

Pflichtenheft

Bearbeiter: Sebastian Dorn

Inhaltsverzeichnis

1 Zielbestimmungen	2
2 Produktübersicht (Soll-Ist-Vergleich)	2
3 Produktfunktionen	2
3.1 Grundsätzliche Änderungen	2
3.2 Rollen-/Rechtekonzept	3
3.3 Prüfungsverwaltung (Prüfung als Lernressource)	3
3.4 Prüfungen (Prüfung als Kursbaustein) inkl. Kursnoten	4
3.5 Elektronische Studentenakte (ESF)	4
3.6 Studiengänge	5
3.7 Module	5
4 Produktdaten	6
5 Produktleistungen	7
6 Qualitätsanforderungen	7
7 Technische Produktumgebung	7
7.1 Software	7
7.2 Hardware	7
8 spezielle Anforderungen	7
9 Ergänzungen	8
10 Quellen	8

1 Zielbestimmungen

OLAT wird mit dem PA-Modul an der Uni Leipzig als Prüfungsverwaltungstool genutzt. Hierbei werden durch den Dozenten/Prüfungsverantwortlichen Prüfungen mit Datum und Ort angelegt. Der Student kann sich hier verbindlich für die Prüfungen anmelden. Nach der Durchführung der Prüfung/Klausur werden durch den Dozenten/Prüfungsverantwortlichen die Ergebnisse der Studenten hier eingetragen. Diese Ergebnisse werden dann in die Studentenakten übertragen. Bisher war die Softwareversion 6.1 im Einsatz. Nun soll ein Update dieser Version auf die aktuelle Versionsnummer 7.2 erfolgen. Hierfür muss nicht nur die Software erneuert werden. Bei der Entwicklung wurde das Springframework verstärkt an das OLAT gebunden und es erfolgte ein Wechsel des Builderstellungstools von Ant auf Maven. Dafür sind folgende Funktionen zu überprüfen (siehe BA von Daniel Gerber):

- Rollen-/Rechtekonzept
- Prüfungsverwaltung
- Prüfungen
- Elektronische Studentenakte (ESF)
- Studiengänge
- Module

2 Produktübersicht (Soll-Ist-Vergleich)

In der Tabelle werden die einzelnen Funktionen und deren Implementierungen in den verschiedenen OLAT-Versionen kurz gegenüber gestellt. Abweichungen sind entsprechend markiert.

OLAT-PA Funktion	OLAT 6.1	OLAT 7.2
Rollen-/Rechtekonzept - Prüfling - Prüfungsamt	BaseSecurityModule, PersistingManager BaseSecurityModule, PersistingManager, SystemRolesAndRightsForm	BaseSecurityModule, BaseSecurityManager BaseSecurityModule, BaseSecurityManager , SystemRolesAndRightsForm
Prüfungsverwaltung	RepositoryHandler	SitesCreator
Prüfungen	CourseNodeConfiguration	
ESF	id.*, util.resource.OresHelper	id.*, util.resource.OresHelper, AfterLoginInterceptor
Studiengänge	UserPropertyHandler	
Module	id.*, util.resource.OresHelper	

3 Produktfunktionen

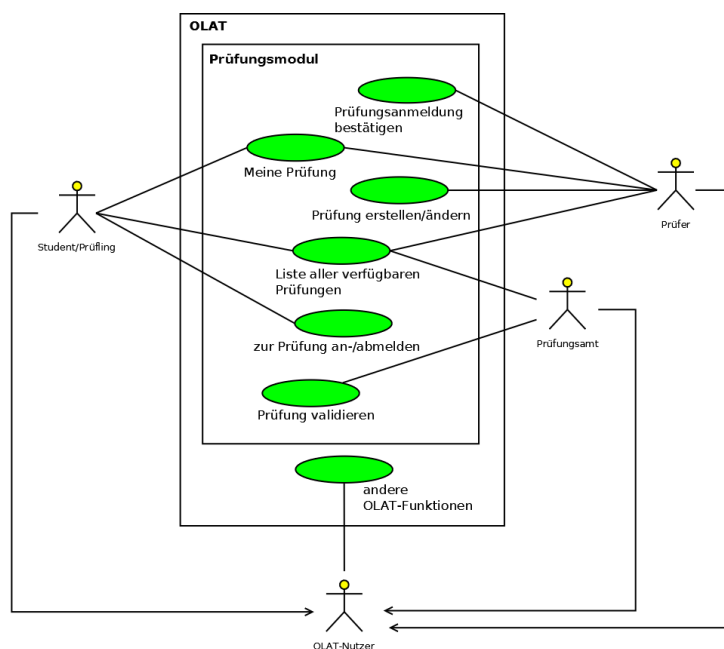
3.1 Grundsätzliche Änderungen

Durch die konsequente Ausrichtung des OLAT 7.2 auf Spring und dessen Architektur muss in allen Paketen des OLAT-PA-Moduls ein neues Unterpaket angelegt werden. Dieses Unterpaket ist wie folgt zu benennen: *"_spring"*. In diesem Paketen werden die *beans* angelegt in Form von XML-Dateien. Auf diese Weise erkennt das OLAT, dass weitere Module zur Verfügung stehen und lädt diese beim Systemstart mit.

3.2 Rollen-/Rechtekonzept

Definition:

Eine Benutzerrolle (oder kurz Rolle) definiert Aufgaben, Eigenschaften und vor allem Rechte eines Benutzers in einer Software. OLAT stellt nur 5 Rollen zur Verfügung, den Administrator, Benutzermanager, Gruppenmanager, Autor und Gast. Die Rollen reichen aber für das PA-Modul nicht, da der Prüfling weniger Rechte haben darf als der Autor, aber mehr haben muss als der Gast. Außerdem muss die Rolle Prüfungsamt zur Laufzeit des System zusätzlich einem Nutzer zugewiesen werden können. Diese Funktionalität wird durch das PA-Modul zur Verfügung gestellt.



Pakete im PA-Modul:

– > de.xman.admin.* nutzt *org.olat.core.gui.control.navigation.SiteInstance*

Implementierung im OLAT 6.1:

In der OLAT-Version 6.1 wurden die Rollen mit dem *BaseSecurityModule* geladen/instanziiert und dann durch den *PersistingManager* erzeugt und die Rechte zugewiesen. Dies wird einmalig beim Systemstart von OLAT durchgeführt. Das dynamische Zuweisen der Rolle "Prüfungsamt" zur Laufzeit musste deshalb über Adaptierung der Klasse *SystemRolesAndRightsForm* erfolgen (Hinzufügen einer Checkbox).

geplante Implementierung im OLAT 7.2:

In der OLAT-Version 7.2 muss dies im Groben ebenso erfolgen. Hierbei wird aber die neue Klasse *BaseSecurityManager* genutzt. Hier werden die Rollen/Rechte durch eine Factory beim Start durch die Methode *public void init()* initialisiert. Die dynamische Zuweisung der Rolle Prüfungsamt muss, wie in der Bachelorarbeit von Daniel Gerber beschrieben, implementiert werden.

3.3 Prüfungsverwaltung (Prüfung als Lernressource)

Definition:

Als Prüfung wird eine schriftlich oder mündlich durchgeführte benotete Leistungsmessung verstanden, welche von einem oder mehreren Individuen, in diesem Fall rechtmäßig immatrikulierten Studenten, über einen fest definierten Zeitraum und ein vorbestimmtes Thema, sowie zu einem im Voraus veröffentlichten Termin durchgeführt wird.

Pakete im PA-Modul:

– > de.xman.exam.* nutzt *org.olat.repository.handlers.RepositoryHandler*

Implementierung im OLAT 6.1:

Für die *RepositoryEntry* Prüfung musste ein neuer *RepositoryHandler* erstellt werden. Der Exam-Handler ist diese Erweiterung und stellt für Erzeugen, Editieren und Starten der Prüfung notwendigen Controller zur Verfügung. Zusätzlich ist der neue *RepositoryEntry* in der *RepositoryHandlerFactory* registriert, der alle Lernressourcen verwaltet und zur Verfügung stellt.

Implementierung im OLAT 7.2:

Da der *RepositoryHandler* ein Interface ist kann er wie in OLAT 6.1 beschrieben weiter eingesetzt werden. Zu prüfen wäre, ob eine Implementierung über den Extensionpoint *org.olat.extensions.sitescreator.SitesCreator* effektiver wäre. Eine Extension, die den SitesCreator als Erweiterung nutzt muss wiederum eine Liste von *SiteDefinitionObjects* besitzen, die jeweils wieder befähigt sind eine *SiteInstance* schaffen.

3.4 Prüfungen (Prüfung als Kursbaustein) inkl. Kursnoten

Definition:

Eine Prüfung als Kursbaustein kann zu jedem Zeitpunkt des Lebenszyklus eines Kurses durchgeführt werden, also auch gegen Ende des Semesters bei der Festlegung der Prüfungstermine. Die Konfiguration der Kursbausteine erfolgt per Konfigurationsdatei und kann zur Laufzeit nicht mehr verändert werden. OLAT bietet für Kursbausteine einen wohldefinierten Erweiterungspunkt.

Pakete im PA-Modul:

- > de.xman.exam.* nutzt *org.olat.repository.handlers.RepositoryHandler*
- > de.xman.protocol.* -
- > de.xman.appointment.* -
- > de.xman.nodes.* nutzt *org.olat.course.nodes.CourseNodeConfiguration*

Implementierung im OLAT 6.1:

Zum Erzeugen eines Prüfungskursbausteins wurde das Paket *ExamCourseNode* implementiert. Dieser Knoten wiederum muss sicherstellen, dass zum Editieren und Starten ein *ExamCourseNodeEdit*- bzw. *ExamCourseNode-RunController* zur Verfügung gestellt wird. Die Konfiguration, wie etwa zugehörige CSS Klassen, wird in der *ExamCourseNodeConfiguration* gekapselt.

Es existieren zwei größere Hauptklassen, ExamImpl und ProtocolImpl. Eine Instanz der Klasse ExamImpl repräsentiert eine vom Prüfungsautor angelegte Prüfung. Die ProtocolImpl repräsentiert dabei das in Papierform vorliegende Prüfungsprotokoll. Diese wird dynamisch in dem Moment instanziiert, in dem sich ein Student zu einem bestimmten Termin einer Prüfung anmeldet. Der Student wird durch die Identity (seine ID) repräsentiert. Eine Unterscheidung der Prüfung auf mündl./schriftl. erfolgt durch den bool *isOral*.

Implementierung im OLAT 7.2:

Grundsätzlich muss die Implementierung im OLAT 7.2 identisch zu der aus OLAT 6.1 erfolgen. Zusätzlich, um der Objektorientierung besser zu genügen, sollte die Klasse *ExamImpl* als abstrakt definiert werden. Die zusätzlichen Subklassen *OralExam* und *WrittenExam* sollten die vorhandenen Methoden implementieren.

3.5 Elektronische Studentenakte (ESF)

Definition:

Eine an genau einen Nutzer gebundene Akte, die der Student selbst im System beantragen muss. Sie enthält eine Reihe von persönlichen Informationen wie zum Beispiel Vorname, Nachname und Studiengang, die er wahrheitsgemäß angeben muss. Die Validierung dieser Informationen beziehungsweise der Akte erfolgt aktuell händisch von einem Mitarbeiter des Prüfungsamtes.

Außerdem enthält die Akte eine Unterstützung zur Verwaltung von Prüfungsergebnissen, Krankenscheinen und Kommentaren. Weiterhin dokumentiert und archiviert sie die Kommunikation zwischen Prüfungsamt und dem entsprechenden Studenten.

Pakete im PA-Modul:

– > de.xman.esf.* nutzt *org.olat.core.id.** und *org.olat.core.commons.persistence.PersistentObject*

Implementierung im OLAT 6.1:

Durch die Implementierung der Interfaces in *org.olat.core.id.** (siehe BA von Daniel Gerber). Die ESF nutzt hier die Identität des Nutzers im System, d.h. jede ESF hat eine eigene OLAT ID, die der entsprechenden Identität des Nutzers zugewiesen wird.

Implementierung im OLAT 7.2:

Grundsätzlich muss an der ESF nichts geändert werden. Bisher musste die ESF durch den Studenten händisch angelegt und zur Validierung beantragt werden. Im OLAT 7.2 steht ein Extensionpoint (*AfterLoginInterceptor*) zur Verfügung. Mit diesem, kann nach dem Login geprüft werden, ob bereits eine ESF existiert. Sollte dies nicht der Fall sein, wird die ESF erstellt. Der Student muss dann noch zusätzlich benötigte Daten angeben, bevor die ESF zur Validierung weitergeleitet wird.

3.6 Studiengänge

Definition:

Studiengang heißt die Gesamtheit der Lehrinhalte eines wissenschaftlichen Studienfaches an einer Hochschule. Um auch den Studiengang in der ESF zu erfassen, soll die Profilsseite von OLAT um eine neues Feld Studiengang erweitert werden. Hier ist anzumerken, dass Studiengänge bereits im bisherigen OLAT System als einfache Zeichenkette angegeben werden können, dies aber nicht akzeptabel ist, da so bei einem potenziell sehr großen Benutzerkreis eine Vielzahl von Bezeichnungen für denselben Studiengang entstehen würde. Ferner muss eine Administrationsoberfläche für Mitarbeiter des Prüfungsamts geschaffen werden, um die Einträge des Studiengangsfeldes im Profilformular verwalten zu können.

Pakete im PA-Modul:

– > de.xman.studyPath.* nutzt zusätzlich implementierte Klasse *org.olat.user.StudyPathPropertyHandler*

Implementierung im OLAT 6.1:

Im OLAT 6.1 wurde das Interface *UserPropertyHandler* genutzt. Das Freitextfeld zur Eingabe des Studienganges wurde durch eine DropBox ersetzt. Der *UserPropertyHandler* wird durch die Klasse *AbstractUserPropertyHandler* erweitert. Die OLAT-PA Klasse *StudyPathPropertyHandler* nutzt diese abstrakte Klasse.

Implementierung im OLAT 7.2:

Es ist zu prüfen, ob der *StudyPathPropertyHandler* auch außerhalb der *org.olat.user - Pakete* implementiert werden kann um diese Funktionalität umzusetzen. Vielleicht kann hier auch der *AfterLoginInterceptor* genutzt werden.

3.7 Module

Definition:

Als Modul wird ein zeitlich und inhaltlich abgeschlossener Verbund von Lehrveranstaltungen mit einem ähnlichen inhaltlichen und thematischen Schwerpunkt verstanden.

Pakete im PA-Modul:

– > de.xman.module.* nutzt *org.olat.core.id.** und *org.olat.core.commons.persistence.PersistentObject*

Implementierung im OLAT 6.1:

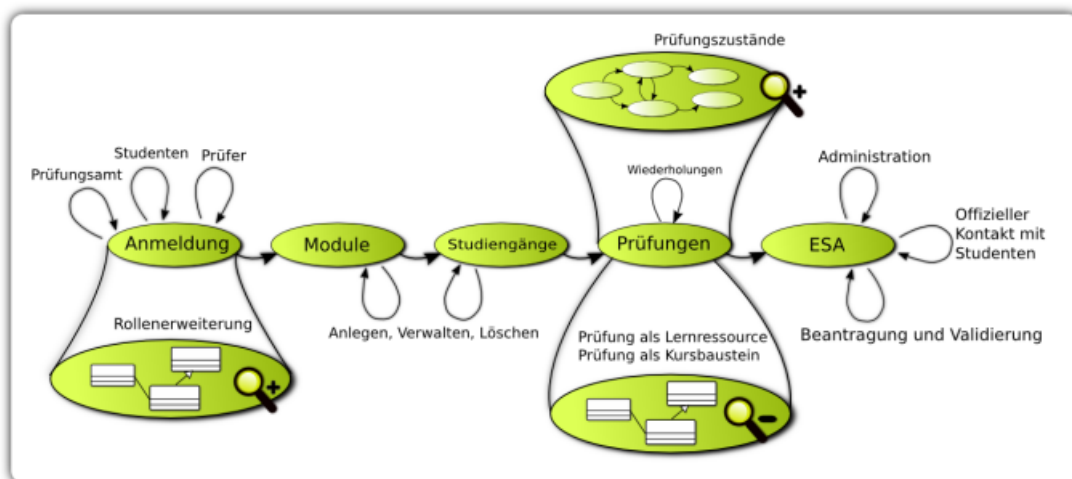
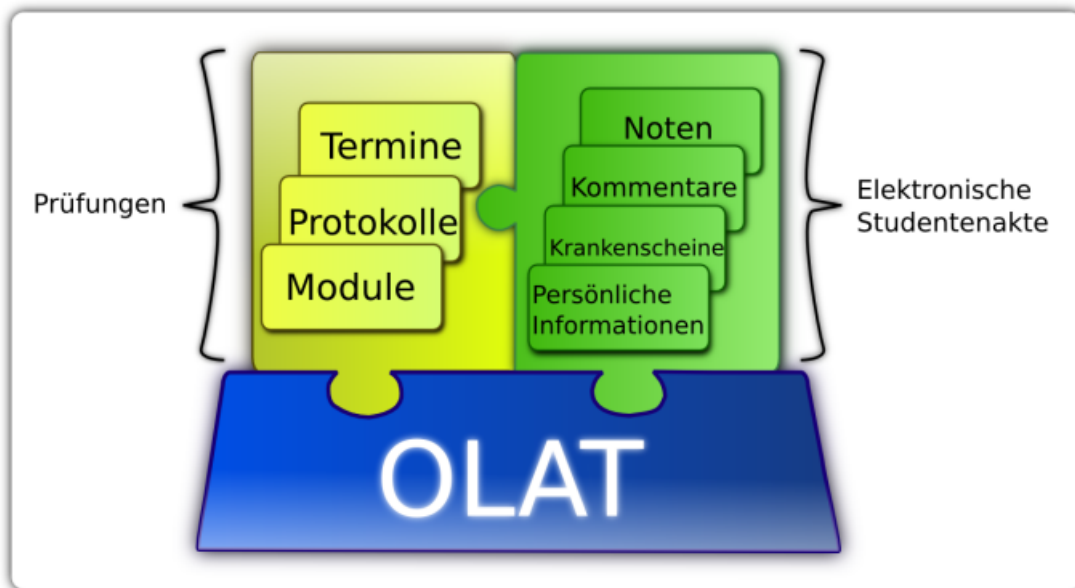
Durch Erweiterung der Interfaces in *org.olat.core.id.** wird jedem Modul eine eigene IdentitätsID (*object key*) zugewiesen, da hier im OLAT eine neue Instanz angelegt wird. Diese Instanz wird dann während des Systemstarts geladen.

Implementierung im OLAT 7.2:

Die Implementierung erfolgt wie im OLAT 6.1.

4 Produktdaten

In den u.a. Abbildung werden wichtige Funktionen und deren Beziehungen zueinander des OLAT-PA dargestellt.



5 Produktleistungen

Das OLAT-PA-Modul soll für neue OLAT Versionen updatefähiger werden. Hierfür sind soweit möglich Schnittstellen und Extensionpoints zu nutze, die durch die OLAT Version 7.2 zur Verfügung gestellt werden.

6 Qualitätsanforderungen

Funktionalität	<i>unverändert</i>
Zuverlässigkeit	<i>unverändert</i>
Benutzbarkeit	<i>unverändert</i>
Effizienz	<i>unverändert</i>
Änderbarkeit	<i>unverändert</i>
Übertragbarkeit	Ist besser, da durch verbesserte Nutzung von Schnittstellen / Extensionspoints bei einem Softwareupdate des OLAT auf eine neuere Version das OLAT-PA-Modul weniger angepasst werden muss.

7 Technische Produktumgebung

Das PA-Modul ist eine Erweiterung für die LMS Software OLAT und ist nicht allein lauffähig. Bei der Entwicklung des Updates wurde folgende Hard- und Softwareumgebung genutzt und ist dann für den weiteren Einsatz notwendig.

7.1 Software

Client	Server
<ul style="list-style-type: none"> - Standardisierter Webbrowser wie Internet Explorer, FireFox, Opera - Java Virtual Machine 	<ul style="list-style-type: none"> - Lauffähiges OLAT Version 7.2 - SQL Datenbank - Tomcatserver ab Version6 - Maven 2 - VeloCity - Mercurial

7.2 Hardware

- internetfähiger PC
- eine über URL/IP erreichbaren Server

8 spezielle Anforderungen

- Entwicklungsumgebung Eclipse / SpringSourceToolSuite
- lokale Entwicklung (Einrichtung lokaler Hardware und Software - Nutzung lokale OLAT-Instanz)
- Nutzung zentrales Kalender-, Aufgaben- und Dokumentenpfegetool (Feng)
- zentraler Upload von funktionierendem Quellcode auf den Praktikumsserver

9 Ergänzungen

keine

10 Quellen

- Bachelorarbeit von Daniel Gerber vom 07.09.2009
- OLAT Development 7.0 (<http://www.olat.org/docu/dev/>)
- Dokumente Entwicklung OLAT PA-Modul im Softwaretechnikpraktikum von 2007 (Gruppe 01-04) (<http://pcai042.informatik.uni-leipzig.de/swp/SWP-07/index.html>)
- Quellcode OLAT-PA, OLAT 6.1, OLAT 7.2