Entwurfsbeschreibung OLAT

Gruppe SWP08-7

20. Mai 2008

Inhaltsverzeichnis

1 Allgemeines **2**

2 Produktübersicht **2**
2.1 Rollenmanagenment 3
2.2 Rechtemanagement 4

3 Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem **5**
3.1 Architekturübersicht 5
3.2 Eingesetzte Frameworks und Technologien 6
3.3 Entwurfsprinzipien 6

4 Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete **7**
4.1 org.olat.core.id.* 7
4.2 org.olat.group.* 8
4.3 org.olat.group.right.* 9
4.4 org.olat.resource.* 9
4.5 Zugriff von Erweiterungspunkten auf Rechte- und Rollenmanagement 10
4.6 Erweiterungspunkt im Rechtesystem 10

1 Allgemeines

Das OLAT (Online Learning And Teaching) ist ein webbasiertes Learning Management System (LMS) mit einer Vielzahl von Funktionen, um einem Campusmanagement mit breitem Einsatz und Anforderungsspektrum zu genügen.
OLAT beinhaltet neben einem flexiblen Kurssystem auch diverse kursunabhängige und kursübergreifende Funktionen. Hierzu zählen insbesondere eine allgemeine Verwaltung von Lernressourcen inklusive Katalogisierung und die Bereitstellung von Editorwerkzeugen für Test, Fragebogen oder Kurse.

2 Produktübersicht

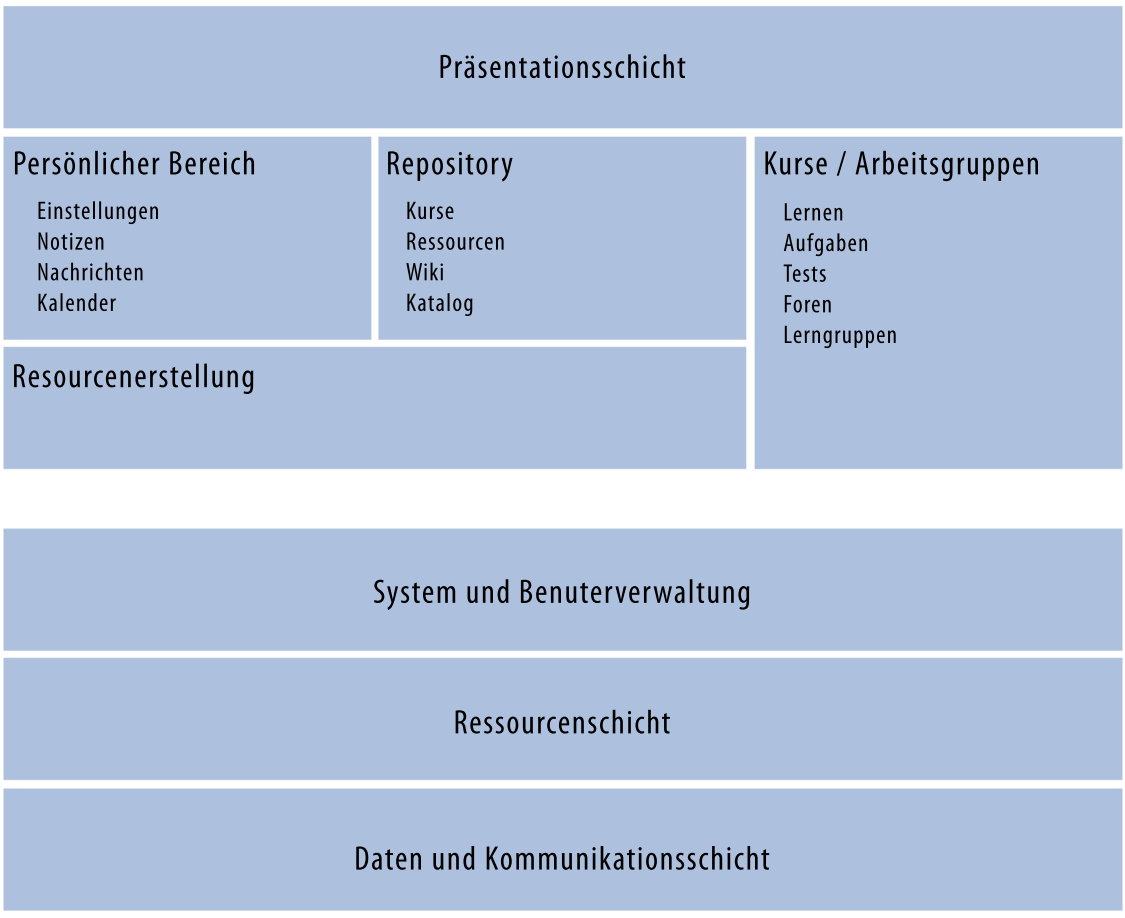

Abbildung 1: Produktuebersicht

2.1 Rollenmanagenment

Die Zuweisung von Zugriffsrechten innerhalb von OLAT erfolgt durch Zuordnung von Rollen an die einzelnen Benutzer. Jede Rolle bringt entsprechende Rechte mit sich. Es existieren die folgenden Rollen:

- Gast (anonymer Nutzer)
- Benutzer
- Autor
- Benutzerverwalter
- Gruppenverwalter
- Systemadministrator

Gast Jeder nichtregistrierte Nutzer nimmt die Rolle des Gastes ein. Ein Gast kann nur auf Ressourcen zugreifen, die explizit für Gäste freigegeben sind. Die Gastrolle bietet einem Besucher die Möglichkeit, einen Überblick über die Funktionsweise von OLAT zu gewinnen.

Benutzer Nach Registrierung im OLAT , bekommt man die Rolle "Benutzerßugewiesen. Als Benutzer steht einem der volle Umfang an Funktionen, die nur die eigene Person betreffen und die Möglichkeiten von anderen Nutzern nicht beeinflussen, zur Verfügung. Ein Gast kann z.B. seinen eigenen Terminkalender bearbeiten, Notizen erstellen, Arbeitsgruppen erstellen, Dateien in seinen persönlichen Ordner laden, usw. . Ein Administrator kann einem Benutzer weitere Rechte zuweisen.

Autor Ein Autor kann hat alle Rechte eines Benutzers und kann zusätzlich Lerngruppen erstellen. Er kann Nutzer in diese Gruppen hinzufügen oder entfernen. Der Autor einer Lerngruppe besitzt innerhalb der Lernressource administrative Rechte.

Gruppenverwalter Dieser kann kursübergreifende Lern- oder Rechtegruppen erstellen und verwalten.

Benutzerverwalter Der Benutzerverwalter kann Benutzer anlegen oder importieren (über eine Tabelle mit Daten). Er kann weiterhin schon bestehenden Benutzern weitere Rechte geben.

Administrator Der Administrator hat alle Rechte eines Benutzers, Autors, Gruppen- und Benutzerverwalters. Weiterhin hat er weitere technisch administrative Funktionen zur Verfügung, welche ihm die Verwaltung des Systems ermöglichen.

Anhand dieser Rollen wird im OLAT unterschieden welche Funktionalitäten dem Nutzer zustehen.

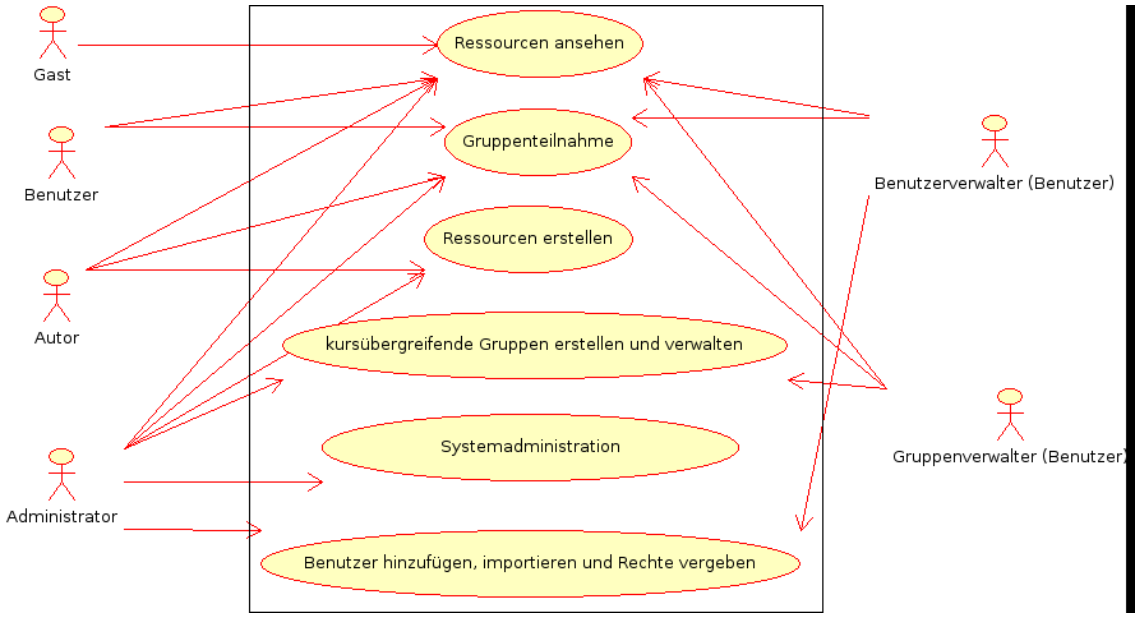

Abbildung 2: Rollenmanagenment

2.2 Rechtemanagement

Das Rechtemanagement ermöglicht das Sperren/Entsperren von Funktionalitäten oder Seiten für ausgewählte Benutzer.

3 Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem

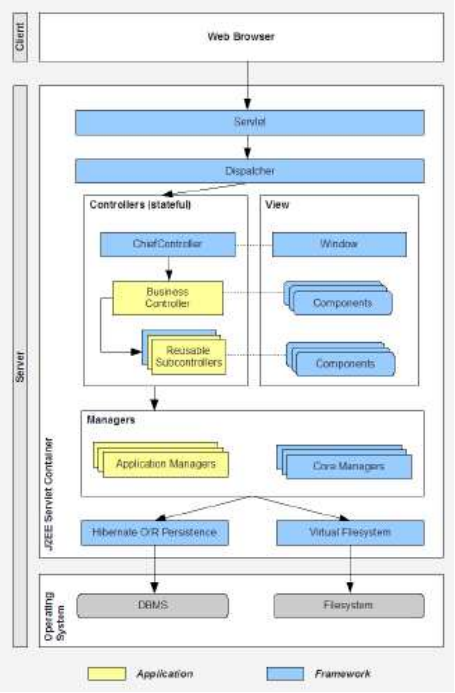

Abbildung 3: Strukturprinzipien

3.1 Architekturübersicht

Das LMS OLAT hat eine klassische 3-Schichten-Architektur und besteht somit aus einer Client- schicht und, auf der Serverseite, aus einer Anwendungs- und einer Datenschicht. Folgende Layers können dabei unterschieden werden:

OLAT Servlet / Dispatcher Initialisierung des OLAT und Weiterleiten von Nutzeranfragen.

Windows/ChiefController Bereitstellung des Inhaltes für Webbrowser und Navigation.

Controllers Verantwortlich für einen speziellen Workflow.

Managers Stellt Basisfunktionen bereit.

VFS (Virtual File System) und Hibernate Datenschicht befinden sich die Datenbanken und das Dateiensystem.

3.2 Eingesetzte Frameworks und Technologien

3.2.1 Apache Velocity

Der Java-Template-Engine Velocity bietet die Grundlage der View-Komponente des Systems. Mittels der Velocity Template Language (VTL), können in eine eigentliche statische HTML- Seite dynamische Textersetzungen durchgeführt werden. Dies geschiet über ein Velocity-Servlet, welches die HTML-Ausgabe generiert.

3.2.2 Apache Commons

Apache Commons stellt eine ganze Reihe nützliche und wiederverwendbare Java-Bibliotheken und -komponenten zur Verfügung.

3.2.3 Hibernate

Das Hibernate-Framework stellt Funktionen bereit, welche das persitente Speichern von Objek- tern in einer Datenbank ermögichen. Der Programmierer muss dabei keine SQL-Abfragen erstel- len, sondern arbeitet mit den Objekten von Hibernate. Dies wird auch als Object-Relational- Mapping bezeichnet.

3.2.4 Spring

Unter Verwendung eines leichtgewichtigem Containers wird ein konsistenter Mechanismus zur Verfügung gestellt um Applikationen in eine J2EE-Umgebung zu integrieren. Mit leichtem Con- tainer wird dabei die Fähigkeit eines Containers gemeint, Applikationscode in verschiedenen Umgebungen zu verwalten, ohne das dafür spezielle Abhängigkeiten im Code bzgl. der Container- API erforderlich sind.

3.3 Entwurfsprinzipien

3.3.1 Komponenten basiert

Eine Webseite wird aus Komponenten zusammengesetzt und durch Traversierung des Kom- ponentenbaumes anschliessend als HTML generiert und als HTML-Seite an den Webbrowser geschickt. Die Komponenten sind einem Business-Controller zugeordnet, welcher die Ablauogik eines mo- mentanen Bildschirmausschnittes steuert.

3.3.2 Event-basiert

Wenn ein Nutzer auf eine Stelle im Webbrowser klickt, so wird diesem Klick immer genau eine Komponente zugeordnet, welche dieses Ereigniss verarbeitet und ein Event an dem Business- Controller, der der Komponente zugeordnet ist, weiterleitet.

3.3.3 wiederverwendbare GUI-Workows/Buisiness-Controllers

Das OLAT-GUI Framework ermöglicht eine einfache Wiederverwendung von Buisiness-Controllers, welche die Workows representieren. Der Ablauf eines Workows sieht aus Sicht der GUI immer gleich aus.

3.3.4 Trennung zwischen Code, Ubersetzung und HTML-Templates

Die meisten Packete enthalten neben den Code-Dateien die zwei Unterordner "content" und "i18n" und eventuell "static" für statische Dateien, wie Bilder oder CSS-Dateien. In "content" benden sich die HTML-Templates und in "i18n" die Ubersetzungsdateien für die verschiedenen Sprachen. Dadurch können neue Sprachen leicht eingebunden werden und die Templates leicht, durch einen Designer, verändert werden ohne sich in den Quelltext reinlesen zu müssen.

4 Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete

4.1 org.olat.core.id.*

Das Packet org.olat.core.id spielt eine wichtige Rolle bei der Verwaltung vom Nutzer- und Rol- lenmanagment, denn es stellt wesentliche Funktionen für die eindeutige Identikation von Benut- zern, deren Rollen und Ressourcen. Die wichtigsten Klassen und Schnittstellen in diesem Paket sind:

4.1.1 Auditable.java

Ist eine Schnittstelle, um für persistente Objekte ein einheitliches Format für die Darstellung von Erstellungs- und zuletzt editierter Zeit zu gewährleisten.

4.1.2 Persistable.java

Stellt die Integrität der Daten sicher.

4.1.3 User.java

Ist ein Interface. Ein Userobjekt, welches dieses Interface implementiert, repräsentiert einen bekannten und angemeldeten Benutzer und hält dessen Daten wie Name, Email etc. Jeder Session eines Benutzers wird ein solches Objekt zugeordnet, um jederzeit seine Daten zur Verfügung zu haben.

4.1.4 Roles.java

Ist eine Klasse die festhält, welchen der fünf in OLAT verfügbaren Rollen ein Benut- zer angehört. Ein und derselbe Benutzer kann mehreren Rollen zugeordnet werden.

4.1.5 Identify.java

Ist eine von Persistable.java und auditable.java abgeleitete Schnittstelle. Gibt das mit ihr assozierte Userobjekt, nur dessen Namen und Informationen über letzte Änderungen, also unkritische Informationen aus.

4.1.6 IdentifyEnvironment.java

Ist ein Container für ein Identify-Objekt, eine Rolle, und weitere Attribute.

4.1.7 OLATResourceable.java

Dieses Interface ist ein eindeutiger Bezeichner für Objekte in OLAT, welcher den Typ der Klasse zurück gibt.

4.2 org.olat.group.*

BusinessGroups sind die Oberklasse aller Gruppen, verwendet wird ein BusinessGroup-Manager um alle Aktionen zu verwalten und Events auszulösen etc.

4.2.1 BusinessGroup

Interface, welches Oberklasse für alle BusinessGroups ist, wie BuddyGroups oder LearningGroups oder RightGroups. Deniert die grundsätzlichen Getter und Setter für alle relevanten Daten solcher Gruppen.

4.2.2 BusinessGroupAddResponse

Leitet 'fired Events Events' vom BusinessGroupManager weiter (also vor allem von addXXX-AndFireEvent).

4.2.3 BusinessGroupFactory

Dient der Erstellung neuer Instanzen von BuddyGroups, LearningGroups und Right-Groups. Dabei wird getestet, ob die Gruppe vielleicht schon exisitert. Weiterhin wird die neue Gruppe gleich persistent gemacht.

4.2.4 BusinessGroupImpl

Implementierung des BusinessGoup-Interfaces, eine solche Gruppe bekommt unter anderem die Attribute: Beschreibung, Name, Typ, minimale und maximale Teil- nehmerzahl, Zeit des letzten Zugriffs, sowie 3 SecurityGroup-Objekte für Besitzer, Teilnehmer und Warteliste.

4.2.5 BusinessGroupManager

Interface, welche folgende Arbeitsabläufe beschreibt: Erstellen einer neuen Gruppen- Instanz, welche persistent ist und als Besitzer die “Identity” enthält, die die Gruppe erstellt hat. Weiterhin kann man Gruppen finden die einer bestimmten “Identity” zugeordnet sind. Geplant ist weiterhin, eine Gruppe zu aktualisieren und zu löschen.

4.2.6 BusinessGroupManagerImpl

Persistente Implementierung des BusinessGroupManager-Interfaces. Sichert die Da- ten in einer Datenbank. Ist eine Singleton-Klasse. Implementiert alle Funktionen des BusinessGroup-Interfaces.

4.2.7 BusinessGroupManagerImplTest

Implementiert org.olat.core.gui.control.WindowControl und erweitert org.olat.core.test.OlatTestCase. Initialisiert sich ein Test-Setup, also spezielle Anwendungsfälle und loggt dabei die Ausgaben, Fehler etc. bei der Benutzung der BusinessGroupManager-Implementierung.

4.2.8 BusinessGroupTest

Erweitert ebenfalls OlatTestCase um mit einem Test die Funktionalität der BusinessGroup-Implementierung zu testen und alle relevanten Informationen zu protokollieren.

4.2.9 GroupfoldersWebDAVProvider

Implementiert das Interface WebDAVProvider und bietet Methoden um Gruppenin- formationen zu transferieren (von anderen Medien, HDDs etc.).

4.3 org.olat.group.right.*

Verarbeitet die Rechtegruppen (RightGroup).

4.3.1 BGRightManager

Ist ein Interface. Gibt die Methodenstümpfe für Abfragen vor, Finden, Setzen und Löschen von Rechten für Rechtegruppen (spezielle BusinessGroup).

4.3.2 BGRightManagerImpl

Implementiert BGRightManager als Singleton-Klasse, überschreibt dabei alle Me- thoden des Interfaces mit sinnvollem Inhalt.

4.3.3 BGRights

Ist ein Interface. Enthält einzig die Methoden .getRights():List zur Ausgabe der ge- speich- terten Rechte des Objektes, sowie transateRight(String):String, das zur Über- setzung (Translation) genutzt werden kann. Der BGRightManager nutzt derzeit al- lerdings dieses Interface für die Rechte nicht, sondern übergibt einfach Strings.

4.4 org.olat.resource.*

Dient der Verwaltung von OLAT-Ressourcen, also Klassen, die org.olat.core.id.OLATResourceable implementieren.

4.4.1 OLATResource

Leeres Interface zum Markieren von Ressourcen, implementiert OLATResourceable, Persistable und Auditable.

4.4.2 OLATResourceImpl

Erweitert PersistentObject und implementiert OLATResource. Wird als Singleton instanziert. Belegt dabei alle für Hibernate relevanten Methoden, sowie toString() und grundlegende Ressourcenmanagement-Methoden.

4.4.3 OLATResourceManager

Wird als Singleton instanziert. Dient zum verwalten von OLATResource-Objekten, diese können erstellt, modiziert, gefunden und gelöscht werden.

4.4.4 OLATResourceManagerTest

Erweitert OlatTestCase und implementiert OLATResourceable um den Resourcen- Manager in einem Testfall zu überprüfen und die Ergebnisse zu loggen.

4.5 Zugriff von Erweiterungspunkten auf Rechte- und Rollenmanagement

Die Instanzen der verschiedenen Manager (Bsp.4.3.1 BGRightManager) lassen sich aus der Erweiterung heraus ermitteln, somit ist auch der Zugriff auf die Funktionalität dieser möglich.

4.6 Erweiterungspunkt im Rechtesystem

Anpassung der Klasse Roles.java aus dem Packet org.olat.core.id.*. Zusätzlich zu den statisch vorhandenen Rollen müssen erweiterbare Rechteattribute (Bsp. Liste von Strings) eingefügt werden. Über ein neue Methode, der eine feste Rollenkonstante übergeben wird, wird die Rolleninstanz auf vorhandene Rechte / Attribute geprüft.