

Einsatz von und Probleme mit Portalsoftware am Beispiel einer universitären Lehr-Lernplattform

Bernhard Strehl

Medienlabor
Institut für Medien und Bildungstechnologie
Universitätsstr. 2
86159 Augsburg
bernhard.strehl@phil.uni-augsburg.de

Abstract: Um den Funktionsumfang von Lehr-Lernplattformen zu erweitern, werden oft mithilfe von Portalsystemen unterschiedlichste Anwendungen unter einer Oberfläche vereint, um diese den Nutzern niedrigschwellig zugänglich zu machen. In diesem Beitrag werden verschiedene Typen von Portalsystemen vorgestellt und anhand konkreter Implementierungen im Projekt „Digicampus“ der Universität Augsburg erläutert. Anschließend wird auf eine Usabilitystudie desselbigen sowie generelle Probleme mit Portalsystemen eingegangen. Abschließend wird der neue Ansatz, bei dem das Learning-Management-System selbst als Portalsoftware fungiert, vorgestellt.

1 Einleitung und Vorstellung der Lehr-Lernplattform

Seit 2007 wird an der Universität Augsburg die zentrale Lehr-Lernplattform „Digicampus“ (www.digicampus.de) durch das Medienlabor implementiert. Zu Projektbeginn wurde der Digicampus nur von wenigen hundert Nutzern einzelner Studiengänge genutzt, mittlerweile wird er an allen Fakultäten eingesetzt. Die ca. 24.000 registrierten Nutzer entsprechen der Gesamt-Belegschaft der Universität. Aus der Tatsache heraus, dass es für viele Problemstellungen bereits gute Software gibt [BS09], wird bei der Entwicklung des Digicampus der Ansatz verfolgt, ein Learning-Management-System (LMS) durch autonome Anwendungen zu erweitern [NRS09], anstatt Funktionen neu zu entwickeln. Als Basissystem für das Veranstaltungsmanagement wurde und wird das Opensource-LMS „Stud.IP“¹ eingesetzt. Dieses war vorab durch die damalige Medienpädagogik evaluiert worden. Heute ist es vornehmlich im deutschsprachigen Raum verbreitet (derzeit an über 50 Hochschulen und Institutionen¹). Zur Abwicklung des Begleitstudiums² wurde das Angebot durch die Opensource-Software „Elgg“³, einer Blog- und Portfoliosoftware, ergänzt. Zur niedrigschwelligen Nutzung der Anwendungen wurden diese in einem Portal integriert.

¹ www.studip.de (29.06.2012)

² www.imb-uni-augsburg.de/studium/begleitstudium-problemloesekompetenz (29.06.2012)

³ www.elgg.org (29.06.2012)

Im folgenden Kapitel werden Kernanforderungen an Portalsysteme genannt, daraufhin folgt im dritten Kapitel eine Übersicht über deren technisch mögliche Realisierungen. Das Vierte widmet sich den bisher an der Universität Augsburg installierten Portalsystemen. Anschließend werden Teilergebnisse einer Evaluation der zuletzt installierten Portalsoftware vorgestellt und im sechsten Kapitel wird ein Überblick über Probleme, die aus Usabilitysicht mit solchen Systemen auftreten, gegeben und erläutert, ob und wie diese Probleme minimiert oder vermieden werden können. Abschließend folgt eine Vorstellung der Neuimplementierung und eine Zusammenfassung dieser Arbeit.

2 Anforderungen an Portalsysteme

Eine Portalsoftware hat die Aufgabe, verschiedene heterogene Anwendungen unter einer Oberfläche zu integrieren, um Nutzern eine einheitliche Anwendung zur Verfügung zu stellen. Dabei muss die Software vielfältige Anforderungen erfüllen:

Zentraler Login: Schon bei einer kleinen Anzahl integrierter Anwendungen stehen die Nutzer vor dem Problem, dass sie sich in einer Sitzung mehrmals, oft mit unterschiedlichen Kennungen und Passwörtern, anmelden müssen. Die Portalsoftware muss deshalb einen zentralen Login anbieten, gegenüber dem sich Nutzer nur einmal authentifizieren müssen. Dieser muss je nach angeschlossener Anwendung auch einen Wechsel zwischen verschiedenen Benutzerkonten, beispielsweise zwischen Administratoren- und Nutzeraccount, ermöglichen (vgl. [NRS09]).

Einheitliche Oberfläche: Durch eine einheitliche Oberfläche fällt es Nutzern einfacher, sich in den angeschlossenen Anwendungen zurechtzufinden, da er eine standardisierte Navigationsstruktur vorfindet (vgl. [BW02],[NRS09]).

Nahtloser Wechsel zwischen den Anwendungen: Nutzer müssen mit einem einfachen Klick zwischen den einzelnen Anwendungen wechseln können (vgl. [NRS09]).

Benutzerfreundliche Bedienbarkeit der Anwendungen: Die einheitliche Oberfläche muss benutzerfreundlich und die Navigationsstruktur verständlich sein (vgl. [NRS09]).

Wartbarkeit der IT-Infrastruktur: Die Portalsoftware muss auch neue Versionen der angebundenen Anwendungen voll unterstützen, ohne dass große Änderungen an der Portalsoftware selbst durchgeführt werden müssen (vgl. [NRS09]).

Datenkonsistenz: Die eingesetzten Webanwendungen bringen in der Regel auch eine eigene Profildatenbanken mit sich. Damit besteht die Gefahr, dass die Aktualisierung der persönlichen Daten in einem der Systeme vernachlässigt wird, was dazu führt, dass in manchen Anwendungen beispielsweise veraltete Kontaktdaten hinterlegt sind. Zur Vermeidung dessen muss die Portalsoftware den Nutzer von der Aufgabe befreien, seine persönlichen Daten in allen Anwendungen aktuell zu halten (vgl. [DE02], [NRS09]).

3 Typen von Portalsystemen

Um Nutzern eine einheitliche Anwendung für ihre Arbeit zur Verfügung zu stellen, ist es vonnöten, die verschiedenen Anwendungen unter einer Oberfläche zu integrieren. Generell ist dabei zwischen zwei Typen von Portalsystemen zu unterscheiden: die *clientseitige Integration* der Anwendungen sowie eine *serverseitige Integration*. Bei der clientseitigen Integration lädt der Browser des Nutzers die einzelnen Anwendungen und integriert diese in einem „Skelett“, so dass es für Nutzer wirkt, als wäre diese kumulierte Seite eine ganzheitliches Anwendung. Bei der serverseitigen Integration versendet der Server an den Browser des Nutzers eine Webseite, die bereits die integrierten Anwendungen beinhaltet [NRS09].

3.1 Clientseitige Integration

Bei der clientseitigen Integration stellt die Portalsoftware grundsätzliche Funktionalitäten wie eine Navigation zum Wechsel zwischen den angeschlossenen Anwendungen und eine Login-Funktionalität bereit, die Anwendungen selbst werden, so wie sie sind, an den Browser des Benutzers ausgeliefert. Dies kann durch Javascript oder Frames beziehungsweise I-Frames realisiert werden. Da jedoch sämtliche Formulare und Links in den Anwendungen in Javascript-Funktionalität konvertiert werden müssten, ist die erstgenannte Variante in der Regel nicht praktikabel [NRS09], weshalb stattdessen Frames genutzt werden. Der grüne Rahmen in Abbildung 1 repräsentiert dabei den Bereich, in dem sich Hauptnavigation und Login der Portalsoftware befinden, der weiße Bereich (links) stellt einen Platzhalter für die angebundene Anwendung A dar.



Abbildung 1: Einbindung der Anwendung A per Frame oder I-Frame

Die clientseitige Integration bietet den Vorteil einer einfachen Implementation, allerdings müssen die einzelnen Anwendungen bereits optisch aneinander angepasst sein. Eine Einbindung per Frame oder I-Frame führt zudem dazu, dass in der Regel keine Links weitergegeben werden können, da der Frame nur die Einstiegsseite lädt.

3.2 Serverseitige Integration

Bei der serverseitigen Integration wird die Ausgabe der Anwendungen durch eine Servertechnologie abgefangen, statt dass die Daten an den Browser des Nutzers gesendet werden.

Diese Daten können anschließend durch die Portalsoftware dahingehend manipuliert werden, dass Elemente der graphischen Oberfläche entfernt, hinzugefügt oder an andere Stellen verschoben werden, das Design geändert, Begrifflichkeiten ausgetauscht oder die Navigation vereinheitlicht wird. Anschließend wird die resultierende Ausgabe an den Browser des Nutzers gesendet. Wie auch bei der clientseitigen Integration repräsentiert der grüne Balken oben auf der Seite in Abbildung 2 die Hauptnavigation, die Anwendung wird vor Auslieferung an den Browser jedoch angepasst, um ein einheitliches Erscheinungsbild zu erreichen [NRS09].

Die serverseitige Integration ist zwar deutlich aufwändiger zu implementieren, allerdings bietet sie gegenüber der clientseitigen Integration den Vorteil, dass eine komplette Webseite inklusive aller Elemente ausgeliefert wird und auch eine Weitergabe von URLs möglich ist.

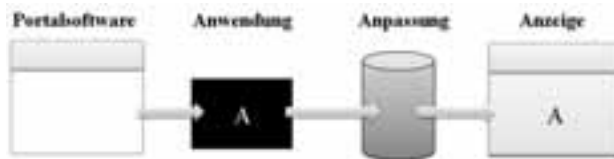


Abbildung 2: Manipulation und Einbindung der Anwendung A per serverseitiger Integration

4 Portalsysteme an der Universität Augsburg

An der Universität Augsburg wurden im Projektverlauf zwei unterschiedliche Portalsysteme implementiert und in Betrieb genommen. Auf eine erste Version mit clientseitiger Integration der Anwendungen Stud.IP und Elgg (2007) folgte 2008 die zweite Version des Digicampus mit serverseitiger Integration.

4.1 Portal mit clientseitiger Integration

Die erste Version der Portalsoftware integrierte die beiden Anwendungen per I-Frames unter einer Oberfläche (Abbildung 3) und ermöglichte einen einfachen Singlesign-On. Dabei wurde ein Nutzer gegenüber dem Rechenzentrum authentifiziert und die Logindaten nach einigen Prüfungen an die (leicht modifizierten) Loginmechanismen von Stud.IP und Elgg weitergereicht. Dieser Loginmechanismus unterstützte jedoch keine Accountverwaltung, so dass die Nutzer zum Wechsel zwischen zwei Rollen sich zunächst ausloggen und anschließend mit abweichenden Daten (ohne Authentifizierung gegenüber dem Rechenzentrum, sondern gegenüber der einzelnen Anwendung) wieder einloggen mussten. Obwohl in Abbildung 3 suggeriert, verfügte das Portalsystem prinzipbedingt auch über keine Funktionen, um die integrierten Anwendungen optisch aneinander anzugleichen. Dies wurde hier durch entsprechende Änderungen der Quellcodes der einzelnen Anwendungen realisiert, wodurch allerdings ein Mehraufwand bei Updates und Pflege der integrierten Systeme entstand.



Abbildung 3: Clientseitige Integration im Digicampus per I-Frame

Auch ein Datenabgleich zwischen den Anwendungen war nicht realisiert worden. Aufgrund dieser unvollständigen Umsetzung der Anforderungen sowie der genannten generellen Nachteile mit der clientseitigen Integration wurde im weiteren Verlauf eine Portalsoftware mit serverseitiger Integration implementiert und 2008 installiert.

4.2 Portal mit serverseitiger Integration

Beim Portal mit serverseitiger Integration wurde, um die verschiedenen Anwendungen unter einer Oberfläche mit einheitlichem Look-and-Feel zu vereinen, zunächst mithilfe der PHP-Technik Output-Buffer⁴ die Ausgabe der entsprechenden Anwendung abgefangen. Anschließend wurde durch einen serverseitigen Parsing-Mechanismus das Menü und der Content der Anwendung extrahiert. Diese Elemente wurden nun optisch einander angepasst, einem Template zugewiesen und das gerenderte Template an den Browser des Nutzers gesendet. Hierdurch wurde ein konsistentes Layout mit konsistenter Navigation ohne Änderung der Quellcodes der angebotenen Anwendungen erreicht (Abbildung 4). Des Weiteren wurde ein zentrales Profil geschaffen, womit persönliche Daten sich an einer zentralen Stelle mit geringem Aufwand aktuell halten ließen [NRS09].



Abbildung 4: Einbindung von Stud.IP und Elgg per serverseitiger Integration

⁴ www.php.net/manual/de/intro.outcontrol.php (29.06.2012)

Später folgte im Startbereich des Portals eine Installation der Opensource-Blog-Software Wordpress⁵, über welche Informationen über Updates und Ähnliches gegenüber den Nutzern kommuniziert wurden. Diese wurde jedoch aus praktikablen Gründen nicht an das Singlesign-On-System angebunden, da aus Sicherheitsgründen nur einzelne Administratoren schreibenden Zugriff darauf erhalten sollten.

Dieses Portalsystem erfüllte viele der in Kapitel 2 genannten Anforderungen. Konzeptuell sollte dieses zudem durch eine Proxylösung ergänzt werden, um auch Anwendungen, die nicht auf dem lokalen Server liegen, integrieren zu können. Beispielsweise sollte die Webseite i-literacy⁶ in die Lehr-Lernplattform integriert werden, um Studierenden schnellen Zugriff auf Informationen und Grundlagen zum wissenschaftlichen Arbeiten zu ermöglichen.

Zunächst folgte im Zeitraum Juni-Dezember 2009 eine (im Rahmen von betacampus [PS09]) durch DFG-Mittel unterstützte Analyse der Usability des Digicampus. Mit dieser sollte sichergestellt werden, dass die Anforderung der benutzerfreundlichen Bedienbarkeit des Systems ebenso wie die anderen Anforderungen erfüllt ist, beziehungsweise um vor der Erweiterung entsprechende Usability-Probleme identifizieren und beheben zu können.

5 Evaluation des Digicampus

In dieser Evaluation wurde eine umfassende Usability-Analyse des Digicampus durchgeführt, wobei die Vorgehensweise auf den Erfahrungen anderer Evaluationen basierte, wie beispielsweise in [SWG05], [CKT01], [TKS05], [MRR08] und [GGS04] beschrieben. Dabei wurden diverse kleinere Probleme mit dem LMS Stud.IP festgestellt, allerdings wurden auch gravierende Probleme mit der Portalsoftware deutlich.

Hinsichtlich der Aussagekraft waren für die an der Portalsoftware aufgedeckten Probleme die Methoden der Webserverlog-Analyse, die Analyse direkten Nutzerfeedbacks sowie die Benutzerbefragung in Form einer anonymen Onlinebefragung am hilfreichsten. Der später durchgeführte Nutzertest (unterstützt durch die Methode des lauten Denkens) bestätigte zumeist vorher vermutete Probleme. Die Ergebnisse der heuristischen Evaluation überschneiden sich zwar teilweise mit der Nutzerstudie, viele der Beanstandungen hatten ihre Ursache aber in Verstößen gegen die in Kapitel 6 genannten Anforderungen.

5.1 Durchführung und Zielsetzung der Evaluation

In einem ersten Schritt wurden die Webserverlogs aus dem Zeitraum Juni bis September 2009 hinsichtlich Benutzungsdauer und der meist aufgerufenen Seiten analysiert. Diese Analyse hatte zum einen zum Ziel, die Fokusgruppe festzustellen sowie mögliche Probleme beim Nutzungsverhalten des Digicampus aufzudecken.

⁵ www.wpde.org (29.06.2012)

⁶ i-literacy.e-learning.imb-uni-augsburg.de (29.06.2012)

Parallel zur Logfileanalyse wurde direktes Nutzerfeedback ausgewertet, um problematische Bereiche zu identifizieren. Dieses ist für Usability-Evaluationen eine sehr wertvolle Ressource, da Benutzer den Support oft nur bei Problemen kontaktieren, die sie von sich aus nicht lösen können [Nie93]. Durch den Aufbau eines zentralen E-Mail-Supports mit Hilfe eines Ticket-Systems (osTicket⁷) stand eine große Anzahl an relevantem Nutzerfeedback zur Verfügung. Diese E-Mails wurden kategorisiert und die Mails im Bereich *vermutete Usabilityprobleme* anschließend detaillierter analysiert, um aus Usabilitysicht problematische Bereiche zu identifizieren.

Im Anschluss wurde ein Online-Fragebogen erstellt und auf der Startseite veröffentlicht. Dieser bestand aus Fragen zum Nutzungsverhalten und kombinierte verschiedenen Likert-Skalen mit nicht-obligatorischen Freitext-Antwortfeldern. Zusätzlich hatten die Nutzer die Möglichkeit, einen Freitext-Kommentar zum Digicampus abzugeben. Ziel des Fragebogens war es, die Fokusgruppe festzustellen und eine Bewertung der subjektiv gefühlten Bedienbarkeit der meistgenutzten Funktionen sowie bei der Analyse des Nutzerfeedbacks als problematisch erkannte Bereiche zu erhalten.

5.2 Ergebnisse der Untersuchung

Insgesamt stand ein großer Pool an Daten zur Verfügung: für die Logfileanalyse wurden knapp 900.000 Seitenaufrufe geloggt und 57.000 unterschiedliche Sitzungen ausgewertet. An direktem Nutzerfeedback standen 800 Datensätze aus E-Mail-Anfragen zur Verfügung. Auch die Anzahl der vollständig ausgefüllten und ausgewerteten Fragebögen war mit 309 (zwei Wochen Laufzeit) sehr hoch.

Die Auswertung der Logfiles ergab, dass die Plattform in der Regel nur kurze Zeit (in über drei Vierteln der Fälle weniger als 15 Minuten am Stück) genutzt wird; 80% der Befragten gaben im Fragebogen an, die Anwendung zumindest wöchentlich aufzurufen.

Die Nutzung des Digicampus geschieht in der Regel also anforderungsgetrieben, um kleine Aufgaben zu erledigen oder zur Informationsbeschaffung. Diesem Ziel stand allerdings offenbar die Struktur des Digicampus entgegen: die mit 15% Gesamtanteil meist aufgerufene Seite war zwar im Untersuchungszeitraum tatsächlich die Seite „Meine Veranstaltungen“ der Anwendung Stud.IP, auf der eingesehen werden kann, ob es in den eigenen Veranstaltungen Neuerungen gibt. Aufsummiert waren jedoch die Plätze zwei und drei (Digicampus-Startseite und Stud.IP-Startseite) für 22% der Gesamtanfragen verantwortlich. Die Aufrufe der Portfoliosoftware Elgg machten insgesamt nur 0,44% der Aufrufe aus. Diese Anwendung wurde also offensichtlich nicht genutzt.

30% der E-Mail-Anfragen betrafen das zentrale Profil, mit dem sich ein Großteil der persönlichen Daten pflegen ließ. Zentral gehaltene Daten waren in Stud.IP nicht direkt veränderbar. Trotz einem entsprechenden Hinweistext waren viele Nutzer derart darüber irritiert, dass sie beispielsweise ihr Fachsemester nicht ändern konnten, dass sie den Support kontaktieren mussten. Dieser Effekt trat auch in der Nutzerstudie auf.

⁷ www.osticket.com (29.06.2012)

In der Onlineumfrage gingen 61 freie Kommentare ein. In vielen dieser Kommentare wurde die Navigationsstruktur bemängelt, in 43% der Anfragen die Übersichtlichkeit beziehungsweise Design. Dies bestätigte sich ebenso in der Nutzerstudie.

6 Konzeptuelle Probleme von Portalsoftware

Sicherlich hätten viele der aufgedeckten Probleme zusammen mit einem Redesign behoben werden können, beispielsweise durch Implementierung eines zentralen Profils für alle Daten und Deaktivierung der Profilsseiten in den Einzelanwendungen.

Bei der Betrachtung allgemeiner Web-Usability-Richtlinien ([NT02], [KBN04], [NL06]) werden jedoch schnell weitere Probleme mit Portalsystemen deutlich, da besonders folgende, wichtige, Richtlinien durch deren konzeptuelle Art gebrochen werden, was in abgewandelter Form auch in der heuristischen Evaluation erkannt worden war:

Auf die Homepage: Das Logo sollte als Backlink auf die Homepage fungieren. Unter-Bereiche der Webseiten sollen keine eigenen „Homepages“ haben, die Homepage fungiert immer als Startseite des kompletten Webangebots.

Da in Portalsystemen immer einzelne Unter-Anwendungen eingebunden sind, haben diese immer auch einzelne „Homepages“ beziehungsweise einen Punkt „Start“ in der Navigation. Zudem bietet oft auch das Portalsystem eine weitere Startseite an. Dass diese zusätzliche Startseite den Nutzer daran hindert, zielgerichtet und schnell die gewünschten Informationen im System zu erhalten, hat sich auch in Kapitel 5.2 gezeigt.

Suche: Es soll sich nicht nur ein Link zur Suche auf der Seite befinden sondern gleich das Eingabefeld. Die Suche sollte auf jeder Seite angezeigt werden, da Nutzer ansonsten erst wieder zur Hauptseite zurückspringen müssen (u.A.).

Eine übergreifende Suche kann ein Portalsystem per se nicht leisten, da eine Portalsoftware unterschiedlichste Anwendungen integrieren können soll, ohne dass tiefgehende Kenntnisse über deren Daten und Datenhaltung vorhanden sind.

Hilfe: Die Hilfe-Funktion sollte sich rechts oben befinden.

Bei einem Klick auf die Hilfe erwarten die Nutzer, dass sie zum aktuellen Kontext Hilfe erhalten. Die Hilfe der aktuellen Anwendung beinhaltet jedoch keine Tipps, dass beispielsweise eine gewisse Aufgabe mit einer anderen eingebetteten Anwendung besser gelöst werden könnte.

Sitemap: Bei komplexen Webapplikationen soll eine Sitemap vorhanden sein. Dies wird insbesondere auch in vielen Checklisten zur Experten-Evaluation von Internetangeboten gefordert⁸

⁸ www.infodesign.com.au/usabilityresources/webevaluation (29.06.2012)

Ein Portalsystem mit mehreren eingebundenen Anwendungen ist sicherlich als komplexe Webapplikation zu verstehen. Eine übergreifende Sitemap über alle integrierten Anwendungen kann es jedoch ohne detaillierte Kenntnisse der Anwendungen nicht liefern.

Portalsoftware verfügt per Definition nicht über tiefreichende Kenntnisse über die eingebetteten Anwendungen, womit Kernfunktionalitäten durch die Anwendungen selbst übernommen werden müssen. Dadurch, dass diese jedoch unter einer fremden Oberfläche integriert sind, kommt es zu zahlreichen Problemen:

Bei den Hilfsfunktionen sowie der Sitemap und Suche müsste das Portalsystem entweder immer auf die Funktionen der aktuell genutzten Anwendung verweisen, oder eine Auswahl der möglichen Bereiche („Suche in Anwendung A / B / C“) anbieten, was den Zugriff auf diese Funktionen erschwert. Hinzu kommt, dass Hilfe und Sitemaps technisch nach Anwendung getrennt und nicht in funktionelle Bereiche gegliedert sind, was der Erwartung der Nutzer widerspricht. Eine Migrierung der Hilfen und Sitemaps aller angebotenen Anwendungen zu einer gesamtheitlichen ist aufgrund des Arbeitsaufwands sowie der nötigen Pflege bei Updates jedoch nicht praktikabel.

In diesem Kapitel hat sich gezeigt, dass Portalsysteme durch ihre konzeptuelle Art Kernfunktionalitäten von Webanwendungen in sehr negativer Weise beeinflussen können. Eine Korrektur dieses Verhaltens ist nur durch Einsatz vieler Ressourcen bei Inbetriebnahme sowie Wartung bei jeder neuen Version einer der eingebundenen Anwendungen möglich. Dies allerdings widerspricht der Grundanforderung „Wartbarkeit der IT-Infrastruktur“.

7 Das LMS als Portalsystem

Sowohl in Kapitel 5 als auch in Kapitel 6 wurden Probleme mit der Portalsoftware deutlich, welche sich nur mit enormem Arbeitsaufwand hätten beheben lassen können. Weiterhin wurde per Logfileanalyse (siehe Kapitel 5.2) deutlich, dass die Portfoliosoftware nicht genutzt wird und somit zumindest derzeit obsolet ist. Zudem stellt die zusätzliche obere Navigationsebene in einem Portalsystem eine Platzverschwendung dar, insbesondere unter dem Gesichtspunkt der immer weiteren Verbreitung von mobilen Geräten wie Smartphones. Auch wirkt es aus Nutzersicht nicht logisch, dass kaum oder selten genutzte eingebundene Anwendungen (wie beispielsweise Elgg) ebenso wie das oft genutzte LMS einen eigenen Navigationspunkt in der Hauptnavigation erhalten. Hinzu kam, dass die Erweiterungsmöglichkeiten in Stud.IP zwischenzeitlich deutlich verbessert worden waren und die Software selbst bereits viele Anforderungen unterstützt, die zu Projektbeginn nicht vorhanden waren. Eine neue Plugin-Schnittstelle erlaubt es, viele bisher im Portalsystem realisierten Funktionalitäten auch im LMS direkt zu implementieren. Beispielsweise können Plugins die Navigation anpassen, indem Navigationselemente hinzugefügt, entfernt oder verschoben werden und es lassen sich Hinweistexte auf bestimmten Seiten anzeigen.

Aufgrund dieser Ausgangssituation wurde schließlich beschlossen, das Portalsystem nach zwei Jahren Laufzeit wieder zu deaktivieren und das LMS in den Rang des Mastersystems zu versetzen. Bei dieser Reimplementierung mussten natürlich alle bisher im Portalsystem vorhandenen Features in Stud.IP-Plugins überführt oder entsprechend neu programmiert werden.

7.1 Funktionsumfang der neuen Version

Diese neue Version wurde Mitte 2011 in Betrieb genommen. Auch in dieser Reimplementierung ist ein Singlesign-Login sowie flexibles Accountverwaltungssystem vorhanden, mit dem sich auch verschiedene Nutzerrollen in (künftig) angebundenen Anwendungen verwalten lassen. Weiterhin können in diesem neuen hybriden System weitere Anwendungen entweder client- oder serverseitig integriert werden.

So wurde beispielsweise im Menüpunkt „Nachrichten“ ein Navigationspunkt „Universitäre E-Mail“ eingeführt. Unter diesem Navigationspunkt ist per I-Frame ein direkter Zugriff auf das Webmail-System des Rechenzentrums direkt über die zentrale Lehr-Lernplattform möglich, ohne dass dessen URL in den Browser eingegeben werden (und gemerkt werden) muss. Dieses Konzept der clientseitigen Integration ist in Abbildung 5 (oben) illustriert.

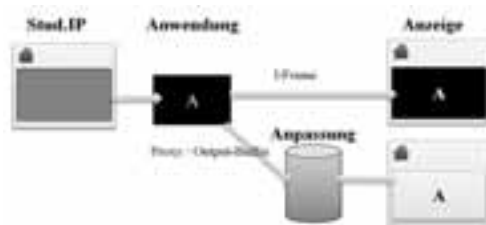


Abbildung 5: Integration weiterer Anwendungen in Stud.IP

Mittels serverseitiger Integration wird beispielsweise die in Kapitel 4.2 genannte Wordpress-Installation eingebunden. Dessen Ausgabe wird über den Output-Buffer in PHP abgefangen, nicht benötigte Elemente wie das HEAD-Element werden entfernt, der Content wird in eine Stud.IP-Seite integriert und das Resultat an den Browser des Nutzers gesendet (Abbildung 5 unten). Durch ein Wordpress-Authentifizierungsplugin, welches auf die Stud.IP-Authentifizierung zurückgreift, sind Administratoren automatisch eingeloggt und können ohne weitere Anmeldung Artikel pflegen.

7.2 Umgehung der konzeptuellen Probleme

Viele der konzeptuellen Probleme aus Kapitel 6 werden dadurch vermieden, dass die Stud.IP-API genutzt wird und das LMS entsprechende Funktionalitäten mit sich bringt:

Generierung der Sitemap: Durch die Einführung neuer Navigationspunkte per API werden diese automatisch in die Sitemap von Stud.IP übernommen. Darin taucht beispielsweise der Punkt Universitäre E-Mail funktionell logisch unter Nachrichten auf.

Nur eine Startseite: Künftig soll per Proxy-Technologie beispielsweise die Webseite i-literacy im Kontext einer Veranstaltung angezeigt werden können, um direkten Zugang zu Hilfe bei der Anfertigung wissenschaftlicher Arbeiten im Kontext einer Veranstaltung zu ermöglichen. Zwar verfügt i-literacy auch über eine Startseite. Dadurch, dass i-literacy dann aber als Lernmodul an eine Veranstaltung angebunden ist, stellt diese keine Startseite im eigentlichen Sinn dar, sondern den Anfang einer Lerneinheit.

Globale Suche: Leider existiert in Stud.IP noch keine globale Suche, somit kann die vorhandene Suchfunktion auch kein Treffer zu beispielsweise i-literacy liefern. Immerhin ist das Verhalten dann jedoch konsistent: Oben auf der Seite wird die – in den meisten Fällen genutzte – Veranstaltungssuche angezeigt, in einer seitlichen Infobox können kontextabhängig einzelne Bereiche durchsucht werden.

Übergreifende, kontextsensitive Hilfe: Beim Einfügen neuer Bereiche wie das Webmail-System lässt sich ein Hilfe-Sprungpunkt in die zentrale Stud.IP-Hilfe einfügen. Diese kann dahingehend erweitert werden, dass grundlegende Informationen zum momentanen Bereich bereitgestellt werden. Beispielsweise verweist die Hilfe im Bereich Universitäre E-Mail auf einige Kurzinformationen zum lokalen Webmail der Universität, obwohl dies eigentlich kein Bestandteil der zentralen Stud.IP-Hilfe ist.

8 Zusammenfassung

In dieser Arbeit wurden die verschiedenen Typen von Portalsystemen vorgestellt, deren Vor- und Nachteile diskutiert und anhand konkreter Beispiele an der Universität Augsburg verbildlicht. Im Anschluss daran wurden Ergebnisse einer Usability-Evaluation des in Augsburg installierten Lehr-Lernsystems vorgestellt. Im sechsten Kapitel schließlich wurde gezeigt, dass einige der aufgedeckten Probleme sowie weitere Probleme mit dem Thema „Portalsoftware“ an sich zu tun haben. Diese jedoch ließen sich nur dann beheben, wenn enorme Ressourcen in die Implementierung des Portals und auch für eine dauerhafte Wartung zur Verfügung stünden. Dies jedoch steht in Widerspruch zu einer der Basis-Anforderungen an Portalsysteme, dass diese einfach gewartet werden können müssen.

Durch die Abschaffung der Inselfösung Portal wurden, nach einer aufwändigen Migrierung aller Features des Portals in Plugins für das LMS, Ressourcen frei, die nun direkt in das LMS zur Verbesserung der Bedienbarkeit eingebracht werden können. Dies kommt zudem auch anderen Hochschulen zugute. Per Plugin lassen sich auch ohne separates Portalsystem Fremdanwendungen in das LMS integrieren und stehen den Nutzern niedrigschwellig zur Verfügung. Andere Anforderungen wie Berücksichtigung des BITV⁹ wurden in dieser Arbeit nicht behandelt, können aber bei Portalsystemen weitere Schwierigkeiten bereiten.

⁹ www.de.wikipedia.org/wiki/Barrierefreie_Informationstechnik-Verordnung (29.06.2012)

Literaturverzeichnis

- [BS09] Boles, D., Schmees, M. (2009), Herausforderungen bei der Anpassung von Open Source Software an neue Einsatzbereiche. In: Stefan Fischer, Erik Maehle, Rüdiger Reischuk (Hrsg.): Tagungsband der Informatik 2009, 39. Jahrestagung der Gesellschaft für Informatik, GI-Edition Lecture Notes in Informatics, 154, Köllen Druck + Verlag, Bonn, S.339-339, 9/2009
- [BW02] Bett, K.; Wedekind, J.: Lernplattformen in der Praxis. Münster: Waxmann, 2002.
- [CKT01] Cunliffe, D., Kritou, E. and Tudhope, D. (2001). Usability evaluation for museum Web sites. *Museum Management and Curatorship* 19: 229-252.
- [DE02] Anforderungen an eine eLearning-Plattform – Innovation und Integration. Studie im Auftrag des Ministeriums für Schule, Wissenschaft und Forschung in NRW, 2002
- [GGS04] Granić A., Glarinić V. and Stankov S. (2004). Usability Evaluation Methodology for Web-based Educational Systems.; 8th ERCIM Workshop "User Interfaces For All" 28-29 June 2004 Palais Eschenbach, Vienna, Austria.
- [KBN04] Koyani, S., Bailey, R., Nall, J., Allison, S., Mulligan, C., Bailey, K. and Tolson, M. (2004). Research-based Web design and usability guidelines. www.usability.gov/guidelines/guidelines_book.pdf (zuletzt besucht: 28.3.2012)
- [MRR08] Martin, L., Roldán Martínez, D., Revilla, O., Aguilar, M.J., Santos, O.C., and Boticario, J. (2008). Usability in e-Learning Platforms: heuristics comparison between Moodle, Sakai and dotLRN. OpenACS and .LRN conference 2008. International Conference and Workshops on Community based environments, Guatemala.
- [Nie93] Nielsen, J. (1993). Usability Engineering. Academic Press San-Diego.
- [NL06] Nielsen, J., Loranger, H. (2006). Web Usability. München: PEARSON EDUCATION DEUTSCHLAND GmbH.
- [NRS09] Noack, P., Rosina, P. & Strehl, B. (2009). Digicampus: Integration von E-Learning-Werkzeugen und Realisierung einer campusweiten Lehr-/Lernplattform. In A. Schwill & N. Apostolopoulos (Hrsg.), Lernen im Digitalen Zeitalter. DeLFI2009 – Die 7. E-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. Bonn: Köllen.
- [NT02] Nielsen, J., Tahir, M. (2002). Homepage usability: 50 enttarnte Websites. München: PEARSON EDUCATION DEUTSCHLAND GmbH.
- [PS09] Pillay, Susanne; Sporer, Thomas (2009): Betacampus – Förderung eines campusweiten Innovationsprozesses durch einen Wettbewerb. In: Kuhlen, Rainer (Hg.): Information: Droge, Ware oder Commons? Wertschöpfungs- und Transformationsprozesse auf den Informationsmärkten; Proceedings des 11. Internationalen Symposiums für Informationswissenschaft (ISI 2009), Konstanz, 1.-3. April 2009. Boizenburg: Hülsbusch (Schriften zur Informationswissenschaft, 50), S. 561–565.
- [SWG05] Sporer, T., Weiß, D., Giesz, M., Striegl, H. (2005). Evaluation des audiovisuellen digitalen Informationsdienstes von Knowledgebay. In Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. Deutsche e-Learning Fachtagung Informatik, Lecture Notes in Informatics, Volume P-66, ISBN: 3-88579-395-4, Bonn: Köllen Druck+Verlag GmbH, S. 399-410.
- [TKS05] Turnbow, D., Kasianovitz, K., Snyder, L., Gilbert, D., & Yamamoto, D. (2005). Usability testing for web redesign: A UCLA case study. *OCLC Systems and Services*, 21(3), 226-234