

Open Source-Lösungen zur Integration von eLearning und Hochschulmanagement

Peter A. Henning

Institute for Computers in Education I.C.E
Hochschule Karlsruhe
Moltkestraße 30
76133 Karlsruhe
p.henning@ice-karlsruhe.de

Abstract: Mit diesem Artikel wird die Notwendigkeit einer Integration von IT-Dienstleistungen einer modernen Hochschule in die elektronische Lernumgebung von Studierenden begründet, sowie konkrete Integrationskonzepte auf der Basis von Open Source-Produkten umrissen. Dabei wird ein unkonventionelles Konzept des direkten Durchgriffs auf Anwendungsdatenbanken vorgeschlagen.

1 Warum Integration ?

Noch vor fünfzehn Jahren erfuhren junge Menschen beim Eintritt in die Hochschule einen „Technologieschub“: Freier Computerzugang, vergleichsweise unbegrenzte Online-Zeit, hohe Bandbreite. Private Internet-Anschlüsse waren eher selten, eigene eMail-Adressen bei Schülern die Ausnahme. Wie sehr sich dies geändert hat, zeigen die systematischen Studien des Medienpädagogischen Forschungsverbundes Südwest MPFS [1-2], nach denen im Jahre 2006 bereits 47% der Jugendlichen zwischen 12 und 19 Jahren regelmäßig mit dem Computer für die Schule lernen – gegenüber 37% noch im Vorjahr. Bei jüngeren Kindern ist dies offenbar noch ausgeprägter. Studienanfänger des Jahres 2007 begreifen Computer als alltägliche Lerngeräte, ohne die eigentlich nichts mehr läuft. Ein Symptom dieses Wandels ist die Verbreitung von Notebook-Computern unter Studienanfängern, die wir vor zwei Jahren auf dieser Tagung berichtet haben [3]. Wesentlich dabei ist auch, dass sich dieser Trend nicht auf technische Fächer beschränkt, sondern Studienanfänger geisteswissenschaftlicher Fächer ebenso an die Nutzung des Computers gewöhnt sind.

Nicht nur fehlt also heutigen Studienanfängern der „Technologieschub“, sondern sie empfinden möglicherweise die IT-Leistungen ihrer Hochschule sogar als inadäquat: Bandbreiten- oder Speicherplatzbeschränkungen, als überzogen empfundene Sicherheitsvorkehrungen und fehlende Funktionalitäten wie Spamschutz sorgen angesichts entsprechender frei verfügbarer Services für ein eher negatives Bild der Hochschul-IT unter den Studierenden. Freemailer“ wie GMX oder WEB.DE stellen heutzutage komfortable kostenlose eMail-Dienste bereit. Junge Menschen verfügen deshalb in der

Regel über solche eMail-Adressen, bevor sie in die Hochschule eintreten – und nutzen diese auch danach.

Es stellt sich deshalb die Frage, welche IT-Leistungen Studierenden *während des Studiums* geboten werden müssen, damit diese darin einen echten Mehrwert sehen. Befragungen von Studierenden ergeben, dass dies erreicht werden kann, indem durch die Studierenden Computer auf breiter Front als unterstützendes Werkzeug für das Studium verwendet werden können, wenn demnach *inhaltliche* Aspekte des Studiums zusammen mit *organisatorischen* Aspekten angeboten werden. Ein Mehrwert für Studierende ergibt sich also, wenn beispielsweise

- Kursanmeldung, Prüfungsanmeldung, Rückmeldung,
- Notenansicht und Notenverbuchung,
- Zwischenzeugnisse, Transcript of Records und Zeugnisausgabe

als Bestandteile der persönlichen elektronischen Lernumgebung abgewickelt werden können. Diese Lernumgebung muss dabei nicht monolithisch sein, sie kann von der Bereitstellung von Manuskripten bis hin zur vollständigen elektronischen Lernplattform reichen. Darüber sollten natürlich Ausweise für die Studierenden multifunktional sein, den Zugang zu geschützten Bereichen ebenso ermöglichen, wie das Bezahlen in der Mensa und die Ausleihe in der Bibliothek.

Die Integration aller IT-Dienste *aus Sicht des studentischen Benutzers* ist deshalb ein wesentlicher Faktor im Wettbewerb der Hochschulen um die besten Studierenden und wird von diesen auch als solcher wahrgenommen.

2 Kernkompetenzen

Es kann jedoch auch die Frage nach den IT-Kernkompetenzen einer Hochschule *aus Sicht der Lehrenden* gestellt werden. Auch diese benötigen zunächst Internet-Zugang und Mailadressen – mit aktuellem Viren- und Spamschutz, hoher Speicherkapazität und weltweitem Web-Zugriff, selbstverständlich auch privater Mitnutzung. Diese Dienste sind hochschulintern zwar leistbar, doch bedingen sie einen hohen technischen und administrativen Aufwand. Dem gegenüber stehen standardisierte Angebote der kommerziellen Provider, die z.B. die Verwaltung von 5000 eMail-Adressen mit allen diesen Features für ca. 100 € pro Monat anbieten können. Es wird deshalb mittelfristig zu überlegen sein, inwieweit nicht ein „Outsourcing“ in diesem Bereich sinnvoll ist.

Zwar denken viele renommierte Hochschulen darüber nach, ob sich durch eine lebenslang(!) verfügbare eMail-Adresse für Absolventen eine langfristige Identifikation der Absolventen mit der Hochschule erreichen ließe. Doch scheint dies angesichts der Ausgangslage weder von den langfristigen Aufwänden her planbar zu sein, noch gibt es

den geringsten Hinweis darauf, dass sich durch eine solche technische Lösung die grandiosen Spendenaufkommen amerikanischer Spitzenhochschulen erzielen ließen [4].

Natürlich umfassen die IT-Leistungen einer Hochschule für ihre Mitarbeiter wesentlich mehr als dieses Basispaket. Doch gibt es nur einen einzigen Dienst, der nicht auch im industriellen Sektor ebenfalls auf breiter Front geleistet werden muss: Die Bereitstellung von Hochleistungsrechnern, etwa im Rahmen eines Grids, für wissenschaftliche Zwecke. Buchstäblich *alle anderen Services* – vom Web-Hosting über eine gesicherte Verwaltungsinfrastruktur – sind auch in großen Firmen vorhanden, werden also zunehmend auch von spezialisierten kommerziellen Unternehmen angeboten.

Die Frage des Outsourcing von IT-Leistungen im Hochschulbereich muss deshalb sogar noch weiter gestellt werden, als oben behauptet – dies soll allerdings nicht Inhalt des vorliegenden Artikels sein. Ein Schluss ist jedoch aus dem Gesagten zu ziehen: Der Betrieb von Rechenanlagen und Netzwerken für *grundlegende* IT-Dienste ist sicher heute keine wissenschaftliche Spezialität mehr – und doch aus Sicht der Hochschule ein wichtiger Wettbewerbsfaktor.

3 Prozesse in der Hochschule

Moderne Hochschulen basieren auf hoch komplizierten Geschäftsprozessen. Von der Behandlung und Auswahl von Studienbewerbern bis zur Ausstellung des Abschlusszeugnisses sind diese Prozesse in der Regel nicht standardisiert, beruhen teilweise auf den Fertigkeiten bestimmter Personen und stellen für Außen stehende oft ein schwer zu entzifferndes Konglomerat aus formalen und informalen Regeln dar. Das Wissen um diese Prozesse stellt also einen großen Teil des intellektuellen Kapitals dar – und sollte entsprechend gemanagt werden. Diese Erkenntnis und ihre Umsetzung in einer professionalisierten Administration verschafft z.B. amerikanischen Hochschulen einen erheblichen Vorsprung im Konkurrenzkampf um die besten Köpfe – allerdings um den Preis hoher Kosten [4]. An deutschen Hochschulen fehlt dieses Wissensmanagement – mit der schwer wiegenden Konsequenz, dass deren Prozesse extrem teuer in Ausbildung und Unterhaltung sowie schwer veränderbar sind.

Aus Sicht der modernen Informationswissenschaften sind deshalb die Geschäftsprozesse einer Hochschule ideale, überreife Kandidaten für die Hebung großer Effizienzreserven, indem sie verschlankt, formalisiert und schließlich durch Software modelliert werden.

Die Vielfalt der Abläufe stellt dabei aber nicht nur Motivation, sondern gleichzeitig eine wesentliche Hürde für eine Modellierung dar – denn sie erschwert die Standardisierung von Software ungemein. Der oft beschrittene Lösungsweg, Geschäftsprozesse an die verfügbare Software eines Anbieters anzupassen, sorgt zwar für eine gewisse Vereinheitlichung. Doch naturgemäß auf niedrigem Niveau und ohne die Möglichkeit zur Anpassung an individuelle Bedürfnisse – und damit nicht mehr zeitgemäß.

Moderner ist die Verwendung von hochgradig konfigurierbarer Software zur Prozessmodellierung. Dies reicht von Content Management-Systemen mit Prozesskomponenten bis hin zu ERP (Enterprise Resource Planning)-Systemen, in der die vorhandenen Geschäftsprozesse abgebildet werden. Typische Beispiele hierfür sind das CMS PiroBase und die Software-Suite von SAP. In beiden Systemen wurden im Rahmen des WUSKAR-Projektes Geschäftsprozesse von Hochschulen modelliert (siehe Bericht auf der DFN-Arbeitstagung 2006 [5]). Für die Zwecke dieses Artikels wollen wir von diesen Systemen als **URP = University Resource Planning-Systemen** sprechen.

Die bei WUSKAR gemachten Erfahrungen mit diesen Systemen zeigen, dass diese Art der Prozessmodellierung an Hochschulen folgende Nachteile schafft:

- hoher Aufwand, der bei Versions- oder Systemwechseln der kommerziellen Softwarekomponenten in kaum planbarer Weise wieder anfallen kann.
- Abhängigkeit von herstellerepezifischen Details der Software, die schlecht oder gar nicht dokumentiert sind.
- Abhängigkeit von extrem spezialisiertem Expertenwissen.

Darüber hinaus ist mit dem Einsatz eines solchen Systems für das Hochschulmanagement keinesfalls das in Abschnitt 1 beschriebene Integrationsproblem gelöst – denn derzeit lassen sich solche kommerziellen prozessmodellierenden Systeme nicht *in* eine elektronische Lernumgebung für Studierende integrieren. Umgekehrt gibt es derzeit keine Bestrebungen der Hersteller, Software für das Hochschulmanagement mit den Funktionalitäten einer elektronischen Lernumgebung auszustatten [6,7].

4 Der vierte Weg

Die zu lösende Aufgabe besteht also darin, die Integration von elektronischen Lernumgebungen mit prozessmodellierender Software zu vollziehen. Natürlich könnte diese Integration auch mit dezidierten Schnittstellen erfolgen, die entweder von den Programmen direkt oder durch eine Middleware bedient werden. Im ersten Fall wären die Lösungen einer Peer-to-Peer, einer Hub-and-Spoke oder einer Bus-Architektur denkbar – mithin also eine enge Integration und damit für heterogene IT-Landschaften ungeeignet.

Die lockere („loose“) Integration kennt außer der Middleware noch andere Verfahren, die aber alle mit Nachteilen verbunden sind. Sowohl die Anwendung einer Plugin-Architektur, als auch die Nutzung von Web-Services erfordern spezielle Anpassungen in den beteiligten Programmen. Ein „Vierter Weg“ zur lockeren Integration besteht darin, zusätzlich zu bestehenden Systemen (Lernumgebung und Management-System) einen Workflow-Server einzusetzen. Als erstes Beispiel zu diesen Möglichkeiten sei die an der Hochschule Karlsruhe verwendete Kombination aus der elektronischen Lernumgebung ILIAS [8,9] und den Softwareprodukten der Fa. HIS [7] vorgestellt.

Bei ILIAS handelt es sich um eine Open Source-Lernumgebung auf LAMP-Basis (Linux, Apache, mySQL und PHP), die an der Universität zu Köln entwickelt wurde und in jeder Hinsicht dem industriellen Standard genügt [10]. Für die hier geschilderten Integrationszwecke ist es deshalb von besonderer Bedeutung, weil es in hohem Maße auf XML basiert und vollständig SCORM-kompatibel ist [11]. Damit ist es prädestiniert für die Aufnahme von Metadaten (etwa zur Kurs- und Vorlesungsorganisation) aus URP-Systemen.

Konkret soll der Fall betrachtet werden, bei dem die Anmeldungen Studierender für einen Kurs im ILIAS eLearning-System aus diesem extrahiert und in die Datenbank des URP-Systems übertragen werden sollen. Durch den direkten Zugriff auf den Source-Code von ILIAS wäre eine entsprechende Entwicklung auf Seiten von ILIAS problemlos möglich. Damit ginge allerdings die Möglichkeit zum einfachen Update verloren.

In Abbildung 1 werden deshalb die Alternativen dargestellt. Zwei dieser Alternativen setzen voraus, dass das URP-System über eine XML-basierte Schnittstelle verfügt. In der ersten werden durch eine externe Instanz XML-Datenobjekte aus ILIAS ausgelesen, einer Transformation unterworfen und an das URP-System weitergeleitet (XML-2-XML). Im zweiten Fall greift diese externe Instanz direkt auf die ILIAS-Datenbank zu (DB-2-XML).

Der dritte alternative Fall schließlich besteht darin, auch bei dem URP-System auf eine dezidierte Schnittstelle zu verzichten und direkt auf den beiden Datenbanken zu operieren (DB-2-DB).

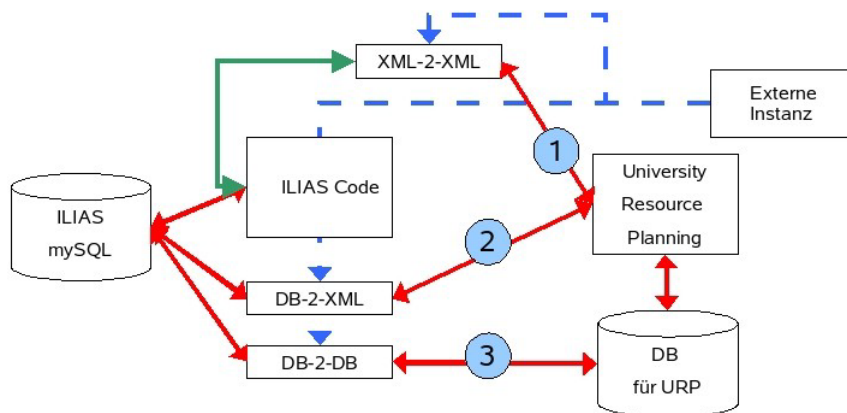


Abb.1. Lose Kopplung eines ILIAS-Systems an ein URP-System, mit den drei im text geschilderten Alternativen. Bevor wir uns der Frage der Prozessdefinition widmen, sei zunächst ein Beispiel für die hier erwähnte „externe Instanz“ genannt.

Dafür hat sich das Open Source-System Cocoon bewährt [12]. Es basiert auf einem *Generator-Transformer-Serializer-Paradigma*, bei dem Daten aus einer Datenquelle gelesen werden, durch eine Kaskade von Transformatoren (die via web-basiertem Formular durchaus komplexe Interaktionen auf den Daten ausführen können) geschickt und „publiziert“. Diese Publikation kann ohne Weiteres wieder im Schreiben einer Datenbank bestehen – aber auch z.B. PDF-Dateien erzeugen.

Insbesondere für den hier diskutierten Zweck einer Service-Erweiterung der Hochschulmanagementsysteme ist dies wichtig – so etwa könnten Transcripts of Record auf diese Weise generiert werden. Abbildung 2 zeigt schematisch das Paradigma von Cocoon.

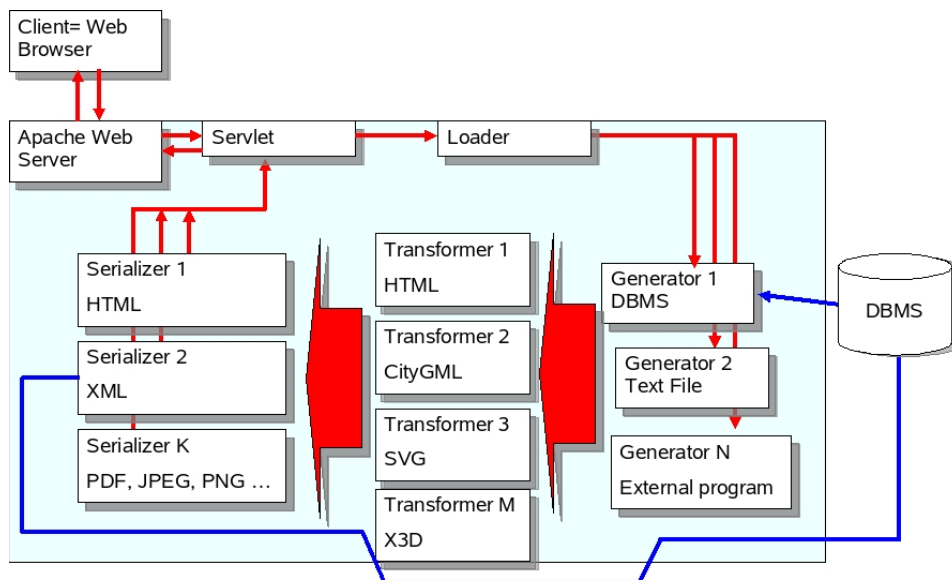


Abb.2: Transformationsparadigma von Cocoon.

Der Nachteil dieser Lösung besteht ganz klar darin, dass für die Definition neuer Prozesse Programmierkenntnisse *und* Prozesswissen notwendig sind. Obwohl Cocoon vollkommen XML-basiert konfiguriert werden kann, sind seine Konfigurationsdateien eher schwer lesbar. Damit wird Cocoon als Integrationsinstanz faktisch zu schwerfällig sein.

5 Prozessdefinitionen

Ein wesentlicher Fortschritt der Informatik in den letzten Jahren besteht in der Entwicklung einer Vielzahl von Beschreibungssprachen, welche die korrekte Annotation von Daten und damit die formale Beschreibung von Datenbeziehungen statischer und dynamischer Natur erlauben.

Für statische Relationen zwischen Daten ist UML schon seit längerem Stand der Technik, seine Erweiterung auf adaptive dynamische Softwaresysteme und Prozesse ist Gegenstand aktueller Forschung [13]. Für den Einsatz in einem - wie oben geschildert - „geschäftskritischen“ System von Prozessen ist UML deshalb nach Ansicht des Autors noch nicht geeignet.

Als Alternative können XML-Anwendungen dienen, die eine formal verifizierbare abstrakte Beschreibung von Prozessen ermöglichen. Unter den verschiedenen Dialekten, z.B. BPMN (Business Process Modeling Notation), WPD (Workflow Process Definition Language) und anderen hat sich XPDL, die XML Workflow Process Definition Language als bemerkenswert stabil und handhabbar herausgestellt [14]. Dieser Standard wird von einem international besetzten Industriekonsortium, der Workflow Modeling Coalition WfMC gepflegt und ist deshalb einigermaßen zukunftssicher [15].

Solche abstrakten Prozessbeschreibungssprachen sind klar abzugrenzen von Sprachen zur Ausführung von Prozessen, wie etwa BPEL (Business Process Execution Language). Letztere befasst sich auf einem sehr grundlegenden Niveau mit Prozessen auf Daten, ist also eigentlich einer prozeduralen Programmiersprache (mit objektorientierten Aspekten) gleichzusetzen. Dem entsprechend befinden sich BPEL-Laufzeitumgebungen und Compiler bereits auf einem vergleichsweise hohen Entwicklungsstand [16]. BPEL-Code kann aus den höherwertigen Sprachen wie BPMN relativ problemlos generiert werden – in umgekehrter Richtung ist dieses (mindestens derzeit) nicht möglich.

Die XML-Anwendung XPDL dient dabei der formalen Spezifikation der Geschäftsprozesse. Sie kennt Teilnehmer, Aktivitäten und Übergänge, die in vielfältiger Weise konditionell miteinander verknüpft werden können. Ein kleines Codefragment, welches aus einer Modellierung des Prüfungsanmeldungsprozesses stammt, ist in der folgenden Abbildung zu sehen.

```

<Package>
  <PackageHeader><XPDLVersion>1.0</XPDLVersion>
  ...
</PackageHeader>
<Participants>
  <Participant Id="prof" Name="Professor">
    <ParticipantType Type="ROLE"/>
  </Participant>
  <Participant Id="student" Name="Student">
    <ParticipantType Type="ROLE"/>
  </Participant>
  ...
</Participants>
<WorkflowProcesses>
  <WorkflowProcess Id="prf_anmeldg" Name="Pruefungsanmeldung">
    <ProcessHeader>...</ProcessHeader>
    <FormalParameters>...</FormalParameters>
    ...
    <Activities>
      <Activity Id="check_number" Name="Kapazität">
        <Implementation>
          <SubFlow Id="newpkg_wp1"/>
        </Implementation>
        <TransitionRestrictions>
          <TransitionRestriction>
            <Split Type="XOR">
              <TransitionRefs>
                <TransitionRef Id="tra3"/>
                <TransitionRef Id="tra2"/>
              </TransitionRefs>
            </Split>
          </TransitionRestriction>
        </TransitionRestrictions>
      </Activity>
      ...
    </Activities>
    <Transitions>
      <Transition From="check_number" Id="tra2" To="R1"/>
      <Transition From="check_number" Id="tra3" To="check_prereq"/>
      ...
    </Transitions>
  </WorkflowProcess>
</WorkflowProcesses>
</Package>

```

Abb. 3: Formale Prozessbeschreibung in XPDL, mit einer ausschließlichen Aufteilung des Prozesses (Type=XOR)

Dabei kann die unübersichtliche XML-Darstellung durch Verwendung eines grafischen Editors vermieden werden – denn in diesem wird der Geschäftsprozess in übersichtlicher Art dargestellt. Abbildung 4 zeigt das eher triviale Beispiel von Abbildung 3 – das als vollständiger XML-Code immerhin mehr als 100 Zeilen umfasst - in Form eines einfachen Diagramms.

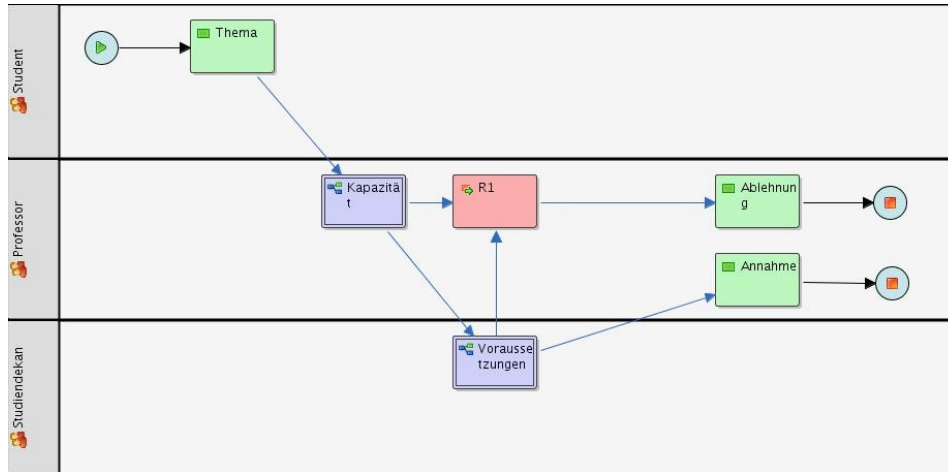


Abb4: Einfaches XPDL-Diagramm zu Abbildung 3.

Diese grafische Erzeugung von Prozessbeschreibungen aber ist genau der Abstraktionsgrad, der zur Lösung der Integrationsproblematik von einem bestimmten Software-Anbieter gewählt werden muss. Sie hat darüber hinaus noch einen wesentlichen Vorteil: Für XPDL existiert mit dem Projekt Enhydra eine hervorragende Implementation eines Workflow-Servers, der die abstrakt modellierten Prozesse in web-basierte Formulare zur Transaktion sowie Datenbankoperationen im Hintergrund umsetzt.

6 Enhydra Workflow Server

Enhydra ist ein relativ umfangreiches Open Source-Projekt, in dem verschiedene Komponenten zur Modellierung und Realisierung von Workflows entwickelt werden [17]. Zu diesen Komponenten gehören Enhydra JaWe als grafischer Editor für die eXtensible Process Definition Language XPDL, Enhydra Shark als zentraler Server für Prozesse sowie mit Enhydra Octopus ein Transformationsserver, der XML-spezifizierte Transformationen zwischen Datenbanken durchführt. Bereits mit Enhydra Shark lassen sich XPDL-Dateien in konkrete Abläufe umsetzen.

Enhydra Octopus erlaubt, die Aktivitäten als *Operationen auf Datenbanken* zu realisieren. Dabei kann es sich um gewöhnliche relationale Datenbanken handeln, um Excel-Dateien oder um objektrelationale Mappings in dem Enhydra-eigenen XML-Dialekt DOML. Diese werden von dem enthaltenen Werkzeug DODS (Data Object Design Studio) erzeugt und gewährleisten den objektorientierten Zugriff auf relationale Daten.

Eine typische Enhydra Octopus-Anwendung in dem hier interessierenden Kontext besteht darin, aus einer Datenquelle eine Gruppe von Studierenden zu extrahieren, deren Daten zu normalisieren und in eine Datensinke zu schreiben. Diese ETL-Sequenz (*Extraction-*

Transformation-Loading) erlaubt also die durch externe Prozessdefinitionen spezifizierte Kopplung zweier datenbankbasierter Systeme, ohne diese direkt zu modifizieren. Schematisch ist dies in Abbildung 5 dargestellt.

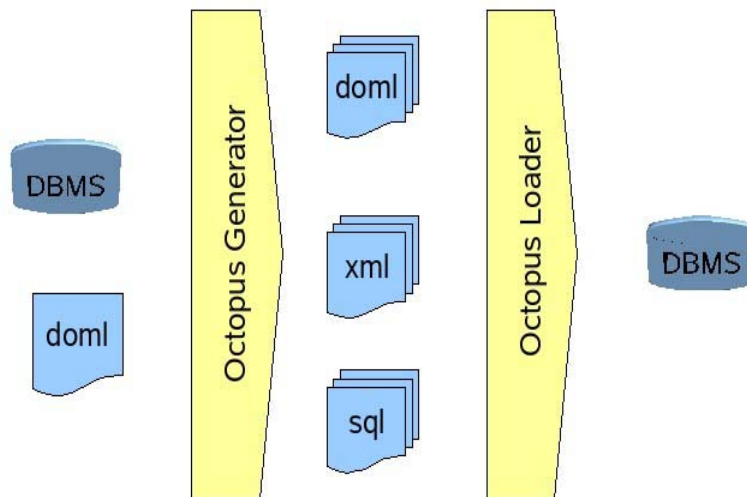


Abb. 5: Schema von Enhydra Octopus: Lesen aus Datenbank A, modifizieren an Hand abstrakter Prozessbeschreibung, Schreiben in zweite Datenbank B.

7 Zusammenfassung und Ausblick

Mit diesem Artikel wurde der Versuch unternommen, die Notwendigkeit einer Integration von IT-Dienstleistungen einer modernen Hochschule in die elektronische Lernumgebung von Studierenden zu begründen, sowie konkrete Integrationskonzepte auf der Basis von Open Source-Produkten vorzuschlagen. Die Diskussion von existierenden Integrationsprojekten – wie im mündlichen Vortrag erfolgt – wurde aus Gründen der Aktualität hier nicht dargestellt, stattdessen sei auf die inzwischen zahlreichen Ansätze großer Hochschulen verwiesen.

Bei den vorgestellten Integrationskonzepten wurde der Fokus auf die Beherrschung der komplexen Prozesse einer modernen Hochschule gesetzt, und für die Verwendung eines externen Prozess-Servers statt modifizierter Anwendungen argumentiert. Dieses Konzept ist sicherlich in hohem Maße gewöhnungsbedürftig, umgeht es doch die Anwendungen, die „normalerweise“ auf die notwendigen Datenbanken zugreifen. Deshalb sind natürlich vor der Implementation entsprechend sorgfältige Überlegungen erforderlich – und das Datenmodell der Anwendungen muss bekannt sein.

Doch treten die gleichen Risiken auch bei den komplexen Anwendungen „eLearning“ und „Hochschulmanagement“ selbst auf. Insofern scheint einerseits der Einsatz eines solchen Systems beherrschbar zu sein – und andererseits gibt es gerade im Hochschulsektor eine Vielzahl von Prozessen, bei denen eine schreibende Modifikation kritischer Datenbanken nicht notwendig ist, die aber andererseits auch mit den gängigen URP-Systemen nur schwerfällig realisierbar sind (z.B. die Erstellung eines Transcript of Records). Für einen solchen Prozess-Server, der sich unmittelbar in die Lernumgebung von Studierenden einfügen lässt, besteht also erheblicher Bedarf.

Schließlich sei ein konkretes Beispiel genannt, das zwar nicht mit den hier genannten Technologien realisiert wurde – aber dennoch dem gleichen Paradigma des Umgehens der Hauptanwendung folgt: Die so genannte ILIAS Management Konsole ist eine Software, die für ILIAS-Installationen zusätzliche Funktionalitäten in der Benutzerverwaltung etc. bietet [18]. Es bleibt abzuwarten, inwieweit dieses unkonventionelle Modell sich in der professionellen Software-Entwicklung durchsetzt.

Literaturverzeichnis

- [1] JIM – Jugend, Information, (Multi-)Media. Basisuntersuchung zum Medienumgang 12-19-Jähriger, Medienpädagogischer Forschungsverbund Südwest (2006-1998), <http://www.mpfs.de>
- [2] KIM – Kinder und Medien, Computer und Internet. Basisuntersuchung zum Medienumgang 6-13-Jähriger, Medienpädagogischer Forschungsverbund Südwest (2006-1998), <http://www.mpfs.de>
- [3] Henning, P. A., Lehr, A.: *Studentische Notebooks statt Poolräumen. Ein Beitrag zur Kostensenkung der Hochschulen?* In: v. Knop, J., Haverkamp, W., Jessen, E. (Hrsg.): *Heute schon das Morgen sehen? Proceedings der 19. DFN-Arbeitstagung über Kommunikationsnetze, GI Lecture Notes in Informatics, Nr. P-73 (2005) p.307 – 320*
- [4] Henning, P.A.: *I18N. Zur Internationalisierung der deutschen Hochschulen*. Liberales Institut der Friedrich Naumann-Stiftung (2007). <http://www.pro-kopf.de>
- [5] Henning, P.A.: *WUSKAR - Die Werkstatt Unternehmenssoftware Karlsruhe*. Vortrag auf der 20. DFN-Arbeitstagung über Kommunikationsnetze. Proceedings im Druck, Vorabdruck unter <http://www.medialab-karlsruhe.de/persons/phenning/start/startpap.html>
- [6] Interne Mitteilung der Fa. SAP im Rahmen des Projektes WUSKAR, 2006
- [7] Mitteilung der Fa. HIS Hochschulinformationssysteme GmbH, LEARNTEC 2007. Firmeninformation unter <http://www.his.de>
- [8] ILIAS-Lernumgebung der Hochschule Karlsruhe, <http://www.ilias-karlsruhe.de>
- [9] ILIAS-Projektseite <http://www.ilias.de>

- [10] Henning, J., Dewal, G., Quenzer, M.: *Die Open Source Lernplattform ILIAS. Handbuch für Studierende, Dozenten und Administratoren* (Verlag uni-edition, Oktober 2007)
- [11] Henning, P.A.: *Taschenbuch Multimedia, 4. Auflage* (Fachbuchverlag Leipzig, September 2007)
- [12] The Apache Cocoon Project, <http://cocoon.apache.org>
- [13] Ben Djemaa, R., Amous, I., Ben Hamadou, A.: *WA-UML: Towards a UML extension for modelling Adaptive Web Applications*, Proc. 8th IEEE International Symposium on Web Site Evolution (WSE'06) pp. 111-117
- [14] Bastida Merino, L., Benguria Elguezabal, G.: *Business Process Definition Languages Versus Traditional Methods Towards Interoperability*. In: *COTS-Based Software Systems*, Lecture Notes in Computer Science (Springer, Heidelberg 2005) 25-35
- [15] Workflow Modeling Coalition <http://www.wfmc.org/>
- [16] Hantschal, R., Ruf, F., Strobek, H.: *Vergleich von BPEL Laufzeitumgebungen* (Institut für Architektur von Anwendungssystemen, Universität Stuttgart 2006)
- [17] Enhydra Workflow Server, <http://www.enhydra.org>
- [18] ILIAS Management Console <http://www.ilias-mc.de/>