



Computergestützte Hochschullehre – das cHL-Konzept

Heinz Lothar Grob, Frank Bensberg

Institut für Wirtschaftsinformatik
Westfälische Wilhelms-Universität Münster
Leonardo-Campus 3
D-48149 Münster
grob@uni-muenster.de
bensberg@uni-muenster.de



Zusammenfassung: 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, dessen Architektur Subsysteme umfasst, die primär die Rollen des Studierenden und des Lehrenden unterstützen. Im Rahmen dieses Beitrags werden die Komponenten der cHL-Rahmenarchitektur skizziert. Von grundlegender Bedeutung für die informationstechnologische Unterstützung universitärer Lehr- und Lernprozesse ist die Verfügbarkeit einer geeigneten Softwareplattform, die administrative Funktionen bereitstellt. Am Institut für Wirtschaftsinformatik wurde zu diesem Zweck das Open Source-Softwareprodukt OpenUSS (Open University Support System) entwickelt, das in diesem Beitrag schwerpunktmäßig dargestellt wird, zumal diese Komponente auch zunehmend international eingesetzt wird.

1 Aufgabenbereich der cHL

Zu den Kernaufgaben der Hochschule gehören Forschung, Lehre und Weiterbildung. Diese Primärbereiche umfassen eine Vielzahl von Aufgaben und Rollenträgern, deren Unterstützung mithilfe von Informations- und Kommunikationssystemen ein wesentlicher Erfolgsfaktor für die universitäre Ausbildung darstellt. Zu diesem Zweck sind Anwendungsarchitekturen notwendig, die im Sinne einer zielorientierten Konstruktion so zu gestalten sind, dass eine effektive und effiziente Unterstützung von Lehr- und Forschungsprozessen sichergestellt wird.^{1,2} Darüber hinaus ist zu berücksichtigen, dass diese Anwendungsarchitekturen notwendigerweise Schnittstellen zur Hochschulverwaltung und begleitenden Diensten aufweisen müssen, die Sekundärprozesse für den akademischen Bereich erbringen. In der folgenden Abbildung wird in enger Anlehnung an DOBERKAT ET AL. ein Referenzmodell dargestellt, das diesen Zusammenhang zum Ausdruck bringt.³

Zur informationstechnologischen Unterstützung der skizzierten Lehr- und Lernprozesse werden in praxi häufig internetbasierte Informationssysteme eingesetzt, die als E-Lear-

¹ Zum Begriff der Anwendungsarchitektur vgl. [AI02], S. 151 ff.

² Das cHL-Konzept bildet den Fokus der an der Universität Münster vertretenen Multimedia-Strategie. Vgl. hierzu [Gr01]. Zur Integration des cHL-Konzepts in die Informations-, Kommunikations- und Medienarchitektur der Universität Münster vgl. [An03].

³ Vgl. [Do02], S. 18. Zum Begriff des Referenzmodells vgl. [Br02].

⁴ In enger Anlehnung an [Do02], S. 18.



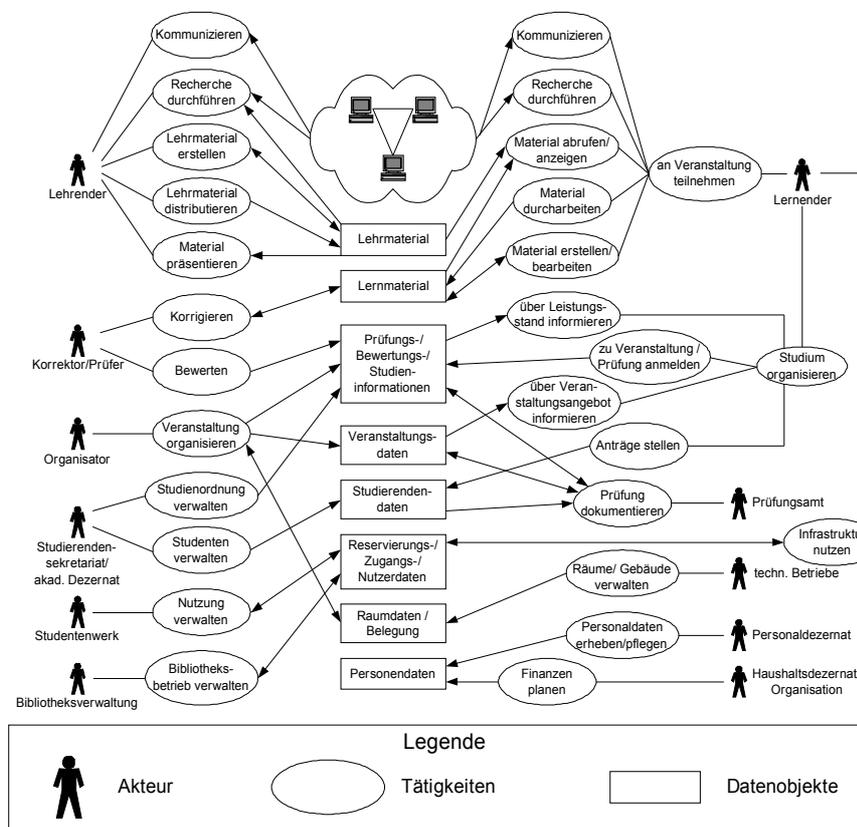


Abbildung 1: Referenzmodell für den Lehrbereich⁴

ning-Plattformen bezeichnet werden und als Kernfunktionen die Distribution und Publikation von Lerninhalten für bestimmte Zielgruppen gestatten. Der Anbietermarkt für E-Learning-Plattformen ist stark fragmentiert und umfasst neben zahlreichen kommerziellen Softwareprodukten auch in zunehmendem Maße Open Source-Lösungen.⁵

Zwar leisten diese E-Learning-Plattformen in Bezug auf das dargestellte Referenzmodell eine Unterstützung der beiden primären Rollenträger *Lehrender* und *Lernender*, doch wird das Problem der Integration mit historisch gewachsenen universitären Anwendungsarchitekturen meist vernachlässigt. Zur Herstellung der Interoperabilität zwischen bestehenden Administrationssystemen und E-Learning-Plattformen müssen jedoch Schnittstellen entwickelt werden, um z.B. ungeplante Redundanzen bezüglich der Datenhaltung zu vermeiden. Ein weiteres Problem bei der Einführung von E-Learning-Plattformen besteht darin, dass es sich meist um monolithisch konstruierte Produkte handelt, die nur ein starres

⁵ Vgl. [Ca01]; [BHM02];[MW03]. Eine Reihe von Open Source-Lösungen steht durch den Mediatoren CampusSource zur Verfügung. Vgl. <http://www.campussource.de>.

Funktionsangebot bieten.⁶ Dabei führen gerade die an Hochschulen anzutreffenden stark divergierenden fachlichen Anforderungsprofile dazu, dass i.d.R. kein konkretes Softwareprodukt die Anforderungskriterien sämtlicher Anwender sinnvoll erfüllen kann.

Infolgedessen sind domänenspezifische Konzepte notwendig, die für eine Integration unterschiedlichster E-Learning-Komponenten sorgen. Zu diesem Zweck wurde am Institut für Wirtschaftsinformatik der Westfälischen Wilhelms-Universität Münster das cHL-Konzept als Rahmenarchitektur für die computergestützte Hochschullehre entwickelt.

2 Das Konzept der computergestützten Hochschullehre (cHL)

Die cHL-Rahmenarchitektur beinhaltet die Klassifizierung von informationstechnologischen Komponenten und deren Nutzung durch die primären und sekundären Rollenträger. Zu den primären Rollenträgern sind die Autoren, Lehrer und Lerner zu zählen. Wichtige Zusatzaufgaben werden von EDV- und Content-Administratoren wahrgenommen. Während der EDV-Administrator informationstechnische Aufgaben durchführt, 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 wird nun anhand eines Ordnungsrahmens verdeutlicht.

Das der Organisationseinheit (z. B. ein Lehrstuhl) 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 eine Hochschule auch als Anbieter anwendungsbezogener Dienste für Dritte (Application Service Provider) fungieren. Auf diese Weise können z. B. Weiterbildungsangebote für hochschulexterne Zielgruppen erschlossen werden. Der Ordnungsrahmen ist in der folgenden Grafik dokumentiert worden (vgl. Abbildung 2).

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

- cHL-Datenbank
- cHL-Administrationssystem
- cHL-Planspiel-Konstruktionssystem
- cHL-Autoren-/Lehrer-/Lernersystem

In der cHL-Datenbank werden multimediale Objekte gespeichert sowie Zugriffsrechte auf diese Objekte verwaltet. Die Objekte können aufgrund der vom Autor vergebenen Zugriffsrechte zum Download angeboten werden. Bei der Bestimmung der Zugriffsrechte sind insbesondere die jeweils geltenden urheberrechtlichen Bestimmungen zu beachten.

⁶ Vgl. [GMS03], S. 2.

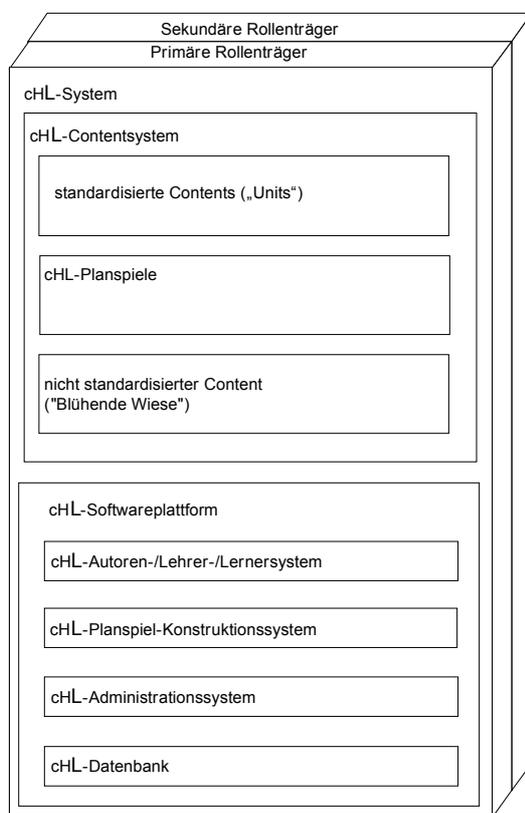


Abbildung 2: cHL-Ordnungsrahmen

Dazu ist festzustellen, dass durch die Publikation und Distribution von rechtlich geschützten Lernmaterialien Vervielfältigungsvorgänge stattfinden, deren urheberrechtliche Bedeutung bislang nur unzureichend berücksichtigt wird. Nach den derzeit geltenden Bestimmungen des Urheberrechtsgesetzes (UrhG) dürfen Lehrende geschützte Werke zur Veranschaulichung „im Unterricht“ öffentlich zugänglich machen (§52a, Abs. 1, Nr.1 UrhG). Dies umfasst nicht nur die Digitalisierung und Großbildprojektion von urheberrechtlich geschützten Werken im Rahmen von Präsenzveranstaltungen, sondern vielmehr auch die Werkwiedergabe im Rahmen moderner Distance Learning-Veranstaltungen über das Internet. Gemäß den Bestimmungen des §52a UrhG ist dabei die öffentliche Zugänglichmachung durch ein Intranet explizit auf einen zu begrenzenden Kreis von Teilnehmern zu beschränken. Diese gesetzliche Forderung ist durch die cHL-Datenbank zu gewährleisten.

Das cHL-Administrationssystem unterstützt die Informationslogistik und die internetbasierte Kommunikation für Lehrveranstaltungen. Das Spektrum reicht von unterstützenden Diensten für Präsenzvorlesungen bis hin zu virtuellen Kursen im Rahmen des Distance Learning. Auf diese Weise können auch hybride Lehr- und Lernkonzepte wie z. B. das Blended Learning verwirklicht werden.

Planspiele sind eine Lehr-/Lernmethode, bei der fiktive Probleme systematisch erzeugt und im Team gehandhabt werden. Die Durchführung moderner Planspiele erfolgt generell computergestützt und setzt die Verfügbarkeit standardisierter Werkzeuge zum Entwurf von On-line- und Off-line-Planspielen voraus. Diesem Zweck dient das cHL-Planspiel-Konstruktionssystem.

Das cHL-Autoren-/Lehrer-/Lernersystem eignet sich zur Erstellung und Nutzung von Lern-/Lehrmodulen, in denen das Wissen zu abgegrenzten Domänen in systematischer Form multimedial repräsentiert wird.

Das auf der cHL-Softwareplattform zu errichtende zweite Teilsystem enthält den Content. Neben beliebig gestalteten Objekten („blühende Wiese“⁷) sind hier vor allem standardisierte Lern-/Lehrereinheiten zu nennen, die – analog zu Büchern – auf einer CD oder DVD sowie im Internet angeboten werden.

Das cHL-System umfasst nicht nur die von der Hochschule und ihren Organisationseinheiten zur Verfügung gestellte cHL-Plattform und das Contentssystem⁸, sondern auch die persönlichen cHL-Systeme der Nutzer, also der primären Rollenträger wie Autoren, Lehrer und Lerner. Deren Systeme resultieren aus den spezifischen Sichten auf das Gesamtsystem, die durch eine zentral gesteuerte Rechtevergabe sowie durch 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 jeder Nutzer grundsätzlich selbst verantwortlich ist. Festzuhalten ist, dass jeder Nutzer somit über ein eigenes cHL-System verfügt.⁹

Zur technischen Implementierung der cHL-Softwareplattform wurden drei eigenständige Softwareprodukte, die unabhängig voneinander als Open Source-Produkte entwickelt worden sind, integriert. Diese bereits im Einsatz befindlichen Systeme werden in der folgenden Abbildung dargestellt. Das Planspiel-Konstruktionssystem befindet sich noch in der Entwicklung.

Als Interaktionsplattform mit den Lehr- und Lerninhalten wird seitens der Lehrenden und Lernenden das Learning Content System (LCS) *Freestyle Learning* genutzt. Die Koordination der Akteure erfolgt auf Basis des Learning Management Systems (LMS) *OpenUSS* (*Open University Support System*), das zur Administration von Lehr- und Lernprozessen dient. Abgerundet wird die Architektur durch eine Dokumentations- und Speicherschicht (Content Storage System, CSS), die mit dem System *MIAMI* (*Münstersches Informations- und Archivsystem für Multimediale Inhalte*) realisiert wird. Das MIAMI-System basiert auf dem Softwareprodukt *MILESS*, das an der Universität Essen entwickelt wurde und im Rahmen der Initiative CampusSource als Open Source-Produkt zur Verfügung gestellt wird. MIAMI ist im Kern als Digitale Bibliothek konzipiert, die vor allem lokal erzeugte digitale Dokumente im direkten Zugriff mit komfortabler Recherchemöglichkeit vorhält.

⁷ Dieser Begriff geht auf E. Doberkat zurück.

⁸ Vergleichbar mit einer öffentlichen Bibliothek.

⁹ Auch hier bietet sich ein Vergleich zwischen einer straff organisierten öffentlichen Bibliothek und einer unbürokratisch geführten privaten Bibliothek an.

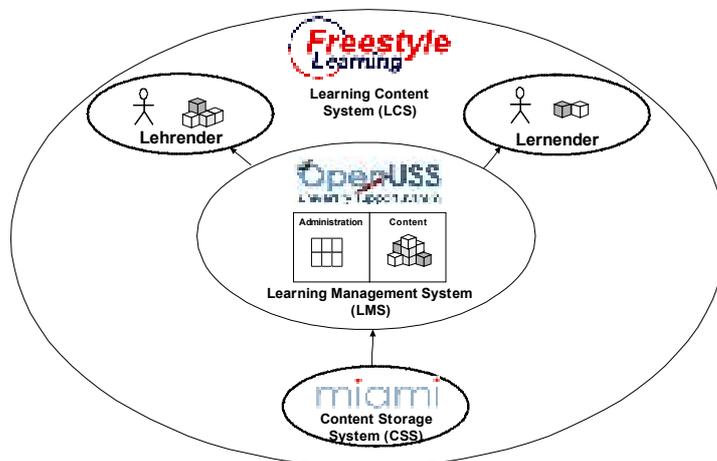


Abbildung 3: Softwareprodukte für das cHL-System

Dabei ist MIAMI auf die gleichberechtigte Behandlung aller medialen Formate (Text, Grafik, Video, Audio) angelegt und kann in andere Systeme als Content Server einbezogen werden kann.

Bei der detaillierteren Darstellung der cHL-Plattform soll im Folgenden auf das spezifisch für die Hochschullehre entwickelte System OpenUSS eingegangen werden.

3 Das cHL-Administrationssystem OpenUSS

3.1 Zielsetzung

Die Präsenzlehre wird seit Mitte der 90er Jahre durch Nutzung des Internets unterstützt.¹⁰ Vorlesungsskripte, Übungsaufgaben und auch Lernsoftware werden per Internet zum Download angeboten. Häufig werden diese zusammen mit beschreibenden HTML-Texten auf einem WWW-Server verfügbar gemacht. Zur Deckung des individuellen Informationsbedarfs müssen Studierende die individuellen Web-Präsenzen der jeweiligen Organisationseinheit (z.B. Lehrstuhl) aufsuchen und die benötigten Materialien herunterladen. 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, stehen für den Publikationsprozess meist keine komfortablen Softwareprodukte zur Verfügung, die ein adäquates Content Management ermöglichen. Vielmehr werden häufig noch Spezialisten mit HTML-Kenntnissen benötigt, die die zu publizierenden Dokumente auf der Web-Präsenz des Anbieters einpflegen. Wegen der üblicherweise anzutreffenden Heterogenität der Internetauftritte kommt das Problem hinzu, dass Studierende unterschiedliche Sichten auf die Lerninhalte kognitiv zu verarbeiten haben. Prozessverlust-

¹⁰ Vgl. [GG95].

te treten somit nicht nur beim Anbieter, sondern auch beim Nachfrager von Informationen auf.

3.2 Funktionen

Zur Handhabung dieser Probleme wurde am Institut für Wirtschaftsinformatik der Universität Münster das Softwaresystem OpenUSS (Open University Support System) als Open Source-Produkt entwickelt.¹¹ Die Software bietet zur Vermeidung von Prozessverlusten konsistente personalisierte Sichten für Studierende und Lehrende. Insgesamt werden durch das System die in der folgenden Abbildung dargestellten Funktionalitäten zur Verfügung gestellt.

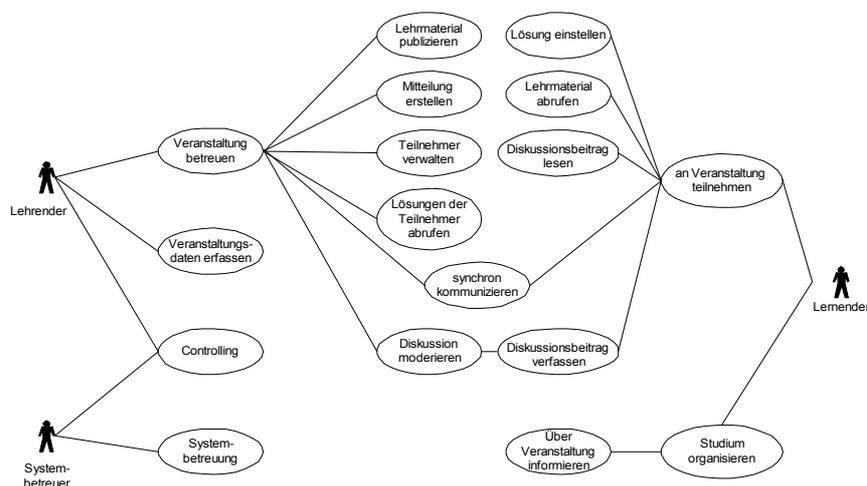


Abbildung 4: Funktionsangebot von OpenUSS¹²

Der Systemzugang erfolgt über einen WWW-Browser über das OpenUSS-Portal (vgl. Abbildung 4).¹³ Bei der ersten Kontaktaufnahme hat sich der Anwender – also der Dozierende bzw. der Studierende – zu registrieren. Während für den Dozenten vorab interne organisatorische Prozesse zur Vergabe von Rechten abzulaufen haben, kann der Studierende im Normalfall ohne Restriktionen Teilnehmer des Systems werden.

Neben dieser rollenbezogenen Personalisierung verfügt das System über weitere Mechanismen, die eine Individualisierung der Lehr- und Lernsichten erlauben. So hat der Anwender die Möglichkeit, das gewünschte Informationsangebot auf Grund seiner persön-

¹¹ Vgl. [GB02].

¹² Die Grafik macht deutlich, dass es sich bei OpenUSS um das Kernsystem des Lehrbereichs handelt. Vgl. Abbildung 1 dieses Beitrags.

¹³ Die OpenUSS-Instanz der Westfälischen Wilhelms-Universität ist über die URL <http://www.openuss.deerreichbar>.



Abbildung 5: Das OpenUSS-Portal

lichen Präferenzen zusammenzustellen. Dies erfolgt auf der Grundlage eines eigenen Profils, das Organisationseinheiten (z. B. Lehrstühle) und Themenbereiche registriert. In der folgenden Abbildung wird ein Beispiel für ein dozentenbezogenes Profil dargestellt, das bestimmte Lehrstühle und Veranstaltungen (z. B. Seminare, Vorlesungen) umfasst.

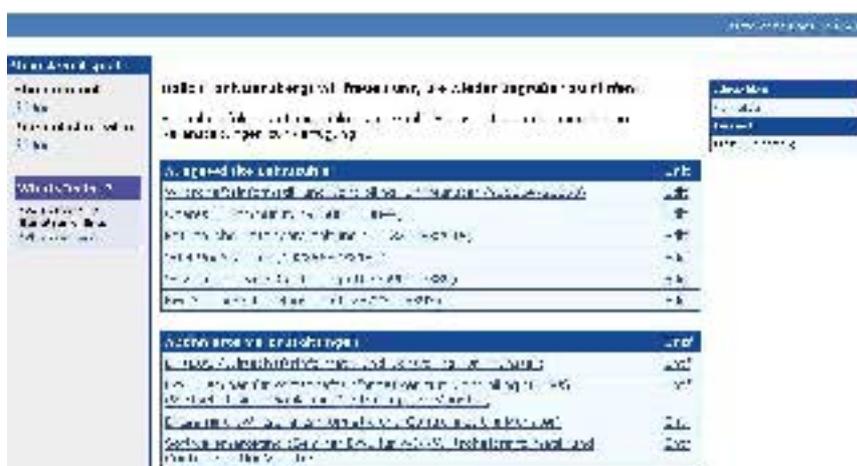


Abbildung 6: Personalisierte Dozentensicht auf Lehrstühle und Veranstaltungen

Nach der Auswahl aus dem registrierten Informationsangebot können entsprechende Administrationsfunktionen wahrgenommen werden. Zu diesen Funktionen gehören Kommunikationsfunktionen wie z. B. der Besuch eines veranstaltungsbezogenen Chat-Bereichs, die aktive oder passive Teilnahme am Diskussionsforum und das Abonnieren eines E-Mail-Verteilers. Darüber hinaus werden nach der Selektion einer Vorlesung automatisch veranstaltungsbezogene Mitteilungen (z. B. Terminänderungen, Vortragsankündigungen) eingeblendet. Auf diese Weise wird für die Studierenden eine homogene Sicht auf sämtliche veranstaltungsrelevanten Informationen erzeugt. Aus der gleichen Sicht kann auch der Zugriff auf die verfügbaren Lernmaterialien der selektierten Veranstaltung vorgenommen werden. Zu jeder vom Dozenten angebotenen Lehrveranstaltung kann der Umfang der Administrationsfunktionen per Customizing zusammengestellt werden. Nicht benötigte Funktionen werden hierdurch deaktiviert.

3.3 Architektur

Zur Realisierung von OpenUSS wurde ein komponentenorientiertes Architekturmodell konzipiert, das aus relativ homogenen Teilmodulen besteht. Zur Infrastruktur gehören Basiskomponenten (Foundation Components) und Erweiterungskomponenten (Extension bzw. Pluggable Components). Diese Komponentenarchitektur wird in der folgenden Abbildung dargestellt.

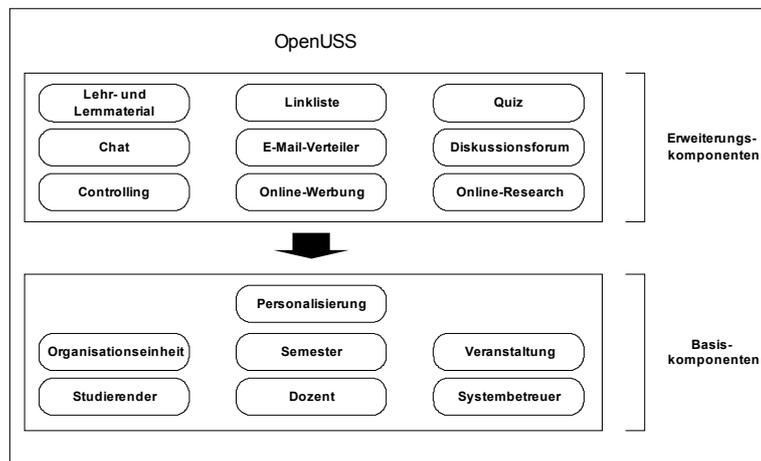


Abbildung 7: Komponentenarchitektur von OpenUSS

Das dargestellte Architekturkonzept ermöglicht zum einen eine einfache Erweiterung der Basiskomponenten. Durch die klar definierten Schnittstellen der Basiskomponenten können neue Erweiterungskomponenten effizient entworfen und implementiert werden. Darüber hinaus kann das Funktionsangebot des Gesamtsystems unter Verwendung externer Komponentensammlungen, die im Rahmen der Open Source zur Verfügung stehen, gene-

risch erweitert werden. In Anwendungsdomänen wie der computergestützten Hochschullehre, die durch dynamische Anforderungsszenarien gekennzeichnet ist, eröffnet dieses Architekturkonzept einen Handlungsspielraum, der eine evolutionäre Weiterentwicklung des Gesamtsystems gestattet.

Zur Förderung von Entwicklungsaktivitäten im Rahmen des Open Source-Prinzips wurde der vollständige Quelltext von OpenUSS über die Entwicklungsplattform SourceForge.net publiziert.¹⁴ Diese Plattform wurde ausgewählt, da sie sämtliche Phasen der verteilten Softwareentwicklung systematisch unterstützt. Hierzu zählt neben der Analyse, der Entwurfs-, Implementierungs- und Testphase insbesondere die Unterstützung bei der Koordination geografisch verteilter Entwicklergruppen. SourceForge stellt hierfür ein Werkzeug zur Revisionsverwaltung (Concurrent Versions System, CVS) zur Verfügung, das die ortsverteilte Arbeit an Programmtexten koordiniert. Sämtliche Quelltexte von OpenUSS werden in einem zentralen Repository gespeichert, aus dem registrierte Entwickler die jeweils neuesten Versionen einzelner Programmdateien beziehen können (Checkout). Nach Modifikation bzw. Erweiterung der Programmdatei kann der Code wieder im Repository gespeichert werden (Commit) und steht den anderen Entwicklern zur Nutzung oder Modifikation Verfügung. Auf diese Weise wurden weltweit zahlreiche sequenzielle – d.h. aufeinander aufbauende – Weiterentwicklungen von OpenUSS durchgeführt, die mittlerweile zum Kernbestandteil der aktuellen OpenUSS-Distribution gehören.

3.4 Systemmanagement

Zur Implementierung von OpenUSS wurde die Technologie Java 2 Enterprise Edition (J2EE) ausgewählt. Hierdurch wird zwar ein hohes Maß an Plattformunabhängigkeit durch Nutzung offener Standards gewährleistet, doch erfordert diese Technologie gerade bei lastintensiven Anwendungen ein proaktives Systemmanagement. Diese Problematik wird anhand der folgenden Abbildung deutlich, die die Anzahl der täglichen Benutzersitzungen der OpenUSS-Instanz der Westfälischen Wilhelms-Universität Münster für den Zeitraum vom 22.08.2002 bis zum 10.04.2003 wiedergibt.

Wie aus der Abbildung deutlich wird, weist das System, das mittlerweile mehr als 10.000 registrierte studentische Nutzer verzeichnet, während des Semesters erwartungsgemäß deutliche Lastspitzen auf. Typischerweise kann diese Situation zu funktionalen Fehlern, Transaktionsverlusten und unzumutbar langen Antwortzeiten auf Nutzerseite führen. Um diese Probleme zu vermeiden, wird eine Lastverteilung (load balancing) auf mehrere Rechnerknoten vorgenommen. Da dieses Verteilungskonzept von der J2EE-Technologie explizit unterstützt wird, können die einzelnen Komponenten von OpenUSS auf unterschiedlichen Serversystemen zur Ausführung gebracht werden. Derzeit wird das System auf der Basis von vier Intel-Dualprozessorsystemen betrieben. Eine Verteilung auf weitere Rechnersysteme ist problemlos möglich, sodass bezüglich der Transaktionen keine mengenmäßigen Restriktionen bestehen und hohe Skalierbarkeit gegeben ist.

Neben dem technisch orientierten Systemmanagement ist zur erfolgreichen Einführung und Nutzung von E-Learning-Plattformen in zunehmendem Maße auch eine fachliche

¹⁴ Vgl. hierzu <http://openuss.sourceforge.net>.

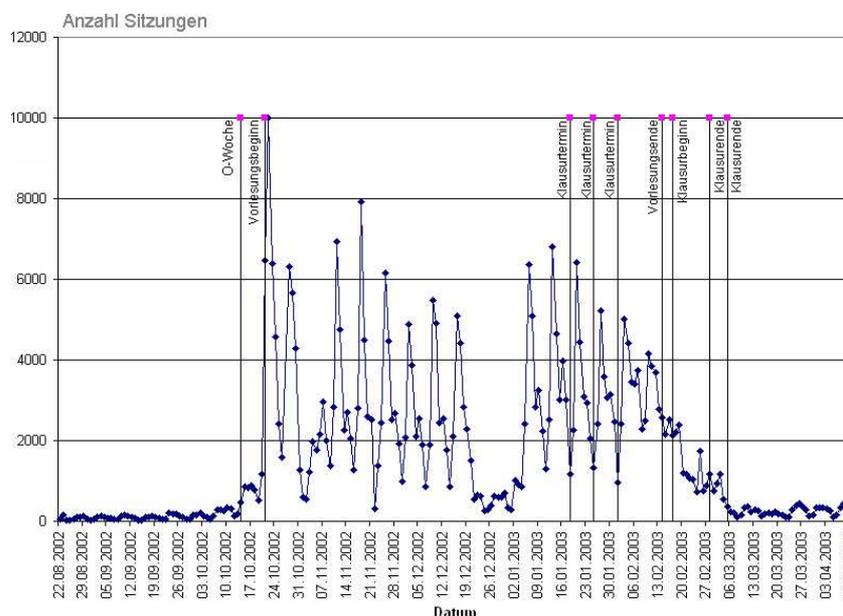


Abbildung 8: Anzahl der OpenUSS-Sitzungen pro Tag

Analyse des Nutzerverhaltens notwendig.¹⁵ So weisen E-Learning-Plattformen die Charakteristik auf, dass die Interaktion zwischen den Rollenträgern mediatisiert erfolgt. Dies führt dazu, dass sich der Studierende dem Wahrnehmungsbereich des Dozenten entzieht. Um jedoch eine zielorientierte Nutzung von E-Learning-Plattformen sicherzustellen, benötigt der Dozent Informationen über das Nutzungsverhalten der Studierenden. Dieses Anforderungskriterium wird vom Leistungsspektrum vieler E-Learning-Plattformen nicht abgedeckt, so dass diese Systeme aus Anwendersicht den Charakter einer Black Box besitzen. Im Rahmen von OpenUSS wurde zu diesem Zweck eine *Controlling*-Komponente entwickelt (vgl. Abbildung 7), die mittels definierter Messpunkte Daten über das systembezogene Verhalten der primären Akteure liefert.

Um Dozenten und Systembetreuern Einblick in die Nutzeroperationen zu liefern, sind geeignete Kennzahlen notwendig, die relevante Sachverhalte in konzentrierter Form erfassen.¹⁶ Zu diesem Zweck ist es zunächst erforderlich, die Dimensionen der kennzahlenorientierten Analyse zu strukturieren. Diese Strukturierung wird in der folgenden Abbildung vorgenommen.

Wie aus Abbildung 9 deutlich wird, bildet die *hierarchische Struktur* von OpenUSS eine erste Bezugsgrößendimension. Während auf der ersten Ebene dieser Dimension das Gesamtsystem OpenUSS angeordnet ist, wird die zweite Ebene durch die einzelnen Organisationseinheiten aufgespannt, die als Mandanten im System gespeichert sind. Diesen

¹⁵ Vgl. [BB99].

¹⁶ Vgl. [Re01], S. 19.

Mandanten werden auf einer dritten Ebene einzelne Veranstaltungen zugewiesen, die wiederum über bestimmte Ressourcen (Lehr- und Lernmaterialien, Diskussionsfäden, E-Mail-Nachrichten und Chat-Räume) verfügen. Die zweite Dimension wird durch die *Nutzer* des Systems aufgespannt. Zu den relevanten Rollenträgern gehören hier die Studierenden und die Dozenten. Durch diese Dimension können gruppenbezogene Kennzahlen ermittelt werden, wie etwa die Nutzungsdauer oder -intensität. Als letzte Dimension ist schließlich die Maßgröße *Zeit* zu erfassen.

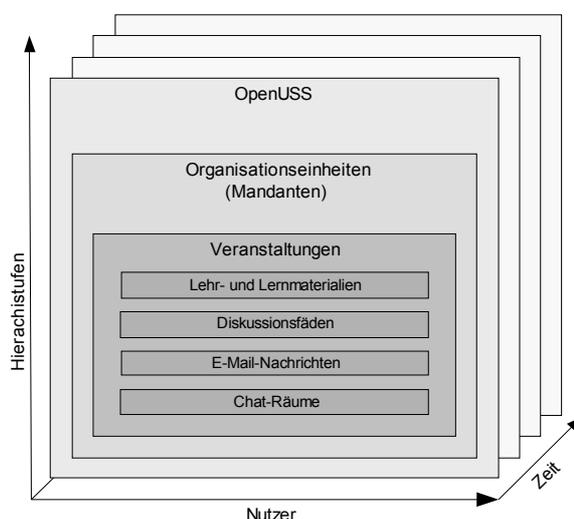


Abbildung 9: Kennzahlendimensionen für OpenUSS ¹⁷

Aus den angeführten Dimensionen erschließt sich eine Vielzahl von Kennzahlen zur Analyse des Nutzerverhaltens. Zur Optimierung der Informationslogistik können für Dozenten Messgrößen bezüglich der Verbreitung von Lehr- und Lernmaterialien relevant sein. Zu diesem Zweck wird mithilfe geeigneter Protokollaufzeichnungen ermittelt, wie viele Studierende einer Vorlesung ein bestimmtes Dokument per Download bezogen haben.¹⁸ Diese Kennzahl, die hier als *MaterialAttendeeAmount* (MAA) bezeichnet werden soll, bildet die informatorische Grundlage zur Messung des Verbreitungsgrades von Dokumenten innerhalb des Adressatenkreises. Zu diesem Zweck ist sie in Beziehung zur Gesamtanzahl sämtlicher Teilnehmer einer Vorlesung zu setzen. Diese Kennzahl sei als *LectureAttendeeAmount* (LAA) bezeichnet. Folglich ergibt sich der relative Verbreitungsgrad eines Dokuments (*MaterialDistributionReach*, MDR) als

$$MDR = \frac{MAA}{LAA}$$

¹⁷ In Anlehnung an [Gr03], S. 14.

¹⁸ Zum Instrument der Protokollaufzeichnung vgl. [Be01], S. 39 ff.

Bei der Ermittlung der Kennzahl MDR ist insbesondere die Analyse mit Bezug auf einen bestimmten Zeitpunkt – z.B. den Beginn einer konkreten Präsenzveranstaltung – interessant. So lässt sich der Verbreitungsgrad eines Dokuments zu einem bestimmten Zeitpunkt t_1 ex post ermitteln (vgl. Abbildung 10). Diese Datengrundlage kann als Informationsgrundlage zur Planung der Vorlaufzeit eingesetzt werden, die notwendig ist, damit zum fixierten Zeitpunkt ein bestimmter Verbreitungsgrad (z.B. 0,5) bei den Teilnehmern erreicht ist. Die Vorlaufzeit berechnet sich aus der Differenz von Zeitpunkt t_1 und dem Publikationszeitpunkt des Materials t_0 .

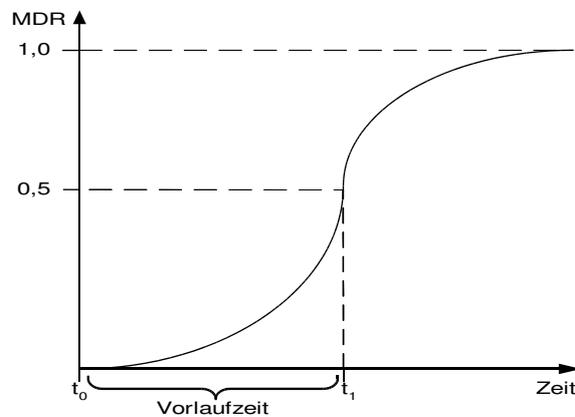


Abbildung 10: Exemplarische Darstellung des relativen Verbreitungsgrades von Dokumenten

Ähnliche Kennzahlen können auch für andere Ressourcen wie etwa Diskussionsfäden und Chat-Foren gebildet werden.

Das dargestellte Beispiel zeigt, dass auf der Grundlage von E-Learning-Plattformen relevante Kennzahlen ermittelt werden können, die das Nachfrageverhalten der Studierenden in Bezug auf bestimmte Ressourcen charakterisieren. Diese Kennzahlen sorgen auch in sogenannten Massenveranstaltungen mit mehr als 1.000 Hörern, wie sie beispielsweise für wirtschaftswissenschaftliche Fakultäten größerer deutscher Universitäten typisch sind, für Transparenz bezüglich der E-Learning-Prozesse und liefern dem Dozenten eine Informationsgrundlage zur Evaluation. Bei der Ausgestaltung des Berichtssystems ist allerdings zu berücksichtigen, dass eine anonyme bzw. pseudonyme Auswertung unter Einhaltung der geltenden datenschutzrechtlichen Bestimmungen erfolgt.¹⁹ Werden mit E-Learning-Plattformen kommerzielle Bildungsmärkte bedient (z. B. hochschulexterne Weiterbildung), sind diese Maßgrößen außerdem notwendig, um ein Mengengerüst für die Abrechnung (Billing) von Bildungsleistungen zu erzeugen.

¹⁹ Von besonderer Bedeutung sind dabei die Bestimmungen des Bundesdatenschutzgesetzes (BDSG) und des Teledienstedatenschutzgesetzes (TDDSG).

3.5 Wirtschaftlichkeitsbetrachtung

Um die Wirtschaftlichkeit der Einführung und Anwendung einer E-Learning-Plattform wie OpenUSS zu untersuchen, sind die Kosten- und Nutzeneffekte zu quantifizieren. Zwar ist der erzielbare Nutzen nur im jeweiligen Anwendungskontext situativ erfassbar (z. B. mit Hilfe von Nutzwertanalysen), doch eröffnet die Anwendung quelloffener Software strategische Kostensenkungspotenziale. So impliziert der Begriff Open Source, dass das entsprechende Softwareprodukt ohne Zahlung von Lizenzgebühren über verteilende Institutionen (z. B. Mediatoren) erhältlich ist. Gegenüber kommerzieller Software (i. d. R. Closed Source) weisen Open Source-Produkte einen Kostenvorteil in Bezug auf die zu zahlenden Lizenzgebühren auf, der sich in langfristigen Wirtschaftlichkeitsbetrachtungen positiv niederschlägt. Dabei ist allerdings zu berücksichtigen, dass der Großteil der softwarebezogenen Gesamtkosten (Total Cost of Ownership, TCO) durch den laufenden Betrieb von Softwaresystemen verursacht wird. Infolgedessen sind im Rahmen von Wirtschaftlichkeitsanalysen im Einzelnen die in der folgenden Abbildung gegliederten Kostenarten zu erfassen:

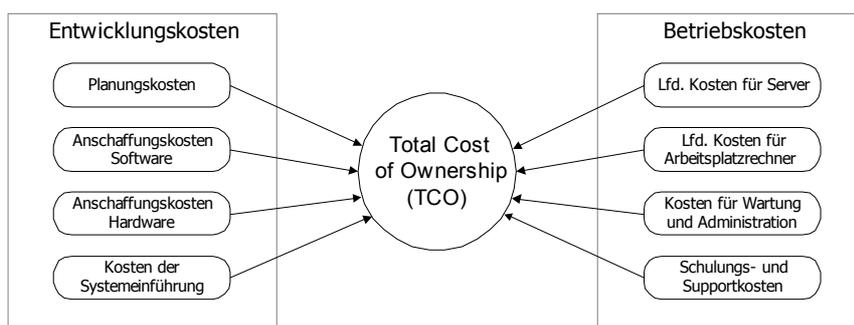


Abbildung 11: Gliederung der Entwicklungs- und Betriebskosten für Softwareprodukte

In Bezug auf die Entwicklungskosten wirkt sich beim Einsatz von OpenUSS die java-basierte Architektur günstig aus. So ist das System auch auf älteren Hardwarearchitekturen lauffähig und setzt keine proprietären Komplementärprodukte voraus, die zusätzliche Lizenzgebühren verursachen. Da OpenUSS auf (theoretisch) beliebig viele Rechnerknoten verteilt werden kann, können bereits bestehende Rechnersysteme integriert werden, so dass die Skalierungskosten gering gehalten werden. Hieraus resultieren Kostensenkungspotenziale für den Bereich der hardware- und softwarebezogenen Anschaffungskosten. Dabei ist allerdings zu berücksichtigen, dass eine Anpassung der Software, etwa in Bezug auf die Anwendungslogik oder die Benutzeroberfläche, entsprechende Programmierleistungen erfordert, die im Rahmen der Wirtschaftlichkeitsrechnung als Kosten der Systemeinführung zu berücksichtigen sind.

Bezüglich der Betriebskosten ist festzustellen, dass Software-Updates per Internet bezogen und installiert werden können, ohne dass zusätzliche Lizenzgebühren anfallen. Außerdem basiert OpenUSS auf einem Thin Client-Konzept, so dass clientseitig lediglich ein WWW-

Browser als Zugangssoftware benötigt wird. Dies macht sich im Systembetrieb vor allem in den Kostenpositionen für Wartung bzw. Administration und den laufenden Kosten für Arbeitsplatzrechner bemerkbar. Darüber hinaus trägt das Open Source-Konzept dazu bei, administrative Kosten des IT-Controlling zu senken. Während beim Einsatz kommerzieller Produkte administrative Tätigkeiten (z. B. Inventarisierung, Abrechnung, technische Vielfältigkeitskontrolle) zur Einhaltung der Lizenzbedingungen im Rahmen des Software Asset Management notwendig sind, fallen diese beim Einsatz von Open Source-Software nicht an.

4 Ausblick

Die cHL-Plattform wurde als Open Source-Software entwickelt. Die wirtschaftliche Bedeutung von Open Source-Software nimmt seit der weltweiten Verbreitung von System- und Anwendungssoftware, wie z. B. Linux und OpenOffice, permanent zu. Auch an Hochschulen wird vermehrt das Open Source-Prinzip aufgegriffen. Zum einen wird dort mit hoher Intensität Software im Rahmen von Forschungs- und Entwicklungsprojekten erstellt; zum anderen werden in zunehmendem Maße Open Source-Produkte für die computergestützte Hochschullehre eingesetzt. Im Idealfall wird sie dann im Rahmen einer Community bedarfsgerecht weiterentwickelt und zweckadäquat in bereits bestehende Anwendungsarchitekturen integriert. Dies gilt grundsätzlich auch für die hier dargestellte cHL-Plattform. Gerade für den Bereich der computergestützten Hochschullehre bietet Open Source-Software eine besondere Chance für Hochschulen. Nicht nur der Vorteil relativ geringer Gesamtkosten, sondern auch der universitätsübergreifende einheitliche Standard, der insbesondere dann relevant ist, wenn die Mitwirkung im Rahmen einer Community erfolgt, ist als Qualitätsmerkmal herauszustellen.

Literatur

- [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., Vieweg Verlag, Braunschweig, Wiesbaden 2002.
- [An03] Anderbrügge, K., Grob, H. L., Held, W., Poll, R.: Das IKM-Konzept der Westfälischen Wilhelms-Universität Münster, Arbeitsbericht, in Vorbereitung, Münster 2003.
- [BB99] Bensberg, F., Bielezke, S.: Web Log Mining bei cHL-Anwendungen. In (Grob, H. L. Hrsg.): Arbeitsbericht Nr. 16 der Reihe CAL+CAT, Münster 1999.
- [Be01] Bensberg, F.: Web Log Mining als Instrument der Marketingforschung – Ein systemgestaltender Ansatz für internetbasierte Märkte, Gabler Verlag, Wiesbaden 2001.
- [BHM02] Baumgartner, P.; Häfele, H.; Maier-Häfele, K.: Learning Management Systeme: Ergebnisse einer empirischen Studie – Evaluationsdesign und Auswahlempfehlungen. In (Bachman, G.; Haefeli, O.; Kindt, M. Hrsg.): Campus 2002 – die virtuelle Hochschule in der Konsolidierungsphase. Waxmann Verlag, Münster et al., 2002; S. 287-296.
- [Br02] vom Brocke, J.: Referenzmodellierung, Gestaltung und Verteilung von Konstruktionsprozessen, Diss., Münster 2002.
- [Ca01] Cap Gemini Ernst & Young (Hrsg.): eLearning Marktstudie 2001, Management Summary, München, 2001.

- [Do02] Doberkat, E. E. et al.: Anforderungen an eine eLearning-Plattform – Innovation und Integration, Dortmund, 2002.
- [GB02] Grob, H. L., Bensberg, F.: Strategische Potenziale von Open Source-Software für die computergestützte Hochschullehre (cHL): dargestellt am Beispiel des cHL-Administrationssystems OpenUSS. In (Bachman, G.; Haefeli, O.; Kindt, M. Hrsg.): Campus 2002 – die virtuelle Hochschule in der Konsolidierungsphase. Waxmann Verlag, Münster et al., 2002; S. 262-276.
- [GG95] Grob, H. L., Griebhaber, W.: Computergestützte Lehre an der Universität. In (Grob, H.L. Hrsg.): Arbeitsbericht Nr. 1 der Reihe CAL+CAT, Münster 1995.
- [GMS03] Gehrke, M.; Meyer, M.; Schäfer, W.: Eine Rahmenarchitektur für verteilte Lehr- und Lernsysteme. Im Internet unter der URL <http://www.campussource.de/projekte/rahmenarchitektur.pdf> [4.6.2003].
- [Gr01] Grob, H. L. : Die cHL-Architektur. In (Grob, H. L. Hrsg.): cHL: computergestützte Hochschullehre - Dokumentation zum cHL-Tag 2000. Lit Verlag, Münster, 2001, S. 9-17.
- [Gr03] Grob, H. L., Bensberg, F., Dewanto, B. L., Düppe, I.: Entwicklung von Controlling-Kennzahlen für Learning Management Systeme. In (Grob, H. L. Hrsg.): Arbeitsbericht der Reihe CAL+CAT, Münster 2003 (in Vorbereitung).
- [MW03] Meister, D.; Wedekind, J.: Lernplattformen im institutionellen Rahmen. In (Bett, K.; Wedekind, J. Hrsg.) Lernplattformen in der Praxis, Waxmann, Münster et al., 2003; S. 210-222.
- [Re01] Reichmann, T. : Controlling mit Kennzahlen und Managementberichten – Grundlagen einer systemgestützten Controlling-Konzeption, 6., überarb. und erw. Auflage, München, 2001.