

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/237427004>

Inhalte und Ergebnisse des Verbundprojekts Wissenswerkstatt Rechensysteme (WWR)

Article

CITATIONS

4

READS

85

5 authors, including:



Ulrike Lucke

Universität Potsdam

220 PUBLICATIONS 922 CITATIONS

[SEE PROFILE](#)



Djamshid Tavangarian

University of Rostock

221 PUBLICATIONS 1,259 CITATIONS

[SEE PROFILE](#)



Matthias Waldhauer

5 PUBLICATIONS 25 CITATIONS

[SEE PROFILE](#)

Inhalte und Ergebnisse des Verbundprojekts Wissenswerkstatt Rechensysteme (WWR)

L. Kornelsen, U. Lucke, D. Tavangarian, D. Voigt, M. Waldhauer

Universität Rostock, Institut für Informatik, Lehrstuhl für Rechnerarchitektur,
Albert-Einstein-Str. 21, D - 18059 Rostock

info@wwr-project.de

Kurzfassung. Der vorliegende Beitrag gibt einen Überblick über das Verbundprojekt Wissenswerkstatt Rechensysteme (WWR), in dem ein Baukastensystem von multimedialen, skalier- und rekombinierbaren Lehr- und Lernmodule entwickelt wird. Anhand der Vorstellung beispielhafter Ergebnisse aus verschiedenen Bereichen des Projektes werden die Schwerpunkte der Projektarbeit vorgestellt. Zu den konzeptionellen Grundlagen des Vorhabens zählen ein innovatives dreidimensionales Modell, das die Skalierung der Materialien erlaubt, sowie dessen Umsetzung in der XML-basierten Sprache <ML>³. Primäre Produkte des Verbunds sind etwa 150 multimediale Lehr- und Lernmodule zur Unterstützung von Vorlesung, Übung, Praktika und Selbstlernen, sowie eine große Zahl darin enthaltener Simulatoren und Medienobjekten. Darüber hinaus wurde begleitend ein umfangreiches Framework aus Werkzeugen für den gesamten Prozess der Erstellung, Evaluierung und Anwendung von Lehr- und Lernmaterialien entwickelt.

1 Einleitung

Nach der Entwicklung proprietärer Einzellösungen [1], spezieller Verbundangebote einzelner Hochschulen oder Ausbildungseinrichtungen [2] und der kompletten Abwicklung eines Studiums über das Internet [3] stehen inzwischen vor allem Aspekte der Modularität, Kombinierbarkeit, Wiederverwendbarkeit, individuellen Adaptierbarkeit und Interoperabilität im Mittelpunkt von aktuellen Forschungsarbeiten. Multimediales Lehr- bzw. Lernmaterial muß flexibel an den jeweiligen Kontext anpassbar sein, um die vergleichsweise hohen Kosten für seine Entwicklung zu rechtfertigen. Dieser Forderung wird heute weitgehend durch den Einsatz von XML begegnet [4].

Im Verbund „Wissenswerkstatt Rechensysteme“ (WWR) [5] entwickeln zwölf deutsche Hochschulen in Kooperation mit Partnern aus GI, ITG und Siemens ein XML-basiertes, multimediales Baukastensystem für das Fachgebiet Technische Informatik. Oberste Maximen des Projektes sind die Modularität und die freie, jedoch kontextbezogene Kombinierbarkeit der Materialien sowie die individuelle Skalierbarkeit der Module bzgl. verschiedener Parameter entsprechend des beabsichtigten Einsatzzwecks. Dieser Artikel gibt in Kapitel 2

zunächst einen Überblick über die theoretischen Grundlagen der Arbeit und stellt anschließend in Kapitel 3 die entwickelten Werkzeuge bzw. in Kapitel 4 das entstandene Lehr- und Lernmaterial als zentrale Ergebnisse des Verbunds vor (siehe dazu auch [6]).

2 Das WWR-Konzept

Kernziel des Verbundprojekts WWR ist die Erstellung von Lehr- und Lernmodulen. Ein Modul ist eine einen gewissen Themenkomplex behandelnde Menge von Lehr- und Lernmaterialien, bestehend aus multimedialen, dynamischen, interaktiven und textuellen Komponenten in einer XML-basierten Beschreibung. Jedes Modul hat einen Gesamtumfang von ca. 8 Stunden Vorlesung und ca. 4 Stunden Übung. Einen Sonderfall stellen die sog. Praktikumsmodule dar, die in ca. 12 Stunden bearbeitbar sind.

Ein Modul besteht aus den eigentlichen Inhalten und Metadaten. Inhaltsobjekte sind Stoffpräsentationen, Übungen oder Tests, die jeweils aus multimedialen Komponenten wie Text, Bild, Animation, Video, Sound usw. zusammengesetzt sind. Metadaten umfassen eine inhaltliche Charakterisierung, technische Aspekte wie Hardware- oder Softwarevoraussetzungen, didaktische Kommentare, Beziehungen zu anderen Modulen usw.

Die Besonderheit der WWR-Module ist ihre mehrdimensionale Skalierbarkeit, die in Abschnitt 2.1 beschrieben wird. Dieses Modell wird in der Sprache <ML>³ implementiert, die anschließend in Abschnitt 2.2 dargestellt ist. Deren Anwendung für die Erstellung von Modulen behandelt Abschnitt 2.3.

2.1 Das dreidimensionale Basismodell

Das Konzept der Modulskalierung bildet das Fundament einer flexiblen Modulbeschreibung. Mit ihm ist die simultane Berücksichtigung von drei vordefinierten Dimensionen – Intensität, Zielgruppe, Ausgabemedium – innerhalb einer einzigen abstrakten Beschreibung des Moduls möglich (siehe Abbildung 1).

Die Intensität ist dabei für die spätere flexible Zusammenstellung der Module zu Kursen und deren breitem Einsatzfeld ausschlaggebend. Sie steuert den zur Vermittlung des Lernstoffs erforderlichen zeitlichen

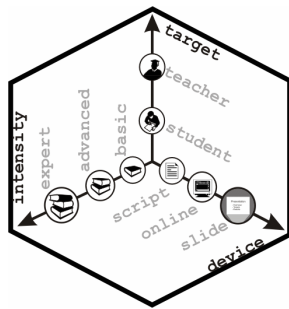


Abbildung 1: Moduldimensionen und Wertebereiche

Aufwand und damit den Schwierigkeitsgrad und die Menge der bereitgestellten Materialien. Eine einzige abstrakte Modulbeschreibung kann auf diese Weise, entweder als Basisversion (*basic*) mit einführendem Inhalt, als Aufbauvariante (*advanced*) mit weiterführenden Informationen oder als Expertenversion (*expert*) mit voller inhaltlicher Tiefe präsentiert werden.

In Ergänzung dazu erlaubt die Dimension des Ausgabemediums, den zu implementierenden Lehrstoff bestimmten Präsentationsformen zuzuordnen. So ist es etwa möglich, *slide*-Versionen für den Einsatz in Präsenzveranstaltungen, vorlesungsbegleitende *script*-Versionen oder *online*-Versionen für die Bildschirmarbeit zu generieren. Letztere bietet aufgrund der starken multimedialen Ausrichtung der Inhalte die Möglichkeit, hochgradig abwechslungsreiche, didaktisch wertvolle Materialien zu generieren. Die möglichen Inhalte der beiden erstgenannten Szenarien sind demgegenüber entweder durch das Medium Papier oder höchstens hypermedial verlinkte Dokumente beschränkt. Im Falle der vorlesungsunterstützenden Folienvariante kommen zudem nur die wesentlichen Textbausteine und essentielle Multimedia-Objekte zum Einsatz.

Das gewählte Ausgabemedium impliziert die hiermit angesprochene Zielgruppe potentieller Nutzer des generierten Lehr- und Lernstoffs. So werden i.d.R. Dozenten primär Foliensätze für die Präsentation in ihren Lehrveranstaltungen benötigen, während Studenten gerne einen vollständigen Ausdruck der Vorlesungsskripte in den Händen halten. In vielen Fällen ist es jedoch sinnvoll, die anzusprechende Zielgruppe explizit zu steuern. Insbesondere Selbstlerner profitieren z.B. von Selbsttestaufgaben zur Kontrolle des Lernfortschritts bzw. von interaktiven Komponenten zur Förderung konstruktivistischer und explorativer Methoden der Wissensvermittlung [7]. Neben der zu diesem Zweck definierten Zielgruppe *student* gibt es weiterhin die sämtliche Dozenten umfassende Zielgruppe *teacher*. Für diese spielen die erwähnten lernerbezogenen Möglichkeiten eine untergeordnete Rolle, sie profitieren vielmehr von didaktischen Hinweisen zur Stoffvermittlung und Musterlösungen zu den gestellten Aufgaben.

Mit Hilfe dieser Dimensionen lassen sich nun verschiedene Modulausprägungen bzw. Sichten auf das Modul festlegen, indem jeder Dimension genau ein Skalierungswert zugeordnet wird und dieser Skalierung

nicht entsprechende Inhaltskomponenten ausgeblendet werden. So ergeben sich durch Kombination von allen Skalierungswerte der drei Dimensionen insgesamt 18 verschiedene Sichten auf das Modul.

2.2 Die Beschreibungssprache <ML>³

XML wurde als technologische Auszeichnungsgrundlage für die im Rahmen der Projektarbeit zu implementierenden Module gewählt. Es stellt ein plattformunabhängiges, textbasiertes, leicht lern- und lesbares Datenformat dar, für das inzwischen eine Vielzahl flexibler Werkzeuge zur Zwischen- und Weiterverarbeitung sowie zur sinnvollen Unterstützung des Paradigmas der Trennung von Inhalt und Präsentation existiert. Die *Multidimensional LearningObjects and Modular Lectures Markup Language* <ML>³ basiert auf grundlegende Konzepte existierender Sprachen aus dem eLearning-Umfeld [8]. Im Folgenden wird zunächst das der Beschreibungssprache zugrunde liegende Metamodell vorgestellt, bevor einige Details zur Umsetzung des Skalierungsmodells erörtert werden.

Metamodell. Das XML-Binding von <ML>³ basiert auf einem Metamodell, das sowohl sachlogische als auch didaktische Strukturierungsformen für ein Modul definiert. Die sachlogische Strukturierung bildet dabei die Grundlage für die Implementierung der Modulinhalte. Mit ihrer Hilfe wird das Modul aus einer Vielzahl einzelner Basisbausteine zusammengesetzt, die Textabschnitte sowie multimediale und interaktive Komponenten repräsentieren. Diese werden dann zu logisch zusammengehörenden Inhaltsabschnitten gruppiert, die entsprechend ihres Einsatzzwecks klassifiziert werden (Algorithmus, Satz, Beweis, Beispiel, Selbsttestaufgabe, Anmerkung, Tipp, Denkanstoß u.a.). Auf nächsthöherer Ebene werden einzelne Inhaltsabschnitte zu Paragraphen zusammengefasst, die wiederum, analog zur Kapitelstruktur eines Buches, innerhalb einer hierarchischen Struktur miteinander verknüpft sind. Während die sachlogische Strukturierung den Aufbau von Printmedien (Einsatzszenario *script*) widerspiegelt, bietet sie nur unzureichende Unterstützung für vielfältig vernetzte Online-Inhalte oder rein sequentiell aufgebaute Foliensätze.

An dieser Stelle kommt die sogenannte didaktische Strukturierung als Ergänzung zum sachlogischen Modulpart zum Tragen. Auf didaktischer Seite ist ein Modul auf oberster Ebene in Lektionen unterteilt, die wiederum selbst aus mehreren stoffvermittelnden Lernschritten zusammengesetzt sind und darüber hinaus spezialisierte Lernschritte, etwa für thematische Einführungen, Zusammenfassungen oder Trainingseinheiten, enthalten können. Diese Abschnitte kapseln dabei die Inhalte nicht direkt, sondern verweisen auf sogenannte Präsentationseinheiten auf der sachlogischen Seite, mit deren Hilfe eine Gruppierung und Zuordnung von Inhaltskomponenten zu Bildschirmseiten (*online*) und Folien (*slide*)

erfolgt. Abbildung 2 gibt einen Überblick über beide Strukturen und ihren Zusammenhang.

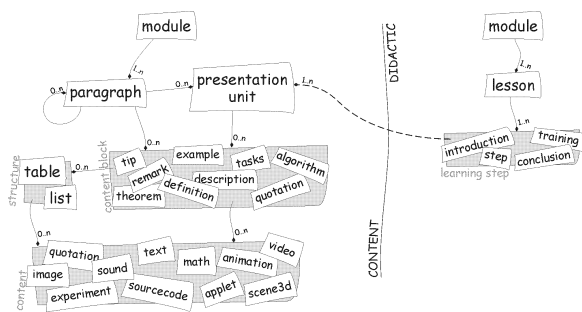


Abbildung 2: Sachlogische und didaktische Strukturierung eines Moduls

Die Abkopplung der didaktischen von der sachlogischen Struktur erlaubt es, einem Modul mehrere didaktische Strukturierungen zuzuordnen, so dass sich je nach den Anforderungen von gegebenen Lehr- / Lernsituationen bzw. von individuellen pädagogisch-didaktischen Verfahrensweisen des Lehrenden passende didaktische Konzepte auswählen oder selbst implementieren lassen, ohne dass Modifikationen an der inhaltlichen Seite notwendig werden. Weiter ist es denkbar, völlig andere didaktische Modelle als das hier vorgestellte über die Referenzierung von Präsentationseinheiten an den sachlogischen Modulpart anzubinden. Hierdurch wird einerseits Kompatibilität zu anderen existierenden Modellen erreicht und andererseits Offenheit für zukünftige Entwicklungen in diesem Bereich sichergestellt.

Skalierung. Die einzelnen Inhaltselemente auf der sachlogischen Seite eines Moduls lassen sich über eine geeignete Attributierung gewünschten Dimensionswerten des dreidimensionalen Basismodells zuordnen. Diese Attributierung basiert auf dem Prinzip der Restriktion. In der baumartigen Struktur eines Moduls werden die Wertebereiche der Skalierungsattribute auf dem Weg von der Wurzel bis zu den Blättern zunehmend eingeschränkt. Da ein Modul per Definition offen für sämtliche Ausprägungen ist, darf auf oberster Ebene der Modulbeschreibung noch keine Einschränkung vorgenommen werden. Skalierungsattribute werden nicht zwingend, sondern nur optional angegeben. Liegt keine explizite Skalierung für ein Element vor, bekommt es implizit die Skalierungswerte seines direkt übergeordneten Elements, also des Vaterknotens im Baum, zugeordnet. Ein semantischer Fehler liegt vor, wenn die über ein Skalierungsattribut angegebenen Werte nicht aus der Menge der zugehörigen Skalierungswerte des Vaterelements stammen, also keine Gleichheit oder Einschränkung vorliegt.

Die im Rahmen des Projektes entwickelte Dokumentenbeschreibungssprache <ML>³ stellt hohe Anforderungen sowohl an den Autor als auch an die Software für die Modulerstellung, -validierung und -transformation. Derartige Werkzeuge sind Gegenstand von Kapitel 3.

2.3 Vorgehensweise bei der Erstellung von Dokumenten

Die Nutzung einer strukturierten Beschreibungssprache für Dokumente erfordert einen strukturierten Erstellungsprozess, um eine hohe Qualität und Wiederverwendbarkeit der Ergebnisse gewährleisten zu können. Dabei sind unterschiedliche Aspekte eines Dokuments zu beachten, wie Abbildung 3 zeigt. Die Arbeitsschritte müssen nicht notwendigerweise streng sequentiell durchlaufen werden, sondern der Autor kann beliebig zwischen ihnen wechseln (Jojo-Verfahren).

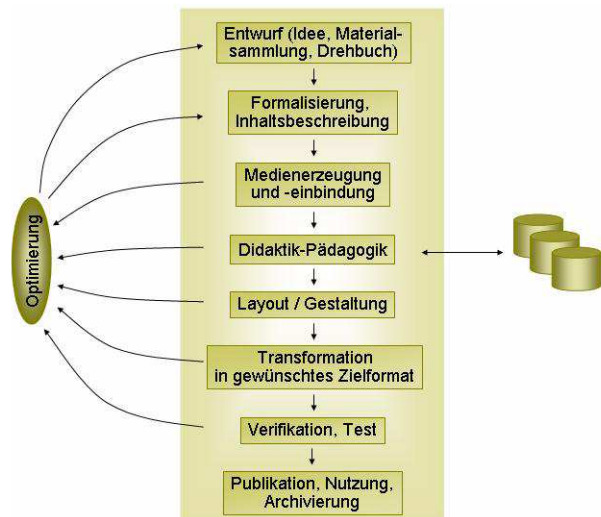


Abbildung 3: Schema des Erstellungsprozesses

Die wichtigste Grundlage des Erstellungsprozesses auf dem Weg von der Idee bzw. dem Thema zum fertigen Dokument ist eine ausgereifte Konzeption, bestehend aus der Sammlung und Strukturierung von Material sowie dessen Umsetzung in ein Drehbuch. Erst danach sollte mit der Formulierung von Inhalten begonnen werden; im XML-Umfeld bedeutet das die Codierung der Inhalte in der gewählten Beschreibungssprache. Weiterhin sind Medienobjekte zu entwickeln und zu integrieren, was oft einen enormen Zeitaufwand erfordert. Dabei ist der didaktisch-pädagogische Aspekt zu beachten: Im Unterschied zu einer wissenschaftlichen Publikation sind alle textuellen und Medienbausteine von Lehr- und Lernmaterialien auf ihre Eignung für die Wissensvermittlung bzw. -aneignung zu prüfen. Getrennt von der Inhaltsentwicklung ist das Layout für verschiedene Einsatzzwecke zu entwerfen; hier bestehen variierende Anforderungen für elektronische und Printmedien. Erst wenn alle Aspekte des Dokuments (Inhalt, Didaktik und Layout) beschrieben sind, kann dessen Übersetzung in ein Zielformat erfolgen, das danach einer Qualitätskontrolle unterzogen wird. Bei positivem Ergebnis kann das Dokument zur Publikation freigegeben werden; anderenfalls ist es (wie auch in allen vorherigen Phasen bereits möglich) einer Verbesserung zu unterziehen.

Dieser komplexe Prozess erfordert eine weitgehende Unterstützung des Autors, weshalb ein wesentlicher

Arbeitsschwerpunkt im Projekt auf die Entwicklung von geeigneten Tools gelegt wurde. Eine Übersicht über die Ergebnisse sowie ausgewählte Beispiele werden im folgenden Kapitel 3 vorgestellt.

3 Entwickelte Werkzeuge

3.1 Übersicht und Klassifizierung

Im Verlauf der Projektarbeit wurde an verschiedensten Stellen des Arbeitsprozesses ein Bedarf an komfortablen Werkzeugen erkannt, um die oft aufwendige Erstellung von XML-basierten und multimedial angereicherten Dokumenten zu unterstützen. Die im Einzelnen entwickelten Tools lassen sich (basierend auf dem in Abschnitt 2.3 vorgestellten Modell des Entwurfsprozesses) wie folgt klassifizieren:

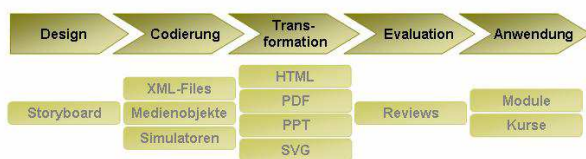


Abbildung 4: Werkzeugklassen und bearbeitete Dokumententypen

Die in Abbildung 4 mit Codierung bezeichnete Phase umfaßt die Umsetzung von Inhalten sowie Medien- und Didaktikelementen. Die Layout-Gestaltung erfolgte für das gesamte Projekt zentral und muß daher nicht vom Autor erbracht werden.

Ergänzend zu bereits vorhandenen Werkzeugen, die zum Teil integriert oder weiterentwickelt wurden, sind für alle fünf o.g. Klassen Eigenentwicklungen bei den Projektpartnern entstanden. Insgesamt wurden über 20 einzelne Tools zur Modulbearbeitung implementiert. Darüber hinaus gibt es zahlreiche Werkzeuge für den Einsatz innerhalb eines Moduls, z.B. Simulatorkerne, interaktive Visualisierungen und Medienobjekte. Im folgenden Abschnitt 3.2 werden herausragende Vertreter kurz vorgestellt.

3.2 Ausgewählte Beispiele

Moduldesign. Für die Konzeption eines Dokuments kann zu großen Teilen noch auf vorhandene Werkzeuge (Gedankensammlung mit MS Mindmapper, gemeinsames Authoring mit einem wiki, usw.) oder auch auf nicht-elektronische Medien (Papier, Tafel, Whiteboard o.ä.) zurückgegriffen werden. Es besteht die Möglichkeit, die Ergebnisse dieses Prozesses in einem Web-Interfaces zu erfassen und daraus automatisch das XML-Grundgerüst eines Moduls zu generieren. Dieses Tool ist leider noch nicht für das aktuelle <ML>³ Schema verfügbar.

Modulcodierung. Um die XML-Implementierung von Modulen auch für technisch nicht versierte Autoren zu

ermöglichen, wird allgemein die Nutzung von Standardanwendungen favorisiert. Dazu wurde das Desktop-Publishing-Programm FrameMaker um Komponenten der <ML>³ Definition erweitert, so dass nahezu der gesamte Sprachumfang unter Nutzung der Möglichkeiten dieses Werkzeugs (z.B. WYSIWYG, Querverweise, Nummerierung) zugänglich ist. Dabei stehen sowohl eine Inhalts- als auch eine Strukturansicht (Baum) des Dokuments zur Verfügung.

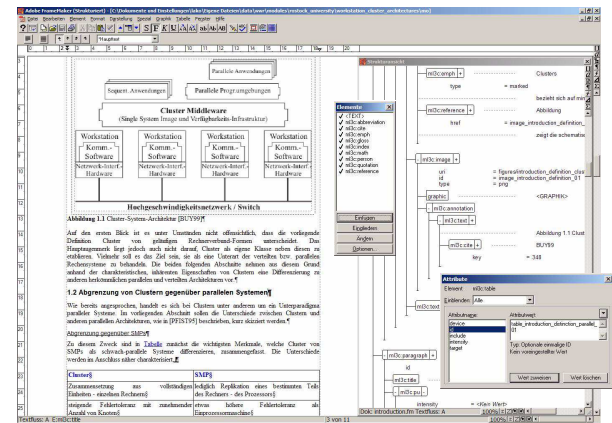


Abbildung 5: <ML>³ Authoring mit FrameMaker

Dem Prozess liegt eine verteilte Infrastruktur zugrunde, die u.a. einen CVS-Server zur Aufnahme aller Module sowie Datenbanken für Literaturdaten, Abkürzungen, Glossareinträge usw. enthält. Andere Tools widmen sich beispielsweise der Erstellung von MathML-Formeln zur Integration in WWR-Module.

Weiterhin existieren Importfilter, die Material aus Fremdformaten nach <ML>³ konvertieren. Das umfasst Dokumente in MS Word, LaTeX oder der ebenfalls XML-basierten Sprache LMML [9]. Mit der Verfügbarkeit von MS Office 2003 wird der Authoring-Prozess voraussichtlich eine weitere Unterstützung erfahren.

Modultransformation. Zur Übersetzung der XML-Quellen eines Moduls in ein konkretes Ausgabeformat existiert eine Reihe von Werkzeugen. Alle basieren auf einer allgemeinen Transformation, die auf syntaktische und semantische Korrektheit prüft, die gewünschte Skalierung der Dimensionen Zielgruppe und Intensität vornimmt und ein Zwischenformat erzeugt, aus dem die einzelnen Ausgabeformate erstellt werden.

Die Online-Version basiert auf XHTML und wird mit Hilfe von Stylesheets erzeugt. Sie enthält den vollen Textumfang sowie alle dynamischen und interaktiven Komponenten eines Moduls. Folien bestehen im Gegensatz dazu nur aus explizit gekennzeichneten kurzen Textpassagen und ausgewählten Medienobjekten. Für die Generierung von Folien gibt es mehrere Werkzeuge: Zum Einen können PPT-Folien erzeugt werden, die jedoch noch eine manuelle Nachbearbeitung erfordern. Daneben können vollautomatisch SVG- und XHTML-basierte Folien erzeugt werden, die in einem Browser betrachtet werden. Schließlich ist die Erzeugung eines

druckbaren PDF-Skripts möglich, das ebenfalls den vollen Textumfang, jedoch nur statische Varianten der dynamischen bzw. interaktiven Elemente beinhaltet.

Für die Verwaltung und Transformation der Module existiert eine integrierte Entwicklungsumgebung auf Basis von IBM Eclipse, die den Authoring-Prozess komfortabel unterstützt.

Modulevaluation. Eine web-basierte Delivery Plattform unterstützt WWR-spezifische Aspekte des Content Managements in einem auf Basis des <ML>³ Workflows und eines Rollenkonzepts dynamisch erzeugten Zugangsportals für Autoren, interne und externe Reviewer und Nutzer von Modulen. Dieses Werkzeug beinhaltet keineswegs die Funktionalität einer Lehr- bzw. Lernumgebung, sondern erlaubt die strukturierte Verwaltung von Modulen von ihrer Erstellung und Transformation über die Evaluation und ggf. Freigabe bis hin zur Recherche nach Modulen.

Modulanwendung. Der Einsatz von Modulen umfasst auch ihre individuelle Anpassung. Eigene Vorstellungen von Dozenten bzgl. der Gestaltung ihrer Präsentation können in einem Editor für SVG-basierte Folien nachträglich umgesetzt werden. Auch die didaktische Struktur eines Moduls kann per Drag and Drop komplett neu arrangiert werden.

Weitere Unterstützung bei der Kurskomposition bietet ein automatischer Manuskriptgenerator, der auf Basis von Informationen über das beabsichtigte Thema, das Vorwissen der Lernenden, die verfügbaren Module mit ihren Querverbindungen etc. einen Vorschlag für die Zusammenstellung von WWR-Modulen in der jeweils geeigneten Intensität erzeugt.

In pädagogischer Hinsicht ist schließlich eine Evaluation des Nutzerverhaltens (User Tracking) in der Laufzeitumgebung von besonderem Interesse, die derzeit implementiert wird.

3.3 WWR-Module und existierende Lernplattformen

Zahlreiche Funktionen einer Lernplattform (z.B. Nutzer-, Kurs- und Rechteverwaltung) sind für den erfolgreichen Einsatz von Lehrmaterial unumgänglich und in den am Markt verfügbaren Systemen hinreichend implementiert. Aus diesem Grund sollte keine neue Lernplattform für das Material der Wissenswerkstatt geschaffen werden. Die vollständige Integration von <ML>³ Modulen in eine verfügbare Lehr- bzw. Lernplattform ist jedoch mit diversen Anforderungen verbunden:

- mindestens die Referenzierung der verschiedenen Ausgabeversionen eines Moduls für den Zugriff durch die Studenten eines Kurses
- möglichst volle Unterstützung von Funktionen wie Bearbeitung und Kontrolle von Übungsaufgaben, Speicherung des Bearbeitungsstands, Bookmarks, Annotationen

- im besten Fall Nachbildung des <ML>³-Workflows und dynamische Generierung der jeweils benötigten Modulinstanz zur Laufzeit
- Zugriff auf mit einem Modul verbundene Werkzeuge (z.B. externe Simulatoren) und allgemeine WWR-Tools wie z.B. zur Kurskomposition

Hinzu kommen weitere Anforderungen wie Plattformunabhängigkeit, leichte Erweiterbarkeit durch offene Schnittstellen, geringe Kosten (möglichst Open Source) etc.

In einer umfangreichen Evaluation wurde festgestellt, dass die vorhandenen Tools zwar einen beeindruckenden Funktionsumfang aufwiesen, jedoch keines davon zur Integration fremder XML-Formate in der Lage war. Die Bindungen zwischen den einzelnen Komponenten der Systeme sowie zu ihren Datenformaten waren zu eng [10].

Als Ergebnis sind innerhalb des Projekts einige Ansätze entstanden, um beliebige Funktionalität mit Hilfe von Web Services auf transparente Weise in eine Lernplattform zu integrieren [11][12]. Diese Ansätze werden derzeit weiter verfolgt.

4 Verfügbare Module

Die Entwicklung einer großen Anzahl von Lehr- und Lernmodulen aus dem Bereich Technische Informatik war neben der Werkzeugentwicklung der zweite Hauptgegenstand der Projektarbeit

4.1 Übersicht und Klassifizierung

Die erstellten Module decken alle Teilgebiete der technischen Informatik ab, welche in sechs *Main Concepts* unterteilt wurden (vgl. Abbildung 6).

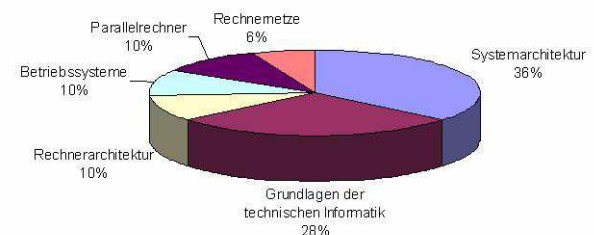


Abbildung 6: Aufteilung der Module auf die Teilbereiche der technischen Informatik

Da jedem dieser Teilgebiete eine große Anzahl Module zugeordnet sind, die selbst jeweils nur einen Teilaspekt davon behandeln, wurde ein zweistufiges Klassifizierungsschema entwickelt. Dieses kann selbst wiederum als ein Zweig der Informatik gesehen werden und ordnet sich diesen unter. Das Schema dient der Einordnung der Module aus Autoren- und Anwendersicht sowie auch aus technischer Sicht. Die Bezeichnung eines Moduls unter Anwendung des Namensschemas enthält mehrere Komponenten: *Main Concept*, *Thema*, *Nummer*, *Modul-*

typ, Version, Sprache. Diese können um Identifikatoren *Target, Intensity, Device* für eine bestimmte Ausprägung erweitert werden. Die Bezeichnung CA.PA.2.L.1.DE besteht beispielweise aus folgenden Komponenten:

CA = Computer Architecture,
PA = Processor Architecture,
lfd. Nr. 2 (innerhalb aller Module in CA.PA),
Lecture,
Version 1,
deutsch.

Über einen derartigen Bezeichner sowie die detaillierte Beschreibung in einem LOM-basierten Metadatensatz [13] können die Module in Katalogen identifiziert werden. Insgesamt werden zu Projektende etwa 150 Module zur Verfügung stehen.

4.2 Ausgewählte Vorlesungsmodule

Mit der Partitionierung des Lehrmaterials in Module ist eine Verbesserung der Organisation und Koordination hinsichtlich einer redundanzfreien Gestaltung der Modulinhalt, aber insbesondere auch die Möglichkeit zur Kombination der Module im Hinblick auf die Generierung vollständiger Kurseinheiten verbunden.

Verzahnung der Module. So existiert beispielsweise im Teilaspekt *Grundlagen der Technischen Informatik* ein Themenkreis für Basismodule zu *Registern und Speichern*, welche die Voraussetzung für die vertiefende Behandlung z.B. von *Prozessorarchitekturen* schaffen können. Die inhaltliche Verzahnung der Module spiegelt sich nicht nur zwischen den Arbeitskreisen sondern auch innerhalb dieser selbst wieder. Im Bereich der *Rechnernetze* werden Grundlagen durch die Beschreibung von *Drahtgebundenen* und *Drahtlosen Netzen* sowie für die Kommunikation benötigter Protokolle (*TCP/IP und UDP, IPv6 und Erweiterungen*) gebildet. Darauf aufbauend können dann höhere Abstraktionsebenen, z.B. die Anwendungsschicht mit den *Internet-Diensten* oder auch das *Netzwerk-Management* eingeführt werden. Ergänzend zu den reinen Vorlesungsmodulen wird eine praktische Behandlung ausgewählter Aspekte durch spezielle Praktikumsmodule realisiert, die in Abschnitt 4.3 detaillierter vorgestellt werden. So existieren etwa Praktikumsmodule zur Netzwerkprogrammierung oder zum Thema *VHDL*, in denen in praktischen Versuchen die theoretischen Grundlagen aus den korrespondierenden Vorlesungsmodulen vertieft werden können.

Beispiel: Rechnernetze. Das Modul *TCP/IP und UDP* gliedert sich in vier große Abschnitte. Nach einer *Einführung*, in der geschichtliche Aspekte, das ISO/OSI-Referenzmodell sowie seine Bedeutung in der Praxis erörtert werden, wird das Protokoll *IPv4* eingeführt. Dabei wird insbesondere auf den Aufbau von IP-Adressen, auf spezielle Adressen, z.B. Broadcast-Adressen, das Routing und den Domain Name Service eingegangen. Im Anschluss erfolgt eine detaillierte Betrachtung der Transport-

schicht-Protokolle *TCP* und *UDP* mit ihren Rahmenformaten, der Realisierung von Verbindungen sowie der Zusammenarbeit mit anderen Protokollen, z.B. *Telnet* und *SMTP* im Fall von *TCP* sowie Echtzeitprotokolle beim *UDP*. Abbildung 7 zeigt korrespondierende Screenshots der Online- und der Slide-Version des Moduls *TCP/IP und UDP*.

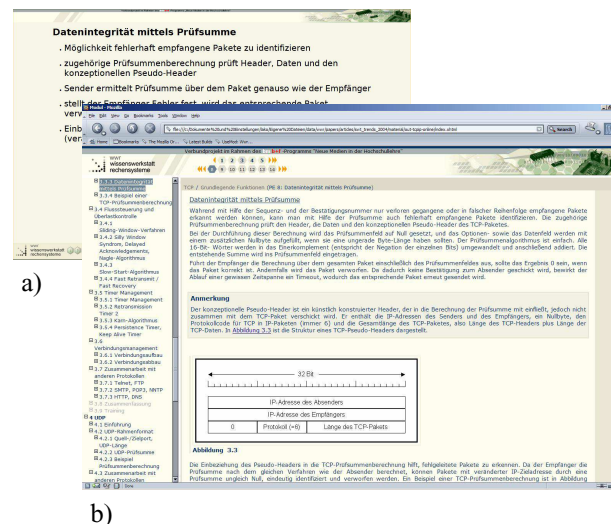


Abbildung 7: Modul „TCP/IP und UDP“
a) Slide- und b) Online-Version

Das Modul *IPv6 und Erweiterungen*, das die nächste Generation des Internet-Protokolls behandelt, basiert auf dem im letzten Absatz vorgestellten Modul *TCP/IP und UDP*. Es beschreibt die grundlegenden Änderungen am Adressschema von *IPv4* für das neue Protokoll sowie sich daraus ergebende, notwendige Modifikationen beispielsweise anderer Protokolle, z.B. des *TCP*, und des *Routings*. Abschließend werden wesentliche Erweiterungen, z.B. die Integration von Sicherheitsmechanismen und von Techniken zur Kommunikation mit mobilen Endgeräten (*Mobile IPv6*) dargelegt.

Ein weiteres Modul, das auf den beiden eben genannten aufbaut, ist die *Netzwerkprogrammierung*. Es ergibt sich also eine mögliche Verkettung der Module (*TCP/IP und UDP – IPv6 und Erweiterungen – Netzwerkprogrammierung*), bei der die ersten beiden Module ein grundlegendes Verständnis zwecks Einführung in die Problematik der *Netzwerkprogrammierung* schaffen.

4.3 Ausgewählte Praktikumsmodule

Umfangreiche praktische Aspekte des zu vermittelnden Stoffs werden als Praktikumsmodule entwickelt, die Experimente und Versuche beinhalten. Es besteht die Möglichkeit, bei Bedarf theoretischen Stoff zur Unterstützung der Aufgabenlösung zu integrieren. Den Erfordernissen entsprechend können externe Hilfsmittel (z.B. Entwicklungsumgebungen) oder durch das Modul (z.B. in Form von Applets oder Flash-Animationen) bzw. Server (z.B. Simulationen) bereitgestellte Werkzeuge

verwendet werden. Im Falle aufwendigerer Vorbereitung von Praktika unter Nutzung externer oder auch bereitgestellter Hilfsmittel sind entsprechende Anleitungen im Modul enthalten.

Beispiel: Schaltungsentwurf mit VHDL. Praktikumsmodule sind unabhängige Module, die aber vorzugsweise in Kombination mit den korrespondierenden Vorlesungsmodulen eingesetzt werden sollen. Im Zusammenhang mit dieser inhaltlichen Unabhängigkeit wurde auch eine Loslösung von der nötigen Übungsplattform angestrebt. Zu diesem Zwecke wurde ein Webserver eingerichtet, welcher den Zugang zu einem VHDL-Compiler und -Simulator (kostenlose Kommandozeilenversionen [14]) über ein Web-Interface ermöglicht. Dabei wurde auf Komfortabilität für den Leiter der Übungen sowie die Übenden bezüglich der Anwendung geachtet. Für den Leiter ergibt sich dies aus der direkten Nutzbarkeit der VHDL-Übungs-Plattform aus dem Modul (in der Online-Variante) heraus. Die Übenden profitieren von einer einfachen Bedienung und für die gestellten Aufgaben ausreichenden Möglichkeiten zur Bearbeitung der Sourcecodes (über Textfelder oder lokale Editierung mit anschließendem Senden der Dateien), Visualisierung und Untersuchung der Ergebnisse (mit Unterstützung durch ein Java-Applet, siehe Abbildung 8). Realisiert wurde diese Plattform mit serverseitigen Skripten und den ausführbaren Dateien für VHDL-Compiler und -Simulator. Zur Bearbeitung der Aufgaben benötigte Dateien (lückenhafte Sourcecodes und Beispiele) werden in Textform innerhalb der Module sowie serverseitig auf der Übungsplattform angeboten. Die Anwendung dieser Online-Entwicklungsumgebung ermöglicht weiterhin eine Mobilität der Anwender, wie sie mit bisherigen Plattformen kaum möglich war, da zur Benutzung eine Internetanbindung und ein (vorzugsweise Java-fähiger) Webbrowser auf einer stationären oder mobilen Arbeitsstation ausreichend sind.

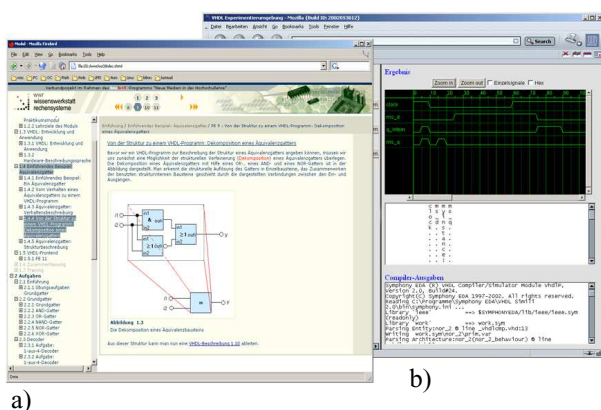


Abbildung 8: VHDL-Praktikumsmodul
a) Online-Variante
b) VHDL-Simulator

Neben diesen vorgestellten Praktikumsmodulen sind innerhalb des Projektes weitere aus anderen Bereichen

entstanden, z.B. zu Themen wie Systementwurf, eingebettete Systeme, Betriebssysteme, Verteilte Systeme, Netzwerkprogrammierung, Prozessorscheduling, Parallele Programmierung, Robotersteuerung oder Assemblerprogrammierung.

In vielen innerhalb des Projektes entstandenen Modulen (Vorlesung und Praktikum) wurde die von <ML>³ gebotene Möglichkeit der Experimentanbindung genutzt, um Nutzern den Zugriff auf vorbereitete Experimente und Simulationen zu bieten. Wichtige Grundlagen zur Einbindung in die Module wurden von der Fachhochschule Wiesbaden aufgezeigt [15]. Dabei werden externe Server bereitgestellt, welche über ein Webinterface die Einbindung von Simulatoren oder Experimenten ermöglichen. Während viele Simulatoren direkt im Modul ausgeführt werden können, werden Experimente meist über einen Experiment-Server angebunden, welcher mit dem Webserver über eine SOAP-Schnittstelle kommuniziert. Dies ist bedingt durch einen für das Experiment oft notwendigen Hardwareaufbau, welcher selten direkt an einen Webserver angebunden ist. Über das Modul ist es dann z.B. möglich, ein Experiment nur zu beobachten (z.B. per Videostream oder übertragene Messwerte) oder es auch zu steuern.

4.4 Ausgewählte Simulatorkerne

Innerhalb der Praktikumsmodule werden zahlreiche Simulatorkerne eingesetzt, von denen herausragende Vertreter kurz vorgestellt werden sollen.

LogiFlash. Zur praktischen Simulation elektronischer Schaltungen in Theorie- und Übungseinheiten wurde an der Universität Frankfurt a. M. der in Flash implementierter Logiksimulator „LogiFlash“ entwickelt (siehe Abbildung 9). Dieser ermöglicht es, digitale Schaltungen interaktiv anzusehen und zu testen oder im Rahmen von Übungsaufgaben selbst zu verdrahten und die Korrektheit der erstellten Schaltung überprüfen zu lassen. Eine auf dem gleichen Simulatorkern basierende umfangreichere Version steht als Schaltungs-Entwurfswerkzeug für den Autor zur Verfügung.

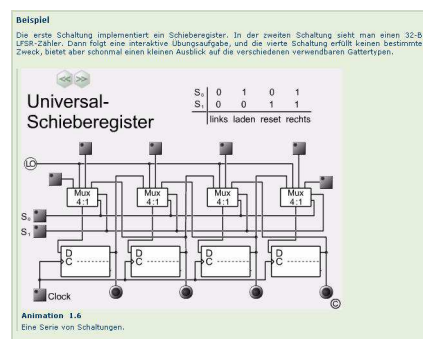


Abbildung 9: LogiFlash in einem Online-Modul

Betriebssysteme. An der TU Chemnitz entstand eine Reihe von Simulatoren, die verschiedene Aspekte von

Betriebssystemen nachbilden, z.B. Speicher- und Dateimanagement, Prozeßscheduling in Windows, Linux und in einem akademischen Betriebssystem (siehe Abbildung 10 b). Mit ihrer Hilfe kann der Lernende die behandelten komplexen Zusammenhänge durch die anschauliche Darstellung und umfassende Möglichkeiten für eine Interaktion leichter verstehen und nachvollziehen.

Mikroprogrammierung. Als Teil eines Praktikumsmodules über Steuerwerke entstand an der TU München ein komplexer Simulator eines mikroprogrammierbaren Rechners. Dieser erlaubt die Entwicklung eigener Mikroprogramme für diesen Rechner, welche dann komplett oder in Einzelschritten getestet werden können. Der Simulator gibt dazu eine Vielzahl an Statusinformationen und die Inhalte des Speichers und der Register aus. Die Eingabe der Mikroprogramme erfolgt dialogbasiert (siehe Abbildung 10 a).

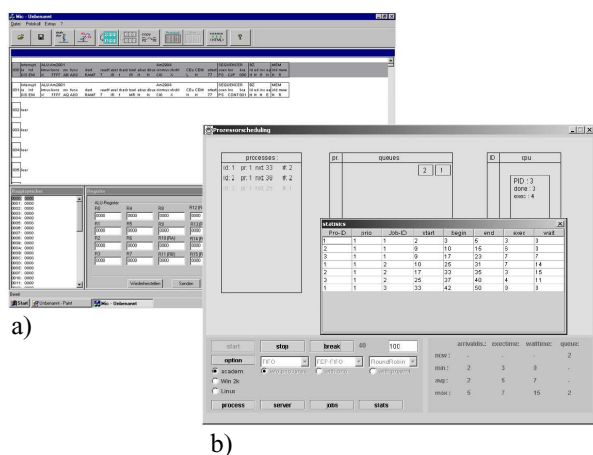


Abbildung 10: a) Mikroprogrammierbarer Rechner
b) Prozeßscheduling

4.5 Ausgewählte Medienobjekte

Im Rahmen des WWR-Projektes konnte eine große Anzahl multimedialer, zum Teil interaktiver Komponenten entwickelt werden. Diese werden vor allem zu Visualisierungszwecken eingesetzt, um komplexe Sachverhalte innerhalb der Module auf verständlichem Wege deutlich zu machen. Ein großer Vorteil dieser Objekte liegt in ihrer Wiederverwendbarkeit, welche zu einem breiten Einsatz innerhalb der erstellten Lehr- und Lernmodule führte.

Beispiele für derartige Visualisierungskomponenten sind ein *interaktives Betrachtungs-Tool für logische Schaltungen*, das eine Teilanwendung des in Abschnitt 4.4 vorgestellten *LogiFlash* darstellt, ein *ECMA-Script-Template für die interaktive Animation logischer Schaltungen auf Gatterebene mit SVG*, ein *Tool für die Generierung animierter UML-Sequenzdiagramme* sowie ein *Java Applet zur Veranschaulichung des MESI-Protokolls*. Auf die beiden letztgenannten Werk-

zeuge soll im Folgenden detaillierter eingegangen werden.

UML-Sequenzdiagramme. *Sequenzdiagramme* sind ein Teil des *UML-Standards*. Sie dienen im Gegensatz zu den Klassen-, Use-Case- und Zustandsdiagrammen der Darstellung des dynamischen Verhaltens. Im WWR-Projekt kommen die Sequenzdiagramme überwiegend in den Modulen aus dem Bereich der *Verteilten Systeme* zum Einsatz. Mit ihnen lassen sich z.B. die Interaktionen zwischen Objekten und Methoden verteilter, objektorientierter Client-/Server-Anwendungen anschaulich visualisieren. Die Beschreibung der Sequenzdiagramme erfolgt durch eine einfache XML-basierte Sprache. Eine anschließende Stylesheet-Transformation überführt die spezielle XML-Syntax in das SVG-Format, wobei auch Navigationskomponenten zum Abspielen, schrittweisen Abspielen, Stoppen und Zurücksetzen der Animation hinzugefügt werden. Ein Fenster dient dem Anzeigen von Informationen zu den ausgetauschten Nachrichten. Die eigentliche Animation wird per ECMA-Script durch Zugriff auf den DOM-Baum des SVG-Dokumentes realisiert. Das Framework zur Erzeugung der Sequenzdiagramme wurde von der FH Wiesbaden entwickelt. Ein Beispiel eines SVG-Sequenzdiagrammes zeigt Abbildung 11 a).

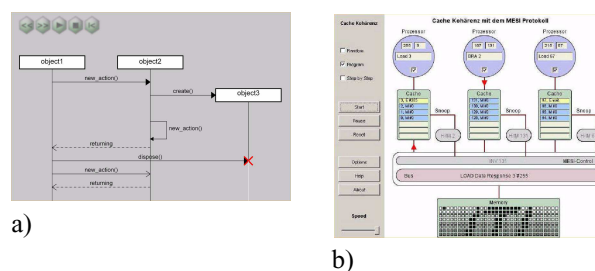


Abbildung 11: a) UML-Sequenzdiagramm
b) MESI-Applet

MESI-Applet. Beim MESI-Protokoll handelt es sich um ein Cache-Kohärenz-Protokoll, das in Shared-Memory-Systemen zur Sicherung der Konsistenz von Daten im Hauptspeicher und in den Caches der Prozessoren eingesetzt wird. Um die komplexen Sachverhalte der korrespondierenden Modifikation von Hauptspeicher und Caches deutlich zu machen, wurde an der Universität zu Lübeck ein Java Applet entwickelt, welches das Vorgehen anschaulich visualisiert (vgl. Abbildung 11 b). Das Applet erlaubt dabei die Simulation mehrerer Prozessoren, denen jeweils ein First Level Cache zugeordnet ist, und die auf einen gemeinsamen Hauptspeicher zugreifen. Die Prozessoren lassen sich in einfachem Assembler programmieren. Dabei werden sowohl die Schreib-/Lesezugriffe als auch die MESI-Steuersignale grafisch auf dem Bildschirm repräsentiert. Die Inhalte der Caches und des Hauptspeichers werden ebenfalls visualisiert. Neben der Ausführung von Programmen auf den einzelnen Prozessoren lässt sich ein zufälliger Ablauf der

Simulation initiieren. Es ist sowohl die schrittweise Ausführung der Simulation als auch der kontinuierliche Ablauf mit einer einstellbaren Geschwindigkeit möglich.

5 Zusammenfassung und Ausblick auf weitere Arbeiten

Der Beitrag stellte die wesentlichen Ergebnisse des Verbundvorhabens WWR vor, zu denen ca. 150 multimediale, mehrdimensional skalierbare Lehr- und Lernmodule sowie eine Reihe von Werkzeugen für deren Bearbeitung zählen. Das den Arbeiten zu Grunde liegende Modell weist eine inhärente Skalierbarkeit bzgl. der drei Dimensionen Intensität, Zielgruppe und Ausgabe-medium auf. Zusätzlich führt die Trennung von Inhalt, Layout und Didaktik zu weiteren Abstraktionsebenen bzw. Dimensionen des entstehenden Materials.

Dieses in der Wissenswerkstatt Rechensysteme entwickelte Konzept ist nicht auf den Einsatz für Lehr- bzw. Lernmodule der Technischen Informatik beschränkt. Die das Konzept umsetzende Beschreibungssprache <ML>³ kann problemlos auch auf andere Fachgebiete angewendet werden. Darüber hinaus ist ein Einsatz nicht nur für den Bereich eLearning, sondern für die Erstellung von digitalen Dokumenten in weitesten Sinne möglich.

Mit der Digitalisierung von Lehr- und Lernmaterial, mit seiner Bereicherung um dynamische und interaktive Multimedia-Elemente zur Steigerung der didaktischen Qualität wird ein Trend zum ubiquitären Einsatz des Materials und damit zu einer Globalisierung des Lernens sichtbar. Dies führt zu einer Stärkung von zentralisierten, virtuellen Formen der Ausbildung.

Eine Grundvoraussetzung für den breiten Einsatz derartiger Systeme in der Lehre stellt allerdings die Verfügbarkeit einer großen Zahl hochqualitativer Inhalte dar, was wiederum wirkungsvoller Mechanismen zur Mobilisierung und Unterstützung der Autoren bedarf. Die Erstellung von Inhalten erfordert zuverlässige und durchgängige, interoperable Werkzeuge, die an bestehende Infrastrukturen (z.B. digitale Bibliothek, Medienportal, Lernplattform, dig. Publisher) angebunden sind. Hier besteht derzeit noch ein enormer Entwicklungsbedarf.

Der mit einer weitreichenden Werkzeugunterstützung für die Erstellung von hochwertigem Multimediainhalt einhergehende Mehrwert ist auf den Prozess des digitalen Publizierens an sich (z.B. von wissenschaftlichen Texten) übertragbar und bietet dort darüber hinaus den Vorteil einer schnellen Publizierbarkeit, leichteren Find-, Zitier- und Wiederverwendbarkeit von elektronischen Dokumenten. Auch didaktische Aspekte oder die automatische Erzeugung von Präsentationsmaterial können so auf diesen Bereich übertragen werden. Werden digitale Publikationen in einem unabhängigen Basisformat wie XML erstellt, aus dem je nach Bedarf verschiedene Ausgabeformate erzeugt werden können,

führt dies zu einer völlig neuen Kultur des elektronischen Publizierens.

Literatur

- [1] R. Mayer, M. Koch, D. Tavangarian: „A Multimedia Learning and Design System for Microprocessors“, 2nd European Workshop on Microelectronics Education (EWME98), Noordwijkerhout, Netherlands.
- [2] O. Neumann, K. Borcea: Homepage „JaTeK“ Java Based Teleteaching Kit.
<http://telet.inf.tu-dresden.de/JaTeK.htm>
- [3] „Virtuelle Universität Hagen“.
<https://vu.fernuni-hagen.de/>
- [4] U. Rinn, K. Bett, J. Wedekind, P. Zentel, D. M. Meister, W.F. Hesse: „Virtuelle Lehre an deutschen Hochschulen im Verbund“, Teil 1: „Eine empirische Untersuchung der Projektkonzeptionen von Vorhaben zur Förderung des Einsatzes Neuer Medien in der Hochschullehre im Förderprogramm Neue Medien in der Bildung“. Institut für Wissensmedien, Tübingen 2003.
- [5] „Wissenswerkstatt Rechensysteme“.
<http://www.wwr-project.de/>
- [6] D. Tavangarian (Hrsg.): „Structured eLearning: Wissenswerkstatt Rechensysteme“, Proceedings zum Projektabschluss-Workshop, Rostock 2004.
- [7] M. Leybold, K. Nölting, M. Röser, D. Voigt: *Learner Centricism and Constructivism - New Paradigms for E-Learning?* submission to Ed-Media 2004.
- [8] „Multidimensional LearningObjects and Modular Lectures Markup Language (<ML>³)“.
<http://www.ml-3.org/>
- [9] Learning Material Markup Language. <http://www.lmml.de/>
- [10] F. Weitzl, A. Wiesner, E. Kelp: „XMLbased content and its integration into existing e-learning solutions: results and consequences of a survey“, World Conference on Educational Multimedia, Hypermedia & Telecommunications (Ed-Media), Denver/Colorado/USA, 2002.
- [11] M. Schmid, J. Stynes, R. Kröger: „Architektur einer dienstebasierten, personalisierbaren Laufzeitumgebung für Lernumgebungen“. DeLFI 2003, GI-Edition - Lecture Notes in Informatics (LNI), Bonn.
- [12] U. Lucke, D. Tavangarian: „Why and How Education Becomes a Service: An eLearning Architecture Based on Web Services“, submission to Ed-Media 2004.
- [13] Bunschowski, M., Lucke, U., Röser, M., Tavangarian, D.: „The role of metadata for the automated generation of educational material“, Proceedings of the European Conference on E-Learning (ECEL), Glasgow/UK, 2003.

- [14] Symphony EDA.
<http://www.symphonyeda.com/>
- [15] Kalfa, W., Kroeger, R., Koehler, F.: „Integrating Simulators and Real-Life Experiments into an XML-based Teaching and Learning Platform“, World Conference on Educational Multimedia, Hypermedia & Telecommunications (Ed-Media), Denver/Colorado/USA, 2002.

Danksagungen

Das Verbundprojekt „Wissenswerkstatt Rechensysteme“ (WWR) wird vom Bundesministerium für Bildung und Forschung im Rahmen des Programms „Neue Medien in der Bildung“ gefördert. Neben der Universität Rostock (Koordination) zählen elf weitere Hochschulen zu den

Mitgliedern des Verbunds: Universität Bayreuth, Technische Universität Chemnitz, Technische Universität Darmstadt, J.-W.-Goethe-Universität Frankfurt a. M., Universität der Bundeswehr Hamburg, Universität Hannover, Universität zu Lübeck, Universität Karlsruhe, Technische Universität München, Universität Passau und Fachhochschule Wiesbaden. Weitere Partner sind die deutschen Fachgesellschaften GI und ITG sowie die Siemens AG.

An dieser Stelle sei den insgesamt weit über 40 aktiv an der Realisierung des Vorhabens beteiligten Wissenschaftlern an den Partner-Hochschulen unser aufrichtiger Dank ausgesprochen. Das Vorhaben wäre ohne ihr unermüdliches Engagement in dieser Form und mit diesen Ergebnissen nicht zu bewältigen gewesen.