besteht, kann durch Definition einer neuen Baseline eine Reduktion der Komplexität des jeweils zu aktualisierenden Materials erreicht werden.

Release: Instanz einer medienunterstützten Lehrveranstaltung, die als Präsenzlehrveranstaltung abgehalten wurde oder als on-line Lehrveranstaltung Studierenden zum Selbststudium zur Verfügung gestellt wird. Durch Evolution von Lehrveranstaltungen entstehen unterschiedliche Releases bzw. Versionen.

Revision: Neuauflage einer Lehrveranstaltung, die auf ihrer Vorgänger-Instanz aufbaut und grundsätzlich dieselben Lehrziele verfolgt. Die neue Revision unterscheidet sich jedoch von der Vorgängerversion durch inhaltliche Aktualisierungen oder Korrekturen.

Variante: Neuauflage einer Lehrveranstaltung, die zwar auf der/einer/(ggf. mehreren) jeweiligen Vorgänger-Instanz(en) aufbaut, sich von dieser/(diesen) jedoch durch geänderte Lehrziele (und entsprechend geändertem Inhalt) unterscheidet

Version: Neuauflage einer Lehrveranstaltung, die zwar auf der/einer/(ggf. mehreren) jeweiligen Vorgänger-Instanz(en) aufbaut, sich von dieser jedoch durch gezielte Veränderungen unterscheidet. Der Grund der Veränderung ist dabei ohne Belang (statischer Sammelbegriff für Revisionen und Varianten).