Service-basierte Integration dynamischer, interaktiver Medien in Lernplattformen

Martin Gläser, Raphael Zender, Ulrike Lucke, Djamshid Tavangarian
Universität Rostock, Institut für Informatik, Lehrstuhl für Rechnerarchitektur
vorname.nachname@uni-rostock.de

Abstract: Zur Unterstützung des individuellen Lernprozesses sind Vorlesungsaufzeichnungen bereits an vielen Hochschulen neben den Foliensatz und das klassische Skript getreten. Neben diesen asynchronen Medien ermöglichen Videokonferenzen und virtuelle Welten auch die synchrone und ortsunabhängige Teilnahme an Lehrveranstaltungen. Durch das breite Medienspektrum erhöht sich die Menge der eingesetzten Werkzeuge für die Erstellung, Verwaltung und Nutzung von Inhalten. Diese Vielfalt erfordert einen systematischen Ansatz zur Kopplung der Systeme. Der Beitrag zeigt, wie das von statischen Lehr-/Lernmaterialien bekannte Konzept der Service-basierten Integration erfolgreich auf die Interoperabilität verschiedener Datenquellen und -senken bei dynamischen, interaktiven Medien übertragen werden kann. Dies wird exemplarisch an einem Prototypen für die Lernplattform Stud.IP gezeigt, über den Studierende an Lehrveranstaltungen sowohl synchron (live) als auch asynchron teilnehmen können.

1 Zum Medieneinsatz an der Hochschule

Die rechnergestützte Präsenzlehre macht derzeit den größten Anteil der an deutschen Hochschulen eingesetzten eLearning-Szenarien aus; dabei kommt eine große Vielzahl heterogener Werkzeuge und Infrastrukturen zum Einsatz [LT07].

Üblich ist der manuelle *Upload von Skripten in Lernplattformen* durch den Dozenten im Anschluss an bzw. im Vorfeld von Lehrveranstaltungen. Mit dieser Verfahrensweise sind zwei grundsätzliche Probleme verbunden. Zunächst ist das Material oft *veranstaltungsspezifisch*. Dies mag aus didaktischen Erwägungen heraus vielfach gewünscht sein, ist aber im Sinne des Bologna-Prozesses für die Vergleichbarkeit erbrachter Studienleistungen und für die Mobilität von Studierenden wie Lehrenden eher hinderlich. Daher wurden zahlreiche Anstrengungen unternommen, um Lehr-/Lernmaterial über Repositorien für einen breiten Nutzerkreis zentral zugängig zu machen [Ari06]. Dabei ist allerdings die nahtlose Integration in bestehende Lernplattformen schwierig, sodass oft eine Reduktion auf ein Minimum unterstützter, einfacher Formate erfolgt – denn hochwertiges Material ist oft *systemspezifisch*. Angesichts eines fehlenden, weithin akzeptierten Standards für die detaillierte Spezifikation vom Lehr-/Lernmaterial (über Packaging-Formate wie SCORM hinaus) spielen viele Lernplattformen ihr Potenzial für interaktives, aus didaktischer Sicht vielversprechendes Material erst und gerade unter Nutzung systemspezifischer Formate aus. Diesem Problem kann mit einem Service-basierten Ansatz begegnet werden, durch den Inhalte nicht statisch in Lernplattformen importiert, sondern mitsamt der assoziierten Funktionalität dynamisch in das System integriert werden [Krö06][Dem07].

Zunehmende Verbreitung findet darüber hinaus das *Lecture Recording*, d. h. das Aufzeichnen von Lehrveranstaltungen (mit einem Video des Vortragenden, seinen Folien sowie Annotation darauf). Ergänzend erfolgt i. Allg. eine Indexierung der Inhalte, sodass eine Volltextsuche über den Medienströmen möglich wird [Zie04]. Diese Anreicherung der Informationsdarstellung mit audio-visuellen Medien erzeugt eine soziale Präsenz und entbindet deshalb von der engen Kopplung an eine konkrete Lehrveranstaltung; dadurch wird neben der Nachbereitung von Vorlesungen auch das reine Selbstlernen unterstützt. Eine organisatorische Unterstützung für die Verantwortlichen bietet die Automatisierung der Publikation von Vorlesungsaufzeichnungen in Lernplattformen [Kop07][Mer07]. Allerdings handelt es sich dabei um *systemspezifische Punkt-zu-Punkt-Verbindungen*. Auch die Möglichkeiten für eine erweiterte Annotation und Suche [SW07][Hür06] sind systemspezifisch und daher kaum plattform- bzw. hochschulübergreifend nutzbar.

Neuere Entwicklungen setzen daher gezielt auf *virtuelle Umgebungen* [Gos08][Voß08], indem sie etablierte eLearning-Plattformen verlassen und sich so einen breiteren Teilnehmerkreis erschließen. In virtuellen Welten lässt sich zudem durch die 3D-Optik und den Live-Charakter eine pädagogische Wirkung weit über die von Vorlesungsmitschnitten hinaus erzielen. Allerdings stecken die Ansätze zur Kopplung dieser Welten mit bestehenden Infrastrukturen der Hochschule noch in den Kinderschuhen [KL06].

Wünschenswert ist deshalb ein systematischer Ansatz zur Kopplung verschiedenster eLearning-Werkzeuge bzw. -Plattformen. Ein Ziel hierfür ist es, manuelle Prozessschritte bei der Produktion und Distribution von Inhalten zu umgehen. Ein weiterer, durch flexible Kombinierbarkeit verschiedener Plattformen entstehender Effekt wäre die transparente Kopplung verschiedener eLearning-Szenarien. So ist es z. B. denkbar, dass eine Lehrveranstaltung automatisch und gleichzeitig ihren Weg in klassische Lernplattformen und sogar virtuelle Welten findet: Während der Veranstaltung werden die im Hörsaal gezeigten Folien präsentiert und ggf. durch ein Live-Video ergänzt. Durch die Möglichkeit zur Interaktion mit dem Dozenten (etwa per VoiceChat) werden synchrone Lernprozesse wie in der Präsenzlehre, jedoch unter Einsatz Web-basierter Plattformen umgesetzt. Im Anschluss an die Veranstaltung kann die Aufzeichnung dann in der Lernplattform (oder der vituellen Welt) asynchron abgerufen werden. Eine derartige Interoperabilität verschiedener Werkzeuge und Technologien ist insbesondere für hochschulübergreifende Lehr- und Lernszenarien wichtig, die künftig vermehrt an Bedeutung gewinnen werden [LT07].

Der Beitrag skizziert ein zunächst prototypisch entwickeltes System zum Umsetzung dieser Vision. Er geht dabei nicht auf virtuelle Welten oder grundsätzliche Aspekte der Kopplung verschiedener Systeme ein, sondern v. a. auf die synchrone und asynchrone Integration von Vorlesungsvideos in eine Lernplattform. Dieses Szenario dürfte für die Hochschullehre derzeit die größere Praxisrelevanz haben.

2 Architekturen und Technologien zur Distribution und Integration dynamischer Medien

Als *Architekturmodell* für das oben geschilderte Szenario einer systematischen Kopplung verschiedener Werkzeuge und Plattformen bieten sich das Broker-Konzept bzw. die Service-Orientierte Architektur (SOA) an. Hier werden Anbieter (Provider) und Nutzer

(Consumer) von Diensten durch Einsatz eines Brokers entkoppelt. Das bedeutet für das System eine einfache Erweiterbarkeit sowie große Flexibilität zur Laufzeit. Wie in Bild 1 am Beispiel der Übertragung von Vorlesungsvideos gezeigt, erfolgt die Kommunikation zwischen diesen Komponenten in drei Schritten: (1) Der Provider registriert seine Dienste und deren Merkmale beim Broker. (2) Der Consumer lässt den Broker nach geeigneten Diensten suchen. (3) Der Consumer baut eine Verbindung zum Provider auf und nutzt dessen Dienst. Diese Architektur vereint verschiedene Anbieter, Nutzer und auch Typen von Diensten. Auf Seiten von Consumer und Provider ist lediglich die Implementierung einer minimalen Service-Schnittstelle erforderlich. Alle weiteren Funktionen werden von einer Service-basierten Middleware ausgeübt.

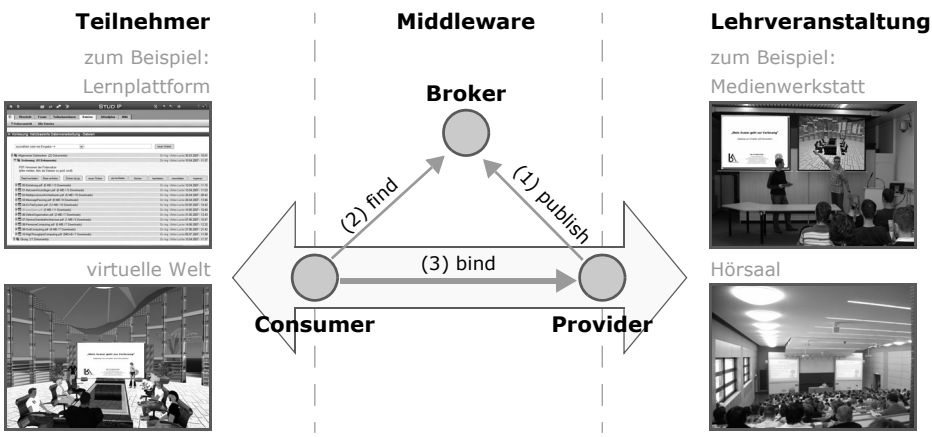

Bild 1 Service-Orientierte Architektur am Beispiel der Vorlesungsübertragung

In der Hochschule kommen einige weitere Eigenschaften des Systems hinzu, die den Einsatz einer SOA verlangen bzw. z. T. auch erst möglich machen. Einerseits sind eine Reihe fester Server für alle Clients bekannt (z. B. Lernplattformen), die als Broker genutzt werden können. Andererseits sind Vorlesungsinhalte i. Allg. nicht sicherheitsrelevant, was die Implementierung des Szenarios vereinfacht. Probleme, die sich aus dem Urheberrecht bzw. Datenschutz ergeben, können durch zusätzliche Basisdienste zur transparenten Authentifizierung und Mechanismen des entsprechenden Lernplattformen behandelt werden. Darüber hinaus stellt die große Zahl von Nutzern und Diensten hohe Anforderungen an die Skalierbarkeit des Systems, die mit herkömmlichen Modellen (z. B. Punkt-zu-Punkt-Kopplungen nach dem Client-Server-Prinzip) nicht umsetzbar sind. Eine weitere Rahmenbedingung, nicht nur für Nutzer aus technik-fremden Fachgebieten, ist eine Client-seitige Nutzung des System ohne vorherige Installation (also z. B. über Web-Browser).

Auf dieser Basis kann die Eignung verschiedener *Technologien* untersucht werden. Populäre SOA-Vertreter sind z. B. Jini [Bad00] und OSGi [OSG07]. Beide stammen aus dem Java-Umfeld und bringen daher inhärente, für das geplante Szenario jedoch nicht benötigte Sicherheitsmechanismen mit. Jini erlaubt zudem einen Einsatz ohne Broker, da das lokale Netzwerk automatisch nach verfügbaren Diensten durchsucht wird. Beide Technologien erfordern aber spezielle Installationen auf dem Client und scheiden daher

aus. Dagegen können gerätespezifische Dienste (u. a. Audio/Video) mit UPnP [JW03] transparent im Web-Browser genutzt werden. Allerdings wird für jedes anzubindende Gerät ein separater HTTP-Server benötigt, und es gibt Probleme beim gleichzeitigen Zugriff mit mehreren Clients auf ein Gerät. Die verbleibende und für das geschilderte Szenario am besten geeignete Technologie sind daher Web Services [Mel07]. Sie können ebenfalls im Web-Browser genutzt werden und erfordern daher keine Client-seitige Installation. Dank effizienter, XML-basierter Kommunikation skaliert das System sehr gut. Zwar werden dedizierte Broker benötigt, was jedoch im Hochschulumfeld kein Problem ist. Es gibt keine inhärenten Sicherheitsmechanismen, aber diese können als ergänzende Dienste bereitgestellt oder über das Web Service Framework selbst ergänzt werden.

Zur *Distribution* der Medienströme aus den Lehrveranstaltungen heraus müssen Gerätespezifische Schnittstellen der eingesetzten Medientechnik genutzt werden. Das bedeutet, dass bereits bei der Beschaffung der Technik auf deren Fernsteuerbarkeit zu achten ist. Weiter soll hier jedoch nicht auf die Spezifika der Geräteanbindung eingegangen werden.

Für die *Integration* der Medienströme in eine Lernplattform sollten Technologien zum Einsatz kommen, die für den Client transparent sind und sich möglichst nahtlos in die vorhandene Nutzungsoberfläche integrieren lassen. Denkbar wäre hierfür z. B. die User Interface Markup Language (UIML) [HA04], die eine abstrakte Spezifikation von GUI-Elementen erlaubt und auf verschiedene konkrete Sprachen bzw. Layouts abgebildet werden kann. Im Detail gibt es jedoch einige Unterschiede, da die Plattformen z. T. ein unterschiedliches Vokabular nutzen. Ähnlich arbeiten Web Services for Remote Portlets (WSRP) [KLT03], die auf abstrakte Weise die Interaktion von Portal-Elementen mit Web Services beschreiben und im GUI als layout-spezifischer HTML-Code umgesetzt werden. Der Fokus liegt hier auf der Aggregation mehrerer Web Services, was in dem geplanten Szenario zunächst unnötig ist. Eine weitere Möglichkeit ist die Synchronized Multimedia Integration Language (SMIL) [Bul07], die eine Kombination von Präsentationen aus verschiedenen Medienobjekten und Steuerelementen erlaubt. Sie wird allerdings nur durch wenige Player unterstützt, und oft übernehmen bereits die Aufzeichnungswerkzeuge die Synchronisation der Audio-/Video-Elemente. Schließlich können mit Asynchronous JavaScript and XML (AJAX) [Bec07] auch Techniken des Web 2.0 zum Einsatz kommen. Der asynchrone Datentransfer beim Nachladen der Elemente von Web-Seiten verhindert eine Beeinträchtigung des Nutzers während der Aktualisierung von Inhalten, was z. B. für die Übertragung von datenintensiven Vorlesungsvideos vor großem Vorteil ist. Zudem basiert AJAX auf JavaScript, was ohnehin oft in existierenden Plattformen verwendet wird, sodass eine nahtlose Integration Server- wie Client-seitig erfolgen kann.

Das folgende Kapitel schildert auf Basis dieser Vorüberlegungen die Umsetzung eines Systems zur Integration von Vorlesungsmitschnitten (sowohl live als auch aus dem Archiv) in die Lernplattform Stud.IP.

3 Systemarchitektur und Realisierung

3.1 Angestrebte Nutzungsmöglichkeiten

Die mit dem o. g. Szenario verbundenen Prozesse sind grundsätzlich in synchron (d. h. die Live-Übertragung einer Lehrveranstaltung) und asynchron (d. h. die Wiedergabe einer Aufzeichnung) zu unterscheiden.

Die *synchrone Übertragung* einer Veranstaltung beginnt in einem mit der benötigten Medientechnik ausgestatteten Vorlesungssaal (nach den üblichen Vorbereitungen, wie dem Öffnen der Folien und der Konfiguration der Präsentationstechnik) mit dem Starten der Aufzeichnung und dem Senden des Live-Videos. Interessenten der Veranstaltung können sich in den angeschlossenen Zielplattformen (z. B. Stud.IP oder Second Life) über den Start der Übertragung informieren und dort auch direkt daran teilnehmen. Mit dem Ende der Veranstaltung endet die Übertragung, und die Aufzeichnung kann durch den Vortragenden archiviert werden.

Bei der synchronen Veranstaltungsteilnahme kann zudem eine *Interaktion* mit dem Vortragenden realisiert werden. Dazu könnten einerseits mit den inhärenten Mitteln der jeweiligen Plattformen (z. B. Chat) Fragen gestellt werden. Voraussetzung dafür wäre jedoch eine gleichzeitige Präsenz des Vortragenden auf all diesen Plattformen, was bei mehreren Systemen und zudem im Rahmen einer regulären Lehrveranstaltung kaum realisierbar ist. Anderseits kann das Feedback von den Teilnehmern aber auch in gleicher Weise wie die Veranstaltung selbst übermittelt werden, also über dynamische Medien (z. B. VoiceChat) zurück in die Präsenzveranstaltung. Der Vortragende wird dabei – wie in einer normalen Vorlesung auch – nur mit Bild und Ton des Fragenden konfrontiert, was sich bedeutend besser in den Ablauf einer Lehrveranstaltung einfügt.

Mit der *asynchronen Wiedergabe* besteht im Nachhinein die Möglichkeit, archivierte Veranstaltungen aus verschiedenen Plattformen heraus anzusehen. Auch der Zugriff auf das Archiv erfolgt Service-basiert und ist damit leicht in eine Vielzahl unterschiedlicher Systeme einzubinden. In Lernplattformen kann dies mit organisatorischen Informationen zur Veranstaltung bzw. Mechanismen zur asynchronen Kommunikation (z. B. per Wiki, Forum) kombiniert werden. Die Möglichkeit zur synchronen Interaktion ist hier natürlich nicht mehr gegeben.

Unabhängig vom Präsenzbetrieb ist darüber hinaus die Variante denkbar, ausschließlich *virtuelle Veranstaltungen* (also z. B. als Video vom Arbeitsplatz des Dozenten oder als Bildsequenz aus einer virtuellen Welt heraus) zur Verfügung zu stellen. Auf die technische Umsetzung hat eine solche abweichende Quelle des Datenstroms keine Auswirkungen.

3.2 Architektur des Frameworks

Aus den bisherigen Überlegungen ergibt sich die in Bild 2 dargestellte Systemarchitektur. Dabei liegt der Fokus auf dem Zugang zu Live- und archivierten Veranstaltungen über eine Lernplattform. Das System beinhaltet die folgenden Komponenten:

- Ausgangspunkt einer Lehrveranstaltung ist eine *Medienwerkstatt*. Hier kommen ein Stream-Provider (zur Bereitstellung des Live-Streams) und ein Autorenwerkzeug

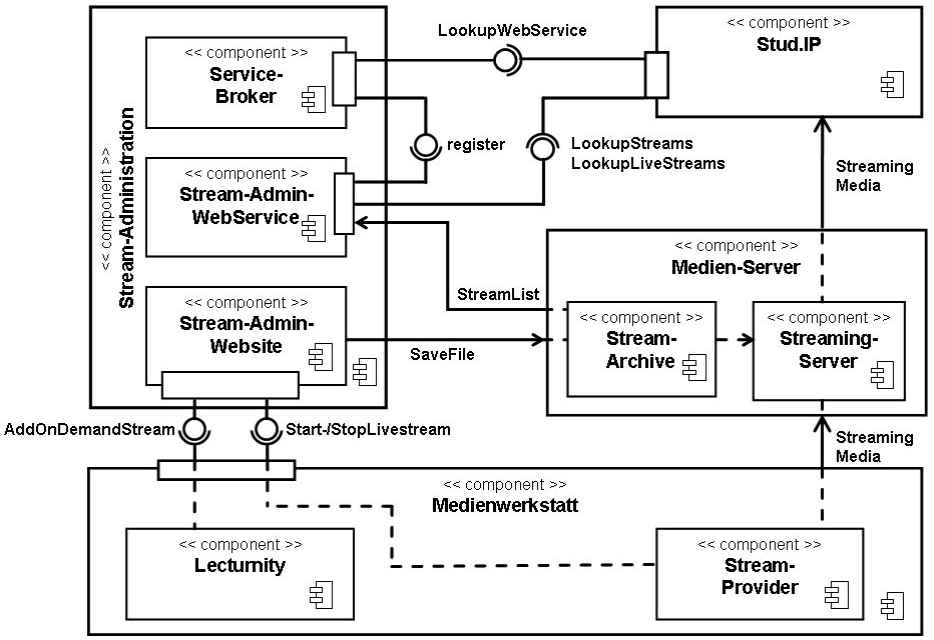

Bild 2 Service-Orientierte Architektur zur flexiblen Integration von synchronen und asynchronen Mediendiensten in eine Lernplattform

(zum Aufzeichnen der Veranstaltung für die anschließende Archivierung, in diesem Fall Lecturnity [imc08]) zum Einsatz.

- Der *Medien-Server* realisiert die Bündelung synchroner und asynchroner Prozesse. Er enthält Stream-Archive und Streaming-Server. In einem (u. U. externen) Archiv vorgehaltene oder live durch einen Stream-Provider produzierte Streams werden über den Streaming-Server veröffentlicht.
- Für die *Administration der Streams* kommen drei Teilkomponenten zum Einsatz, die technologisch auf Web Services beruhen. Kern ist der Stream-Admin-WebService, der auf Anfrage eine Liste der auf dem Streaming-Server verfügbaren Streams liefert. Der Dozent signalisiert hierfür über die Stream-Admin-Website (eine einfache, Web-basierte Oberfläche) zuvor den Start eines Live-Streams sowie Metadaten zur Veranstaltung. Der Web Service ist bei einem Service-Broker (realisiert mittels WS-Inspect [Bal01]) registriert, welcher den zentralen Anlaufpunkt für die Suche nach Mediendiensten darstellt.
- Als Stream-Consumer kommt hier *Stud.IP* [dat08] zum Einsatz. Mit der Liste der verfügbaren Veranstaltungen ist die Lernplattform in der Lage, einen ausgewählten Stream vom Medien-Server abzurufen und direkt in die Anzeige zu integrieren.

Aufgrund ihrer Modularität ist diese Architektur vielfältig erweiterbar. Einerseits sind neue Datenquellen (z. B. Hörsaal oder Labor; über Einbindung dedizierter Dienste) sowie Datensenken (z. B. andere Lernplattformen oder auch virtuelle Welten) integrierbar. Hier spielt die Service-Orientierte Architektur ihre Vorteile aus [Leh08]. Dabei wird von den

zur Produktion bzw. Nutzung des Materials eingesetzten Werkzeugen (hier Lecturnity und Stud.IP) völlig abstrahiert. Auch bereits bestehende Archive können in das System aufgenommen werden. Ferner kann der Einsatz weiterer Streaming-Server sinnvoll sein – sowohl zur Leistungssteigerung als auch zur Erweiterung um neue Datenformate. Selbst die Nutzung einer anderen Service-Technologie ist parallel zur bestehenden Stream-Administration möglich, ohne bestehende Komponenten zu modifizieren. Das System verbindet also eine hohe Flexibilität mit einem geringen Aufwand für die Implementierung und folglich auch Wartung. Es passt sich deshalb gut in bestehende Infrastrukturen ein.

Zur Realisierung der Architektur wird weitestgehend auf bestehende Lösungen zurückgegriffen. Dies betrifft die Medienwerkstatt, den Streaming-Server und Stud.IP. Die Komponenten zur Stream-Administration, der Medien-Server sowie nötige PlugIns für die Lernplattform Stud.IP wurden hinzugefügt. Die beiden folgenden Abschnitte beschreiben die Realisierung von Medien-Server und Stud.IP-Anbindung. Die Infrastruktur für die Service-basierte Kopplung von beliebiger Plattformen für Präsenz- und virtuelle Lehre wird detailliert in einem separaten Beitrag beschrieben [Leh08].

3.3 Umsetzung von Archiv- und Streaming-Server

Aus den in Abschnitt 3.1 geschilderten Nutzungsmöglichkeiten resultieren drei zu implementierende Anwendungsfälle: Durchführen, Archivieren und Abrufen einer Veranstaltung. Die Arbeit des Medienservers soll hier nur ausschnittweise anhand des letzten, komplexesten Falls erläutert werden. Bild 3 zeigt im Sequenzdiagramm die beteiligten Systemkomponenten und deren Interaktionen.

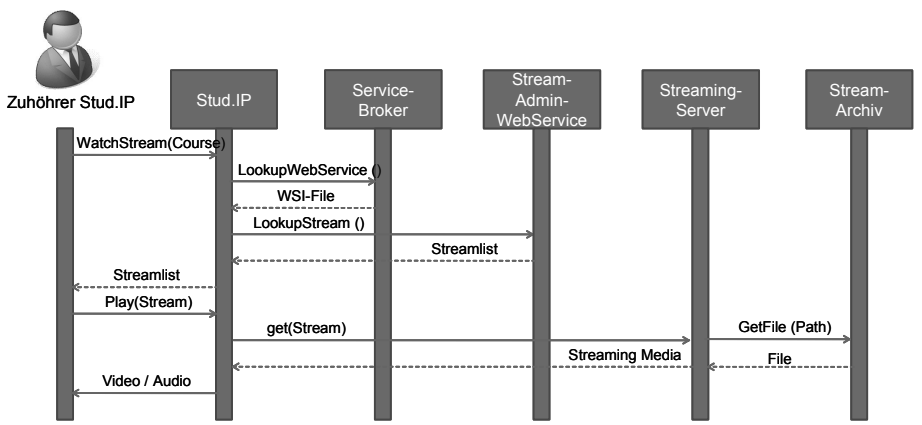

Bild 3 Abrufen einer Veranstaltung vom Medien-Server durch den Nutzer in Stud.IP

Betritt der Nutzer einen Veranstaltungsbereich im Stud.IP, wird beim Broker eine Anfrage nach hierzu registrierten Mediendiensten ausgelöst. Daraufhin wird vom verantwortlichen Stream-Admin-WebService eine Liste der für diese Veranstaltung vorhanden Daten (Live-Streams oder aus dem Archiv) erfragt, aus der sich der Nutzer das gewünschte Material auswählt. Diese Daten werden vom Streaming-Server angefordert, der sie sich wiederum

vom Stream-Provider (bei synchroner Übertragung) oder aus dem Archiv (bei asynchroner Wiedergabe) holt. Die Audio-/Video-Daten werden dann direkt im Stud.IP angezeigt.

Dabei stellt sich die zunächst trivial erscheinende Frage nach einem geeigneten Datenformat. Es existiert bereits eine Reihe von Formaten für das Live-Streaming (z. B. Real, Windows Media oder Quicktime) und für den Download (z. B. Real, Windows Media oder Flash, aber auch native Formate ausgewählter Applikationen wie Lecturnity). Für diese Formate sind Player für nahezu alle Plattformen verfügbar, sei es als Browser-PlugIn oder als eigenständige Applikation. Einschränkungen ergeben sich jedoch, sobald der Bereich klassischer Lernplattformen (d. h. Browser-Anwendungen) verlassen wird. So unterstützt Second Life als weitere mögliche Zielplattform für Videos nur Quicktime. Ein anderes Spezifikum ist die im Lecturnity-Format mögliche Volltextsuche nach Folieninhalten. Streaming-Formate unterstützen das generell nicht, reduzieren aber die Wartezeiten beim Datentransfer deutlich. Als Konsequenz bleibt folglich, mehrere Formate bereitzustellen und dem Nutzer die Wahl zu überlassen (sowohl für einen schnellen Einblick per Stream als auch für die vertiefende Arbeit mit dem Material nach komplettem Download).

Für den Streaming-Server ist eine Reihe von Lösungen verfügbar, die sich neben dem Preis auch deutlich in der Menge der unterstützten Datenformate und gleichzeitigen Streams unterscheiden. Open-Source-Lösungen können in diesen beiden für den Einsatz in der Lehre wichtigen Aspekten i. Allg. nicht mit professionellen Systemen mithalten. Aufgrund der großen Leistungsspektrums wurde deshalb für den Prototypen eine Demo-Version des Helix Server Unlimited [Rea08] eingesetzt.

3.4 Anbindung von Stud.IP

Angesichts der in Kapitel 2 diskutierten Vor- und Nachteile einzelner Technologien wurde für die Integration der Vorlesungsmitschnitte in Stud.IP AJAX ausgewählt. Es handelt sich zwar um eine Speziallösung nur für diese Lernplattform, die aber trotzdem einfach auf andere HTML-basierte Plattformen übertragen werden kann. Parallel können die über das System bereitgestellten Dienste z. B. auch mittels WSRP oder UIML in weitere Zielplattformen integriert werden.

Bild 4 zeigt einen Screenshot der modifizierten Nutzungsoberfläche von Stud.IP. Dort wurde der Datei-Bereich mit den nun treffenderen Begriff *Medien* überschrieben (1) und die Ordnerstruktur um die Möglichkeit zum Zugriff auf Vorlesungsmitschnitte ergänzt. Dies erfolgt über eine chronologisch sortierte Liste der zur Lehrveranstaltung bereits im Archiv verfügbaren Aufzeichnungen (2) sowie ggf. eines Live-Streams (3). Dabei erhält der Nutzer mit Datum und Titel weitere Informationen zur Aufzeichnung. Für jeden Mitschnitt kann er dann zwischen mehreren Formaten wählen. Anschließend öffnet sich ein Player (4) mit dem ausgewählten Material.

Für den Vortragenden wird im Stud.IP ein Link zur einfachen Administration der Mitschnitte (d. h. zur Stream-Admin-Website) vor bzw. nach der Veranstaltung bereitgestellt. Während der Veranstaltung erfolgt die Konfiguration der Medientechnik bzw. -dienste lokal in der Medienwerkstatt.

Das entwickelte Stud.IP-PlugIn nutzt JavaScript unter Verwendung der bereits im System definierten Layout-Templates. Es ist somit nicht speziell für die aktuelle Version

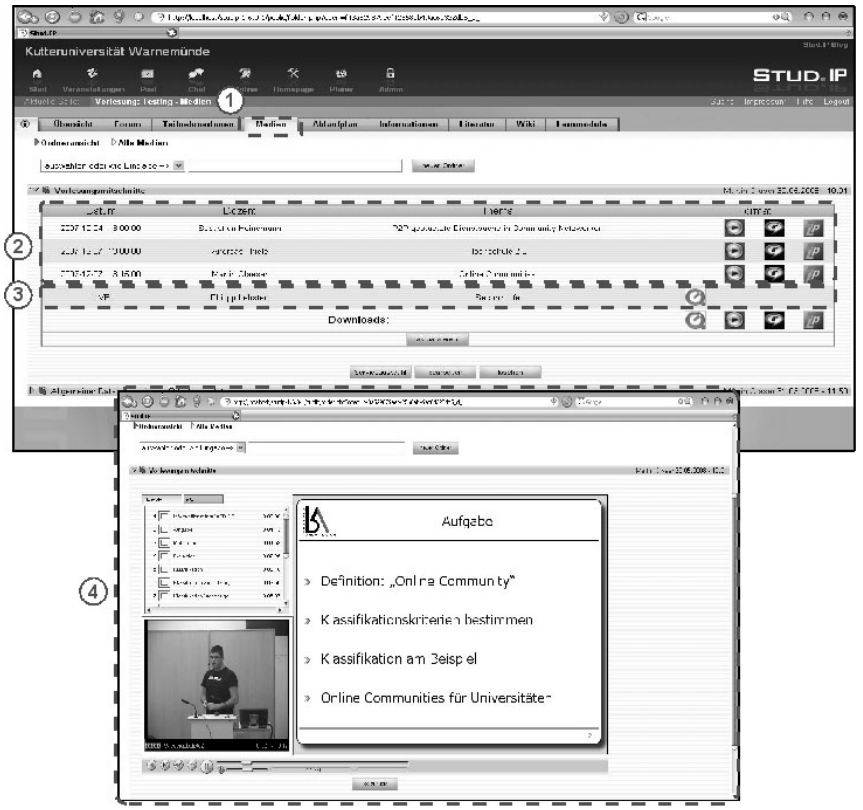

Bild 4 Modifizierte Stud.IP-Oberfläche zur nahtlosen Integration dynamischer Medien

konzipiert, sondern bleibt auch für zukünftige Weiterentwicklungen lauffähig (solange die prinzipielle Struktur und Funktionsweise erhalten bleiben).

Die beschriebene prototypische Umsetzung stellt zunächst nur eine Anbindung von Stud.IP dar. Durch den Service-basierten Ansatz ist aber eine einfache Integration in beliebige weitere Plattformen möglich.

4 Bewertung und Erweiterungsmöglichkeiten

Mit dem vorliegenden Prototypen sind zwei wesentliche Vorteile gegenüber den in Kapitel 1 vorgestellten, bisherigen Entwicklungen verbunden:

- Die Nutzung einer allgemeingültigen, Service-basierten Schnittstelle erlaubt die Kopplung verschiedenster Teilsysteme – sowohl Provider- als auch Consumer-seitig. Das geht deutlich über die bisher bekannten, systemspezifischen Punkt-zu-Punkt-Verbindungen hinaus.
- Die transparente Einbindung dynamischer Medienströme und die damit verbundenen Interaktionsmöglichkeiten erlauben nun auch einen Einsatz von Lernplattformen für

synchrone Szenarien der Präsenzlehre. Dies bisherige Beschränkung auf die Vor- und Nachbereitung von Lehrveranstaltungen wird aufgehoben.

Zwar greift die Stud.IP-Realisierung nur auf *eine* Lernplattform aus der Vielzahl der an Hochschulen derzeit eingesetzten Systeme zurück. Jedoch ist der Aufwand zur Integration eines Service-PlugIns, das nur die Funktionalität eines zentral vorgehaltenen Dienstes weitergibt, vergleichsweise einfach in Relation zur aufwändigen Integration der gesamten Funktionalität in ein System. Dadurch kann das dargestellte System flexibel um weitere Nutzer und auch Anbieter von Diensten erweitert werden. Darüber hinaus stellt diese Architektur eine Systemkopplung jenseits der lokalen Grenzen von administrativen Verantwortungsbereichen dar, was insbesondere im Hochschulumfeld zu einer größeren Akzeptanz als eine zentralisierte Struktur führen wird. Weiterhin ist die Lösung mit den generellen Vorteilen einer SOA verbunden: geringerer Wartungsaufwand, redundantes Angebot verwandter Dienste, damit Verfügbarkeit von Ausweichlösungen im Fehlerfall, größere Flexibilität für den Nutzer, schnellere Reaktion des Systems auf Veränderungen usw.

Denkbare Erweiterungen des dargestellten Systems betreffen eine größere Menge der aufgezeichneten Lehrveranstaltungen, genutzten Vortragsumgebungen, angesteuerten Lernplattformen, beteiligten Institutionen, ... – gerade in solch heterogenen Umgebungen spielt der Service-basierte Ansatz seine Stärken aus, da ohne Änderungen an der Kern-Infrastruktur ein deutlicher Mehrwert für die Nutzer erzielbar ist. Nicht nur Studenten würden von einer großen Zahl an Vorlesungsmitschnitten profitieren, insbesondere wenn diese nahtlos in die gewohnte Lernumgebung integriert sind. Auch die Dozenten der Veranstaltungen werden einerseits von organisatorischen Aufgaben entlastet und andererseits um neue Inhalte bereichert.

Die Interaktion mit dem Dozenten während einer Live-Veranstaltung ist technisch bzw. organisatorisch unproblematisch. Aus Sicht der Didaktik und Usability bleiben jedoch geeignete Mechanismen zu verifizieren, wie Feedback von entfernten Teilnehmern in die Präsenzlehre integriert werden kann, ohne den Veranstaltungsablauf zu stören.

Zudem wäre eine Übertragung des Konzeptes auf andere Dienste der Hochschule (nicht nur für das eLearning/eTeaching, sondern auch für eScience und eAdministration) im Sinn einer Service-Orientierten Universität [Fis07] mittelfristig realisierbar und auch sinnvoll.

Literatur

[Ari06] Ariadne Foundation for the European Knowledge Pool: „Ariadne“, letztes Update: 2006. http://www.ariadne-eu.org/

[Bad00] H. Bader: „Jini : die intelligente Netzwerkarchitektur in Theorie und Praxis“, Addison-Wesley, 2000.

[Bal01] K. Ballinger, P. Brittenham, A. Malhotra, W. A. Nagy, S. Pharies: „Web Services Inspection Language (WS-Inspection) 1.0“, GXA Specification, November 2001. http://www.serviceoriented.org/ws-inspect.html

[Bec07] T. Beckert: „Web 2.0 und Ajax : Ein erster Einstieg ins neue Internet“, Saarbrücken : VDM Verlag, 2007.

[Bul07] D. Bulterman, J. Jansen, S. Mullender, M. DeMeglio, J. Quint, P. Vuorimaa, S. Cruz-Lara, H. Kawamura, D. Weck, E. Hyche, X. García Pañeda, D. Melendi, T. Michel, D. Zucker: „Synchronized Multimedia Integration Language, SMIL 3.0“, W3C Working Draft, 2007.

[dat08] data-quest GmbH: „Stud.IP Portalseite“, letztes Update: 2008. http://www.studip.de/

[Dem07] E. Demidova, P. Karger, D. Olmedilla, S. Ternier, E. Duval, M. Dicerto, C. Mendez, K. Stefanov: „Services for Knowledge Resource Sharing & Management in an Open Source Infrastructure for Lifelong Competence Development“, 7th IEEE Int. Conference on Advanced Learning Technologies, (ICALT), Juli 2007.

[Fis07] S. Fischer: „SOAs an Hochschulen – practice what you preach!“, Workshop 'Pervasive University' im Rahmen der 37. GI Jahrestagung, September 2007.

[Gos08] KVHS Goslar: „Kreisvolkshochschule Goslar im Second Life“, letztes Update: 2008. http://www.vhs-secondlife.de/

[HA04] J. Helms, M. Abrams (Hrsg.): „User Interface Markup Language (UIML) Specification“, Working Draft 3.1, 2004.

[Hür06] W. Hürst, T. Lauer, R. Kaschuba: „Interfaces for interactive audio-visual media browsing“, 14th ACM Int. Conference on Multimedia, Oktober 2006.

[imc08] imc AG: „Rapid Authoring Tool LECTURNITY“, letztes Update: 2008. http://www.lecturnity.de/

[JW03] M. Jeronimo, J. Weast: „UPnP Design by Example: A Software Developer's Guide to Universal Plug and Play“, Intel Press, 2003.

[KL06] J. Kemp, D. Livingstone: „Putting a Second Life 'Metaverse' Skin on Learning Management Systems“, Second Life Education Workshop at SL Community Convention, San Francisco, August 2006.

[KLT03] A. Kropp, C. Leue, R. Thompson: „Web Services for Remote Portlets (WSRP) Specification“, OASIS Standard, 2003.

[Kop07] S. Kopf, F. Lampi, T. King, M. Probst, W. Effelsberg: „EDL-Editor: Eine Anwendung zur automatischen Aufbereitung von Vorlesungsvideos“, Die 5. e-Learning Fachtagung Informatik (DeLFI), September 2007.

[Krö06] R. Kröger, U. Lucke, M. Schmid, D. Tavangarian: „Web Services for the Integration of XML-based Content into Learning Platforms: a Three-level Model“, 2nd Int. Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA), November 2006.

[Leh08] P. Lehsten, A. Thiele, R. Zilz, E. Dressler, R. Zender, U. Lucke, D. Tavangarian: „Dienste-basierte Kopplung von virtueller und Präsenzlehre“, Die 6. e-Learning Fachtagung Informatik (DeLFI), September 2008.

[LT07] U. Lucke, D. Tavangarian: „Aktueller Stand und Perspektiven der eLearning-Infrastruktur an deutschen Hochschulen“, Die 5. e-Learning Fachtagung Informatik (DeLFI), September 2007.

[Mel07] I. Melzer: „Service-orientierte Architekturen mit Web Services : Konzepte – Standards – Praxis“, 2. Auflage, Spektrum Akademischer Verlag, 2007.

[Mer07] R. Mertens, M. Ketterl, O. Vornberger: „The virtPresenter lecture recording system: Automated production of web lectures with interactive content overviews“, Int. Journal of Interactive Technology and Smart Education, 4 (1), Februar 2007.

[OSG07] OSGi Alliance: „About the OSGi Service Platform“, Revision 4.1, 2007. http://www.osgi.org/documents/collateral/OSGiTechnicalWhitePaper.pdf

[Rea08] Real Networks Inc.: „Media Servers“, letztes Update: 2008. http://www.realnetworks.com/products/media_delivery.html

[SW07] H. Sack, J. Waitelonis: „Osotis – kollaborative, inhaltsbasierte Video-Suche“, Die 5. e-Learning Fachtagung Informatik (DeLFI), September 2007.

[Voß08] S. Voß, T. Vuong, D. Krömker, R. Müller: „E-Learning im Second Life. Eine Veranstaltung 'IT-Projektmanagment'“, Die 6. e-Learning Fachtagung Informatik (DeLFI), September 2008.

[Zie04] P. Ziewer: „Navigational Indices and Full Text Search by Automated Analyses of Screen Recorded Data“, World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education (E-Learn), November 2004.

Danksagung

Diese Arbeit wird teilweise im Graduiertenkolleg „Multimodal Smart Appliances for Mobile Applications“ durch die DFG gefördert. Die Autoren danken darüber hinaus dem studentischen Projektteam „Komplexe Software-Systeme“ vom Wintersemester 2007/ 2008 für ihre Unterstützung bei der Konzeption und Umsetzung der Dienste-Schnittstelle.