

Digicampus: Integration von E-Learning-Werkzeugen und Realisierung einer campusweiten Lehr-/Lernplattform

Patrick Noack, Peter Rosina, Bernhard Strehl

Medienlabor
Institut für Medien und Bildungstechnologie
Universitätsstraße 2
86159 Augsburg
{vorname.nachname}@its.uni-augsburg.de

Abstract: Der Digicampus ist die zentrale Plattform zur virtuellen Unterstützung von Lehre und Studium an der Universität Augsburg. Es handelt sich um ein Webportal, in das bestehende Plattformen und Tools im Bereich E-Learning und Campus Management unter einer einheitlichen Benutzeroberfläche integriert werden. Ein zentraler Loginmechanismus erlaubt es, sich gleichzeitig in die integrierten Applikationen an- oder abzumelden. In diesem Paper werden die Anforderungen an eine zentrale Lehr-Lernplattform umrissen und deren Realisierung im Rahmen des Digicampus näher vorgestellt.

1 Einleitung

Im Rahmen des DFG-geförderten Projekts „Aufbau eines IT-Servicezentrums“ wird an der Universität Augsburg am Medienlabor des Instituts für Medien und Bildungstechnologie die zentrale Lehr-Lernplattform „Digicampus“ (www.digicampus.de) implementiert. Das im Jahre 2007 begonnene Projekt hat die Integration aller E-Learning-Anwendungen an der Universität Augsburg zum Ziel und basiert auf fundierten Forschungs- und Evaluationsergebnissen zu Learning-Management-Systemen [Sch00; BH02; Gru08]. Der Digicampus befindet sich seit Ende 2007 im Einsatz, wird kontinuierlich weiterentwickelt und steht allen Fakultäten offen. Finanziell wird das Projekt und dessen Entwicklung derzeit vom Institut für Medien- und Bildungstechnologie sowie aus Mitteln des genannten DFG-Projekts getragen.

Die in dem Projekt entwickelte Webapplikation ist eine Lehr-Lernplattform zur Verwaltung von Veranstaltungen und zur virtuellen Begleitung der Präsenzlehre. Die Anwendung stellt Methoden zur Verfügung, um verschiedene heterogene Systeme unter einer einheitlichen Oberfläche mit identischem Look-and-Feel zusammenzuführen und Datenredundanzen sowie -Inkonsistenzen zu vermeiden. Die Benutzer sollen also von den Hürden befreit werden, Daten an vielen Stellen aktuell zu halten sowie neue Navigationsstrukturen unterschiedlicher Systeme zu erlernen [BW02]. Die enthaltenen Systeme werden bei diesem Vorgang nicht verändert, sondern bleiben eigenständig und austauschbar um langfristige Erweiterbarkeit zu gewährleisten.

Im folgenden Kapitel wird auf die Herausforderungen bei der Implementierung einer universitätsweiten Lehr-Lernplattform eingegangen; im dritten Kapitel werden dann verschiedene Lösungsansätze der im zweiten Kapitel beschriebenen Herausforderungen dargestellt und analysiert. Das vierte Kapitel handelt von der konkreten Umsetzung des Digicampus. Das fünfte Kapitel fasst die Ergebnisse dieses Papers zusammen.

2 Anforderung an die zentrale Lehr-Lernplattform

Eine Universität mit ihren unterschiedlichen Fakultäten und Fachbereichen, Studierenden verschiedenster Studiengänge und Mitarbeitern in Verwaltung, zentralen Einrichtungen und Lehrstühlen ist als hochgradig heterogene Organisation aufzufassen [KB04]. Dadurch entstehen vielfältige Anforderungen, die bei der Entwicklung des Digicampus berücksichtigt werden müssen. Diese reichen von einer angepassten Darstellung der Teilnehmerlisten zu Veranstaltungen über die Möglichkeit, spezielle Werkzeuge wie Wikis oder Blogs einzubinden, bis hin zur Bereitstellung wissenschaftlicher Texte als Ergänzungsmaterialien. Da es kaum ein System gibt, das alle diese Funktionalitäten mitbringt, wird an Universitäten eine Vielzahl unterschiedlicher Systeme eingesetzt. Bei dieser großen Anzahl zu integrierender Systeme muss allerdings beachtet werden, dass kein zu komplexes, unübersichtliches und nicht mehr wartbares Gesamtsystem entsteht [DE02].

Wartbarkeit der IT-Infrastruktur. Die Wartbarkeit der IT-Systemlandschaft ist aus zwei Gesichtspunkten zu betrachten. Zum einen die Wartbarkeit durch Administratoren, die regelmäßig aktuelle Versionen der verwendeten Software einspielen, um Sicherheitslücken zu schließen und neu entwickelte Features den Nutzern zugänglich zu machen. Updates haben jedoch in der Regel zur Folge, dass jene Änderungen an den Systemen jedesmal erneut eingepflegt werden müssen, die durchgeführt wurden, um lokalen Gegebenheiten Rechnung zu tragen. Zum anderen sollte trotz großer Funktionsvielfalt für den Nutzer die Übersichtlichkeit gewahrt bleiben.

Benutzerfreundliche Bedienbarkeit der Systeme. Eine einheitliche Lehr-Lernplattform erfordert daher, dass ein Wechsel zwischen den darin beteiligten Systemen möglichst nahtlos vonstatten geht. Wird beispielsweise von einer Veranstaltung in das zugehörige e-Portfolio verlinkt, so ist es von Vorteil wenn Seitenlayout und Navigation identisch aufgebaut sind. Denn durch eine standardisierte Navigationsstruktur fällt es den Nutzern leichter, sich in den angeschlossenen Systemen zurechtzufinden.

Datenkonsistenz bei den Personendaten. Die eingesetzten Webanwendungen bringen in der Regel immer einen eigenen Login-Mechanismus sowie eigene Profildatenbanken mit sich. Schon ab einer kleinen Anzahl von Systemen stehen die Nutzer vor dem Problem, dass sie sich in einer Sitzung mehrere Male, oft mit unterschiedlichen Kennungen und Passwörtern, anmelden müssen. Mit der Anzahl der Systeme steigt so auch der administrative Aufwand für den Nutzer und es besteht die Gefahr dass die Aktualisierung in einem der Systeme vernachlässigt wird und somit in manchen Systemen falsche Daten angezeigt werden [DE02].

Datenkonsistenz bei den Veranstaltungsdaten. Bei weitem wichtiger als die Kontaktdaten einzelner Anwender sind für die Studierenden jedoch die Informationen zu einer Veranstaltung. Werden Grunddaten von Veranstaltungen sowohl in ein Content-Management-System als auch in das Learning-Management-System eingetragen, so ist bei einer Änderung der Daten (beispielsweise des Raumes oder der Veranstaltungszeit) diese Änderung bereits an zwei Stellen durchzuführen. Wird dies (teilweise) vergessen, so werden Studierende versuchen, die Veranstaltung im falschen Raum zu besuchen oder kommen zu einer falschen Zeit am Veranstaltungsort an. Derartige Fehlinformationen verringern die Akzeptanz der eingesetzten Systeme und führen zu Unmut bei den Nutzern.

3 Zielsetzung und Lösungsansatz des Digicampus

Da die von uns getesteten Learning-Management-Systeme allesamt nur einen kleinen Teil der geforderten Funktionalität erfüllen konnten, wurden im Vorfeld der Entwicklung des Digicampus 2007 zwei verschiedene Ansätze diskutiert.

3.1 Entwicklung eines eigenen Learning-Management-Systems (LMS)

Die Entwicklung eines eigenen LMS bietet die Chance, das System von vornherein so zu planen, dass Funktionen und Usability exakt auf die Strukturen und Prozesse innerhalb der eigenen Universität ausgelegt werden können. Der Entstehungsprozess eines solchen Systems ist jedoch langwierig. Zunächst müssen die Strukturen in der Universität und die darin stattfindenden Prozesse genau analysiert werden. Die digitale Abbildung eben dieser Prozesse muss dann für alle beteiligten Benutzergruppen gleichermaßen verständlich sein und allgemein Arbeitserleichterung schaffen, um auch universitätsweit Akzeptanz zu finden. Erschwerend kommt hinzu, dass die verschiedenen Fachkulturen unterschiedliche Anforderungen an E-Learning-Lösungen haben und diese unterschiedlichen Bedürfnisse eine stetige partielle Anpassung des LMS erforderlich machen, was den Entwicklungsprozess zusätzlich verlängert. Ein weiterer Nachteil ist, dass zunächst erhebliche Aufwendungen gemacht werden müssen und die erzielten Ergebnisse erst vergleichsweise spät nutzbar sind und somit die Universität während der finanzierten Entwicklungszeit nicht von den Aufwendungen profitieren kann. Darüber hinaus muss auch die langfristige Weiterentwicklung gesichert sein, und diese auch an die Bedürfnisse anderer Universitäten angepasst werden, damit das System nicht zu einer Insellösung wird. Gleichzeitig zeigt sich, dass mit der Zeit das Bedürfnis innerhalb der einzelnen Lehrstühle wächst, Verwaltungsaufgaben wie beispielsweise Übungen und deren Abgaben über ein LMS abzuwickeln. Dort, wo noch kein etabliertes System vorhanden ist, greifen die Lehrstühle auf eigene, teils selbstentwickelte Lösungen zurück. Die Bereitstellung einer einheitlichen und umfassenden Lösung für die Lehr- und Lernverwaltung muss also sehr zeitnah erfolgen, da mit einer steigenden Anzahl von Einzelsystemen die Migration der Daten und die Umstellung auf ein einheitliches System immer schwieriger wird.

Viele der genannten Risiken lassen sich minimieren, indem auf bereits bestehende Frameworks und Content-Management-Systeme (CMS) zurückgegriffen wird. Die Implementierung kann sich dann stärker auf die Modellierung der Prozesse und Strukturen an der Universität konzentrieren. Der Einsatz eines CMS erfordert jedoch an einigen Stellen Kompromisse bei der Umsetzung administrativer Prozesse, da diese auf die Benutzer- und Gruppenverwaltung des CMS sowie der daran angeknüpften Rechte- und Rollenverteilung abgebildet werden müssen. Diese technischen Vorgaben hätten somit auch bei Verwendung eines CMS einen längeren Entwicklungszeitraum und Probleme bei der Migration bestehender Daten mit sich gebracht.

Aus diesen Gründen fiel die Entscheidung zugunsten der Weiterentwicklung eines bestehenden LMS.

3.2 Weiterentwicklung eines bestehenden Learning-Management Systems

Hierfür stand eine Vielzahl verschiedener LMS zur Auswahl, von denen allerdings die meisten nur unzureichende Verwaltungsfunktionen für die komplexen Organisationsstrukturen der Universität Augsburg bereithielten. Auf der anderen Seite waren die Systeme, die nahezu alle Prozesse und Strukturen der Universität abbilden konnten, nur schwierig erweiterbar. Die Entscheidung für oder gegen ein bestimmtes LMS ist zudem nicht nur von den funktionalen Anforderungen der Universität Augsburg abhängig, sondern bezieht ebenso wie bei einer Eigenentwicklung die Verbreitung an anderen Universitäten mit ein. Die verfügbaren Systeme wurden daher vor der Entwicklung des Digicampus durch das Projekt „LMSNews“ [Gru08] des Lehrstuhls für Medienpädagogik evaluiert. Ausgehend von den Ergebnissen wurde die Auswahl auf drei geeignete LMS (*Olat*¹, *Moodle*² und *Stud.IP*³) eingegrenzt. Ein wichtiger Faktor war dabei ein offener Quellcode, um lizenzrechtliche Schwierigkeiten bei der Anpassung und Weiterentwicklung des Systems zu vermeiden [GB02; Kie02]. Letztendlich fiel die Entscheidung auf Stud.IP, da diese Plattform an der Universität Augsburg bereits erfolgreich in der Lehre eingesetzt wurde und einen hohen Verbreitungsgrad an anderen deutschen Universitäten aufweist⁴.

Das LMS Stud.IP stellt jedoch lediglich die Basis für ein universitäres Gesamtsystem dar. Es erfüllt alle grundsätzlichen Funktionen zur Verwaltung von Vorlesungen, Lehrstühlen, Dozenten, Studenten und deren Zuordnungen innerhalb der verschiedenen Einrichtungen, Fakultäten und Fachbereiche. Die virtuelle Zusammenarbeit und die Unterstützung der Lehre durch interaktive Lerninhalte ist in Stud.IP allerdings für den an der Universität Augsburg bestehenden Bedarf nicht ausreichend umgesetzt. Gleiches gilt für die Verwaltung von Übungsgruppen, Noten- und Punkteverteilung sowie speziellen Aufnahmemodalitäten. Auch eine Anbindung an das campusweite Identity-Management ist mit Stud.IP alleine nicht möglich. Aus diesem Grund müssen weitere Anwendungen in den Digicampus integriert werden.

¹ www.olat.uzh.ch

² www.moodle.de

³ www.studip.de

⁴ ein Drittel der deutschen Universitäten nutzt Stud.IP: siehe blog.studip.de

3.3 Erweiterungsmöglichkeiten der Basisfunktionen des Digicampus

Um über das Basissystem von Stud.IP hinausreichende Funktionen in den Digicampus hinzuzufügen wurden die im Folgenden beschriebenen Ansätze verfolgt:

Erweiterung des Basissystems von Stud.IP. Die Weiterentwicklung von Stud.IP hat den Vorteil, dass die Universität die Erweiterungen an der Software auch anderen Hochschulen zur Verfügung stellen kann und so einen Beitrag zur Verbesserung des Open-Source-Systems leistet. Allerdings sind alle unispezifischen Veränderungen nur schwer im allgemeinen Projektzweig von Stud.IP unterzubringen. Dies bezieht sich insbesondere auf die Integration bestimmter bereits bestehender Anwendungen, wie Blog, E-Portfolio und Login-System. Werden diese Zusatzanwendungen an die Benutzer- und Rollenfunktionen von Stud.IP gekoppelt, so ist eine langfristige Austauschbarkeit dieser Anwendungen sowie deren Verwendung in anderem von Stud.IP unabhängigen Kontext nicht mehr gewährleistet. Für die Entwicklung von Erweiterungen bietet Stud.IP daher ein Plugin-System. Dieses ermöglicht es, eigene Module zu entwickeln ohne dass Änderungen am Quelltext des zugrunde liegenden Basissystems notwendig sind.

Erweiterungen des Digicampus um autonome Systeme. Bei diesem Ansatz wird versucht, die im Digicampus integrierten Programme möglichst eigenständig und somit austauschbar zu halten. Der Vorteil dieser Architektur ist die hohe Flexibilität und eine vereinfachte Einbettung neuer Anwendungen. Die Implementierung erweist sich jedoch als schwierig, da die daran beteiligten Anwendungen jeweils über ein eigenes System zur Benutzerverwaltung und –Authentifizierung verfügen. Dies muss ebenso berücksichtigt werden wie eventuell mehrfach vorhandene Funktionen zur Benutzerkommunikation und Profilverwaltung. Auch hier gibt es die Möglichkeit, die Systeme an sich so zu verändern, dass diese ihre Benutzerdaten vom Digicampus beziehen sowie Funktionen zu deaktivieren, die in anderen zum Einsatz kommenden Systemen besser gelöst sind.

Die Anpassung sämtlicher zu integrierender Anwendungen erfordert jedoch einen immensen Wartungsaufwand, da die Veränderungen auch nach einem Update noch funktionieren müssen. Aus Benutzersicht genügt es, wenn eine Funktion nicht aus der Programmlogik einer Anwendung entfernt wird, sondern lediglich der Zugriff auf diese innerhalb des Benutzerinterfaces. Werden die Veränderungen an den Teilsystemen auf diese Anforderung reduziert, sinkt der Wartungsaufwand erheblich, da nur Teile der Benutzeroberfläche ausgetauscht werden müssen, anstelle elementarer Funktionen im Programmcode.

3.4 Benutzerverwaltung und Login

Um mehrere Systeme in die Lehr-/Lernplattform zu integrieren, muss ein Benutzer beim Login in den Digicampus simultan in alle daran beteiligten Webanwendungen mit den ihm zugewiesenen Rechten eingeloggt werden. Problematisch ist dies, wenn ein Benutzer in einem System verschiedene Rollen verkörpert (z.B. Dozent und Student) und infolgedessen mehrere Benutzeraccounts in einer im Digicampus integrierten Anwendung benötigt. Um einen simultanen Login in allen Systemen zu gewährleisten, wurden im Digicampus zwei verschiedene Ansätze geprüft:

Zentraler Login und zentrale Benutzerdatenbank. Hier gibt es einen zentralen Authentifizierungsserver (z.B. *LDAP*⁵, *Shibboleth*⁶, *OpenID*⁷, *Webauth*⁸), an dem der Benutzer sich mit seiner Benutzerkennung und seinem Passwort einloggt. Alle beteiligten Webanwendungen erfragen dabei die Benutzer, deren Rechte und Rollen sowie den Login-Status von dem verwendeten Authentifizierungsserver. Diese Authentifizierung setzt allerdings voraus, dass die verwendeten Webanwendungen allesamt einen solchen Login-Mechanismus unterstützen, was bislang aber nur vereinzelt der Fall ist.

Dezentraler Login und dezentrale Benutzerdatenbank. Bei einem dezentralen Login-Mechanismus weiß die Anwendung selbst nichts von einem Login-Server. Loggt sich der Benutzer am Authentifizierungsserver ein, so erstellt dieser die für den Zugriff notwendigen Sitzungen in der Datenbank der jeweiligen Systeme und setzt für jedes System das erforderliche Authentifizierungs-Cookie im Browser des Benutzers. Zudem ist für den Login-Server keine Kenntnis des Benutzerpasswortes erforderlich, da dieser den Login-Status direkt in der Datenbank der Anwendung setzen kann, in die der Benutzer eingeloggt werden soll, ohne dass die Anwendung selbst eine Passwortüberprüfung anstößt. Die Abfrage des „richtigen“ Passwortes geschieht daher nur einmalig am Authentifizierungsserver.

3.5 Funktionale Verknüpfung von Anwendungen

Wünschenswert wäre auch die Darstellung unterschiedlicher Informationen aus verschiedenen Anwendungen auf einer Webseite. Ein Beispiel hierfür ist die Anzeige neuer Einträge eines Portfolio-Systems beim Aufruf der entsprechenden Vorlesungsseite in Stud.IP. Um dies zu erreichen, müssen die Darstellung der angezeigten Webseite sowie deren Inhalte voneinander separiert und in Form eines Mashups rekombiniert werden. Die Möglichkeit, verschiedene Anwendungen durch Mashup unter einer einheitlichen Benutzeroberfläche miteinander zu verzahnen, ist nicht trivial. Im Folgenden werden drei mögliche Ansätze zur Realisierung erläutert.

⁵ www.isode.com/whitepapers/ldap-standards.html

⁶ www.shibboleth.internet2.edu

⁷ www.openid.net

⁸ webauth.stanford.edu

Clientseitige Integration über Javascript und AJAX. Bei dieser Variante wird in jeder am Digicampus beteiligten Anwendung ein Javascript integriert. Ruft der Benutzer die Anwendung auf, so wird das Javascript aktiv, und verhindert zunächst, dass das Benutzerinterface des aufgerufenen Systems angezeigt wird. Stattdessen wird der gesamte Inhalt der aufgerufenen Seite clientseitig im Browser des Benutzers nochmals durchgeparst, die störenden redundanten GUI-Elemente entfernt, und Style- und Layout-Angaben durch ein Corporate Layout ersetzt. Um Fremdanwendungen zu integrieren, können diese via AJAX zur Laufzeit nachgeladen, und ebenfalls geparst werden. Wichtig ist dabei, dass der Benutzer selbst in jeder der beteiligten Anwendungen eingeloggt ist. Diese Technik hat den entscheidenden Vorteil, dass sämtliche Prozesse zur Integration der Anwendungen ineinander auf den Rechner des Benutzers ausgelagert werden, und der Server somit stark entlastet wird. Nachteilig ist jedoch, dass technisch versierte Benutzer den eigentlichen Quellcode einsehen und modifizieren können und somit Zugriff auf eigentlich gesperrte Funktionen erhalten.

Serverseitige Integration eines Parsers. Die serverseitige Integration der Parsing-Funktion funktioniert über einen in PHP vorgesehen Mechanismus, der die Ausgabe einer Webanwendung in einen „OutputBuffer“ genannten Zwischenspeicher schreibt, anstatt diese direkt an den Browser des Benutzers weiterzuleiten. Somit ist es möglich, die gesamte Seite auch serverseitig durchzuparsen, und dort wesentliche GUI-Elemente auszutauschen, bevor die Seite beim Benutzer angezeigt wird. Das zugrunde liegende Prinzip bleibt also dasselbe wie bei der Javascript basierten Lösung. Es erfordert jedoch einige Tricks, um zu verhindern, dass Anwendungen selbst den „OutputBuffer“ überschreiben oder Programmabbrüche dazu führen, dass auch der Parser vorzeitig beendet wird. Diese Technik erzeugt eine erhöhte Serverlast durch das nochmalige Durchparsen der aufgerufenen Seite, hat jedoch den Vorteil, dass zum Browser des Benutzers keine Daten übertragen werden, die der Benutzer nicht einsehen darf.

Serverseitige Integration über Parser mit vorgeschaltetem Proxy. Die Vorschaltung eines Proxies ermöglicht es, dass auch Anwendungen, die auf anderen Servern liegen in den Digicampus integriert werden können. Mit dieser Technologie kann einerseits eine Lastverteilung erzielt werden, indem alle im Digicampus beteiligten Systeme auf ein Netzwerk von Einzelservern für jede Anwendung verteilt werden. Andererseits ist auch die Integration von Anwendungen möglich, auf deren Server kein Zugriff möglich ist. Die Ausgabe des Proxies wird wie im vorhergehenden Absatz erläutert in einen „OutputBuffer“ geschrieben und danach durch einen Parser geleitet, der die GUI-Anpassungen durchführt.

4 Technische Umsetzung

Im Folgenden wird die Realisierung der im zweiten Kapitel dargestellten Anforderungen am Beispiel der Augsburger Lehr-Lernplattform „Digicampus“ beschrieben.

4.1 Single-Sign-Login

Die vielen verschiedenen Benutzerprofile und Login-Mechanismen in den Anwendungen wurden durch das Konzept eines Single-Sign-Logins mit einheitlicher Profildatenbank ergänzt. Anstelle eines separaten Logins für jede Applikation reicht ein übergeordnetes System im Digicampus die Login-Informationen der Nutzer an den zentralen Authentifizierungsserver (Webauth) des Rechenzentrums der Universität Augsburg weiter (vgl. Abb. 1). Nach erfolgreicher Anmeldung wird dem Benutzer, sofern er in einer der Anwendungen mehrere Accounts besitzt und keinen Standard-Account festgelegt hat, eine Auswahlmöglichkeit gegeben, aus der er die Accounts wählt, die er in dieser Sitzung verwenden möchte. Zudem ist es möglich, während einer Sitzung zwischen bestehenden Accounts zu wechseln so dass beispielsweise ein Administrator die getätigten Änderungen gleich aus der Sicht eines Studierenden sehen kann. Dieses Verfahren ist nötig, da in den meisten Systemen nicht Benutzern verschiedene Rollen zugewiesen werden, sondern verschiedene Rollen existieren und einzelne Accounts genau einer Rolle zugeordnet sind. Beispielsweise hat bei Stud.IP ein Account entweder den Status Administrator oder Student – was beispielsweise dazu führt, dass eine studentische Hilfskraft sich mit dem Administratoren-Account nicht mehr zu Vorlesungen anmelden kann.

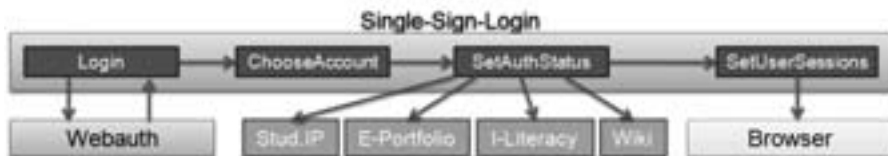


Abbildung 1: Single-Sign-Login im Digicampus – vereinfachte Darstellung

4.2 Datenkonsistenz

Verwaltung der Benutzerdaten. Im Zuge des oben genannten Login-Mechanismus wurde auch eine zentrale Profilverwaltung eingeführt. Im zentralen Profil werden benutzerbezogene Daten, die in mehreren Systemen benötigt werden, einheitlich gespeichert. Beim Speichern dieses Profils werden die Datensätze in den unterschiedlichen Systemen neu gefüllt. Dadurch dass alle vier Monate eine Zwangsaktualisierung des Profils angestoßen wird, ist gewährleistet, dass Studierende nicht vergessen, beispielsweise ihre Fachsemester zu aktualisieren.

Mehrfachverwendung der Veranstaltungsdaten. Neben dem Abgleich persönlicher Daten wurde zudem eine Schnittstelle für gespeicherte Veranstaltungsdaten geschaffen. Basierend auf einem XML-Service, dessen Ausgabe mithilfe eines einfachen Wizards konfigurierbar ist, lassen sich Veranstaltungsdaten unter Zuhilfenahme von AJAX auf Fremdseiten, also auch im Content-Management-System der Universität einbinden. Da diese Daten auch serverseitig verarbeitet werden können, ist ein Update der Daten beispielsweise in der Notenverwaltungssoftware des Prüfungsamtes möglich.

4.3 Erweiterung des LMS Stud.IP um eigene Plugins

Ein praktisches Beispiel für die Umsetzung eines Plugins ist das an der Universität Augsburg zum Einsatz kommende System „LectureReg“, das ein erweitertes Anmeldeverfahren für Vorlesungen und Übungsgruppen innerhalb von Stud.IP bereitstellt. Stud.IP bringt von Haus aus nur zwei einfache Anmeldeverfahren mit sich: Dozenten haben zum einen die Möglichkeit, die Anmeldung zu einer Veranstaltung so einzurichten, dass die Studierenden einen Platz erhalten, die sich als erstes anmelden. Dieses System führt aber zu einer erheblichen Serverlast, da alle Interessenten unmittelbar zum Anmeldestart die Seite aufrufen und der Server somit sehr viele parallele Seitenaufrufe bearbeiten muss. Zum anderen gibt es ein Anmeldeverfahren über ein Lossystem, was aber zu einer undurchsichtigen Verteilung der Studierenden führt. Bei der Entwicklung von LectureReg wurde daher Wert auf eine gerechte Verteilung bei geringerer Serverlast gelegt. Zudem sollte das Plugin für die Benutzer einfach zu bedienen sein und für die Entwickler ohne Umstände modifiziert werden können.

Neue Anmeldeverfahren. LectureReg bietet eine Prioritätswahlliste an. Die Studierenden können unterschiedliche Veranstaltungen mit Prioritäten versehen, gemäß denen die Verteilung vollautomatisch nach Anmeldeende erfolgt. Neben den angegebenen Prioritäten kann der Dozent noch weitere Parameter hinzuziehen, welche die Verteilung beeinflussen: das Fachsemester des Studierenden, um Studierenden in höheren Fachsemestern Vorrang zu gewähren, einen studiengangbezogenen Koeffizienten und eventuell auch das Anmeldedatum. Die errechnete Verteilung kann jederzeit manuell abgeändert werden, um etwaige Sonderwünsche zu berücksichtigen. Die Studierenden werden nach erfolgter automatisierter Verteilung per Email über den Ausgang informiert und sind dann bereits als Teilnehmer der entsprechenden Veranstaltungen in Stud.IP eingetragen worden.

Die in der Weboberfläche des Plugins dargestellten Steuerelemente, basierend auf dem Javascript-Framework *Dojo*⁹, werden über AJAX angesprochen um eine flüssige Bedienung zu gewährleisten. So können die Prioritäten der Kurse einfach per Drag&Drop in einer Liste festgelegt werden. Nach jeder Transaktion erhält der Benutzer ein Feedback. In jedem Systemzustand wird dem Benutzer situationsbezogene Hilfe angeboten, sowohl als Tooltip als auch durch einen aufrufbaren detaillierteren Hilfetext. Die kompetente Darstellung ist zudem templatebasiert¹⁰, um das Layout einfach modifizieren zu können.

Verbesserte Übungsverwaltung. Die neben dem Anmeldeverfahren zweite Aufgabe des Plugins ist die Verwaltung von Übungen und Hausarbeiten. Für die Zulassung zu einer Klausur oder dem Erhalt eines Scheins muss oft eine Voraussetzung erfüllt werden, z.B. das Erreichen von 50% aller Punkte in den wöchentlich abzugebenden Übungsblättern oder das Halten von drei Referaten etc. LectureReg ermöglicht das Anlegen von Übungsblättern und Zulassungskriterien. Die Tutoren oder Dozenten können die Punkte/Noten online eintragen.

⁹ www.dojotoolkit.org

¹⁰ www.smarty.net

Diese sind dann nur für den jeweiligen Studenten sichtbar und er erhält schnell einen Überblick welche Leistung er noch zu erbringen hat, um die Zulassung zu erreichen. Die Kriterien lassen sich dabei relativ frei festlegen, in der Form „mindestens x von y“, „maximal x mal“ etc. So könnte man für einen Kurs X festlegen, dass der Studierende maximal dreimal fehlen darf, zweimal selbst vorzurechnen hat und insgesamt mehr als 50% aller Punkte erreicht werden müssen.

4.4 Integration autonomer Anwendungen

Einheitliche Benutzeroberfläche aller Systeme. Der Digicampus bietet dem Benutzer ein Benutzerinterface, welches über mehrere Anwendungen hinweg eine konsistente Benutzerführung aufweist. Der Benutzer sieht den Digicampus als ein einheitliches Gesamtsystem an Stelle der darin integrierten Einzelanwendungen. Im Abschnitt 3.3 wurden bereits Ansätze erläutert, wie die Einbettung von Zusatzanwendungen erfolgt, ohne dass dazu deren Quellcode verändert werden muss.

Alle Anwendungen innerhalb des Digicampus sind prinzipiell eigenständig und werden auch nur teilweise von den Entwicklern am Medienlabor betreut. Beim Aufruf einer Anwendung innerhalb des Digicampus wird deren Ausgabe in einen OutputBuffer geschrieben und über einen Webservice an einen Layout-Manager übergeben (siehe Abb. 2). Der Layout-Manager parst nun die gesamte Seite durch, und fügt die relevanten Inhaltselemente anhand vordefinierter Regelsätze in das einheitliche Layout-Template des Digicampus ein. Bei diesem Vorgang wird auch der Zugriff auf bestimmte Funktionen, wie beispielsweise die Benutzerverwaltung des Teilsystems unterbunden, indem die dafür notwendigen Steuerfunktionen aus der Benutzeroberfläche entfernt werden. Der Layout-Manager seinerseits gibt die „fertige“ Seite wieder an den Browser des Benutzers zurück.

Funktionale Verknüpfung der Teilsysteme. Die Verknüpfung zwischen den Systemen untereinander wird über sogenannte „Widgets“ realisiert. Widgets sind kurze Informationsblöcke aus einem anderen System, z.B. eine Projektfortschrittsübersicht aus dem E-Portfoliosystem, die innerhalb einer Kursverwaltungsseite von Stud.IP angezeigt wird. Diese Widgets werden aus Performancegründen erst zur Laufzeit vom Browser des Benutzers über AJAX an relevanten Stellen in einem anderen System nachgeladen. Das Aussehen der Widgets und die darin enthaltenen Funktionen werden ebenfalls über Regeln im Layout-Manager festgelegt. Widgets werden im Unterschied zu den Anwendungen selbst über ein Proxysystem (vgl. Abb. 2) aufgerufen, um zu gewährleisten, dass die Ursprungs-URL die gleiche ist wie die des gerade aufgerufenen Systems.

Mittels dieser Widgets ist es auch möglich, Seiten in bestehenden Systemen um neue Funktionen zu erweitern. So wurde beispielsweise ein Tool zur Erstellung von E-Portfolios in das LMS integriert und Informationen daraus werden an entsprechenden Stellen innerhalb der Veranstaltungsseiten angezeigt. Ebenso findet sich im Digicampus ein Helpdesk-System zur Abwicklung von Supportanfragen sowie eine Anwendung, in der Lehrtexte ähnlich einem Wiki präsentiert werden können.

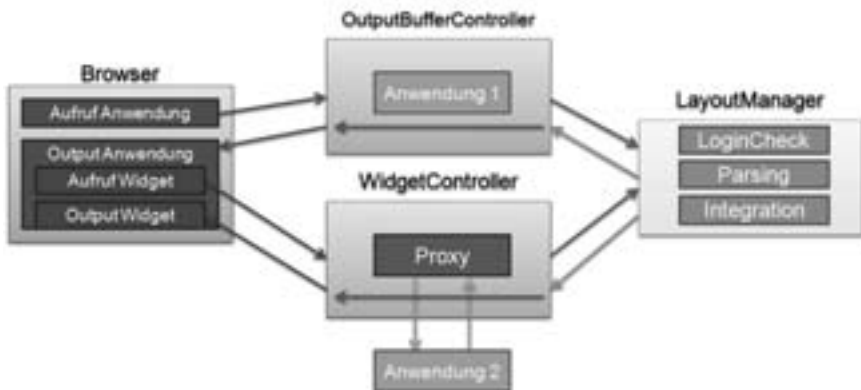


Abbildung 2: Konzept zur Integration autonomer Anwendungen im Digicampus

5 Zusammenfassung

Im Digicampus wurde erreicht, dass die eingesetzten Lehr-/Lern- und E-Learning-Systeme eigenständig bleiben, die für den Benutzer sichtbare graphische Ausgabe jedoch in einer einheitlichen Benutzeroberfläche vereinigt ist. Dies ermöglicht es, mit vergleichsweise geringem Aufwand neue Anwendungen optisch und funktional in die zentrale Plattform zu integrieren oder diese auszutauschen. Darüber hinaus gestaltet sich auch die Administration der im Digicampus integrierten Systeme einfacher, da diese unabhängig gewartet und auf separate Server verteilt werden können.

Trotz der vielen unterschiedlichen Systeme ist eine Datenkonsistenz durch ein übergeordnetes Login-System und eine zentrale Profilverwaltung gewährleistet, so dass die Benutzer von der Aufgabe befreit werden, ihre persönlichen Daten in den integrierten Systemen aktuell zu halten. Die zentrale Komponente des Digicampus stellt das LMS Stud.IP dar, in dem die Verwaltung der Einrichtungen und Veranstaltungen erfolgt. Aus diesem werden auch die Veranstaltungsdaten über einen XML-Service an das CMS der Universität und nicht integrierte Systeme exportiert.

Neben den E-Learning-Anwendungen verfügt der Digicampus über ein integriertes Ticket-Supportsystem, das den Benutzern schnelle Hilfe bei offenen Fragen oder Problemen bietet. Die aus den Support-Anfragen gewonnenen Daten werden genutzt, um Schwierigkeiten aufzudecken und den Digicampus stetig zu verbessern. Inzwischen wird der Digicampus von verschiedenen Lehrstühlen an fünf von sieben Fakultäten der Universität Augsburg eingesetzt. Mittelfristig ist die Ausweitung auf alle Fachbereiche der Hochschule und die Einbettung weiterer bis dato autonomer Webanwendungen, wie ein Webmail-System, die Webseite der Bibliothek oder Tools für Projektmanagement und Aufgabenverwaltung geplant.

Alte vereinzelt genutzte Inselfösungen werden durch die Bereitstellung neuer Funktionen obsolet und die Integration bereits etablierter Anwendungen wie des Prüfungsverwaltungssystems FlexNOW¹¹ macht den Digicampus zunehmend attraktiver für Studierende und Lehrende, da sie dort alle studienrelevanten Informationen vorfinden.

Literaturverzeichnis

- [BH02] Baumgartner, P; Häfele, H; Maier-Häfele: K.: E-Learning Praxishandbuch: Auswahl von Lernplattformen. Marktübersicht - Funktionen - Fachbegriffe. StudienVerlag, Innsbruck, 2002 <http://peter.baumgartner.name/publications-de/pdfs/evaluation-lms.pdf>
- [BW02] Bett, K.; Wedekind, J.: Lernplattformen in der Praxis. Münster: Waxmann, 2002.
- [DE02] Doberkat, E.; Engels, G.; Hausmann, J.H.; Lohmann, M.; Veltmann, C.: Anforderungen an eine eLearning-Plattform – Innovation und Integration. Studie im Auftrag des Ministeriums für Schule, Wissenschaft und Forschung in NRW, 2002 http://www.uni-paderborn.de/cs/ag-engels/Papers/2002/Studie_elp.pdf
- [GB02] Grob, H. L., Bensberg, F., Strategische Potenziale von Open Source-Software für die computergestützte Hochschullehre (cHL). In (G. Bachmann, O. Haefeli, M. Kindt): Campus 2002 – Die Virtuelle Hochschule in der Konsolidierungsphase. Münster: Waxmann, 2002; S. 262-276
- [Kie02] Kiedrowski, J.: Open-Source-Software - E-Learning zum Nulltarif?. In (A. Hohenstein; K. Wilbers): Handbuch E-Learning. Expertenwissen aus Wissenschaft und Praxis. Stand: 9. Erg.-Lfg. Verlag Deutscher Wirtschaftsdienst, Köln, 2002.
- [KB04] Kubicek, H.; Breiter, A.; Fischer, A.; Wiedwald, C.: Organisatorische Einbettung von E-Learning an deutschen Hochschulen. Bremen: Institut für Informationsmanagement Bremen GmbH, 2004 http://www.ifib.de/publikationsdateien/MMKH_Endbericht_2004-05-26.pdf
- [Sch00] Schulmeister, R: Selektions- und Entscheidungskriterien für die Auswahl von Lernplattformen und Autorenwerkzeugen. Gutachten für das BM: BWK ...Österreich, 2000 <http://www.zhw.uni-hamburg.de/pdfs/Plattformen.pdf>
- [Gru08] Grünwald, S.: Learning Management Systeme im universitären Betrieb. Lulu.com, Deutschland, 2008.

¹¹ flexnow.uni-bamberg.de