Lastenheft

1. Zielbestimmung

Das OLAT-PA-Portal ist eine zur einfachen Prüfungsorganisation modifizierte OLAT-Installation. Seit dem Sommersemester 2009 steht die im SWT-Praktikum 2007 entwickelte Prüfungsmanagement Lösung in der weiterentwickelten Version des PA-Moduls als eigenständige Instanz, wie es vom Prüfungsausschuss gefordert wurde, an der Fakultät für Mathematik und Informatik zur Verfügung. Nach zweijährigen Betrieb der auf OLAT 6.1 aufsetzenden Installation haben sich neue Anforderungen herausgebildet, und auch die Architektur von OLAT, mittlerweile in Version 7, hat sich mit dem klaren und architekturkonformen Einsatz des Spring-Frameworks und neuen Build- und Verwaltungswerkzeugen grundlegend verändert. Damit diese modernen Entwicklungen auch beim PA-Modul genutzt werden können, ist ein umfassendes Upgrade auf die neue OLAT Version 7.2 erforderlich. Hierbei müssen die Erweiterungsansätze des PA-Moduls architekturkonform angepasst und bisher nicht konform realisierte Lösungen an die neu vorhandenen Erweiterungspunkte angedockt werden.
Des weiteren soll die Liste von Anforderungen, welche sich als Defizit im laufenden Betrieb ergaben, abgearbeitet werden.

2. Produkteinsatz

Das OLAT-PA-Modul, auf dem Entwicklungsstand von Olat.7, soll an der Universität Leipzig Anwendung finden. Es ist für das Institut für Mathematik und Informatik gedacht. Die Zielgruppen sind zu einem die Studenten, die sich im Internet mit Hilfe von diesem Portal an den Prüfungen zum Abschließen ihrer Module anmelden können und zum Anderen die für die Prüfungsorganisation verantwortlichen Dozenten, welche Informationen über einzelne Prüfungen herausgeben und auch das Prüfungsamt, das Daten über die Prüfungen verwaltet.

3. Produktübersicht

Das vorhandene Produkt, OLAT-PA, ist stark an OLAT 6.1 gekoppelt, wodurch die bereitgestellten Mittel besser ausgeschöpft werden können, was jedoch auch den Nachteil hat, dass das Modul abhängig von dieser Version ist und nicht ohne weiteres auf die neue Version portiert werden kann. Im folgendem soll ein kurzer Gesamtüberblick über das bestehende System gegeben und danach die einzelnen Pakete genauer betrachtet werden. Dabei wird die bestehende Anbindung an OLAT ausfindig gemacht und der erforderliche Aufwand der Änderungen abgeschätzt.

3.1. Gesamtüberblick

Das PA-Modul besteht aus zwei großen Teilsystemen, der Prüfung und der Elektronischen Studentenakte(ESF). Die Komponente Prüfung setzt sich wiederum aus kleineren Paketen, wie den Terminen, den Modulen und natürlich den Examen selbst zusammen. Die Komponente ESF besteht aus den Unterkomponenten persönliche Informationen, Prüfungsinformationen, Kommentare und Krankenschein. Die beiden Komponenten sind so entworfen, dass sie grundlegend voneinander unabhängig sind, jedoch nur zusammen sinnvoll eingesetzt werden können. Beispielsweise könnte man ohne die Prüfungskomponente zwar Kommentare über Studenten erfassen jedoch keine Prüfungsergebnisse verwalten. Die Anbindung an OLAT ist jedoch bei beiden Komponenten ähnlich und wird im folgendem für jedes einzelne Paket dargestellt.

3.2. Pakete

- de.xman.admin

Das Paket admin dient zur Erzeugung der Seite inklusive Navigation für Admins/ExamAdmins (siehe Funktionen) zur Verwaltung von Studentenakten, Prüfungen und Studiengängen. Enthalten sind Controller, welche die administrativen Arbeitsabläufe steuern und neue Controller aus den Paketen esf, module, studypath und exam zur Bearbeitung erzeugen. Verwendet wurden hierfür hauptsächlich Klassen aus den OLAT Paketen core.gui und core.util, welche sich gar nicht oder nur gering in Version 7.2 verändert haben. Veränderungen in den OLAT Klassen selbst konnten nicht ausfindig gemacht werden, weshalb von einer einfachen Portierung auf die neue Version ausgegangen werden kann. Lediglich die Rolle ExamAdmin, welche den Zugriff auf die ExamAdmin Seite erlaubt wurde direkt im OLAT Code hinzugefügt, weshalb deren Neuimplementierung aufwändiger sein wird.

- de.xman.esf

Das Paket esf stellt, wie der Name vermuten lässt, sowohl die Elektronische Studentenakte als Datenobjekt und GUI Element, als auch die Controller, welche die Funktionen zum Bearbeiten und Erstellen der ESFs anbieten, sowie einen Manager für den Datenbankzugriff zur Verfügung. Verwendet werden hierbei vornehmlich Klassen aus den OLAT Paketen core.id, core.persistence, core.ressource und core.gui. Der Änderungsbedarf ist hier wohl etwas größer, da auch Änderungen direkt in OLAT vorgenommen wurden. So wurde im HomeMainController ein neuer Eintrag zum Anfordern der ESF Seite erstellt. Dieser direkte Eingriff in OLAT könnte wohl durch Nutzung einer ActionExtension verhindert werden. Des weiterem haben sich auch einige verwendete Klassen in der neuen Version aufgrund der Anpassung an das Spring Framework etwas verändert, wodurch einige Änderungen in den ESF Klassen von Nöten sind.

- de.xman.illness

Das Paket illness enthält das Datenobjekt Krankenschein, einen Container zum Kapseln mehrerer Krankenscheine, die Tabelle zur Darstellung und eine Manager-Klasse zur Ein- und Ausgabe in die Datenbank. Verwendet werden Klassen des Java API und Klassen aus den OLAT Paketen core.persistence, core.logging, core.id und core.gui. Der Änderungsaufwand geht hierbei gegen Null, da alle Klassen ohne größere Veränderungen auch noch in der neuen Version von OLAT vorhanden sind. Somit kann das Paket

einfach übernommen werden.

- package de.xman.home

Innerhalb dieses Paketes existieren zwei Klassen, HomeMainControllerExtension und MyValidationExtension, welche die Extensions myExams und myValidation im HomeMainController hinzufügen.
Beim Aufruf des Konstruktors wird eine Instanz von Extensionelements erstellt und dieser durch das jeweilige HomeMainControllerExtension- oder MyValidationExtension Objekt spezifizierte ExtensionElement hinzugefügt.
Der Aufruf der Funktion homeMainControllerEntry erstellt jeweils eine Actionextension und gibt diese zurück.
Die benötigten von OLAT importierten Klassen wurden keinen Änderungen unterzogen, somit sollte der Implementationsaufwand als moderat eingestuft werden.

- package de.xman.schedule

Dieses Paket beinhaltet nur die Klasse ESFMailJob, welche sich zur Zeit nicht in Nutzung befindet und dazu diente, alle 24h an alle User mit der Rolle >>exam manager<< eine Email zu verschicken, die die Anzahl der Studentenakten mit dem Status "non-validated" enthält. Da sämtliche funktionalen Inhalte auskommentiert wurden, kann der zu erwartende Aufwand zur Implementation als äußerst gering angesehen werden.

- de.xman.studyPath

Sämtliche Klassen innerhalb dieses Paketes dienen dem Erstellen und Verwalten von Studiengängen, sowie deren Verwaltung in der Datenbank.
Das Interface Studypath ist eine Erweiterung von ModifiedInfo, CreateInfo, Persistable und OLATResourceable, welche allesamt aus dem OLAT-code entstammen.

Beim initialen Aufruf der Funktion getinstance() innerhalb der Managerklasse wird ein studypath angelegt, mit einem I18nKey versehen und anschließend gespeichert. Zur Zeit werden initial auch sämtliche in TEMPcreateAllStudyPaths() gelisteten Studiengänge erzeugt, dies soll jedoch laut Kommentar entfernt werden.
Die Klassen in den Unterordnern form und table dienen der internen Verwaltung von durch den Manager vorgenommen Veränderungen bzw. dem Erstellen einer grafischen Oberfläche zur leichteren Verwaltung.

Für die Bereitstellung der Funktionalitäten der einzelnen Klassen mussten keine Änderungen innerhalb des OLAT-Codes vorgenommen werden. Somit ist der zu erwartende Aufwand als moderat einzuschätzen.

- de.xman.exam und de.xman.appointment

Eine Prüfung muss als Lernressource interpretiert werden. Auch in OLAT Version 7 steht diesbezüglich kein Erweiterungspunkt zur Verfügung, so dass es nicht möglich ist, den erforderlichen Code von OLAT selbst zu trennen.
Betroffen sind auf oberer Ebene die Klassen RepositoryMainController, welche unter anderem der Erzeugung und Löschung von Lernressourcen dient, und RepositoryHandlerFactory, welche der Erzeugung der RepositoryHandler dient, die die zu einer bestimmten Lernressource gehörenden Controller zum

Erstellen, Einfügen, Bearbeiten und Löschen bereitstellt.
Die Architektur dieser Klassen hat sich in OLAT Version 7 zur Version 6.1 nicht dergestalt verändert, die eine umfassende Anpassung nötig machte.
Der Migrationsaufwand für dieses Paket ist als moderat einzuschätzen.
Appointment verwaltet die Termine zu einer Prüfung. Appointment wird von ExamHandler benutzt.

- de.xman.nodes

Dieses Paket stellt eine Prüfung als Kursbaustein dar. Dabei werden in ExanCourseNodeConfiguration die Intefaces OlatExtension sowie CourseNodeConfiguration und in ExamCourseNode die abstrakte Klasse AbstractAccessableCourseNode(welche über die abstrakte Klasse GenericCourseNodes das Interface CourseNodes realisiert) implementiert. Diese Implementierung kann weitesgehend wieder übernommen, einzige die Interface-Klasse OLATExtension ist in Olat7 nicht mehr vorhanden, daher muss deren Funktionalität in ExamCourseNodeConfiguration realisiert werden. Die restliche Funktionalität kann allerdings übernommen werden da ansonsten die Architektur der Interface-, Import- und Oberklassen identisch bleibt. Der Gesamtaufwand zur Migration des Gesamtpakets ist daher als moderat einzuschätzen.

- de.xman.comment

Dieses Paket ist Teil der ESA und stellt deren Kommentarfunktion zur Verfügung. Die Verwaltung der einzelnen Einträge erfolgt in CommentEntry und CommentImpl(implementiert das Interface Comment). Die Interaktion mit der Datenbank wird dabei in CommentManager realisiert. CommentTableModel modelliert das Erscheinungsbild der Kommentarfunktion. Da die Architektur der verwendeten Importe, Interfaces und Oberklassen nahezu identisch geblieben ist kann die Funktionalität weitesgehend übernommen werden. Der Migrationsaufwand des Pakets ist daher bestenfalls als moderat einzuschätzen

- de.xman.upgrade

Die Klasse dieses Paketes wurde dazu genutzt um beim Upgrade von OLAT 5 auf OLAT 6 entsprechende Änderungen an der Table-darstellung innerhalb der Datenbank vorzunehmen. Da dieses Paket vermutlich nicht gebraucht wird ist der Aufwand maximal als gering einzuschätzen.

- de.xman.module

Im Paket Module gibt es 4 Klassen. Erstens die Klasse Module.java, welche Interface für verschiedene Funktionen ist, die in der Klasse ModuleImpl beschrieben sind. Alle importierten Klassen dieser beiden Klassen existieren bereits in OLAT. Die Klasse ModuleManager stellt Funktionen zum Erstellen, Speichern, Updaten, Löschen, Auflisten und finden von Modulen bereit, die in der Klasse CreateAndEditModuleForm aus dem Admin-Paket verwendet werden. Die vierte Klasse des Module-Pakets ist ModuleTableModel.java. Alle von ihr importierten Klassen existieren bereits in OLAT. Sie wird von der Klasse ExamAdminModulesController, ebenfalls aus dem Admin-Paket, verwendet.

Das Module-Paket kann zum Teil mit den Extentionpoint org.olat.persistence.DB implementiert werden. Dies bedeutet ein etwas höherer Aufwand. Der größte Teil der Implementierung kann allerdings übernommen werden.

- de.xman.catalog

Das Catalog-Paket besteht aus einen Controller und ein Table. Die Klasse CatalogEntryTableModel enthält sowohl Model aus auch View des Catalog. Sie erbt nur von Klassen die bereits in OLAT zur Verfügung stehen. Der Controller ExamCatalogController.java benutzt die Klasse CatalogEntryTableModel zum Erstellen des CatalogTableModel. Der Controller importiert viele Klassen, die fast alle in OLAT bereits vorhanden sind. Weiterhin benötigt sie die Klasse Exam aus den Paket Exam und eine Klasse, die das OLAT-PA-Modul zur Verfügung stellt.
Da alle importierten Klassen bereits in OLAT vorhanden sind, muss der OLAT-Code nicht gravierend geändert werden und die Implementierung kann größtenteils übernommen werden. Der Aufwand sollte auch dementsprechend gering sein.

- de.xman.calendar

Die Klasse in diesem Paket verwaltet die Kalendereinträge für die Prüfungsanmeldungen. Die weitesgehend gleichgebliebene Architektur der verwendeten Importe sowie die Tatsache dass das Paket nur eine Klasse enthält lassen einen eher geringen Migrationsaufwand vermuten.

- de.xman.protocol

Die Klassen dieses Paketes sammeln alle relevanten Informationen zu einer Prüfung zusammen und bieten das Model sowie die nötigen getter und setter zu deren Verwaltung an. Im Manager wird die Datenbankkommunikation geregelt. Der Migrationsaufwand dieses Pakets ist moderat einzuschätzen.

4. Produktfunktionen

Für den Großteil der Pakete ist momentan davon auszugehen das neben der Anpassung der Architektur sowie dem Festlegen der Beandefinitions keine weiteren schwerwiegenden Änderungen zu erwarten sind. Die dennoch zu realisierenden Modifikationen sind im folgendem dargestellt

- ## Zusätzliche Rolle des ExamAdmin

Ist:

Die Architektur von OLAT 6.1 lässt es nicht zu, das Rollenkonzept einfach zu erweitern, weshalb die entsprechenden Klassen selbst erweitert werden mussten.
Ausgangspunkt ist die Klasse BaseSecurityModule, welche das Singleton PersistingManager erzeugt. Hier werden die System-Level-Gruppen(Menge an Benutzern mit gleicher Rolle) angelegt, sofern diese noch nicht existieren. Erweitert wurde diese Klasse um die Gruppe ExamAdmin.
Damit die Rolle auch jemanden zugewiesen werden kann, musste noch ein neuer Eintrag in das Formular SystemRolesAndRightsForm gemacht werden. Wird die entsprechende Checkbox aktiviert, erzeugt PersistingManager eine SecurityGroupMembershipImpl, welche der Gruppe ExamAdmin zugeordnet wird. Wenn sich eine Person anmeldet, wird für diese geprüft, ob sie sich in der Gruppe befindet und ihr dann ein entsprechendes Objekt Roles zugeteilt.

Neu:

Auch in OLAT 7.2 ist das Rollenkonzept weitestgehend statisch, sodass die entsprechenden Klassen selbst verändert werden müssen. Die oben verwendeten Klassen haben sich in der neuen Version gering verändert; beispielsweise hat der BaseSecurityManager den PersistingManager ersetzt. Die Implementierung ist trotzdem auf ähnliche Weise, wie oben dargestellt möglich.

- ## Menüeintrag „Meine Studentenakte“ im HomeMainController

Ist:

Im MenuTree der Home Seite eines Studenten befindet sich der Eintrag „Meine Studentenakte“, welcher einen ESFLaunchController erzeugt, wenn er angeklickt wird. Dieser Eintrag wird direkt durch eine Codeergänzung im HomeMainController angelegt.

Neu:

Dieser direkte Eingriff in OLAT kann verhindert werden, indem der Link im Menü mit Hilfe einer ActionExtension erzeugt wird.

5. Produktdaten

Persistente Daten werden von den Klassen des Pakets org.olat.core.commons.persistence in einer MySQL Datenbank gespeichert. Xman Pakete, die Zugriff auf die Datenbank benötigen, enthalten eine eigene Manager-Klasse, welche die benötigten Objekte mit Hilfe der OLAT Klassen aus der Datenbank laden und in die Datenbank schreiben. Bei den zu speichernden Objekten handelt es sich zum einen um die Studentenakten, inklusive Kommentare, Krankenscheine und Prüfungsergebnisse, und zum anderen um die Prüfungen selbst, sowie die Studiengänge, Termine und Module.

6. Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität			X	
Zuverlässigkeit	X			
Benutzbarkeit	X			
Effizienz			X	
Änderbarkeit		X		
Übertragbarkeit				X

7. Ergänzungen

Verwendete Extensionkonzepte

- org.olat.home.HomeMainController

Über diesen Erweiterungspunkt lässt sich ein Link mit Text in der Home-Navigationsleiste erstellen.
Durch Definition eines Controllers, welcher dann durch den AutoCreator via Reflection intantiiert wird, ließen sich dann die Einträge für die Elektronische Studentenakte anzeigen.

- org.olat.persistence.DB

Dieser Erweiterungspunkt dient dazu, weitere Mappings für Hibernate hinzuzufügen. Notwendig wird dies, um hinzugefügte Objekte, wie zum Beispiel die Prüfungen, auf die Datenbank abbilden zu können. Das Mapping selbst erfolgt über eine xml-Datei, welche die Attribute des Objekts auf die Spalten der relationalen Datenbank abbildet.
Die Schnittstelle zur Datenbank wird von den Managerobjekten implementiert, welche Objekte in der Datenbank speichern, manipulieren oder löschen. Da die Managerklassen nach dem Singleton-Entwurfsmuster realisiert wurden, ist gewährleistet, dass Transaktionen betreffend eines Objektes getrennt voneinander ablaufen es also zu keinen Inkonsistenzen kommt.