

Architektur einer dienstebasierten, personalisierbaren Laufzeitumgebung für Lernumgebungen

Markus Schmid^{+#}, Jeanne Stynes⁺, Reinhold Kröger[#]
{mschmid|jstynes}@cit.ie, kroeger@informatik.fh-wiesbaden.de

⁺Department of Maths and Computing, CIT, Cork, Irland

[#]FB Informatik, Labor für Verteilte Systeme, FH Wiesbaden

Abstract: Das Papier analysiert Schwachpunkte aktueller LMS-Architekturen und stellt als Lösungsvorschlag ein dienstebasiertes Architekturmodell für eine erweiterbare, personalisierbare LMS-Laufzeitumgebung vor. Die Architektur definiert ausschließlich Schnittstellen zur Interaktion von Komponenten des Laufzeitsystems und lässt damit Entwicklern größtmögliche Freiheiten bei der Implementierung. Ziel des Architekturmodells ist eine vereinfachte Wiederverwendung von Lehrinhalten, Personalisierbarkeit sowie die Unterstützung der Kooperation verschiedener Hochschulen.

1 Einleitung

In den letzten Jahren haben Web-basierte Lernplattformen (*Learning Management Systems - LMS*) zur Präsentation multimedialer Lehrmedien immer stärkere Bedeutung erlangt. Die Aufgabengebiete von LMS können grob in drei funktionale Kategorien unterteilt werden: Erstellung/Pflege von Kursinhalten (Authoring), System-, Nutzer- und Kursverwaltung (Administration), sowie Bereitstellung der eigentlichen Lehrinhalte und ergänzende Funktionalität (Laufzeitumgebung). LMS-Laufzeitumgebungen unterstützen in der Regel Lerneinheiten und definierbare Lernpfade, sie bieten sowohl Feedback- als auch Lernkontrollmechanismen, um didaktische Inhaltskonzeptionen zu ermöglichen.

Um angebotene Lehrinhalte einem möglichst breiten Publikum in geeigneter Weise nahe bringen zu können, ist die Personalisierbarkeit der Laufzeitumgebung ein wichtiges Kriterium für die Auswahl eines LMS. Personalisierung unterstützt den Lernenden durch das Bereitstellen einer für seine Bedürfnisse optimierten Umgebung.

Viele Hochschulen und Unternehmen evaluieren derzeit Lernplattformen, wobei der Markt derzeit insgesamt sehr stark in Bewegung ist ([Sch03]). Längerfristiges Ziel der Hochschulen ist auch die Schaffung von hochschulübergreifenden Lernportalen, in die die beteiligten Institutionen ihre Inhalte einbringen können.

Die Entscheidung für ein LMS ist in der Regel mit hohen Investitionskosten verbunden und bedeutet, bedingt durch den hohen Migrationsaufwand zwischen Systemen, meist eine längerfristige Bindung an einen Hersteller. Vor allem die Migration von Kursinhalten und den damit verbundenen didaktischen Konzepten stellt eine große Herausforderung dar.

Das vorliegende Papier analysiert in Abschnitt 2 Defizite existierender LMS-Plattformen in Bezug auf Erweiterbarkeit und Personalisierung und beschreibt in Abschnitt 3 einen daraus resultierenden neuen Architekturansatz.

2 Analyse

Die Produktion von didaktisch hochwertigen Inhalten für LMS ist sehr teuer und zeitaufwendig. Mit der wachsenden Menge von durch Hochschulen bereitgestellten Lehrinhalten, die teilweise auch im Verbund mehrerer Institutionen entstehen (z.B. [ULO2]), und auf Grund der derzeit unklaren Marktsituation wächst der Wunsch, einmal erzeugte Inhalte in Verbindung mit LMS unterschiedlicher Hersteller verwenden zu können. Zur Unterstützung des Austauschs u.a. von Kursinhalten zwischen verschiedenen LMS existieren darum mittlerweile mehrere, sich teilweise ergänzende bzw. aufeinander aufbauende Spezifikationen. Gemeinsame Basis bildet der *LOM (Learning Object Metadata)* Standard [LTS02], der Metadaten zur Beschreibung von Lernobjekten definiert. Die in LOM definierten Datenelemente finden sich sowohl in den *IMS-Spezifikationen* [IMS02] als auch im *Sharable Content Object Reference Model (SCORM)* wieder [Adv01].

IMS definiert im Ganzen zehn Spezifikationen, die sich mit verschiedenen Aspekten eines LMS von der Definition von Tests bis zu einem Datei-Austauschformat für Kursinhalte befassen. SCORM hingegen beschreibt neben einem auf LOM basierenden Datenmodell ein API, über das ein Inhaltsobjekt mit dem LMS kommunizieren kann. Das API dient hauptsächlich dem Auslesen und Ablegen von benutzerbezogenen Daten, die vom LMS über Sessiongrenzen hinweg gespeichert werden können. Zur Definition eines Austauschformats für Kursinhalte greift SCORM im Wesentlichen auf die äquivalenten IMS-Spezifikationen zurück.

Zusätzlich befindet sich derzeit die *IEEE LTSA (Learning Technology Systems Architecture)* [LTS01] in der Spezifizierungsphase, die auf abstrakter Ebene Akteure und Systemkomponenten von Lernsoftware identifiziert sowie Interaktionsmöglichkeiten zwischen diesen definiert und so eine einheitliche Sicht auf Lernsoftware erreichen will.

Auf Grund des oben erwähnten unterschiedlichen Funktionsumfangs existierender LMS kann sich eine Migration von Kursinhalten schwierig gestalten. Insbesondere die Übertragung didaktischer Konzepte bedeutet einen Aufwand, der um ein Vielfaches höher ist als der für die Migration des eigentlichen Inhalts, der in der Regel aus Texten und Multimediaobjekten besteht, da man auf bestimmte Merkmale des eingesetzten LMS angewiesen ist (z.B. Navigations- und Kommunikationsmöglichkeiten).

Im Rahmen der Personalisierung einer LMS-Laufzeitumgebung aus Benutzersicht ist eine Erweiterbarkeit und Anpassbarkeit des verwendeten LMS sinnvoll ([Sch02], [Ker01]). Personalisierung ist daher ein Konzept, das keiner einzelnen funktionalen Komponente der LMS-Laufzeitumgebung allein zugeordnet werden kann, sie bezieht vielmehr alle Funktionen des Systems mit ein.

Als problematisch erweisen sich für beide Ziele die monolithische Struktur und fehlende standardisierte APIs bestehender Lernplattformen, was die Anpassung und Erweiterung

eines eingesetzten Systems durch den Kunden behindert. Ein weiteres Manko ist, dass sich hochschulübergreifende Lernportale mit unterschiedlichen LMS-Systemen in den beteiligten Institutionen mit den existierenden Systemen nur schwer realisieren lassen.

Mit der "Open Knowledge Initiative" (OKI) des Massachusetts Institute of Technology [Ope02] existiert seit 2002 ein erster Ansatz zur Definition von APIs innerhalb eines LMS. OKI hat hauptsächlich das Ziel, einzelne LMS-Komponenten zur Entwicklungszeit zu entkoppeln und so allgemein den modularen Aufbau eines LMS zu ermöglichen. Nach der Installation arbeitet ein OKI-basiertes System als monolithische Applikation. Konzepte wie die Unterstützung der Personalisierung der LMS-Laufzeitumgebung oder der Aufbau eines hochschulübergreifenden Portals sind nicht Fokus von OKI.

3 Architekturmodell

Das im Folgenden vorgestellte Architekturmodell einer LMS-Laufzeitumgebung greift speziell die in Abschnitt 2 dargestellten Defizite existierender Lernplattformen auf, wobei der Personalisierbarkeit der Plattform besondere Aufmerksamkeit geschenkt wird.

Das Architekturmodell ist komponentenbasiert, wobei die einzelnen Komponenten funktionale Anforderungen an eine LMS-Laufzeitumgebung repräsentieren. Im Modell werden lediglich die Schnittstellen zwischen den einzelnen Komponenten definiert, so dass Hersteller maximale Freiheit bei der Implementierung erhalten, gleichzeitig aber sichergestellt bleibt, dass Komponenten verschiedener Hersteller untereinander interagieren können.

Der Zugriff auf durch das Portal bereitgestellte Inhalte wird an dieser Stelle nicht diskutiert, da dieser Bereich bereits gut durch existierende Lernplattformen abgedeckt wird. Im Folgenden werden vielmehr die für Personalisierung benötigte Funktionalität sowie die den Lernprozess unterstützenden Leistungsmerkmale einer Laufzeitumgebung in den Mittelpunkt gestellt.

Im Modell werden die einzelnen Komponenten als Services bezeichnet, wobei *Infrastructure Services* und *E-Learning Services* unterschieden werden (Abbildung 1).

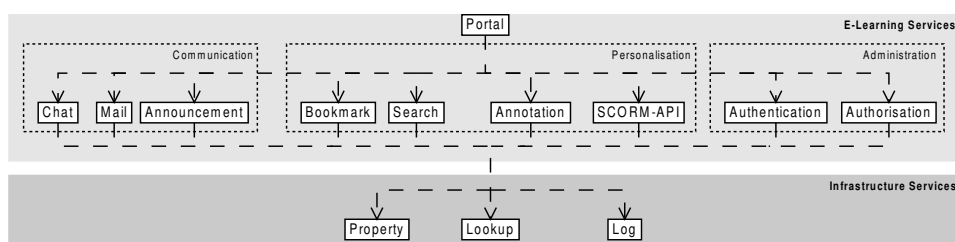


Abbildung 1: Services repräsentieren die funktionalen Komponenten im Architekturmodell

Infrastructure Services sind nicht von anderen Services abhängig. Sie erfüllen einfache Aufgaben und werden von den *E-Learning Services* benutzt. *E-Learning Services* erfüllen funktionale Anforderungen an das System und können direkt mit Benutzern interagieren.

Um ihre Aufgabe zu erfüllen, können *E-Learning Services* andere Services (*E-Learning* oder *Infrastructure Services*) benutzen.

Die Architektur sieht das aus diesen Servicetypen zusammengesetzte LMS als verteiltes System, d.h. Services können – im Unterschied zum monolithischen OKI-Ansatz – bei verschiedenen Hochschulen existieren und lose gekoppelt miteinander interagieren. Dies trifft allerdings im Wesentlichen auf die Schicht der *E-Learning Services* zu, diese nutzen dann jeweils lokale *Infrastructure Services* zur Datenablage. Ein *Lookup Service* kann zu diesem Zweck mit *Lookup Services* anderer Institutionen kooperieren und Referenzen auf Service-Instanzen austauschen. Diese Flexibilität erfordert, dass Services der Plattform zur Laufzeit dynamisch bekannt gemacht und auch wieder entzogen werden können.

Die Benutzung von Services ist auf Benutzer- oder Gruppenebene konfigurierbar, so dass die Plattform dienstübergreifend personalisiert werden kann. Personalisierung ist somit nicht nur auf die Funktionalität innerhalb eines einzelnen Dienstes beschränkt, je nach Nutzer kann das sich hinter dem Portal befindende System im Extremfall aus völlig verschiedenen Komponenten bestehen.

Durch eine Vererbungshierarchie wird sichergestellt, dass alle implementierten Services die gleiche Grundfunktionalität bereitstellen. Von einem allgemeinen *Search Service* kann man z.B. weitere, auf ein eingeschränktes Aufgabenfeld spezialisierte Suchdienste ableiten. So kann eine Plattform einen Service auch mit dem Interface seines Vaters ansprechen und auf diese Weise zumindest ein Subset der bereitgestellten Funktionalität verwenden, was eine effektive Kopplung von Services unterschiedlicher Hersteller ermöglicht. Konkrete Implementierungen von Services können so Funktionalität bereitstellen, die über die im Interface definierte hinausgeht.

Ziel des vorgestellten Architekturansatzes ist es, eine Vereinheitlichung der intern von einem LMS benutzten Schnittstellen zu erreichen. Dadurch werden auf der Designebene die Grundvoraussetzungen dafür geschaffen, komplexere didaktische Konzepte zwischen verschiedenen LMS-Implementierungen einfacher portieren zu können. Durch die Einführung einer flexiblen Service-Struktur erfolgt gleichzeitig die Ablösung des bisherigen monolithischen LMS durch ein offenes verteiltes System. Dies kann als erster Schritt hin zur Etablierung institutionsübergreifender Hochschulportale gesehen werden.

Gleichzeitig ist es möglich, bestehende Spezifikationen in die Architektur einzugliedern, so kann beispielsweise die durch SCORM definierte Schnittstelle als ein Service innerhalb der beschriebenen Hierarchie implementiert werden. Die Architektur berücksichtigt, soweit zutreffend, auch Festlegungen des LTSA-Drafts, allerdings sind hier eher die Anbieter der einzelnen Services in der Pflicht, die geforderte Semantik zu implementieren.

4 Stand der Arbeit

Das hier vorgestellte Architekturmodell einer dienstbasierten LMS-Laufzeitumgebung ist Teil einer Master's Thesis am Cork Institute of Technology, Cork, Irland. Im Rahmen dieser Arbeit werden derzeit wesentliche Teile des Systems unter Nutzung von WebServices und CORBA prototypisch implementiert. Hierbei sind die E-Learning Services sowohl als

WebServices als auch über CORBA ansprechbar, Infrastructure Services verfügen lediglich über ein CORBA-Interface. Auf diese Weise können über Hochschulgrenzen hinweg genutzte Services von der Flexibilität der Webservices-Architektur profitieren, wohingegen innerhalb einer Hochschule eng gekoppelte Services von der durch CORBA bereitgestellten Zusatzfunktionalität (Transaktionen, Security usw.) Gebrauch machen können.

Zum Teil werden im Rahmen der Implementierung existierende Dienste mit einer zusätzlichen, im Architekturmodell definierten Service-Schnittstelle ausgestattet bzw. mit einem entsprechenden Wrapper versehen (als Suchdienste Greenstone Digital Library, Google-WebService, für das SCORM API Teile der ADLnet SCORM-Referenzimplementierung, für Infrastructure Services div. CORBA-Services). Teilweise entstehen aber auch eigene Implementierungen, wie z.B. ein zum W3C Annotea Ansatz (vgl. [KKPS01]) kompatibler Annotations-Service, der Anmerkungen zu Web-Seiten speichern kann.

Im Zuge einer Verfeinerung und Vervollständigung der Schnittstellen-Spezifikation wird untersucht, inwieweit sich die im Rahmen der OKI-Initiative des MIT definierten Dienste als *Infrastructure Services* in das hier beschriebene Architekturmodell eingliedern lassen. Eine klare Untergliederung in notwendige und optionale Funktionalität sowie die Definition von typischen, diensteübergreifenden Abläufen erscheint ebenfalls sinnvoll.

Eine Diskussion der Eignung der verwendeten Technologien für die Implementierung des beschriebenen Modells sollte ebenfalls Ergebnis der Arbeit sein.

Literatur

- [Adv01] Advanced Distributed Learning Initiative. *Sharable Content Object Reference Model Version 1.2 – The SCORM Overview*, Okt 2001.
- [IMS02] IMS Global Learning Consortium, Inc. *Getting Started in IMS*, Jul 2002.
- [Ker01] M. Kerres. *Multimediale und telemediale Lernumgebungen*. Oldenbourg Verlag, 2001.
- [KKPS01] J. Kahan, M. Koivunen, E. Prud’Hommeaux, and R. Swick. Annotea: An Open RDF Infrastructure for Shared Web Annotations. In *Proc. of the WWW10 International Conference*, Mai 2001.
- [LTS01] IEEE Learning Technologies Standards Committee. *Learning Technology Systems Architecture (LTSA) - Draft 9*, 2001.
- [LTS02] IEEE Learning Technologies Standards Committee. *Learning Object Metadata (LOM) Standard VI.0*, 2002.
- [Ope02] Open Knowledge Initiative, MIT. *What is the Open Knowledge Initiative?*, Sep 2002.
- [Sch02] R. Schulmeister. *Grundlagen hypermedialer Lernsysteme – Theorie - Didaktik - Design*. Oldenbourg Verlag, 3rd edition, 2002.
- [Sch03] R. Schulmeister. *Lernplattformen für das virtuelle Lernen: Evaluation und Didaktik*. Oldenbourg Verlag, 2003.
- [UL02] D. Tavangarian U. Lucke. Turning a Current Trend into a Valuable Instrument: Multi-dimensional Educational Multimedia based on XML. In *Proc. of ED-Media 2002*, Jun 2002.