

Informationsverarbeitung in der Hochschullehre

Heinz Lothar Grob

Lehrstuhl für Wirtschaftsinformatik und Controlling
Westfälische Wilhelms-Universität Münster
Leonardo-Campus 3
48149 Münster
grob@uni-muenster.de

Abstract: cHL stellt ein an der Universität Münster entwickeltes E-Learning-Konzept dar, mit dem das Ziel verfolgt wird, die Qualität der Lehre und des Lernens zu verbessern. cHL ist ein wirtschaftsinformatischer Ansatz, der in erster Linie die Rolle der Studierenden, der Lehrenden und der Autoren von Lern-/Lehreinheiten unterstützen soll.

Komponenten der cHL-Architektur sind neben einer multimedialen Datenbank insbesondere ein Autoren-/Lehrer-/Lerner (ALL)-System sowie ein Administrationssystem. Softwaremäßig wurden diese drei Komponenten durch OpenSource-Produkte realisiert. Hierzu gehören primär die am Institut für Wirtschaftsinformatik der Universität Münster entwickelten Produkte OpenUSS und Freestyle Learning (FSL). Im Mittelpunkt des Beitrages steht eine Kurzbeschreibung des Systems OpenUSS, das für die Informationsverarbeitung in der Hochschullehre erstellt worden ist.

1 Zielsetzung

Die öffentliche Verwaltung tritt zunehmend ins Blickfeld der Wirtschaftsinformatik. Eine besonders reizvolle Aufgabe ist darin zu sehen, für die Institution der *Hochschule* Informations- und Kommunikationssysteme zu entwickeln, zu deren Kernaufgaben Forschung, Lehre und Weiterbildung gehören. Der folgende Beitrag hat die Informationsverarbeitung in der Lehre zum Inhalt. Beschrieben wird ein an der Universität Münster entwickeltes E-Learning-Konzept: die computergestützte Hochschullehre (cHL). Auf die Wiederverwendbarkeit und Übertragbarkeit von Konzept und Software auf andere Hochschulstandorte wird abschließend eingegangen.

Mit der Einführung des cHL-Konzepts durch die Universitätsleitung wird das Ziel verfolgt, die Lern- und Lehrbedingungen in den Fakultäten zu verbessern [Sch01]. Erwartet wird, dass sich die investiven Maßnahmen in studienbezogenen Kennzahlen (z. B. Prüfungsleistungen, Studiendauer, Zufriedenheit) widerspiegeln. Das Konzept ist gleichzeitig Erkenntnisobjekt einer von Psychologen vorgenommenen Wirkungsforschung [Sc03].

2 Ein Referenzmodell zur Informationsverarbeitung in der Hochschullehre

In einem Referenzmodell für die Hochschullehre werden wieder verwendbare Konstruktionsleistungen zur Entwicklung von Informationssystemen für die Lehre dokumentiert [Br03, S. 34-37]. Aus dem in Abbildung 1 dargestellten Modell gehen die Rollenträger, die Aufgaben und die Datenbestände hervor, die bei einer weit gefassten Betrachtung des Lehrbereichs relevant sind.

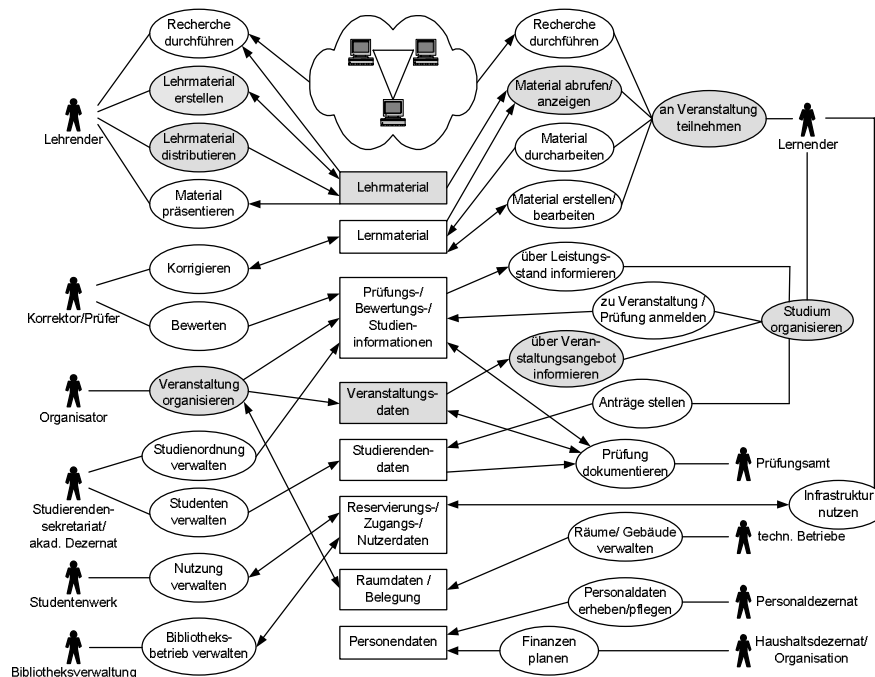


Abbildung 1: Referenzmodell für den Lehrbereich [Do02, S. 18]

Der von Doberkat [Do02] entwickelte Ansatz umfasst neben den Kernfunktionen des E-Learnings, zu denen insbesondere die Lehre und das Lernen gehören, auch die indirekt mit der Lehre zusammenhängenden Aufgaben. Hierzu zählt die Abwicklung von Prüfungen, die eine Zusammenarbeit zwischen den Hauptakteuren und dem Prüfungsamt erfordert. Darüber hinaus ist auf die Funktionen der Verwaltung und der technischen Betriebe hinzuweisen, die für die in der Lehre benötigten Ressourcen zuständig sind. Die Funktionen des Kernbereichs sind in der Abbildung grau markiert worden. Das Referenzmodell für den Lehrbereich ist selbstverständlich offen und um weitere Funktionen ergänzbar.

3 Konzept der computergestützten Hochschullehre (cHL)

3.1 Das cHL-System als Wissensmanagementsystem

cHL stellt ein Referenzmodell zum Forschungsgebiet „E-Learning“ dar, das neben den Kernfunktionen Lehren und Lernen auch die Entwicklung von Lehr-/Lerneinheiten (Authoring) zum Inhalt hat. Der Terminus Authoring wurde gewählt, da kein vergleichbarer deutscher Begriff existiert. Diese Funktionskette zur Wissensverarbeitung wird in Abbildung 2 dargestellt.



Abbildung 2: Funktionskette zur Wissensverarbeitung

Den drei Kernfunktionen sind die Rollen Autor, Dozent (Lehrer) und Studierender (Lerner) zugeordnet. Ein System, das diese drei genannten Akteure unterstützt, soll als Autoren-/Lehrer-/Lerner (ALL)-System bezeichnet werden.

Während in der Bezeichnung cHL der Begriff der *Lehre* betont wird, steht bei E-Learning das *Lernen* im Vordergrund. Hierin sollte jedoch kein Widerspruch gesucht werden. Bei Verwendung des Begriffes *Lehre* wird die mittlere Phase der dreiteiligen Funktionskette eines Wissensverarbeitungsprozesses fokussierend herausgestellt, während bei Verwendung des Begriffes *Lernen* die letzte Phase zielorientiert betont wird.

In jedem Fall haben das generelle Konzept E-Learning und das konkret ausgestaltete Referenzmodell cHL sämtliche Phasen des Wissensmanagements für den Aufgabenbereich der Hochschullehre zum Inhalt. Mit dem Begriff *Wissen* ist eine Bestandsgröße gemeint, die durch die Bewegungsgrößen *Informationen* bzw. *Desinformationen* verändert wird [GB98, S. 8]. Die Funktionen des Wissensmanagements sind in Abbildung 3 wiedergegeben worden.

Das allgemeine Schema zum Wissensmanagement setzt sich aus zwei Teilen zusammen. Das Gesamtsystem enthält die vorgegebenen bzw. vereinbarten Ziele sowie die Bewertung des im Realisierungsprozess verarbeiteten Wissens. Der Realisierung der Ziele ist ein eigenes Teilsystem gewidmet worden.

Das erste Element des Teilsystems „Realisierung“ stellt die Wissensidentifikation dar. Subjektiv wahrgenommenes Wissen ist letztlich in objektiver Form zu repräsentieren. Hierbei geht es darum, den spezifischen Beitrag des Wissens zur Lösung von Problemen zu ergründen. Wissensidentifikation steckt dabei den Rahmen ab, der durch Wissenserwerb zu füllen ist. Beim Wissenserwerb kann zwischen internen und externen Quellen unterschieden werden. Zu den internen Quellen eines Unternehmens gehören insbesondere die Datenbestände des Rechnungswesens sowie des Controllingsystems. Externe

Quellen betreffen spezielle Marktforschungsergebnisse, aber auch das breite Spektrum webbasierter Informationen.

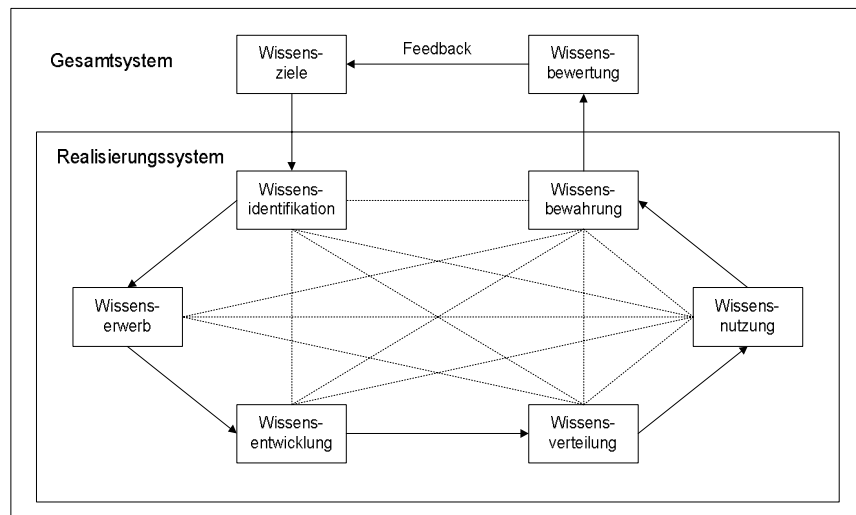


Abbildung 3: Bausteine des Wissensmanagements [PRR99, S. 56]

Bei der Wissensentwicklung geht es primär um die Gestaltung von Rahmenbedingungen zur Erzeugung von Kreativität [Al02, S. 59]. Wissensverteilung beinhaltet, dass das Wissen einer Organisation einzelnen Mitgliedern oder Gruppen verfügbar gemacht wird. Ob dies top-down oder bottom-up oder in einer Mischform geschieht, ist eine Entscheidung, die insbesondere von den situativen Determinanten einer Unternehmung abhängig ist, wie z. B. vom Umfang der Routinearbeit im Vergleich zur innovativen Tätigkeit, vom Führungsstil und vom kognitiven Stil der Vorgesetzten bzw. von der Motivation und der Wissensverarbeitungsfähigkeit der Mitarbeiter. Letztlich geht es um die Wissensnutzung zur Verbesserung operativer und strategischer Prozesse. Wissensnutzung muss stets die Steigerung der Effektivität bzw. der Effizienz zum Ziel haben. Bei der Wissensbewahrung geht es nicht nur um die informationstechnische Archivierung des Wissens, sondern auch um die Verankerung relevanter Wissensbausteine „in den Köpfen von Individuen“.

Das allgemeine Konzept des Wissensmanagements bildet eine geeignete Grundlage zur Gestaltung des Wissens- bzw. Informationsverarbeitungsprozesses in der Hochschullehre. Dabei ist zu bedenken, dass das Wissensmanagement nur einen Teilaspekt der mit der Hochschullehre verbundenen Ziele abdeckt. Neben Wissen, zu dem nicht nur Fakten- und Modellwissen, sondern auch das Problemlösungswissen gehört, sollten an Hochschulen auch Schlüsselqualifikationen zur Persönlichkeitsbildung vermittelt werden.

Im Unterschied zum Wissensmanagement einer Unternehmung, in der eine Vielzahl von Akteuren mit heterogenen Eigenschaften beteiligt ist, ist an der Institution Hochschule ein einfacheres Rollenschema gegeben. Dies zeigt sich im Prozess der Wertschöpfung,

zu dessen Visualisierung die in Abbildung 3 dargestellte Funktionskette um die Bestandsgröße „Wissen“ erweitert worden ist. Dabei soll verdeutlicht werden, dass die Abgrenzung von Wissen im jeweiligen Kontext vorgenommen wird.

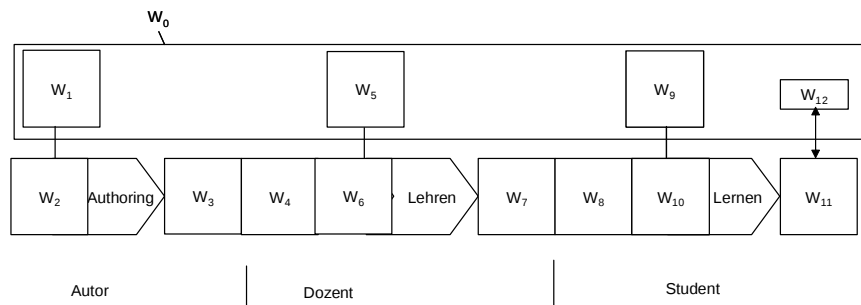


Abbildung 4: Erweiterung der Funktionskette um rollenspezifische Wissensbausteine

Während des Informationsverarbeitungsprozesses in der Hochschullehre wird der Wissensbestand eines jeden Rollenträgers beeinflusst. Das aus der Grundgesamtheit der Wissensdomäne W_0 von einem Autoren wahrgenommene Wissen W_1 wird von ihm extrahiert (W_2) und formal in einer Lern-/Lehreinheit repräsentiert (W_3). Anschließend erfolgt der Wissenserwerb unter Verwendung der Lern-/Lehreinheit W_3 durch den Dozenten, der W_4 wahrnimmt. Seiner Präsentation liegt das Wissen W_6 zugrunde, bei dem das neue Wissen W_4 mit seinem bereits vorhandenen Wissen W_5 vernetzt wird. Durch die Lehre wird es in das von ihm zu präsentierende Wissen W_7 transformiert. Von den Lernenden wird dieses Wissen nicht in gleicher Weise, sondern subjektiv modifiziert aufgenommen (W_8) und mit dem eigenen Wissen W_9 zu W_{10} verknüpft. Das Ergebnis dieses Informationsverarbeitungsprozesses stellt das durch den Lehr-Lernprozess erworbene Wissen W_{11} des Studierenden dar. Im Rahmen von Prüfungen, in denen das Wissen (W_{12}) gefordert wird, finden Wissensvergleiche und Abweichungsanalysen statt. W_{12} kann sich entweder an W_3 oder aber an W_7 oder an W_6 oder gar an W_0 orientieren.

3.2 Der Ordnungsrahmen

Autoren, Lehrer und Lerner sind die *primären* Rollenträger des cHL-Systems. Unterstützt werden sie durch *sekundäre* Rollenträger, zu denen z. B. der EDV-Administrator und der Content-Administrator gehören. Während der EDV-Administrator für informationstechnische Abläufe zuständig ist, hat der Content-Administrator Aufgaben bezüglich der Administration der Inhalte inne, zu denen vor allem Lernmaterialien, Learning by Doing-Einheiten und Simulationsprogramme gehören. Im Idealfall wird die Rolle des Lehrers durch einen Professor ausgeübt, während die des Content-Administrators durch seine Sekretärin wahrgenommen wird. Denkbar ist aber auch, dass Lehrer und Content-Administrator in Personalunion agieren. Die Perspektive der Rollenträger auf das cHL-System und dessen Komponenten werden nun anhand eines Ordnungsrahmens verdeutlicht [Br03, S. 128].

Der Ordnungsrahmen klassifiziert die cHL-Komponenten und deren Nutzung durch die primären und sekundären Rollenträger. Das der Organisationseinheit (z. B. einem Lehr-

stuhl) zur Verfügung stehende Gesamtsystem setzt sich aus der cHL-Softwareplattform und dem -Contentsystem zusammen. Die Plattform wird im Normalfall von einer zentralen Einrichtung der Hochschule, z. B. dem Rechenzentrum, administriert und steht einer Vielzahl von Organisationseinheiten zur Verfügung. Dabei kann die Einrichtung auch als Application Service Provider für Dritte fungieren.

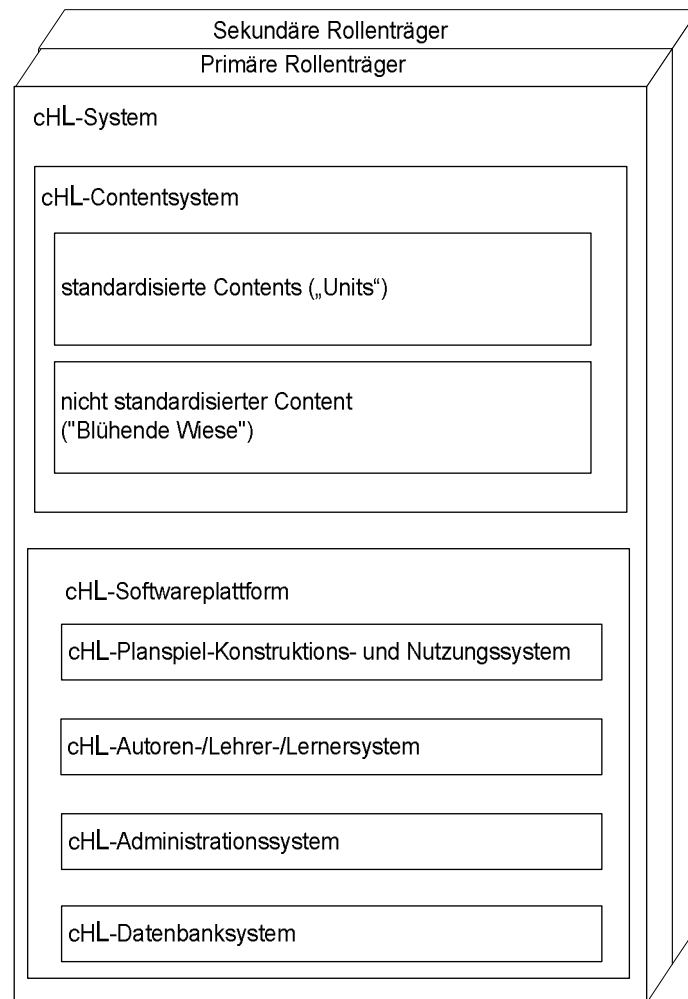


Abbildung 5: cHL-Ordnungsrahmen

Die Darstellung des Ordnungsrahmens dient als Grundlage für Detailübersichten (vgl. Abbildung 6). Als Beispiel wird gezeigt, welche Nutzer mit dem Autoren-/Lehrer-/Lerner (ALL)-System arbeiten.

Komponenten	Administrator	Autorenteam				Lehrteam			Lerner
		Autor	Designer	Programmierer	...	Professor	Content-Administrator	Internet-Mentor	
ALL-System	X	Y	–	–	–	X	X	–	Y

Legende

- X vollständige Berechtigung
- Y eingeschränkte Berechtigung
- keine Berechtigung

Abbildung 6: Detaillierung des Ordnungsrahmen (Auszug)

Zunächst ist die Funktionalität der zentral zur Verfügung gestellten cHL-Softwareplattform kurz zu erörtern. Sie enthält vier über Schnittstellen verbundene Komponenten.

- cHL-Datenbanksystem
- cHL-Administrationssystem
- cHL-Autoren-/Lehrer-/Lernersystem
- cHL-Planspiel-Konstruktions- und Nutzungssystem

Im cHL-Datenbanksystem werden multimediale Objekte gespeichert sowie Zugriffsrechte auf diese Objekte verwaltet. Die Objekte können aufgrund der vom Autoren vergebenen Zugriffsrechte zum Download angeboten werden.

Das cHL-Administrationssystem unterstützt die Logistik und die internetbasierte Kommunikation für Lehrveranstaltungen. Das Spektrum reicht vom Support für Präsenzvorlesungen bis hin zu virtuellen Kursen im Rahmen von Distance Learning.

Das cHL-Autoren-/Lehrer-/Lernersystem dient der Erstellung und Nutzung von Units, in denen das Wissen zu abgegrenzten Domänen in systematischer Form multimedial repräsentiert wird.

Das cHL-Planspiel-Konstruktionssystem dient der standardisierten Generierung von Planspielen. Aus dem in das System einzugebenden Content (insbesondere Dokumente, digitale Goodies) sowie den Elementen und Relationen des Reaktions- und des Abrechnungsmodells können Planspiele für die Präsenz- und Internetlehre erzeugt werden, die über das Planspielnutzungssystem durchgeführt werden können.

Zur Realisierung der cHL-Softwareplattform wurden bereits drei Softwareprodukte als OpenSource-Produkte entwickelt. Das vierte Produkt, das der Konstruktion und Nutzung von Planspielen dient, ist noch in der Entwicklung befindlich.

Teilsystem	Softwareprodukt
cHL-Autoren-/Lehrer-/Lernersystem	Freestyle Learning
cHL-Administrationssystem	OpenUSS
cHL-Datenbank	MIAMI

Abbildung 7: In Betrieb befindliche Softwareprodukte für das cHL-System

Zur Steigerung der Benutzerfreundlichkeit (Usability) wurde eine gegenseitige Aufrufbarkeit der Programme mit einer einmaligen Registrierung des Benutzers und einem jeweils einmaligen Login organisiert. Wo es sinnvoll erschien, wird nicht in das „Eingangsporta“ des aufgerufenen Programms, sondern in einen spezifischen Bereich des Administrationssystems verzweigt. Beispielsweise ist die Freestyle Learning Unit *VOFI* in denjenigen Teil des Administrationssystems OpenUSS verlinkt, in dem ein Diskussionsforum und ein Archiv zum Thema *VOFI* zur Verfügung gestellt wird [Gr01].

Das auf der cHL-Plattform zu errichtende zweite Teilsystem der Institution enthält den Content. Neben beliebig gestalteten Objekten („blühende Wiese“) sind hier vor allem standardisierte Units zu nennen, die – analog zu Büchern – auf einer CD („FSL Home“) oder DVD sowie im Internet („FSL Web“) unter Verwendung von OpenUSS angeboten werden. Das Contentangebot wird außerdem um Planspiele ergänzt, die beiden Teilsystemen zugerechnet werden können.

Der Content kann in der Datenbank MIAMI archiviert und verwaltet werden und ist durch das Administrationssystem unter Beachtung der Bestimmungen des Urheberrechts distribuierbar. In zunehmendem Maße wird der Content auch als Add-on zu einem Buch oder in Form einer CD oder DVD von wissenschaftlichen Verlagen vertrieben. Als Beispiel sei auf die im Rahmen des Wissensnetzwerks Controlling erarbeitete Publikation hingewiesen [Gr03].

Das cHL-System umfasst nicht nur die von der Hochschule und ihren Organisationseinheiten zur Verfügung gestellte cHL-Plattform und das Contentssystem – vergleichbar mit einer öffentlichen Bibliothek –, sondern auch die persönlichen cHL-Systeme der Nutzer, also der Autoren, Lehrer und Lerner. Deren Systeme resultieren aus den spezifischen Sichten auf das Gesamtsystem, die durch eine zentral gesteuerte Rechtevergabe sowie durch eine individuelle Sammlung von Medienobjekten zustande gekommen sind. Zum cHL-System der Nutzer zählen also nicht nur die über das Netz ansteuerbaren Plattformen sowie die zentral zur Verfügung gestellten Inhalte, sondern auch die lokal gespeicherten Daten und Programme, für deren Pflege und Wartung der Nutzer selbst verantwortlich ist. Zu seiner Unterstützung werden Empfehlungen gegeben, wie die FSL-Units individuell weiterverarbeitet werden und wie Verzeichnisstrukturen zur Organisation vorlesungsbezogener Contents aufzubauen sind. Festzuhalten ist also, dass jeder Nutzer somit über ein eigenes cHL-System verfügt. Auch hier bietet sich ein Vergleich zwischen einer straff organisierten öffentlichen Bibliothek und einer unbürokratisch privat geführten Bibliothek an.

Zur detaillierteren Darstellung einer Komponente der cHL-Plattform soll im Folgenden auf das spezifisch für die Hochschullehre entwickelte System OpenUSS eingegangen

werden, da dies die grundlegende Informationsverarbeitung in der Hochschullehre unterstützt.

3.3 Darstellung des cHL-Administrationssystems

3.3.1 Die Ausgangssituation

In wachsendem Maße wird die Präsenzlehre durch Nutzung des Internets unterstützt. Vorlesungsskripte, Übungsaufgaben und auch Lernsoftware werden dabei zum Download angeboten. Häufig werden diese zusammen mit beschreibenden HTML-Texten auf einem WWW-Server verfügbar gemacht. Die Studierenden „wandern“ üblicherweise von Homepage zu Homepage, um sich das vorlesungsspezifische Informationsangebot anzusehen und ggf. herunterzuladen. Dieser Ansatz führt sowohl bei den Hochschullehrern als auch bei den Studierenden zu einem unverhältnismäßig hohen Administrationsaufwand. Während die zu publizierenden Dokumente häufig mit Standardsoftwareprodukten (z. B. Microsoft PowerPoint) erzeugt werden, ist für den Publikationsprozess ein Spezialist mit HTML-Kenntnissen erforderlich, der auf der Homepage des Anbieters einen Verweis auf das herunterzuladende Dokument einfügt. Die Aktualität muss überwacht werden. Nur begrenzt ist eine Automatisierung möglich. Wegen der üblicherweise anzutreffenden Heterogenität der Internetauftritte kommt das Problem hinzu, dass Studierende unterschiedliche Sichten auf die Lerninhalte kognitiv zu verarbeiten haben. Prozessverluste treten somit nicht nur beim Anbieter, sondern auch beim Nachfrager von Informationen auf.

3.3.2 Ziele und Funktionen

Zur Handhabung dieser Probleme wurde am Institut für Wirtschaftsinformatik der Universität Münster das Softwaresystem OpenUSS (Open University Support System) als OpenSource-Produkt entwickelt. Die Software bietet konsistente personalisierte Sichten für Studierende und Lehrende, um die oben angesprochenen Prozessverluste zu vermeiden. Das OpenSource-Produkt, das von der Initiative CampusSource gefördert wurde, wird auf der Basis von GPL distribuiert. Bereits bei der Konstruktion von OpenUSS wurde von der Vision einer weltweiten Entwickler-Community ausgegangen.

OpenUSS implementiert ausgewählte Funktionen des Referenzmodells, die bereits in Kapitel 1 dargestellt wurden. Abbildung 8 stellt ein Anwendungsfalldiagramm für die Hauptakteure dar, das diese Funktionen spezifiziert. Asynchrone und synchrone Kommunikationskomponenten, wie beispielsweise Diskussionsforen und Chat, werden im Vergleich zum Referenzmodell zusätzlich berücksichtigt.

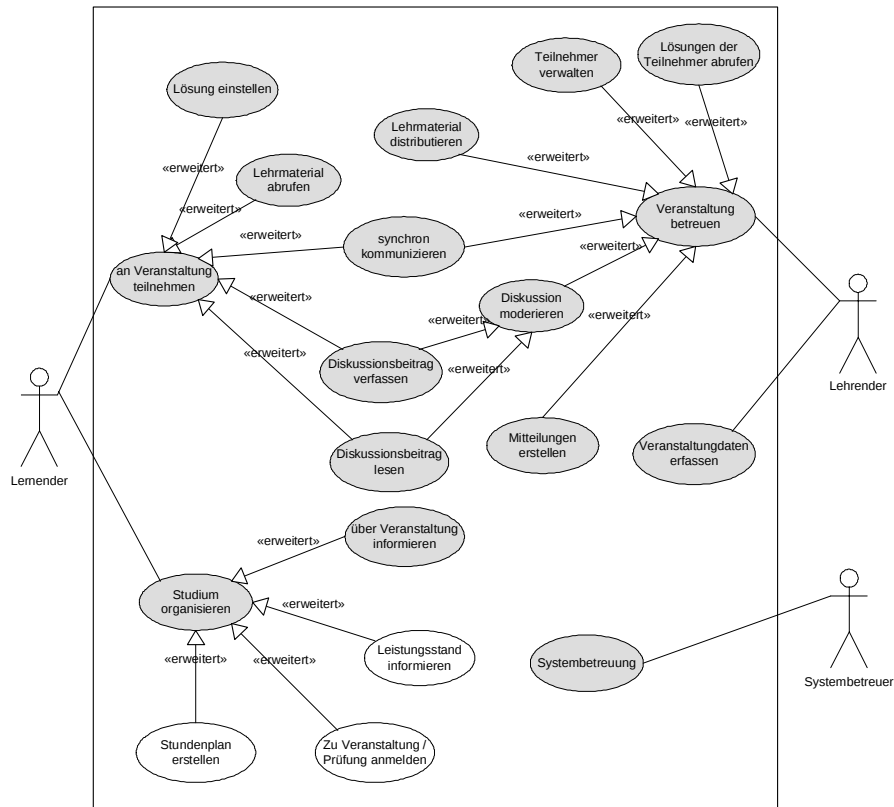


Abbildung 8: Anwendungsfalldiagramm OpenUSS

Zur Realisierung dieser Funktionen ist ein komponentenorientiertes Architekturmodell konzipiert worden [Br03, S. 200-211], das aus relativ homogenen Teilmodulen besteht. Zur Infrastruktur gehören *Service*-, *Basis*- und *Erweiterungskomponenten*. Die Servicekomponenten stellen die Infrastrukturdienste des gesamten Systems dar. So dient die Mailservice-Komponente dem Versenden von E-Mails, zu denen auch diejenigen gehören, die bei bestimmten Systemereignissen (z. B. „Ein neuer Diskussionsbeitrag ist geschrieben worden“) erzeugt werden. Die Basiskomponenten stellen die für den cHL-Bereich spezifischen („domänenabhängigen“) Komponenten dar. Die Erweiterungskomponenten dienen als domänenunabhängigen Komponenten, da sie nicht nur in einem cHL-Administrationssystem eingesetzt werden können. So kann die Komponente Diskussionsforum z. B. auch in einem Customer Relationship Management-System (CRM) genutzt werden. Diese Einteilung erfolgte im Hinblick auf die Wiederverwendung bereits entwickelten Komponenten in OpenUSS. Als Beispiel ist hierfür die Chat-Komponente „Babylon Chat“ zu nennen, die mittlerweile in OpenUSS integriert worden ist. Die Unabhängigkeit der Erweiterungskomponenten von Basiskomponenten stellt somit eine wichtige Eigenschaft dieser Architektur dar. Abbildung 9 präsentiert die Komponentenarchitektur von OpenUSS grafisch.

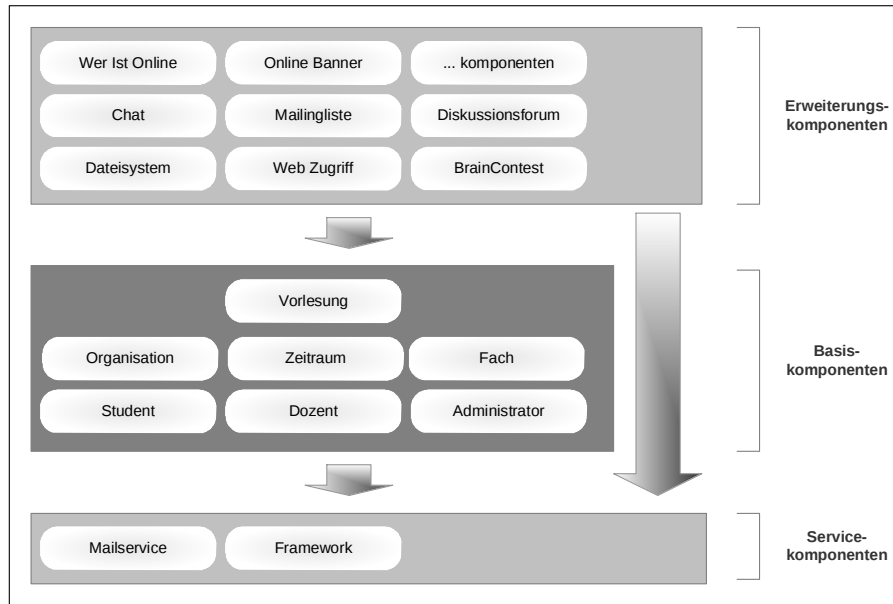


Abbildung 9: Komponentenarchitektur von OpenUSS (in Anlehnung an [BD01])

Zur Implementierung der Komponenten wurde Java 2 Enterprise Edition (J2EE) als Technologie ausgewählt. Damit sind nicht nur Plattformunabhängigkeit und Nutzung offener Standards, sondern auch die Verwendung der mehrschichtigen Architektur, die eine Einteilung in *Präsentations*-, *Geschäftsprozess*- und *Daten-Schicht* vorgibt, gewährleistet. Die Präsentations-Komponenten laufen innerhalb des Servlet-Containers und dienen ausschließlich zur Darstellung des Sachverhalts. Innerhalb des EJB-Containers (Enterprise Java Beans) werden die Geschäftsprozess-Komponenten ausgeführt und stellen die Implementierung der Geschäftslogik dar. Schließlich werden die Daten in einem Datenbanksystem gespeichert. Diese Einteilung ermöglicht eine gemeinsame Verwendung der Geschäftsprozesse mit diversen Präsentationstechniken, wie beispielsweise HTML-Browser und WAP-fähiges (Wireless Application Protocol) Handy, und wirkt sich somit auf die Wiederverwendung der bereits implementierten Geschäftsprozesse aus. Auch hier ist das Prinzip der Unabhängigkeit gewährleistet.

Zur Speicherung der Daten wurde eine relationale Datenbank aufgebaut. Auch hier besteht Unabhängigkeit. So können zur Datenspeicherung durchaus kommerzielle Datenbanksysteme (z. B. Oracle), aber auch OpenSource-Produkte (z. B. Firebird) verwendet werden. Auch bezüglich der untersten Ebene – der Betriebssystemplattform – besteht Unabhängigkeit, sodass auch das Betriebssystem Linux zum Einsatz kommen kann.

Eine weitere wichtige Eigenschaft ist in der Skalierbarkeit des Systems zu sehen, deren Vorteil darin besteht, dass keine mengenmäßigen Restriktionen bezüglich der Transaktionen mit dem System zu beachten sind. Durch Integration eines Verteilerprogramms (Load Balancer) ist der Einsatz einer beliebigen Anzahl leistungsfähiger Server möglich. In Abbildung 10 stehen beispielsweise vier Präsentations- und Geschäftsprozess-Schich-

ten redundant zur Verfügung, sodass die Anfragen der Benutzer zügig bearbeitet werden können. Im Produktionsbetrieb an der Universität Münster werden Enhydra als Servlet-Container, JOnAS als EJB-Container und Firebird als Datenbanksystem eingesetzt, die alle als OpenSource-Produkte zur Verfügung stehen.

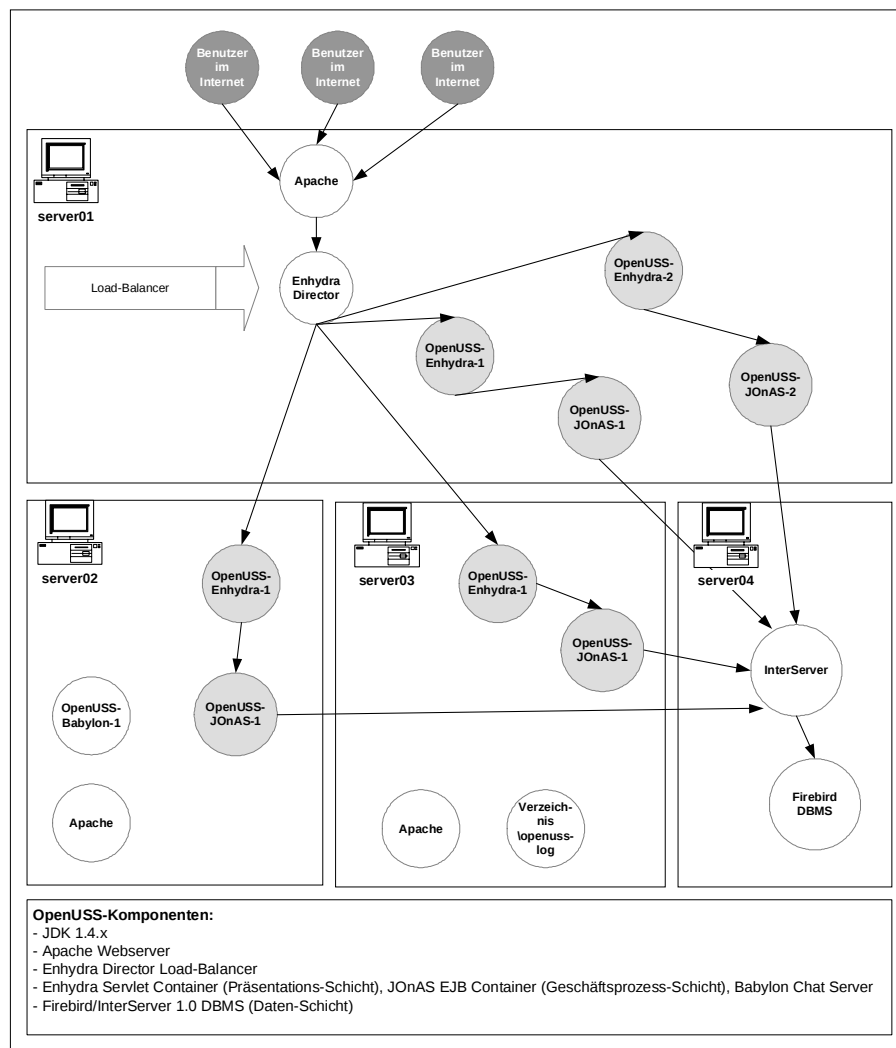


Abbildung 10: Verteilte Architektur von OpenUSS

Als Hardwarevoraussetzung für die Benutzer sieht OpenUSS grundsätzlich einen mit einem Standard-HTML-Browser ausgestatteten PC mit Internetanschluss vor (vgl. Abbildung 11). Auch für innovative Kommunikationstechnologien, wie z. B. WAP und SMS (Short Message Service), wurden Schnittstellen geschaffen, sodass Studierende und Dozierende auch mit Mobiltelefonen News zu ihren jeweiligen Veranstaltungen empfangen können.

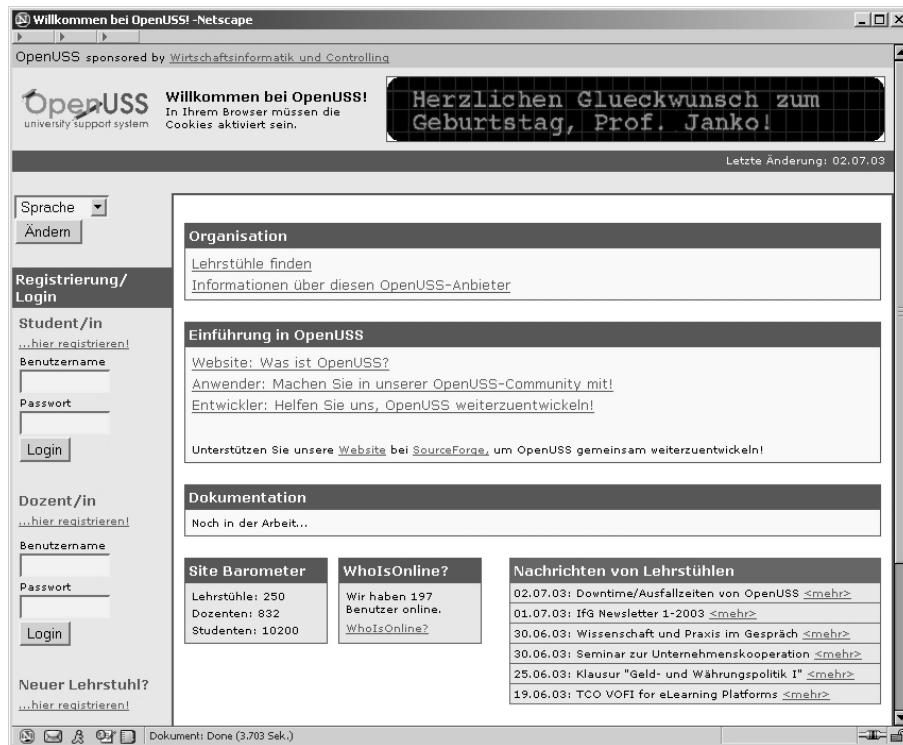


Abbildung 11: <http://www.openuss.de>

Zur verteilten Entwicklung von OpenUSS wurde die Entwicklungsplattform SourceForge.net ausgewählt. Auf dieser Plattform sind mehr als 40.000 Projekte für über 430.000 Entwickler gespeichert. SourceForge unterstützt sämtliche Phasen der verteilten Softwareentwicklung. Hierzu zählt neben der Analyse, der Entwurfs-, Implementierungs- und Testphase insbesondere die Hilfe bei der Koordinationsaufgabe. Aufgrund der Dokumentationsfunktion und der Möglichkeit, Fehler zu verfolgen, wird die Qualitätssicherung gefördert. Bereits in der Anfangsphase können Entwickler integriert werden.

Aufgrund von Protokollanalysen kann die Anzahl der Downloads sowie der Aufruf bestimmter Seiten festgehalten werden – indes sind diese Kennzahlen nur wenig geeignet, um die Leistungsfähigkeit zur Beurteilung der Weiterentwicklung eines OpenSource-Produktes beurteilen zu können. Trotz der offenen Philosophie ist eine hierarchische Entwicklerstruktur zur Steigerung der Effektivität und Effizienz des Projektes erforderlich. Von besonderer Bedeutung ist hierbei die Präsentation im Rahmen von Fachtagungen und -publikationen, um die Community durch neue Akteure zu erweitern, die bereit sind, weitere Komponenten zu entwickeln. Um eine unwirtschaftliche Parallelentwicklung zu vermeiden, müssen Absprachen getroffen werden, deren Ergebnisse im Rahmen eines Internet-Administrationssystems transparent zu machen sind. Dass sich hierfür OpenUSS als Plattform für eine verteilte Weiterentwicklung eignet, ist evident.

4 Ausblick

Die Bedeutung von OpenSource-Software in der Wirtschaft nimmt seit der weltweiten Verbreitung von Systemsoftware, wie z. B. Linux, permanent zu. Auch an Hochschulen wurde die OpenSource-Idee aufgegriffen. Zum einen wird dort mit hoher Intensität Software im Rahmen von Forschungs- und Entwicklungsprojekten erstellt; zum anderen werden in zunehmendem Maße OpenSource-Produkte für die computergestützte Hochschullehre eingesetzt. Im Idealfall wird sie dann im Rahmen einer Community bedarfsgerecht weiterentwickelt. Dies gilt grundsätzlich auch für die hier dargestellte cHL-Plattform.

Der OpenSource-Gedanke ist eng verbunden mit Richard Stallmann, der das Projekt GNU initiierte und dafür die General Public License (GPL) konzipierte. GPL ist nach wie vor von aktueller Bedeutung. Die Lizenz erlaubt unter bestimmten Bedingungen die Erstellung und Verbreitung von Kopien des Quellcodes eines Softwareproduktes. Zu den Bedingungen gehört insbesondere, dass der Copyright-Vermerk des Entwicklers bestehen bleibt. Auch sind Haftungsfragen ausgeschlossen.

Dem potenziellen Nutzer wird die Software zum Download ohne Lizenzgebühren zur Verfügung gestellt. Kosten im Sinne von Gebühren können für Distributionsleistungen des Anbieters, wie z. B. Beratungen, Schulungen, Zurverfügungstellung von Manuals, erhoben werden. Insofern bieten OpenSource-Produkte auch die Möglichkeit, spezifische Geschäftsmodelle zu entwickeln. Diese Option gilt nicht etwa nur für den Entwickler, vielmehr kann sich jeder Softwareberater auf einem solchen Geschäftsfeld betätigen. Für den Entwickler ist es – und hierin ist ein nicht zu unterschätzender ideeller Anreiz zu sehen – in jedem Fall ein erheblicher Reputationsgewinn, wenn „seine“ Software von Dritten genutzt wird.

Die GPL gestattet nicht nur die Nutzung von Kopien, sondern auch die Weiterentwicklung der Software. Die Modifikationen müssen allerdings deutlich gemacht werden. Die erweiterte Software kann vervielfältigt und distribuiert werden – die Verbreitung darf jedoch nur im Rahmen der GPL erfolgen.

Aufgrund der Besonderheit des Entwicklungsprozesses weist OpenSource-Software spezifische technologische Eigenschaften auf: Die Transparenz des Entwicklungsprozesses führt regelmäßig zu einem relativ hohen Sicherheitsniveau, da die Softwareentwickler Sicherheitslücken kontrollieren und kommunizieren. Die modulare Architektur begünstigt nicht nur die Erweiterung des Quellcodes, sondern auch die Wiederverwendung. Dabei lässt sich feststellen, dass die Softwarekosten bei der Erstellung wieder verwendbarer Komponenten regelmäßig höher sind als bei einer so genannten „Einwegversion“. Positiv herauszustellen ist die Verwendung offener Standards bei OpenSource-Produkten. Sie vermeiden herstellersistem spezifische Systembindungen (Lock-In-Effekte). Systemspezifische Investitionen sind wegen der Plattformunabhängigkeit nicht erforderlich. Die OpenSource-Software ist also nicht nur kostenlos, sondern auch kostengünstig im Hinblick auf Folgeinvestitionen. Dieses Phänomen begünstigt den Total Cost of Ownership (TCO) von OpenSource-Software [GL03].

Für den Bereich der computergestützten Hochschullehre bietet OpenSource-Software eine besondere Chance für Hochschulen. Nicht nur der Vorteil eines relativ geringen TCOs, sondern auch der universitätsübergreifende einheitliche Standard, der insbesondere dann relevant ist, wenn die Mitwirkung im Rahmen einer Community erfolgt, ist als positives Moment herauszustellen.

Literaturverzeichnis

- [Al02] Alpar, P., Grob, H. L., Weimann, P., Winter, R.: Anwendungsorientierte Wirtschaftsinformatik – Strategische Planung, Entwicklung und Nutzung von Informations- und Kommunikationssystemen, 3., überarb. u. erw. Aufl., Braunschweig, Wiesbaden 2002.
- [BD01] Bensberg, F.; Dewanto, L.: Entwurfsmuster bei der Implementierung von OpenUSS, Teil 1: Mustergültig, in: Java Magazin, Ausgabe 12/2001, S. 70-74.
- [Br03] Brocke, J. vom: Referenzmodellierung, Gestaltung und Verteilung von Konstruktionsprozessen, Berlin 2003, zugl. Univ. Diss., Münster 2002.
- [Do02] Doberkat, E. E. et al.: Anforderungen an eine eLearning-Plattform – Innovation und Integration, Dortmund, 2002.
- [Gr01] Grob, H. L.: Einführung in die Investitionsrechnung – Eine Fallstudiengeschichte, 4., vollst. überarb. u. erw. Aufl., München, 2001 (mit FSL Unit VOFI).
- [Gr03] Grob, H. L.; Brocke, J. vom; Lahme, N.; Wahn, M. (Hrsg.): Wissensnetzwerk Controlling mit DVD, München, 2003.
- [GB98] Grob, H. L.; Bielezke, S.: Aufbruch in die Informationsgesellschaft, 2., vollst. überarb. u. erw. Aufl., Münster, 1998.
- [GL03] Grob, H. L.; Lahme, N.: Total Cost of Ownership-Analyse mit vollständigen Finanzplänen, in: Zeitschrift Controlling, 2003 (in Vorbereitung).
- [PRR99] Probst, G.; Raub, S.; Romhardt, K.: Wissen Managen – Wie Unternehmen ihre wertvollste Ressource optimal nutzen, 3. Aufl., Wiesbaden u. a., 1999.
- [Sch01] Schmidt, J.: Eröffnungsbeitrag zum cHL-Tag 2000, in: cHL – computergestützte Hochschullehre, Dokumentation zum cHL-Tag 2000, Alma Mater Multimedialis, Hrsg.: H. L. Grob, Münster u.a. 2001, S. 3-5.
- [Sc03] Schmidt, B.: Das Konzept der computergestützten Hochschullehre: Wirkungen, Chancen und Risiken – Ergebnisse einer Expertenbefragung (Teil I), Arbeitsbericht Nr. 23 der Reihe „CAL+CAT“, Hrsg.: H. L. Grob, Münster 2003 (in Vorbereitung).