

Softwaretechnik-Praktikum SS 2008

Recherchebericht

1.Begriffe

IMS: Das IMS Global Learning Consortium,Inc (IMS) hat sich zum Ziel gesetzt, offene Spezifikationen für den Bereich E-Learning zu entwickeln.

J2EE Java Platform, Enterprise Edition ist die Spezifikation einer Softwarearchitektur für die transaktionsbasierte Ausführung von in Java programmierten Anwendungen. OLAT basiert auf J2EE.

OLAT: Online Learning and Training und wurde zum Zweck entwickelt, Vorlesungen und Übungen über das Internet anbieten zu können. Die größte Gewichtung liegt im Aufbau und der Durchführung von Lehrveranstaltungen. OLAT ist eine Webapplikation, ist Open Source und liegt ab der Version 3.x in einer neu implementierten JAVA-Version vor. Im Wintersemester 99/00 kam OLAT am Institut für Informatik an der Universität Zürich bei erstmals zum Einsatz.

HTML Hypertext Markup Language. Es ist eine textbasierte Auszeichnungssprache um Inhalte wie Text, Bilder und Hyperlinks in Dokumenten darzustellen. HTML-Dokumente bilden dabei die Basis des Internets und werden von einem Webbrowser dargestellt.

SCORM steht für Sharable Content Object Reference Model. Es umfasst eine (Variablen-) Sammlung von Standards und Spezifikationen aus verschiedenen Quellen, um einfache Austauschbarkeit, einen allgemeinen Zugriff und Wiederverwendbarkeit in verschiedenen Umgebungen von web-basierenden Lerninhalten (E-Learning) zu ermöglichen

Learning Management System(Lernplattform) ist ein Programm, das die Betreuung von Lehrinhalten ermöglicht. Dabei bietet das LMS eine Benutzerverwaltung, sowie Rollen- und Rechtevergaben an, stellt Arbeitswerkzeuge und Kommunikationsmethoden wie Foren, Wikis etc. zur Verfügung und bietet Möglichkeiten zur Kursverwaltung, für Onlineevaluationen und

-bewertungen. Natürlich sorgt das LMS außerdem für die Verfügbarkeit und die Präsentation der Lehrinhalte in einem normalen Browser. Das Open-Source LMS OLAT ist Gegenstand dieses Projektes, weshalb zumindest der Begriff geklärt sein muss.

Servlets/JSP Servlets zu Deutsch etwa "Server-chen", stellen Anwendungen für Server zur Verfügung oder erweitern diese. Alle Servlets implementieren die Java-Klasse `javax.servlet` oder eine davon abgeleitete Klasse, um anschließend innerhalb eines J2EE Applicationservers zu laufen. Servlets arbeiten auf dem request-respond Prinzip und können auf zahlreiche requests antworten, hauptsächlich werden sie jedoch verwendet um mittels HTTP die Anwendungsmöglichkeiten eines Webservers zu erweitern. Zu diesem Zweck existiert auch die von `javax.servlet` abgeleitete Klasse `javax.servlet.http.HttpServlet`. Dies ermöglicht das Erstellen von dynamischen Web-Inhalten. Die JavaServer Pages erweitern die Servlet-Technologie, indem sie einfaches, dynamisches Erstellen von HTML und XML-Ausgaben ermöglichen. Damit wird zum einen wiederum die Darstellung von der Logik getrennt und zum anderen einfaches Erstellen von dynamischen Websites ermöglicht. OLAT als Webanwendung basiert auf Servlets und den JSPs, weshalb diese Begriffe eine zentrale Rolle erhalten.

Velocity ist ein Open-Source-Projekt und entstammt der Apache Jakarta-Projektgruppe. Es handelt sich hierbei um ein Java-basiertes Template, das es ermöglicht Java-Objekte in verschiedenste andere Formate, wie Text, XML, SQL, Post Script oder HTML einzubinden. Da es streng nach dem MVC-Konzept konzipiert wurde und nicht sonderlich schwer zu erlernen ist, erlaubt Velocity quasi paralleles Arbeiten an funktionalen Code und einem ansprechenden Äußeren. Die Hauptanwendungen von Velocity sind XML-Transformationen, das Erstellen von dynamischen Webseiten und Webanwendungen. OLAT nutzt Velocity aufbauend auf der Präsentationslogik, so dass der Client die erhaltenen Daten nur noch darstellen muss. Es wird deshalb ein wichtiges Hilfsmittel im Projekt sein.

Framework Als Framework bezeichnet man eine Menge von verknüpften Klassen, welche zusammen ein wiederverwendbares und erweiterbares Gerüst für die Entwicklung von Software eines bestimmten Typs bilden. Frameworks geben eine bestimmte Anwendungsarchitektur vor. Es werden konkrete Implementierungen eingebunden, die dann durch das Framework gesteuert werden

XML Bei XML handelt es sich um eine Auszeichnungssprache ("Extensible Markup Language") für Datenrepräsentation in Textform. Dabei stellt XML nur eine Metasprache dar, aus der mittels Schemasprachen wie DTD die eigentliche Sprache gewonnen wird. Zu den auf XML basierenden Sprachen gehören u.a. XHTML, WML und RSS. Die repräsentierten Daten sind relativ leicht durch Maschinen zu verarbeiten, bleiben aber auch für Menschen lesbar und änderbar. Die Anwendungen von XML sind entsprechend vielfältig, am häufigsten wird XML jedoch zur Speicherung Daten oder

zur intermaschinellen Kommunikation genutzt. Im Projekt OLAT taucht XML u.a. für die Konfiguration und Kommunikation auf.

2. Konzepte

OLAT Systemarchitektur Die OLAT Systemarchitektur beschreibt auf einer abstrakten Ebene die Module und den Aufbau des Learning Management Systems (LMS) OLAT. OLAT kann in drei Schichten erklärt werden, der Integrationsschicht, der Serviceschicht und der Applikationsschicht.

OLAT Kurskonzept Das OLAT Kurssystem ist für die Erstellung und den Betrieb von E-Learning Kursen gebaut worden. Es grenzt sich von Konkurrenzprodukten durch ein äusserst modulares und flexibles Konzept ab und lässt auf didaktischer Ebene die Umsetzung von fast allen didaktischen Konzepten zu. OLAT wendet dabei ein Arbeitskopie Modell an: im Editor arbeitet man an einer Arbeitskopie, Änderungen werden erst nach einem selektiven Publikationsprozess in die Laufzeitversion des Kurses übernommen. So ist jederzeit ein in sich konsistentes Laufzeitverhalten sichergestellt.

Didaktisches Konzept Das didaktische Konzept eines Kurses ist in OLAT nicht vorgegeben. Der Autor kann aus einer Reihe von so genannten Kursbausteinen (Course elements) eine hierarchische Kursstruktur aufbauen und die Sichtbarkeit und den Zugang zu den Kursbausteinen detailliert steuern. Dies ermöglicht die Umsetzung von beliebigen didaktischen Vorgehensweisen. Kursbausteine sind die wichtigste Einheit und werden über eine gemeinsame Schnittstelle angesprochen. Diese Abstraktion erlaubt die Erweiterung des Kurssystems um neue Funktionen auf genial einfache Art und Weise. Die Kursbausteine können beliebige Funktionen anbieten und können in der Regel in eine der drei Kategorien Content/Structure, Assessment und Collaboration eingeteilt werden. Neue Kursbausteine können per Konfiguration dem System hinzugefügt werden. Jeder Kursbaustein verfügt zudem zur Steuerung des individuellen Lernpfades über so genannte Sichtbarkeits- und Zugangsbeschränkungen. Dies sind Regeln, die zur Laufzeit mit den Daten des Benutzers durchlaufen werden und so einen Kursbaustein für einen Benutzer freigeben oder sperren. Regeln können zusammengesetzt werden aus Datumsregeln, Punkteregeln, Gruppenregeln oder Attributprüfungen.

Technische Struktur und Softwarearchitektur Das LMS OLAT wird unter Verwendung moderner Technologien und Hilfsmittel entwickelt und ist in der Programmiersprache Java geschrieben. Dies ermöglicht einen Betrieb unter verschiedenen Betriebssystemen wie Windows,

SWP08-6-OLAT

Projektleiter: Johannes Kreidler ,Verantwortlicher :Nie Mao

21.April 2008

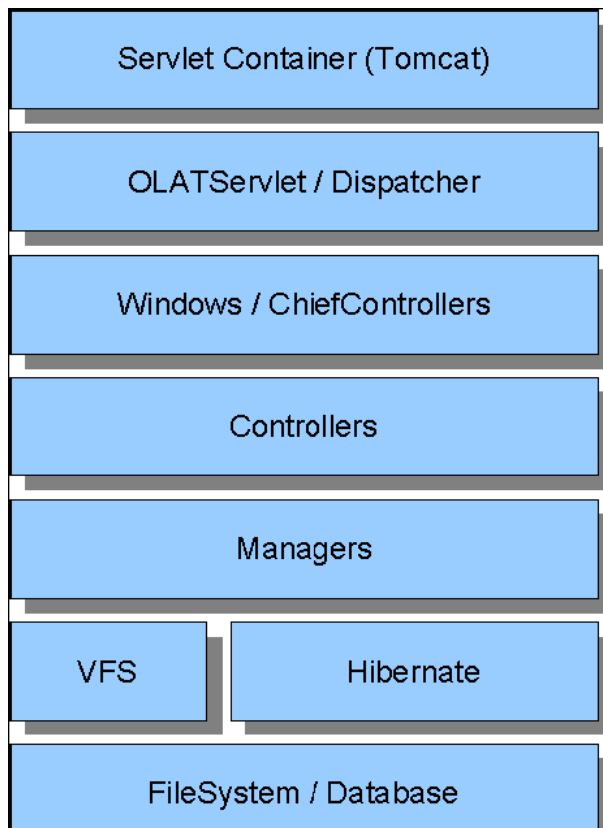
MacOSX, Linux, BSD, Unix oder Solaris, ohne Anpassungen vornehmen zu müssen. Zur Persistierung der Daten können ebenfalls verschiedene Datenbankmanagementsysteme wie zum Beispiel MySQL, Postgres oder Oracle zum Einsatz kommen. OLAT setzt auf der Java 2 Enterprise Edition (J2EE) auf und verwendet eine Servlet basierte Architektur. Ein eigens entwickeltes Model-View-Controller (MVC) Framework (welches weit über das hinausgeht, was z.B. JSF oder Spring MVC leisten) erlaubt eine moderne, fehlerreduzierte und schnelle Entwicklung mit strikter Trennung zwischen Darstellungslogik, Ablauflogik, Businesslogik und der Datenhaltung. Um eine möglichst hohe Wartbarkeit und Erweiterbarkeit zu erzielen wird bei der Entwicklung stets ein spezielles Augenmerk auf eine saubere Entkoppelung und eine hohe Wiederverwendbarkeit der verschiedenen Softwarekomponenten gelegt.

Software Schichtenmodell und Tiers

OLAT hat im klassischen Sinne eine 3-Tier-Architektur:

- User Tier
- Via Webbrowser bedient der User das LMS
- Ein kleines Set von Interaktionskontrollen geschieht via JavaScript.
- Business Tier
- Die Prozess- und Ablauflogik wird hier gesteuert
- Benutzerdaten Validierung (Formulare)
- Verwaltung der Benutzersitzungen
- Persistierung (Create, Read, Update, Delete für DB und Filesystem)
- Data Tier
- Mit Hibernate wird der Zugriff auf die Datenbank abstrahiert
- Mit dem Virtual Filesystem wird der Zugriff auf das Filesystem abstrahiert
- Mit JDBC wird die Datenbank angesprochen
- Enthält die Datenbank sowie das Filesystem

Der User Tier läuft auf dem Client (Webbrowser), Business und Data Tier können auf einem oder mehreren Servern verteilt sein. Der Business- und Data Tier von OLAT können wie folgt in Schichten dargestellt werden:



Programmier-Konzept

- Komponenten-basiert
- Event-basiert
- Wiederverwendbare GUI-Workflows / Business-Controllers
- Klare Trennung zwischen Code, Übersetzung, und HTML Templates

Rechte und Rollen-Konzept

Das Konzept von OLAT lehnt sich stark an das Policy - Konzept von Java an. Es gibt:

- Identities (User)
- Gruppen mit Identities darin (sogenannte Securitygroups, nicht zu verwechseln mit Businessgroups)
- Rechte
- Ressourcen / Objekte
- Policies

Codestruktur

Damit sich ein Entwickler von OLAT möglichst schnell zurechtfindet, folgt bei OLAT die Codestruktur einem festen Muster. Generell kann folgendes über die Codestruktur

ausgesagt werden:

- Unterhalb eines Source-Code Verzeichnisses finden sich alle notwendigen Dateien inklusive der HTML-Vorlagen, der i18n Dateien etc.
- Unter org.olat.core befindet sich der Framework- und Util Code
- Unter org.olat.* befindet sich der OLAT Learning Management System Code

Test Konzept

- Unit Tests
- Automated GUI Tests

Wichtigste eingesetzte Bibliotheken

Sämtliche eingesetzten Bibliotheken sind

Open-Source (LGPL, Apache BSD und ähnliche Lizenzen) und dürfen mit der OLAT-Distribution gebündelt werden. Alle eingesetzten Bibliotheken sind etabliert, stabil, und haben sich in der Praxis bewährt.

Bibliothek	Einsatzgebiet
Hibernate	Persistierung der Business-Objekte
SpringFramework	Konfiguration und Dependency Injection
Apache Commons	Utility Klassen
Apache Velocity	Template Engine zur Berechnung der HTML-Seiten
Xalan, Xerces, Jdom	XML und XSLT Processing
Lucene	Volltextsuche

3.Rahmenapplikation

Funktionen von OLAT

PERSÖNLICHER BEREICH

Die für den Benutzer selbst bestimmten Funktionen, fasst das OLAT im Home Bereich zusammen. Hier werden diese auf einer Portal-Seite zusammengefasst und organisiert. Im Einzelnen stehen den Usern folgende Features zur Verfügung:

-Profil

- optionales eintragen von persönlichen Information, persönliche systemweite Einstellungen setzen, persönliches Passwort ändern

-Benachrichtigungen

- Neuigkeiten/Änderungen abonniertes Ordner, Foren, Wikis werden hier angezeigt

-Kalender

- eintragen von privaten Terminen

-Bookmarks

- einstellen und löschen von Links um Kurse schnell zu erreichen

-Ordner

- privater und persönlicher Ordner als Ort für Dateien die anderen Usern zur Verfügung stehen sollen oder nur dem Benutzer

-Notizen

- private Notizen und Notizen aus Kursen

-Leistungsnachweise

- Überblick über erbrachte Leistungen aus Kursen die diese Option aktiviert haben

-Visitenkarte

Der Nutzer kann persönliche Informationen, sowie ein Foto öffentlich für andere Nutzer machen. Dies wird Homepage genannt und bietet ausserdem die möglich für andere dem nutzer eine Email zu schicken und auf seinen persönlichen Ordner zuzugreifen

-Chat

- der nutzer kann mit anderen auf der Plattform kommunizieren und seinen Status (verfügbar, abwesend...) einstellen

-Persönliche Kontakte

-separates Menü in dem Mitglieder einer Arbeitsgruppe aufgeführt werden, und darüber schnell erreichbar sind

SWP08-6-OLAT

Projektleiter: Johannes Kreidler ,Verantwortlicher :Nie Mao

21.April 2008

Die kollaborativen Funktionen des OLAT gliedern sich in kollabortive Bereiche und Werkzeuge. Dabei stehen bestimmte Werkzeuge innerhalb bestimmter Bereiche zur Verfügung:

Bereiche:

-Kurse

-Kurs in dem Kollaboration zum didaktischen Konzept gehört

-Gruppen

-Kollaboration erfolgt aus Eigeninitiative

-Lerngruppen: werden kursintern oder übergreifend verwendet,

Den kursbetreuern steht die Administration der Lerngruppen zur Verfügung

-Rechtegruppen: Kursautor wählt Personen die selektiv Zugang zu Kurswerkzeugen

haben

-Arbeitsgruppen: Können von allen registrierten Nutzern erstellt werden. Sie werden

vom

Ersteller administriert

Werkzeuge:

-Information

-eine einfache Information wird allen Teilnehmern angezeigt

-Gruppenkalender

-verwaltet Termine einer Gruppe

-Mitglieder

-Liste der Teilnehmer und/oder Betreuer

-E-Mail

-direktes kontaktieren von Teilnehmern/Betreuern

-Ordner

-Ort für Dateien die allen Teilnehmern zugänglich seien sollen

-Forum

-eröffnung von Themen zur asynchronen Kommunikation

-Wiki

-einfaches Content Management System zum kollaborativen erarbeiten von Wissen

-Chat

-Funktion zur Synchronen Kommunikation

ÖFFENTLICHER BEREICH:

Den öffentlichen Bereich des OLAT bildet der Katalog. Hier werden die für Benutzer freigegebenen

Lehrressourcen in einer thematisch gegliederten Struktur wiedergegeben. Der Katalog kann das Vorlesungsverzeichnis, eine Sitemap oder den Lehrplan abbilden. Außerdem kann man die Volltextsuche zum öffentlichen Bereich zählen. Sie sucht nach Lernressourcen, Inhalten aus Foren, Ordnern und Word/PDF/Excel Dateien.

Erweiterbarkeit von Olat Olat ist in einem Schichtenmodell aufgebaut. Insgesamt sind es drei verschiedene Schichten. Die "Olat Applikationsschicht", die "Core Services Schicht" und die "Integrationschicht".

In der "Olat Applikationsschicht" findet sich die GUI (Graphical User Interface) und die Seiten, die die verwendeten Applikationen anzeigen. Dabei sind alle Einzelkomponenten auf einem hohen Abstraktionsgrad gehalten um eine Wiederverwertung leicht möglich zu machen.

In der "Core Services Schicht" sind die Dienste zusammengefasst, die der Applikationsschicht zur Verfügung stehen.

In der "Integrationschicht" wird das ganze an das Dateisystem und die Datenbank angebunden.

Die Erweiterungen betreffen die "Applikationsschicht". Hier können diverse Applikationen unter Nutzung der "Core Services Schicht" hinzugefügt werden. Das bedeutet also, dass die Erweiterungen keinen direkten Zugriff auf die Integrationschicht haben sondern „nur“ über bereits bestehenden Service auf Datenbanken und Dateisystem zugreifen können.

Um eine Erweiterung möglichst einfach und Unabhängig vom Rest Olats zu machen bietet Olat spezielle Erweiterungspunkte an denen Erweiterungen ohne Änderungen des Original Quellcodes vorgenommen werden können. Es gibt insgesamt 5 Erweiterungspunkte:

1. org.olat.dispatcher.DispatcherAction

Diesem Erweiterungspunkt wird ein Mapper übergeben, d.h. ein Weg (bzw. eine Wegbeschreibung) der zu einem bestimmten Ort im Sourcecode führt

2. org.olat.home.HomeMainController

Damit lässt sich ein Link erstellen, der ein Event auslöst (welches noch separat erstellt werden muss oder welches Olat bereits zu Verfügung stellt).

3. org.olat.gui.components.Window

Hier lässt sich der Name einer CSS Datei hinterlegen (die natürlich auch erstellt werden muss). Dadurch lässt sich das Design der Grafischen Oberfläche anpassen.

4. org.olat.persistence.DB

Dieser Erweiterungspunkt ermöglicht den Zugriff auf zusätzliche oder bereits existierende Datenbanken. Dabei ist zu beachten, dass je nach Datenbank noch zusätzliche Dateien erstellt werden müssen (sql Scripts).

5.org.olat.gui.control.generic.dtabs.DTabs

Dieser Erweiterungspunkt ermöglicht das Hinzufügen einer neuen Seite wie z. B. "Home" oder "Gruppen". Diese Erweiterung erfordert noch einige zusätzliche Dateien, die im Detail und unter Verwendung von Olatobjekten das Aussehen und die Funktionsweise der Seite spezifiziert.

Die Erweiterungspunkte stellen umfangreiche Möglichkeiten einer Erweiterung zur Verfügung. Allerdings hängen die Möglichkeiten stark von der "Core Services Schicht" ab. Wenn diese keine flexiblen konfigurierbaren Dienste bereitstellt ist eine Erweiterung (an den Erweiterungspunkten) festgelegt auf die Einstellung dieser Dienste und damit stark eingeschränkt.

Verzeichnisstruktur Wenn man OLAT installiert, werden Verzeichnisse in Form eines Baumes angelegt. Normalerweise, zumindest auf einem Linux-System, ist die Wurzel das Verzeichnis „olat“, welches in /user/local zu finden ist. Direkte Nachfolgeverzeichnisse, also Söhne, von „olat“ sind beispielsweise „installs“, „bin“, „backups“, „olatlms“ und „olatdata“. Dabei befinden sich in „installs“ alle Software Pakete und in „bin“ nützliche Hilfsscripte für linux und macOSX. Unter „backups“ werden Sicherheitskopien der Datenbank gespeichert. „olatlms“ ist für die Installation von OLAT und enthält selbst die folgenden Unterverzeichnisse: „conf“ mit nötigen Konfigurationsvorlagen für apache und instant-messaging, „database“ mit den Vorlagen für die Datenbank, um Upgrades von früheren Versionen zu erzeugen, „doc“ mit der Dokumentation, „monitoring“ mit Vorlagen für MRTG und Setupdateien, sowie „webapp“ mit der OLAT Web Applikation.

Aus Performance- und Strukturgründen, werden einige Daten auf dem Filesystem gehalten. Um Eindeutigkeit zu garantieren, wird der Primärschlüssel eines Datenbank Eintrags als Name des Ordners verwendet. Zu dieser Sorte Daten gehört das bereits erwähnte Verzeichnis „olatdata“. Es enthält die Laufzeitdaten verteilt auf die Unterverzeichnisse: „calendars“ mit den Kalenderdateien, „logs“ mit den technischen Logdateien und den Logdateien von Apache und dem JK Connector, „monitoring“ mit den Statistikdaten der Systemerwachung, „recording“ mit den Daten des Recording/Test Frameworks, „system“ mit den installierten Upgrades, „tmp“ für temporäre Daten und „broot“. Letzteres ist wiederum gegliedert in die Verzeichnisse: „course“ für die Informationen

über die Kurse, „cts“ für die Gruppen, „forum“ für Dateianhänge zu Forumseinträgen, „homepages“ für die Benutzerhomepages, „homes“ für Informationen über die Benutzer, „repository“ für die Lernressourcen, zum Beispiel QTI-Tests oder Selbsttests, „scorm“ für die

SCORM-Laufzeitdaten, „search_index“ mit dem Index der Volltextsuche und einem weiteren Verzeichnis „tmp“, welches auch für die Ablage temporärer Daten benutzt wird.

Dabei sind "logs", "monitoring" und "recording" für den Betrieb einer Olatinstanz nicht notwendig sondern nur für Erweiterungen

Eine verbreitete Möglichkeit Webseiten dynamisch zu gestalten sind Servlets, welche Anwendungen für Server zur Verfügung stellen oder erweitern. Wenn der Client eine Anfrage stellt, wird diese vom Servlet entgegengenommen, bearbeitet und dann weiter an die Datenbank oder das Filesystem geleitet. Beide sind dem "Operating System" zuzuordnen und enthalten die eben beschriebenen Verzeichnisse. Genauso, nur in umgekehrter Richtung, wird mit den Antworten verfahren. Dadurch könne die Inhalte dynamisch manipuliert werden.

Rechtssystem von OLAT

OLAT ist ein rechtebasiertes System mit mehreren Hierarchiestufen. Dem Benutzer stehen Funktionen zur Verfügung zu welchen er in einem Kontext berechtigt ist. Es existieren im OLAT die folgenden Systemrollen:

Gäste Sie sind anonyme Benutzer mit eingeschränkten Rechten. Gäste können ihre Benutzeroberfläche nicht anpassen und weder Tests absolvieren noch Forumsbeiträge verfassen.

Benutzer Sie können ihre Benutzeroberfläche selber gestalten und einen Kurs als Teilnehmer starten.

Autoren Zusätzlich zu den Berechtigungen eines Benutzers können Autoren Lernressourcen herstellen, kopieren, archivieren, löschen etc.

Gruppenverwalter Mitglieder dieser Systemrolle können kursübergreifende Lern- und Rechtegruppen verwalten dies zusätzlich zu den Rechten eines Benutzers.

Benutzerverwalter Sie können neue Benutzer erstellen oder importieren und ihnen Rollen zuweisen.

Administratoren Berechtigt die Benutzer dieser Gruppe zu administrativen Tätigkeiten im gesamten OLAT-System. Beinhaltet ebenfalls die Rechte aller anderen Systemrollen.

Das Konzept von OLAT lehnt sich stark an das Policy-Konzept von Java an. Es gibt:

Identities (User)

Gruppen mit Identities darin (sogenannte Securitygroups, nicht zu verwechseln mit

Businessgroups

Rechte

Ressourcen / Objekte

Policies

User sind in einer oder mehreren Gruppen. Eine Policy ist die Kombination von einer Gruppe,

einem Recht, und einer Resource. Ein Benutzer hat dann ein gewisses Recht (z.B. lesen) auf einen gewisse Resource (z.B. RepositoryEntry), wenn er in mindestens einer Gruppe ist, die von einer Policy referenziert wird, welche das Rechte "lesen" auf die Resource hat, d.h. die Policy hat einen Fremdschlüssel auf die Gruppe, einen Fremdschlüssel auf die OLAT-Resource, und einen Text für das Recht

Die Rechte sind aktiv und positiv:

aktiv Alle Rechte von allen möglichen Gruppen, in denen ein User ist, werden zusammengezählt, und ergeben so als Summe die Rechte dieses Benutzers

positiv Alle Rechte beschreiben die Erlaubnis, etwas zu tun. Die Systemrollen (also z.B. Autor) sind demzufolge nichts anderes als Gruppen mit bestimmten Rechten.

Im OLAT existieren verschiedene Arten von Gruppen:

Lerngruppen werden in Kursen verwendet um Personen aus administrativen oder pädagogischen Gründen zu gruppieren, z.B. um Ihnen Lernmaterialien in einem geschützten Bereich zugänglich zu machen.

Rechtegruppen werden in Kursen verwendet um Personen zielt spezielle Rechte innerhalb eines Kurses zuzuteilen, z.B. um das Bewertungswerkzeug zu bedienen.

Arbeitsgruppen können von allen Benutzern selbst erstellt werden um z.B. gemeinsam an einem Projekt zu arbeiten oder Dokumente auszutauschen. Sie haben keinen Zusammenhang zu einem Kurs.

All diesen Gruppen liegt ein generisches Businessgroup Konzept zugrunde. Die drei spezifischen Gruppen sind lediglich konfigurierte Ausprägungen ,diesen generischen Gruppenimplementation. Alle Gruppen verfügen über so genannte kollaborative Werkzeuge. Diese Werkzeuge stehen allen Gruppen zur Verfügung um gemeinsam Diskussionen zu führen oder an Dokumenten zu arbeiten. Durch den modularen Ansatz ist es gut möglich neue kollaborative Werkzeuge in Gruppen zu integrieren. Gruppen, die zusammen in eine Kurs verwendet werden bilden einen so genannten Gruppenkontext. Ein Gruppenkontext hat immer eine Beziehung zu einem oder mehreren Kursen. So ist es möglich Kurs interne wie auch Kurs übergreifende Gruppen zu modellieren oder mit einem Gruppenkontext von einem Kurs zum nächsten zu ziehen. Eine BusinessGroup kann eine Lern- Rechte- oder Arbeitsgruppe sein und eine SecurityGroup enthält User und wird z.B. einer Policy angehängt für die Rechteverwaltung.