

Recherchebericht

Inhaltsverzeichnis

1. Glossar	2
1.1. Servlet	2
1.2. Erweiterungspunkt	2
1.3. Groupwaretools	2
1.4. AJAX	2
1.5. OLAT-Core	2
1.6. LMS/CMS	2
1.7. CVS	2
1.8. Framework	3
1.9. XML	3
1.10. YAML	3
2. Konzepte	3
2.1. Open Source	3
2.2. Java	3
2.3. Modularer Aufbau	4
2.4. Servlets	4
2.5. MVC	4
2.6. Client-Server-Modell	4
2.7. Schichtenkonzept	5
2.8. Rechtekonzept	5
2.9. Hibernate	5
3. Beschreibung der Applikation	5
3.1. Funktionalitäten	6
3.2. Rechtekonzept	6
3.3. Erweiterbarkeit	7
3.4. Verzeichnisstruktur	8
4. Quellen	9

1. Glossar

1.1 Servlet

Ein Servlet ist eine Java Klasse, deren Instanz innerhalb eines Java-Webrowsers arbeitet. Sie werden zur Erweiterung der Serverfunktionalität und für interaktive Web Applikationen verwendet. Der Begriff Servlet setzt sich aus den Worten Applet und Server zusammen. Man spricht also auch von einem serverseitigen Applet.

1.2 Erweiterungspunkt

Gerade im Bereich von Open Source bieten sich einfache Möglichkeiten an, das Produkt nach kundenspezifischen Bedürfnissen zu erweitern. Eine Schnittstelle für Entwickler, im Falle von OLAT, sind so genannte Erweiterungspunkte (Extension Points). Sie erlauben es OLAT um benötigte Funktionen zu erweitern, ohne dabei das Grundsystem (OLAT-Core) zu verändern.

1.3 Groupwaretools

In einem LMS sollen verschiedene Gruppen von Personen mit einander arbeiten. Um dies zeitlich- und räumlichunabhängig zu erreichen, kommen Groupwaretools zum Einsatz. Unter dem Begriff versteht man Anwendungen die der Kommunikation, Organisation und dem Austausch von Dateien zwischen mehreren Personen dienen. Dazu zählen unter anderem Foren, Chat, Kalender, Wiki, E-Mail oder Dokumentenablagen.

1.4 AJAX

AJAX („Asynchronous JavaScript and XML“) ist ein Konzept der asynchronen Datenübertragung zwischen Browser und Server. Das Ziel ist es, nur einen Teil der dargestellten Website zu verändern, ohne diese komplett neu zu laden. Die dynamische Darstellung der Inhalte wird durch JavaScript realisiert.

1.5 OLAT-Core

Der Quellcode von OLAT teilt sich in zwei Eclipse Projekte: olatcore und dem LMS olat3. Der OLAT-Core ist das Grundgerüst (Framework) auf dem das OLAT aufbaut. Für die Entwicklung an OLAT benötigt man nur olat3. Der „Core“ ist standardmäßig integriert.

1.6 LMS/CMS

Ein LMS (Learning Management System) ist ein Softwaresystem, dass der Bereitstellung von Lerninhalten und der Organisation von Lernvorgängen dient.

Ein CMS (Content Management System) ist ein Anwendungsprogramm, welches die gemeinschaftliche Erstellung und Bearbeitung von Inhalten ermöglicht und organisiert.

1.7 CVS

Das Concurrent Version System (CVS) ist ein Softwaresystem für die Versionsverwaltung von Dateien. Solch ein System kommt vor allem bei großen Open Source Projekten zum Einsatz und dient der zentralen Verwaltung von Quellcode. Die Dateien im Repository, also dem zentralen Speicher, können verändert werden, wobei alte Daten erhalten bleiben.

1.8 Framework

Als Framework bezeichnet man ein Entwurfsmuster, welches viele verknüpfte Klassen zusammenfügt und ein wieder verwendbares und erweiterbares Gerüst für Software eines bestimmten Typs ergibt. Das Framework definiert den Kontrollfluss und die Schnittstellen für die konkreten Klassen. Es ist jedoch nicht nur als eine reine Klassensammlung anzusehen. Vielmehr steuert es die in einem Programm eingebundenen Implementierungen.

1.9 XML

Zur einfachen Datenrepräsentation in Textform, werden häufig Auszeichnungssprachen verwendet. HTML und XML (Extensible Markup Language) sind zwei von vielen Beispielen. Das Ziel solcher Auszeichnungssprachen ist es, Daten sowohl für Menschen, als auch Maschinen leicht les- und änderbar zu machen.

1.10 YAML

OLAT setzt seit 2008 auf das YAML CSS Framework um die Darstellung von Tabellenbasierten zu einem CSS-basierten Layout zu wechseln. Diese strategische Entscheidung hatte das Ziel, das grafische Design komplett zu erneuern und forderte die Verwendung von CSS. Die Kooperation mit YAML kam zustande, um ein bereits bestehendes Framework zu verwenden. Darin sieht man den Vorteil der modularen Programmierung und damit auch der Verwendung von Frameworks. Das OLAT Team hat nun die Möglichkeit, sich auf die Entwicklung der Anwendung zu konzentrieren.

2. Konzepte

2.1 Open Source

Unter Open Source versteht man im Allgemeinen ein Produkt, welches ohne Lizenzgebühren auskommt und dessen Quellcode nicht nur einsehbar ist, sondern auch für die Nutzung – partiell oder im ganzen – in eigene Produkte eingebunden werden kann.

Das Konzept hinter Open Source ist jedoch nicht nur die lizenzfreie Nutzung, sondern auch die Arbeitsteilung in der Entwicklung. Es können viele verschiedene und vor allem unabhängige Entwickler an einem Produkt arbeiten. Während der Nutzung von Open Source Produkten ist man außerdem unabhängig von anderen Firmen.

2.2 Java

Java ist eine einfache objektorientierte Programmiersprache die von Sun Microsystems entwickelt wird und mittlerweile in der Version 6 frei erhältlich ist. Es ist eine leistungsfähige, architekturneutrale und portable Programmiersprache und läuft so auf allen gängigen Systemen, wie Linux, Mac OS X, Windows und Solaris.

Java-Programme werden in Bytecode übersetzt und in der Laufzeitumgebung mittels einer Virtual Machine (VM) interpretiert und bei Bedarf kompiliert. Die VM macht die Sprache sehr sicher, da es die ausgeführten Programme zur Laufzeit überwacht und so beispielsweise Zugriffsfehler verhindert.

2.3 Modularer Aufbau

Der Modulare Aufbau einer Anwendung ähnelt dem Konzept der objektorientierten Programmierung. Ziel ist es, ein Projekt in einzelne Module zu teilen und bereits geschriebene Module wieder verwenden zu können und so die Planungs-, Definitions- und Entwicklungsphase zu vereinfachen. An diesen Modulen kann nun unabhängig gearbeitet und getestet werden.

Änderungen innerhalb von Modulen sollten sich nicht auf andere Module auswirken. Die Kommunikation zwischen den Modulen erfolgt über genormte Schnittstellen.

2.4 Servlets

Servlets sind Java Klassen, deren Instanz innerhalb eines Java-Webserver Anfragen von Clients entgegennimmt und beantwortet. Sie sind bereits fester Bestandteil aller Java-EE-Anwendungsserver. Die Antwort eines Servlets kann sowohl aus Text (HTML, XML), als auch aus Binärdaten bestehen.

Ablauf einer Kommunikation mit einem Servlet:

1. Nutzer sendet ausgefülltes Formular im Browser ab.
2. Browser sendet Formulardaten an den Webserver.
3. Webserver übersetzt den Namen der Aktion in den Namen der Servlet-Klasse.
4. Webserver ruft die Methode request des Servlets auf und übergibt dabei die Anfragedaten als Parameter.
5. Die Methode request erzeugt dann eine Ausgabe, die der Webserver zurück an den Browser sendet.
6. Der Browser stellt die Antwort dar.

2.5 MVC

MVC (Model-View-Controller) ist ein Architekturmuster zur Gliederung der Softwaresysteme in drei Einheiten. Hierbei gibt es eine strikte Trennung von Anzeige (View), Steuerung (Controller) und Verarbeitung (Model).

Die Eingabe aus dem View wird an den Controller weiter gegeben, welcher die Eingabe interpretiert und an das Model weiter gibt. Nach der Verarbeitung werden die Ergebnisse dann wieder im View dargestellt.

Das Ziel ist ein flexibler Programmentwurf, der spätere Änderungen oder Erweiterungen erleichtert und die Wiederverwendbarkeit einzelner Komponenten erlaubt.

2.6 Client-Server-Modell

Das Client-Server-Modell dient der Aufgabenteilung in einem Netzwerk.

Der Server bietet Dienste an, während ein Client diese nutzen kann. Der Server verhält sich hierbei passiv und wartet auf Anfragen eines Clients, der aktiv die Dienste zur Informationsgewinnung nutzt. Bei der Übermittlung der Daten werden Protokolle eingesetzt, welche Syntax und Semantik der versendeten Daten festlegen.

Sinn einer solchen Aufgabenteilung ist es, dass der Client sich nicht um die Bearbeitung der Daten kümmern muss. Der Client ist also nur für die Eingabe und Ausgabe entscheidend. Änderungen an der Bearbeitung der Daten auf Seiten des Servers haben keinen Einfluss auf den Client.

2.7 Schichtenkonzept

Schichtenmodelle unterteilen ein System hierarchisch in verschiedene Stufen, wobei die Kommunikation zwischen den einzelnen Stufen über Schnittstellen erfolgt.

OLAT ist in drei Schichten unterteilt:

1. Applikationsschicht (OLAT application Layer)
Enthält Hauptfunktionen, die im OLAT direkt im GUI dargestellt werden können. Die Anwendung bietet so eine Möglichkeit, Funktionsbereiche mit eigenen Schnittstellen hinzuzufügen.
Wichtige Funktionsbereiche in OLAT sind beispielsweise der Bereich Home, das Lernressource Repository, das Kurssystem und die Gruppenverwaltung.
2. Serviceschicht (OLAT service Layer)
Stellt wesentliche Dienste und Komponenten kontextfrei zur Verfügung. Die hier definierten Dienste und Komponenten sind nicht für ein spezielles Einsatzgebiet erstellt, sondern sind meist Bereichsunabhängig.
Alle Module und Dienste greifen auf ein zentrales Benutzer- und Sicherheitsframework zu.
3. Integrationsschicht (Integration Layer)
Anbindung an das Dateisystem. Dienste werden in einer abstrahierten Form an die Serviceschicht weitergegeben.

2.8 Rechtekonzept

Jedem Benutzer wird eine Anzahl von Rechten/Funktionen, je nach Rolle des Nutzers, zur Verfügung gestellt. Die Rolle die ein Nutzer in einem Funktionsbereich einnehmen kann, reicht vom einfachen Benutzer bis hin zum Administrator. Anders als bei einem Administrator, haben andere Nutzergruppen nur einen eingeschränkten Zugriff.

2.9 Hibernate

Hibernate wird dazu verwendet, gewöhnliche Objekte mit Attributen und Methoden in relationalen Datenbanken zu speichern und aus entsprechenden Datensätzen wieder Objekte zu erzeugen. Hinzu kommt die Möglichkeit, auf Datenbanken zuzugreifen zu können, ohne diese Zugriffe explizit in SQL programmieren zu müssen.

Dies hat den Vorteil, dass die eigene Java Anwendung von der gewählten Datenbank unabhängig ist und man Entwicklungszeit spart, die man brauchen würde, um komplexe Objekte manuell über SQL und JDBC zu verwalten.

3. Beschreibung der zu studierenden Applikation

OLAT („Online Learning And Training“) ist ein webbasiertes Lernmanagementsystem, dass seit 1999 an der Universität Zürich entwickelt und eingesetzt wird. Das OpenSource Projekt ist in der Version 6.3 frei erhältlich.

Es bietet ein flexibles Kurssystem und unterstützt das kollaborative Arbeiten in Gruppen, mittels einer einfach zu bedienenden und mehrsprachigen Benutzeroberfläche, sowie verschiedenen Groupwaretools.

OLAT ist Servlet-basiert, wodurch man eine modulare, leicht anpassbare und modifizierbare Architektur erhält. Aufgrund der Verwendung von Erweiterungspunkten, bietet es Entwicklern eine Möglichkeit, OLAT mit eigens zugeschnittenen Funktionen und Komponenten zu erweitern. Da es fast ausschließlich in Java geschrieben und somit Plattformunabhängig ist, bedarf es nur geringe Anpassungen um es auf beliebige Systeme installieren und betreiben zu können.

3.1 Funktionalitäten

Das OLAT besteht aus drei großen Funktionsbereichen:

Privater Bereich

Der private Bereich dient dem Nutzer dazu, sein OLAT nach eigenen Vorstellungen und in eine für ihn produktive Umgebung zu gestalten.

- Profil, Einstellungen und Passwort anpassen
- Portal planen und organisieren
- Kalender verwalten und Notizen führen
- Bookmarks
- Persönlicher Ordner
- Persönliche Leistung im besuchten Kurs anschauen
- Einschreibung in Veranstaltungen

Kollaborativer Bereich

Der kollaborative Bereich dient der Kommunikation zwischen den Benutzern und dem Austausch, sowie der Präsentation von Informationen. Außerdem bietet er Funktionen für eine übersichtliche Organisation von Gruppen:

- Gruppenkalender
- Mitgliederlisten anschauen
- Benachrichtigungsservice via E-Mail
- Gruppenkommunikation via RSS, Chat, Wiki oder Forum
- Dateien in Ordner austauschen

Öffentlicher Bereich

Der öffentliche Bereich dient der Administration zur Verwaltung von Gruppen und einzelnen Nutzern.

- Durch Gruppenmanagement Kurse verwalten
- Durch Rechtemanagement den Zugang zu Werkzeugen verteilen
- Datenarchivierung
- Durch Bewertungswerkzeuge Punkte von Kursteilnehmern anschauen und editieren

3.2 Rechtekonzept

OLAT liegt ein rechtebasiertes System mit mehreren Hierarchiestufen zu Grunde. Ein Nutzer kann dabei folgende System-Rollen einnehmen:

Gast

- anonymer Benutzer mit eingeschränkten Rechten
- Keine Anpassung der Benutzeroberfläche möglich
- Keine Beteiligung an Tests oder das Schreiben von Forenbeiträgen

Benutzer

- Anpassen der Benutzeroberfläche
- Arbeitsgruppen erstellen
- Kurs als Teilnehmer starten

Autor

- Lernressourcen herstellen, kopieren, archivieren, löschen, etc.

Gruppenverwalter

- Lern- und Rechtegruppen verwalten

Benutzerverwalter

- Erstellen/Importieren von neuen Benutzern
- Zuweisen von Rollen

Administrator

- Administrative Tätigkeiten im gesamten OLAT-System

Anwendung finden diese System-Rollen vor allem in Kursen. Dort werden Gruppen eingesetzt um die Verteilung der Rollen zu organisieren.

Die **Lerngruppe** erlaubt zwei System-Rollen: Betreuer und Teilnehmer. Der Betreuer erhält automatisch auch die Administrationsfunktionen für diese Gruppe zur Verwaltung der Teilnehmer und Eigenschaften der Gruppe.

Die **Rechtegruppe** wird vom Kursautor administriert. Dieser kann einem Teilnehmer die Berechtigung zu bestimmten Funktionen und Werkzeugen erteilen, ohne die Person zum Kursadministrator zu ernennen.

Die **Arbeitsgruppe** kann von allen Systembenutzern (außer dem Gast) erstellt werden und hat keinen Bezug auf einen Kurs. Einen Betreuer gibt es in dieser Gruppe nicht. Stattdessen dient sie nur dem kollaborativen Arbeiten innerhalb eines Projektes.

3.3 Erweiterbarkeit

Die Extension-Points erlauben die Modifizierung von OLAT, ohne Zugriff auf den Quellcode (OLAT-Core). So lässt sich OLAT nach den Bedürfnissen des Kunden erweitern. Das Grundsystem bleibt dabei unangetastet. Dies erleichtert zudem die Installation von Updates.

Die Erweiterungen werden als .jar Paket in den Ordner WEB-INF/lib abgelegt und in der olat_extensions.xml definiert.

Folgende Extension-Points sind derzeit definiert:

Interface: org.olat.extensions.globalmapper
 Erweiterungspunkt: org.olat.dispatcher.DispatcherAction
 Klasse: org.olat.dispatcher.DispatcherAction
 Beschreibung: Die Erweiterung kann den verwendeten Mapper einholen und erhält den Pfad mit dem er assoziiert wurde.

Interface: org.olat.extensions.action.ActionExtension
 Erweiterungspunkt: org.olat.home.HomeMainController
 Klasse: -
 Beschreibung: Bietet einen Link mit Text und Beschreibung und definiert die Aktion hinter dem Link.

Interface: org.olat.extensions.css.CSSIncluder
 Erweiterungspunkt: org.olat.gui.components.Window
 Klasse: org.olat.gui.css.CSSGenerator
 Beschreibung: Neue CSS Stylesheets.

Interface: org.olat.extensions.hibernate.HibernateConfigurator
 Erweiterungspunkt: org.olat.persistence.DB
 Klasse: org.olat.persistence.DB
 Beschreibung: Hinzufügen zusätzlicher hibernate mappings.

Interface: org.olat.extensions.sitescreator.SitesCreator
 Erweiterungspunkt: org.olat.gui.control.generic.dtabs.DTabs
 Klasse: org.olat.FullChiefController
 Beschreibung: Zum Anlegen neuer Seiten.

Interface: org.olat.login.SupportsAfterLoginInterceptor
 Erweiterungspunkt: org.olat.login.AfterLoginInterceptorController
 Klasse: org.olat.user.UserModule
 Beschreibung: Für Module, die direkt nach dem Login arbeiten.

3.4 Verzeichnisstruktur

Für den Betrieb einer OLAT Instanz notwendige Verzeichnisse:

/bin	Hilfsskripte
/cluster	Skripte zum Clustering des OLAT
/conf	Konfigurationsvorlagen
/database	Datenbankvorlagen für SQL
/monitoring	Statistik der Trafficüberwachung
/olatdata	OLAT Laufzeitdaten
/bcroot	
/course	Kursdaten
/cts	Gruppenordner
/homepages	Websites aller Nutzer
/homes	Persönlicher Ordner aller Nutzer
/repository	Angelegte Lernressourcen
/tmp	Temporäre Daten
/calenders	Kalenderdaten
/customizing	Anpassungen
/logs	Log Dateien
/monitoring	Statistik der Systemüberwachung
/system	Installierte Upgrades
/tmp	Temporäre Daten
/interfaces	Vorlagen für den grafischen OLAT Aufbau
/temp	Temporäre Daten
/webapp	Webapplikationen
/examples	Kursbeispiele und OLAT Demo
/help	Hilfe Dateien
/static	Layoutdateien
/WEB-INF	Konfigurationsdateien von OLAT (z.B. olat_extensions.xml)
/classes	Java-Klassendateien
/lib	Verwendete Java-Bibliotheken

Für die Entwicklung an OLAT notwendige Verzeichnisse:

/doc	OLAT Dokumentation
/htdocs	Anpassbare Fehlerberichtseiten
/webapp	
/WEB-INF	Konfigurationsdateien von OLAT (z.B. olat_extensions.xml)
/patchesSrc	Quelldateien für Patches
/src	Quelldateien des OLAT

Anhand der Verzeichnisstruktur und den Vergleich zu einfachen Servlet-Anwendungen, erkennt man, dass OLAT eine sehr umfangreiche und komplexe Servlet-Anwendung ist. Außerdem werden eine ganze Reihe von XML Dateien für die Konfiguration und den Betrieb verwendet. Einfache Servlets benötigen für den Betrieb oft nur eine einzige Konfigurationsdatei und jeweils einen Ordner für die Java-Klassen und die benötigten Bibliotheken.

4. Quellen

www.wikipedia.org

http://www.olat.org/website/en/html/unit_development.html

http://www.olat.org/website/en/html/download_documentation.html

<http://java.sun.com/products/servlet/>

http://www.jboss.com/pdf/HibernateBrochure-03_07.pdf

<http://blog.yaml.de/post/78/learning-management-software-olat-and-yaml-a-success-story/>

<http://pcai042.informatik.uni-leipzig.de/swp/SWP-09/index.html>