

# Das elatePortal-Projekt – Produktlinien- entwicklung im Bereich E-Learning und E-Testing

Thorsten Berger<sup>+</sup>, Thomas Riechert<sup>\*\*</sup>, Steffen Dienst<sup>++</sup>  
Universität Leipzig, Institut für Informatik  
Abteilung Betriebliche Informationssysteme  
Johannissgasse 26  
04103 Leipzig

## Zusammenfassung

Der Beitrag stellt die Entwicklung der elatePortal-Plattform vor. Das Projekt hat zwei Dimensionen, die jeweils einer funktionalen und einer nichtfunktionalen Herkunft zuzuordnen sind. Einerseits besitzt das elatePortal umfangreiche Funktionen zur Durchführung elektronischer Klausuren unter Examensbedingungen und befindet sich derzeit im Einsatz an drei Fakultäten der Universität Leipzig. Andererseits besitzt es einen Forschungscharakter zur Untersuchung der Softwareproduktlinien-Entwicklung im Bereich E-Learning und zur Entwicklung von Integrationsansätzen im Zusammenspiel verschiedener Hochschulinformationssysteme.

Der Beitrag geht speziell auf die Entwicklungsmethodik und die Architektur des elatePortals ein, beschreibt aktuelle Entwicklungen sowie zu lösende Probleme und gibt einen Ausblick über die nähere Zukunft des Projekts.

## 1 Einleitung

Mittlerweile steht eine erfreulich große Auswahl an E-Learning-Systemen und -Umgebungen im Open-Source-Bereich zur Verfügung. Während viele einen inzwischen beachtlichen Funktionsumfang zum Anstoßen nutzerorientierter Lernprozesse bieten, existieren weitere Anforderungsfelder, für die bisher nur unzureichende Lösungen existieren. So ist eine an Bedeutung gewinnende Anforderung der Einsatz computergestützter Prüfungen unter Examensbedingungen als Kriterium für die Vergabe von Leistungsnachweisen oder als Bestandteil akademischer Prüfungen. Obwohl auch einige Systeme im Open-Source-Bereich Funktionen für Tests bereitstellen, bieten sie oft nur eine unzureichende Unterstützung für vollständig integrierte Prüfungsabläufe. Eine Lösung für typische Prüfungen im universitären Umfeld sollte einen Ablauf unterstützen, der von Einschreibung, inklusive der Gewährleistung von Zulassungsvoraussetzungen, über

---

<sup>+</sup> mail@thorsten-berger.net

<sup>\*\*</sup> riechert@informatik.uni-leipzig.de

<sup>++</sup> steffen.dienst@gmx.de

Ausführung, semiautomatische Korrektur bis hin zu Einsichtnahme und Nachkorrektur führt.

### **1.1 Bedarf an serviceorientierten Architekturen im E-Learning-Bereich**

Ein weiteres Defizit bestehender Systeme stellt häufig deren eingeschränkte und nicht standardisierte Erweiterbarkeit dar. Oft ist eine Integration mit bestehenden und teilweise sehr heterogenen Hochschul-Informationssystemen notwendig, wofür Funktionen wie Single Sign On (SSO) und ein Identitätsmanagement bereitgestellt werden müssen. Da zentralisierte Architekturen, insbesondere im Hinblick auf den aktuellen Strukturwandel an Universitäten, keine ausreichende Flexibilität bieten, geht die Entwicklung derzeit in Richtung integrierter, dienstorientierter Systeme (iSOA) [Fr+06]. SOA bietet hier Möglichkeiten zur Fortsetzung des aus dem Software Engineering bekannten Prinzips *Separation of Concerns* in die Granularität ganzer Systeme.

Insbesondere stellt sich in der Praxis oft das Problem der Datenredundanz und -konsistenz wenn ein E-Learning-System zum Einsatz kommt und seine eigene Kursverwaltung mitbringt. Genau für diese Funktionalität werden in der Regel Systeme der HIS GmbH<sup>1</sup> benutzt, die in der Universitätsverwaltung verankert sind und wesentlich komplexere Modelle z.B. für die modularisierten Studiengänge Bachelor/Master besitzen. Aus diesem Grund besteht Bedarf an einer Trennung der Belange Kursverwaltung, Lernmanagement, Kommunikation und E-Testing.

### **1.2 elatePortal-Projekt**

Das Ziel des Open-Source-Projekts elatePortal<sup>2</sup> ist die Bereitstellung einer Plattform mit Fokus auf E-Testing sowie die Entwicklung von Lösungsansätzen für die genannte Integrationsproblematik. Dazu verfolgt das Projekt nicht die Neuentwicklung eines weiteren E-Learning-Systems, sondern besitzt durch Umsetzung eines Produktlinien-Ansatzes vielmehr einen Baukastencharakter.

Das elatePortal besteht derzeit aus zwei Systemen (siehe Abbildung 1): einem Portal zur Organisation und Verwaltung von Lehrveranstaltungen und einem eigenständigen Prüfungsserver. Das wichtigste Artefakt stellt das Aufgaben-Framework dar, welches die E-Testing-Funktionalität bereitstellt. Dies sind Übungsaufgaben/-klausuren im Portal sowie Prüfungsklausuren im Prüfungsserver.

---

<sup>1</sup> <http://www.his.de>

<sup>2</sup> <http://www.elateportal.de>

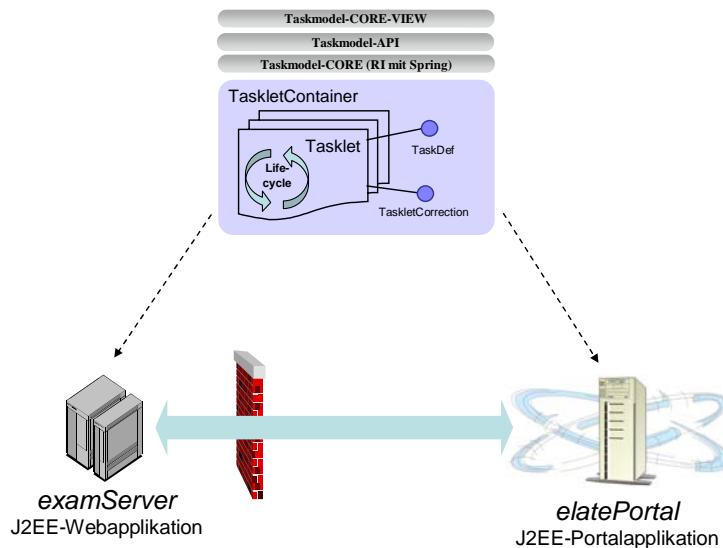


Abbildung 1: Überblick über die Systeme des elatePortal-Projekts

### 1.3 Weitere Arbeiten

Das Gesamtprojekt ist bereits in [BW06] vorgestellt worden. Der Fokus lag auf einer Erläuterung der Ausgangssituation für computergestützte Prüfungen an der Erziehungswissenschaftlichen Fakultät der Universität Leipzig, der Darstellung von grundlegenden Anforderungen an das System sowie einer Skizzierung der Basisarchitektur des elatePortals.

Einen Ansatz für eine stärkere Serviceorientierung universitärer Informations- und Verwaltungssysteme durch eine bessere Integration einrichtungsübergreifender Geschäftsprozesse stellen Freudenstein et al. [Fr+06] vor. Das Konzept orientiert sich daran, dass sich die Informationsversorgung im universitären Umfeld nach dem Prinzip der Nutzersteuerung richtet. Damit stellt es eine Alternative zu Zentralisierungsbestrebungen dar. Wie bereits erläutert erscheint ein derartiger Paradigmenwechsel auch zur Integration von E-Learning-Systemen sinnvoll.

## 2 Softwareproduktlinien-Entwicklung

Die Softwareproduktlinien-Entwicklung (SPLE) hat sich in den letzten zehn Jahren als erfolgreichster Ansatz für geplante Software-Wiederverwendung etabliert. Gegenüber der reaktiven Entwicklung von (Einzel-) Applikationen nach Bedarf verfolgt sie die proaktive Entwicklung einer flexiblen und erweiterbaren Basisplattform. Deren Bestandteile sind einzelne Plattform-Artefakte, aus denen wiederum konkrete Applikationen erstellt werden.

Die zentralen Konzepte der SPLE sind die Einführung von *Variabilität* in die Plattform-Artefakte sowie eine Trennung der beiden Prozesse *Domain* und *Application Engineering* [[WL99]. Abbildung 2 zeigt vier grundlegende Belange, mit denen sich die Produktlinien-Forschung beschäftigt. Die folgenden Betrachtungen des Beitrags konzentrieren sich auf die Architektur des elatePortals und dessen Entwicklungsprozess.

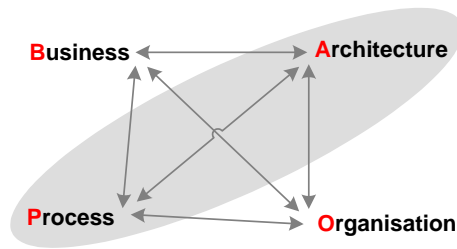


Abbildung 2: BAPO-Modell (nach [Li02])

Die Einführung der Produktlinienentwicklung in Unternehmen ermöglicht eine Verkürzung von Entwicklungszeiten, die Einsparung von Kosten sowie eine Verbesserung der Produkt-Qualität. Es existieren verschiedene Studien, die den Erfolg von Produktlinien in großen Organisationen mit kommerziellem Hintergrund belegen [CN01]. Während auch Erfahrungsberichte zur Einführung von Produktlinienentwicklung in kleinen und mittelgroßen Unternehmen verfügbar sind [Kn+00, VK05], ist mittlerweile noch keine Fallstudie bekannt, welche die Anwendbarkeit des Ansatzes im Rahmen von Open-Source-Projekten untersucht, die oft auf weiterer freier Software basiert und die teilweise ihren eigenen Gesetzen und Regeln folgt.

Ein derartiges Projekt stellt das elatePortal dar. Eine ausführliche Untersuchung des Domain Engineerings im Rahmen der Entwicklung des elatePortals ist in [Be07] durchgeführt worden. Dazu war es notwendig ein Produktlinien-Framework durch ein Tailoring auf die speziellen Bedingungen des elatePortal-Projekts anzupassen, woraus wiederum spezifische Problemstellungen der Produktlinien-Forschung in diesem Bereich identifiziert und Lösungsansätze entwickelt werden konnten.

## 2.1 Methodik

Abbildung 3 zeigt den Referenzprozess des verwendeten Produktlinien-Frameworks von Pohl, Böckle und Van der Linden [PBL05], in dem die beiden Hauptprozesse *Domain* und *Application Engineering* mit ihren jeweiligen Teilprozessen sowie den zu erstellenden Plattform-Artefakten dargestellt sind. Von Bedeutung, aber nicht in der Abbildung zu erkennen, sind außerdem Feedback-Schleifen zwischen den einzelnen Teilprozessen.

Das Framework selbst basiert auf den Ergebnissen der drei großen europäischen Forschungsprojekte ESAPS<sup>1</sup>, CAFÉ<sup>2</sup> und FAMILIES<sup>3</sup>. Es kam zum Einsatz, da es als eine der derzeit aktuellsten Methodiken für die SPLE bezeichnet werden kann und sehr gut dokumentiert ist.

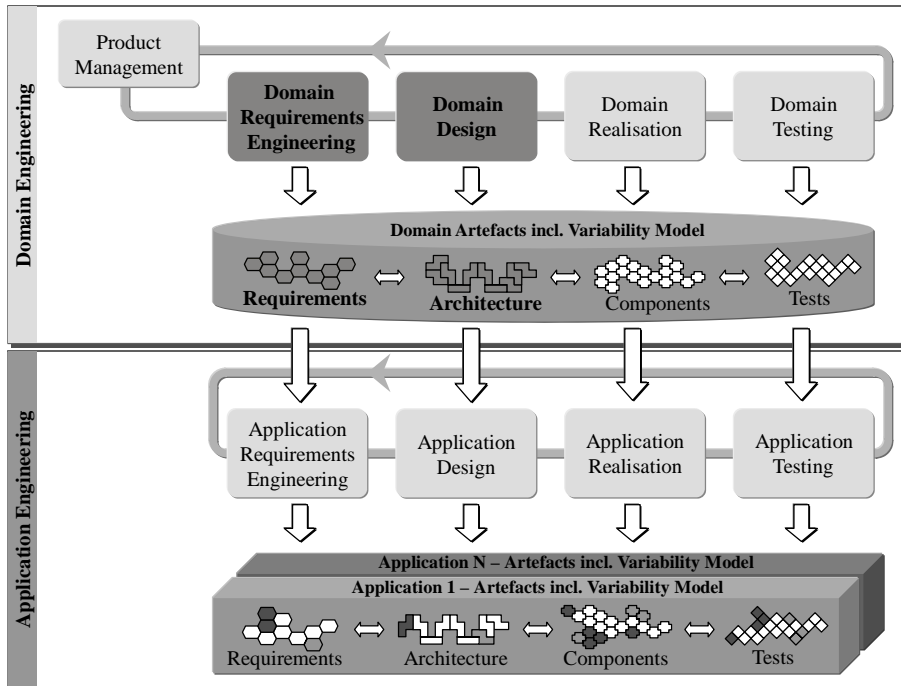


Abbildung 3: Referenzprozess des Produktlinien-Frameworks [PBL05]

Die folgenden beiden Kapitel skizzieren die Teilprozesse Domain Requirements Engineering und Domain Design anhand des elatePortal-Projekts.

## 2.2 Technologien

Das gesamte Projekt wurde überwiegend mit Frameworks und Technologien aus der J2EE-Plattform sowie deren Umfeld entwickelt. Von zentraler Bedeutung für das elatePortal war die Verwendung eines Portalservers in Form von Apache Jetspeed-2<sup>4</sup>, der zum Java-Portlet-Standard JSR168 [AH03] kompatibel ist.

Der Vorteil von Portalen liegt in der Bereitstellung einer grundlegenden Infrastruktur für das elatePortal sowie in der Erweiterbarkeit mit fremden, JSR168-kompatiblen Portlets. Letztere stellen in Portalen die eigentliche und auch beliebig umfangreiche Funktionalität dar. Sie werden wiederum zu sog. Portletapplikationen zusammengefasst. Außerdem stellen Portale eine einheitliche, personalisierbare Oberfläche, SSO-Funktionalität und Sicherheitsmodelle bereit.

<sup>1</sup> <http://www.esi.es/esaps>

<sup>2</sup> <http://www.esi.es/Cafe>

<sup>3</sup> <http://www.esi.es/Families>

<sup>4</sup> <http://portals.apache.org/jetspeed-2>

Das elatePortal, der Prüfungsserver und das Aufgaben-Framework verwenden als Komponenten-Technologie das Spring-Framework [JH04] als Alternative zum J2EE-Standard Enterprise Java Beans.

Darüber hinaus kamen bei der Entwicklung des elatePortal-Projekts u.a. auch aspektorientierte Programmierung (Spring-AOP) und Generative [CE00] bzw. Modellbasierte Ansätze (JAXB<sup>1</sup>, JAXFront<sup>2</sup>) zum Einsatz.

## 4 Domain Requirements Engineering

Das Requirements Engineering ist im Produktlinienkontext umfangreicher als für die Einzelsystementwicklung, da eine Vielzahl von Applikationen betrachtet werden muss. Eine der Hauptaktivitäten im Domain Requirements Engineering ist die Identifizierung gemeinsamer (*Commonality Analysis*) und variabler Anforderungen (*Variability Analysis*) an die Produktlinie. Letztere bezeichnen Anforderungen, die nur für einen Teil der geplanten Applikationen gültig sind und mit Variabilität versehen werden müssen. Dazu erfolgt die Definition von Variationspunkten und möglichen Varianten.

Neben der Erstellung von Requirements-Modellen muss die Variabilität der Produktlinie in einem Variabilitätsmodell modelliert werden. Das verwendete Produktlinien-Framework bietet dazu ein orthogonales Variabilitätsmodell [Ba+03], das die konsistente Definition von Variationspunkten und Varianten sowie die Einführung von Traceability-Links ermöglicht, ohne dabei vorhandene Metamodelle von Diagrammen erweitern zu müssen.

Für das elatePortal stellte die Verwendung von Goal- und Feature-Listen sowie die textuelle Beschreibung von Requirements eine solide Ausgangsbasis für die Commonality und Variability Analysis dar. Auch die aus agilen Methodiken bekannten Story Cards kamen zum Einsatz. Die anschließende Verfeinerung der Requirements-Artefakte erfolgte iterativ und mit Beachtung des Feedbacks aus dem nachfolgenden Teilprozess Domain Design.

Insbesondere mussten die ursprünglichen Anforderungen verfeinert und differenzierter strukturiert werden, da durch Architekturbetrachtungen im Domain Design die Entwicklung zweier Systeme, des elatePortals und des Prüfungsservers, fokussiert wurde. Diese Entscheidung ist maßgeblich von hohen Sicherheitsanforderungen bestimmt worden, die eine strikte Trennung zwischen sensiblen Prüfungsdaten und anderen ebenfalls sensiblen studentischen Daten aus dem Portal vorsahen.

---

<sup>1</sup> <http://java.sun.com/webservices/jaxb>

<sup>2</sup> <http://www.jaxfront.com>

Requirements	Erziehungswissenschaftliche Fak.	Abteilung Betriebliche Inf.-Systeme	Abteilung für Allgemeinmedizin
Unterstützung eines vollständigen Prüfungsablaufs	notwendig		notwendig
Vermeidung von Medienbrüchen	notwendig	notwendig	
Sicherung der Prüfungsklausuren gegen Betrug, Täuschung und Manipulation	notwendig		notwendig
sichere Identifikation der Teilnehmer	notwendig		notwendig
Aufgabentypen: <i>Datei hochladen, Multiple Choice, Zuordnung, Lückentext, Freitext, Graphische Darstellung</i>	notwendig		
Aufgabentypen: <i>Datei hochladen, Multiple Choice, Freitext, Remote</i>		notwendig	
Aufgabentypen: <i>Multiple Choice</i>			notwendig
Erweiterbarkeit mit versch. Aufgabentypen	notwendig	notwendig	
Trennung Prüfungsdaten von Einschreibungs- u. Studentendaten			
.....	.....	.....	.....
Benutzerverwaltung	notwendig	notwendig	notwendig
Authentifizierung der Benutzer gegen eigene Datenbank	notwendig	notwendig	notwendig
Authentifizierung der Benutzer gegen LDAP-Server	notwendig		notwendig
Authentifizierung der Benutzer gegen Shibboleth			
Eigenen Web-Content hinzufügen und verwalten (CMS)	notwendig		notwendig
Erweiterbarkeit mit vorhandenen Webapplikationen	notwendig	notwendig	notwendig

Tabelle 1: Application Requirements-Matrix

Zur Durchführung der Commonality bzw. Variability Analysis favorisiert das Produktlinien-Framework folgende drei Methoden: *Application Requirements-Matrix*, *Priority-Based Analysis* und *Checklist-Based Analysis* [PBL05]. Die einfacheren Ansprüchen genügende und auch für das elatePortal verwendete Methode ist die Application Requirements-Matrix, welche auszugsweise in Tabelle 1 dargestellt ist. Sie enthält grobe funktionale Anforderungen an die drei geplanten Applikationen. Die Matrix wird analog zu den anderen Requirements-Artefakten iterativ verfeinert und dient als Grundlage für die Einführung von Variationspunkten im orthogonalen Variabilitätsmodell. Für das elatePortal sind dabei auch viele Variationspunkte definiert worden, die möglicherweise erst in Zukunft von Relevanz sein könnten. Derartige Aspekte spielten eine wichtige Rolle bei der Entwicklung der Plattform.

## 5 Domain Design

Der Teilprozess Domain Design nutzt die im vorhergehenden Domain Requirements Engineering erarbeiteten Requirements-Artefakte zur Entwicklung einer Referenzarchitektur für die Produktlinie, indem Anforderungen mit technischen Lösungen verknüpft werden. Da die Referenzarchitektur die Umsetzbarkeit aller

späteren Applikationen aus der Produktlinie bestimmt, sind die an sie gestellten Anforderungen hoch. Den größten Einfluss auf die Architektur haben die Qualitätsanforderungen Erweiterbarkeit, Entwicklungsfähigkeit und Wartbarkeit, neben den bereits im Requirements Engineering erarbeiteten. Ihre Umsetzung stellt einen zentralen Erfolgsfaktor für die Produktlinie dar.

## 5.1 Referenzarchitektur

Zur Verdeutlichung der groben Architektur der Systeme im elatePortal-Projekt ist in Abbildung 4 eine physische Sicht in Form eines UML-Verteilungs-Diagramms dargestellt. Zu erkennen sind zwei Serversysteme, die jeweils den J2EE Servlet Container Apache Tomcat<sup>1</sup> enthalten.

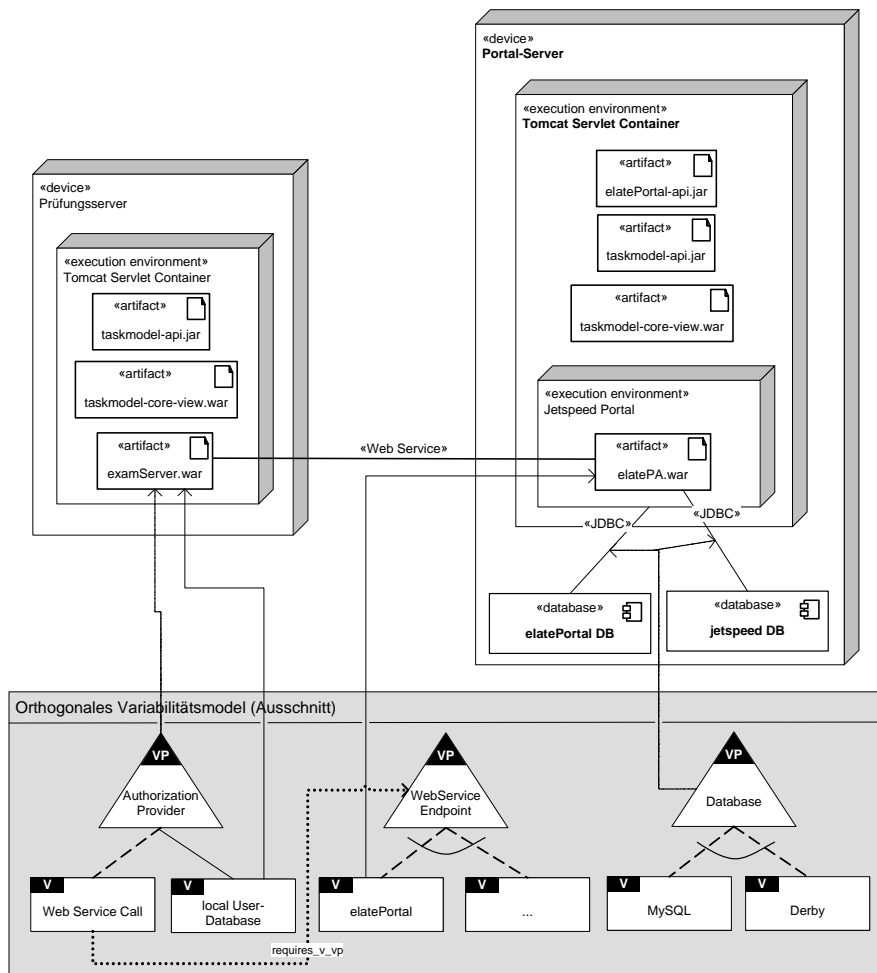


Abbildung 4: Verteilungs-Diagramm

<sup>1</sup> <http://tomcat.apache.org>

Dabei benötigt der Jetspeed2-Portalserver keinen vollständigen Application Server, was insbesondere an der Verwendung des leichtgewichtigen Komponenten-Frameworks Spring statt Enterprise Java Beans liegt. Unter dem Verteilungsdiagramm ist das orthogonale Variabilitätsmodell mit einer grobgranularen Definition von Variationspunkten und deren Traceability-Links enthalten.

## 5.2 Aufgaben-Framework

Das Aufgaben-Framework stellt eines der zentralen Artefakte im elatePortal-Projekt dar und ist zur Umsetzung der Funktionalität für elektronische Klausuren entwickelt worden.

Das Framework selbst ist wiederum durch verschiedene Variationspunkte erweiterbar und besteht aus folgenden drei Teilen:

- einer *API*, welche die Struktur des Frameworks beschreibt und den Zugriff auf die Implementierung kapselt,
- einer *Referenzimplementierung (RI)*, die alle Interfaces implementiert und, je nach Anwendung, die benötigten Aufgabentypen unterstützt, sowie
- einer *Webapplikation als Präsentationsschicht*, welche die Interaktion mit den Nutzern durchführt.

Im Wesentlichen stellt das Aufgabenframework eine Container-Lösung dar, in welcher ein Container den Lifecycle von sog. *Tasklets* steuert, die in Ausführung befindliche Aufgaben bezeichnen. Jedes *Tasklet* referenziert wiederum eine sog. *TaskDef*, die seine Aufgabendefinition enthält.

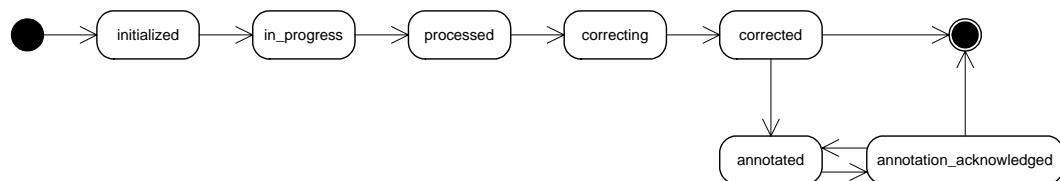


Abbildung 5: Aufgaben-Lebenszyklus

Von besonderer Bedeutung im Framework ist der für Prüfungsklausuren relevante Typ *Complex*, der wiederum aus verschiedenen Teilaufgaben, wie Multiple-Choice-, Zuordnungs-, Lückentext-, Freitext- oder Graphik-Aufgaben besteht. Für ihn ist der in Abbildung 5 skizzierte Lebenszyklus definiert.

Damit die Referenzimplementierung flexibel in das elatePortal, den Prüfungs-server oder in Fremdsysteme integriert werden kann, wurden Persistenzaufgaben in Komponenten gekapselt, die spezifisch für das Host-System implementiert werden müssen.

## 5.3 elatePortal

Bei der Entwicklung der Referenzarchitektur des elatePortals spielten zwei Gesichtspunkte eine wichtige Rolle. Zum einen mussten die erläuterten Qualitäts-Anforderungen Erweiterbarkeit, Entwicklungsfähigkeit und Wartbarkeit in angemessener Weise berücksichtigt werden und zum anderen hatte die Verwen-

derung von Portaltechnologie in Form des Portalservers Jetspeed-2 einen signifikanten Einfluss auf das Design.

In Abbildung 6 ist die grobe Schichtenarchitektur des elatePortals in Form eines UML-Paketdiagramms dargestellt. Darin ist auch zu erkennen, in welche Ebenen die einzelnen Bestandteile des Aufgaben-Frameworks in das Portal integriert worden sind. Unter dem Paketdiagramm befindet sich die Abbildung eines n-Tier-Modells [Ar05], bezüglich dessen die dargestellten Schichten den Middle-Tier darstellen.

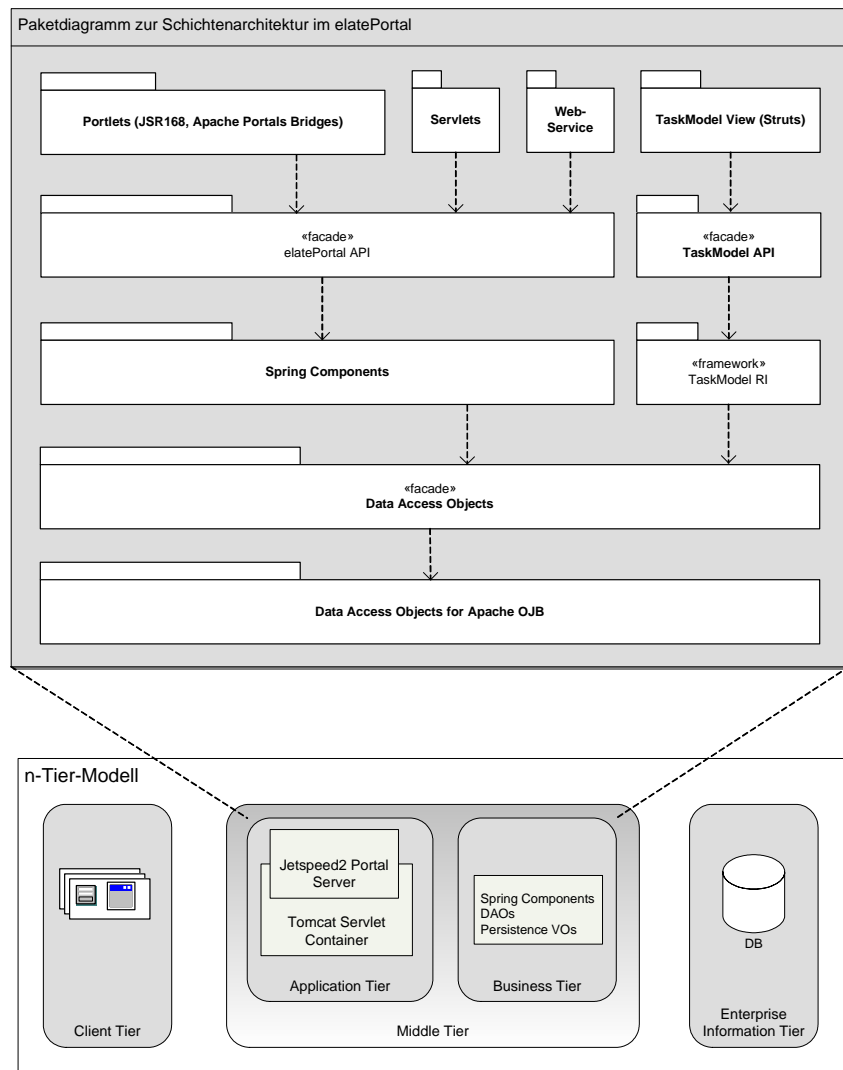


Abbildung 6: Grobe Architektur des elatePortals

Die in Abbildung 6 dargestellte oberste Schicht des elatePortals dient in erster Linie der Realisierung der Benutzeroberfläche, aber auch der Bereitstellung von Web-Services zur Anbindung anderer Systeme, wie des Prüfungsservers.

In dieser Schicht spielt der Einfluss von Portaltechnologie die größte Rolle. Sie unterscheidet sich teilweise maßgeblich von der Präsentationsschicht normaler Webapplikationen. Das lässt sich auf die spezielle Art des Deployments, den genauer definierten Lebenszyklus, die unterschiedliche Anfragebehandlung sowie die Einbettung von Portlets in Portal-Seiten zurückführen [AH03]. Portlets erzwingen durch die Spezifikation von separaten Action- und Render-Requests die Einhaltung des MVC-Patterns, wofür bei Servlets bekanntermaßen Web-MVC-Frameworks dienen. Viele Portlets wurden außerdem, je nach Komplexität, mit Hilfe eines weitergehenden Web-Frameworks konzipiert. Zur Anwendung kam dafür die Struts Bridge aus dem Apache Portals Bridges-Projekt<sup>1</sup>, mit der Portlets nahezu wie normale Struts-Webapplikationen<sup>2</sup> entwickelt werden konnten.

Servlets wurden in der Präsentationsschicht für Funktionen benötigt, die Portlets aufgrund ihres Konzeptes nicht ausführen können. Dazu zählen beispielsweise Download-Funktionen oder die Unterstützung von AJAX-Funktionalität [Ga05].

#### 5.4 Prüfungsserver

Der Prüfungsserver wurde in Form einer Webapplikation konzipiert mit dem Ziel eines einfachen Deployments in einen Servlet Container. Für die Erstellung wurde der so genannte Kickstarter AppFuse<sup>3</sup> verwendet, der eine Integrationslösung zur agilen Entwicklung von Webapplikationen bereitstellt. Darin sind bereits verschiedene Grundfunktionen, wie z.B. ein Sicherheitsmodell mit Benutzerverwaltung enthalten. Darüber hinaus war AppFuse durch die Verwendung von Spring sehr gut an die Anforderungen des Prüfungsservers anpassbar.

Die Integration des Aufgaben-Frameworks in den Prüfungsserver war durch dessen Design mit sehr geringem Aufwand möglich. Die Hauptaufgabe bestand in einer spezifischen Implementierung der Persistenz-Schicht.

## 6 Erweiterungen des Aufgaben-Frameworks

Da das Aufgaben-Framework die zentrale funktionale Komponente des elatePortals darstellt, wird an dieser Stelle genauer auf die aktuellen Entwicklungen eingegangen. In seiner Grundfunktionalität stellt es fünf Aufgabentypen zur Verfügung (vgl. Kap. 5.2) [BW06]. Während sich das Framework seit über zwei Jahren in Prüfungen unter Examensbedingungen bewährt hat, sind für erweiterte Einsatzszenarien umfangreichere Aufgabentypen und differenziertere Lebenszyklen notwendig. Dazu gehören z.B. Aufgaben aus der theoretischen oder praktischen Informatik, deren Auswertung weitaus aufwendiger ist. Bezüglich ersteren wird eine Anbindung an das System Autotool [WR02] und bzgl. zweiten die Verwendung von Unit-Test-Frameworks zur dynamischen Verifizierung von Programmieraufgaben angestrebt.

---

<sup>1</sup> <http://portals.apache.org/bridges>

<sup>2</sup> <http://struts.apache.org>

<sup>3</sup> <http://www.appfuse.net>

Die Basis der weiteren Arbeiten am Aufgaben-Framework ist eine Weiterführung des Produktlinien-Ansatzes unter Evaluierung und voraussichtlicher Anwendung des *OSGi-Frameworks* [OS03], das ein dynamisches Modulsystem in Form einer serviceorientierten Architektur bereitstellt. OSGi versteht sich selbst als „universal middleware“. Es stellt die Spezifikation für eine Plattform bereit, welche nach dem Prinzip der SOA das dynamische Zusammensetzen einer Applikation aus einzelnen Bundles ermöglicht. Der Standard umfasst den Aufbau der Bundles, ihren Lifecycle, Versionierung, Standardservices, Sicherheitsaspekte und das Management der Anwendungen. Damit ist es möglich, einzelne Dienste unabhängig voneinander zu entwickeln und auf einer großen Bandbreite von Plattformen von eingebetteten Systemen bis hin zu Serverfarmen einzusetzen, ohne Änderungen an der Codebasis ausführen zu müssen. In der Nomenklatur von Softwareproduktlinien bedeutet dies, dass die Bindung von Variationspunkten durch das Deployment sog. Service-Bundles erfolgt. Durch diesen Ansatz wird ein flexibleres Hinzufügen neuer Aufgabentypen verfolgt.

Die Verwendung von OSGi als dynamisches Komponentenframework zusammen mit Spring wird in Zukunft auch einige Veränderungen an der elatePortal-Basisplattform verursachen, welches dann z.B. zu einer neuen Qualität des Hot-Deployments führen wird. Auch sollte sich so die Integrierbarkeit von elatePortal-Artefakten in andere Plattformen unkomplizierter gestalten.

Weitere Untersuchungen sind jedoch noch bezüglich der Integration völlig neuer Aufgaben-Workflows in das Framework notwendig. Das muss momentan noch in der eigentlichen Codebasis erfolgen.

## 7 Zusammenfassung und Ausblick

Im elatePortal-Projekt konnte erfolgreich eine flexible Plattform zur Unterstützung von Prüfungsprozessen realisiert werden, deren Artefakte einen Gesamtumfang von derzeit ca. 65 kSLOC aufweisen. Knapp die Hälfte entfällt auf generative Ansätze.

Das verwendete Produktlinien-Framework stellte einen guten formalen Rahmen bei der Entwicklung dar. Dabei ist anzumerken, dass der initiale Aufwand zur Entwicklung und Etablierung der Plattform durch die verwendeten Technologien weitaus geringer ausfiel als zunächst vermutet. Die im Produktlinien-Framework definierte Vorgehensweise kann als adaptiv bezeichnet werden, die sich gut an unterschiedliche Rahmenbedingungen anpassen lässt. Als Kritikpunkt stellte sich die Praxistauglichkeit des orthogonalen Variabilitätsmodells heraus, für das derzeit geeignete Tools fehlen. Jedoch bildete es einen sehr konsistenten theoretischen Rahmen zur Modellierung von Variabilitätsaspekten.

Bei der Entwicklung des elatePortal-Projekts kam überwiegend Open-Source-Software zum Einsatz. Dabei erwies sich insbesondere die Verwendung von J2EE-Portaltechnologie als größter Enabler der Produktlinien-Entwicklung. Sie gewährleistete eine sehr entwicklungsfähige Architektur und ermöglichte eine umfangreiche Erweiterbarkeit des Portals durch das standardisierte Deployment weiterer Portlet-Applikationen. Außerdem setzen Portlets den Komponentenge-

danken bis an die Präsentationsschicht fort und ermöglichen dadurch eine einfache Umsetzung von Variabilität in der Oberfläche.

Die Ziele der weiteren Entwicklung des elatePortal-Projekts sind eine umfangreichere Unterstützung von Prüfungsprozessen durch weitere Aufgabentypen (vgl. Kap. 6) und eine bessere Integration in bestehende (Hochschul-) Informationssysteme. Außerdem ist eine Erweiterung des Portals zur umfangreicheren Verwaltung von Lehrveranstaltungen mit Anbindung an Web-Services der HIS-Systeme geplant. Derzeit wird auch über weitergehende Sicherungsmaßnahmen der elektronischen Klausuren nachgedacht, die eine spätere Manipulation (seitens der Betreiber) von Eingabedaten durch kryptographische Prüfsummen einschränken. Eine noch höhere (Manipulations-) Sicherheit wäre nur durch die Verwendung asymmetrischer Verschlüsselung möglich, wobei der private Schlüssel lediglich dem Studierenden bekannt sein darf (bspw. durch Nutzung des Krypto-Prozessors auf Studentenausweisen). Das ist zwar derzeit technisch noch zu aufwendig, wird aber in Zukunft bei stärkerem Einsatz elektronischer Prüfungen notwendig sein.

## Literaturverzeichnis

- [AH03] Abdelnur, A.; Hepper, S.: Java Portlet Specification. <http://www.jcp.org/en/jsr/detail?id=168> (12.02.2007). 10 2003
- [Ar05] Armstrong, E.; Ball, J.; Bodoff, S.; Carson, D.B.; Evans, I.; Green, D.; Haase, K.; Jendrock, E.: *The J2EE™ 1.4 Tutorial*. Santa Clara, CA, USA: Sun Microsystems, 12 2005
- [Ba+03] Bachmann, F.; Goedicke, M.; Leite, J.; Pohl, K.; Ramesh, B.; Vilbig, A.: Managing Variability in Product Family Development, In: *Proceedings of the 5th International Workshop on Product Family Engineering (PFE-5)*. Siena, Italy: Springer-Verlag, 11 2003
- [Be07] Berger, T.: *Softwareproduktlinien-Entwicklung – Domain Engineering: Konzepte, Probleme und Lösungsansätze*. Betrachtung im Rahmen einer Fallstudie über die Entwicklung eines Portals und eines Frameworks zur Unterstützung elektronischer Prüfungsabläufe. Diplomarbeit. Institut für Informatik, Universität Leipzig, 04 2007
- [BW06] Berger, T.; Wollersheim, H.-W.: Eine dienste- und komponentenbasierte Architektur zur elektronischen Durchführung von Prüfungen und zum Management von Lehrveranstaltungen. In: Hochberger, C.; Liskowsky, R. (Hrsg.): *Beiträge der 36. Jahrestagung der Gesellschaft für Informatik e.V (INFORMATIK 2006)*. Dresden, Deutschland: GI-Edition - Lecture Notes in Informatics (LNI), P-93, 10 2006
- [CE00] Czarnecki, K.; Eisenecker, U.W.: *Generative Programming. Methods, Tools and Applications*. Addison-Wesley, 06 2000

- [CN01] Clements, P.; Northrop, L.M.: *Software Product Lines: Practices and Patterns*. Addison Wesley, 08 2001
- [Fr+06] Freudenstein, P.; Liu, L.; Majer, A.; Maurer, A.; Momm, C.; Ried, D.; Juling, W.: Architektur für ein universitätsweit integriertes Informations- und Dienstmanagement. In: Hochberger, C.; Liskowsky, R. (Hrsg.): *Beiträge der 36. Jahrestagung der Gesellschaft für Informatik e.V (INFORMATIK 2006)*. Dresden, Deutschland: GI-Edition - Lecture Notes in Informatics (LNI), P-93, 10 2006
- [Ga05] Garrett, J.J.: Ajax: A New Approach to Web Applications. <http://www.adaptivepath.com/publications/essays/archives/000385.php> (23.03.2007). 02 2005
- [JH04] Johnson, R.; Höller, J.: *J2EE Development Without EJB*. Wiley & Sons, 07 2004
- [Kn+00] Knauber, P.; Muthig, D.; Schmid, K.; Widen, T.: Applying Product Line Concepts in Small and Medium-Sized Companies. In: *IEEE Software* 17 (2000), Nr. 5, Seiten 88-95
- [Li02] Van der Linden, F.: Software Product Families in Europe: The Esaps & Café Projects. In: *IEEE Software* 19 (2002), Nr. 4, Seiten 41-49
- [OS03] OSGi Alliance: *OSGi Service Platform*. IOS Press, 12 2003
- [PBL05] Pohl, K.; Böckle, G.; Van der Linden, F.: *Software Product Line Engineering: Foundations, Principles and Techniques*. Springer-Verlag. 09 2005
- [VK05] Verlage, M.; Kiesgen, T.: Five years of product line engineering in a small company. In: *Proceedings of the 27th international conference on Software engineering (ICSE '05)*. St. Louis, MO, USA, 2005, Seiten 534-543
- [WL99] Weiss, D.M.; Lai, C.T.R.: *Software Product-Line Engineering – A Family-Based Software Development Process*. Reading, Massachusetts, USA: Addison-Wesley, 1999
- [WR02] Waldmann, J.; Rahn, M.: The Leipzig autotool System for Grading Student Homework. In: Hanus, M.; Krishnamurthi, S.; Thompson, S. (Hrsg.): *Proceedings Functional and Declarative Programming in Education (FDPE-02)*. Kiel, 2002