

## SWT-Praktikum 2008 Aufgabenblatt 2

### 1. Begriffe

#### **WebDAV:**

WebDAV steht für "Web-based Distributed Authoring and Versioning". Es ist ein offener Standard zur Bereitstellung von Dateien im Internet, der es den Nutzern ermöglicht, auf ihre Daten wie auf eine Online-Festplatte zuzugreifen. WebDAV stellt eine Erweiterung des HTTP-Protokolls dar und garantiert einen einfachen und schnellen Datenaustausch. Bestes Beispiel ist das GMX MediaCenter.

#### **OLAT:**

OLAT steht für "Online Learning And Training". Es ist eine Open-Source-Lernplattform, die seit 1999 an der Universität Zürich entwickelt wird. Diese dient dazu Vorlesungen, Übungen und Tests online anzubieten und den Nutzern eine Möglichkeit zu geben, sich einfach und schnell in sogenannten Lerngruppen auszutauschen. Seit der Version 3.x basiert OLAT komplett auf Java und zeichnet sich durch eine sehr modulare, anpassbare und erweiterbare Architektur aus. Im April 2008 wird bereits die Version 6.0 veröffentlicht.

#### **Velocity:**

Velocity ist ein Open-Source-Projekt und entstammt der Jakarta-Projektgruppe von Apache. Das Tool wird als eine Java-basierte Schablonen-Maschine bezeichnet, die es ermöglicht, Java-Objekte in verschiedene andere Formate, wie Text, XML, SQL, Post Script oder HTML einzubinden. Damit eignet es sich besonders gut zum Erstellen von dynamischen Web-Seiten. Dadurch, dass es streng dem Modell-View-Controller-Prinzip folgt und einfach zu erlernen ist, erlaubt Velocity das gleichzeitige Arbeiten an funktionalem Code und einer ansprechenden Optik.

OLAT benutzt Velocity aufbauend auf der Präsentationslogik, so dass der Client die erhaltenen Daten nur noch darstellen muss.

#### **Ajax:**

Ajax steht für „Asynchronous JavaScript and XML“. Es bezeichnet ein Konzept der asynchronen Datenübertragung zwischen einem Server und dem Browser und ist eine Schlüsseltechnik zur Umsetzung des Web 2.0. Es ermöglicht, innerhalb einer HTML-Seite eine HTTP-Anfrage durchzuführen, ohne die Seite komplett neu laden zu müssen. Das heißt, dass bei Bedarf sukzessive gewisse Teile der Seite oder auch Nutzerdaten nachgeladen werden können. OLAT nutzt Ajax in seinem GUI-Framework, um nur die relevanten Komponenten der Oberfläche neu zu laden und so die Performance beim Seitenaufbau zu verbessern.

#### **Hibernate:**

Hibernate kommt aus dem Englischen und heißt übersetzt „Winterschlaf halten“. Es bezeichnet ein Open-Source-Persistenz-Framework für Java. Das Framework ermöglicht es, den Zustand eines Objekts mit Attributen und Methoden in einer relationalen Datenbank zu speichern und aus entsprechenden Datensätzen wiederum Objekte zu erzeugen. Die Beziehungen zwischen den Objekten werden auf entsprechende Datenbank-Relationen abgebildet. Dabei ist die spätere Anwendung unabhängig von der zugrundeliegenden Datenbank, da Hibernate mit fast allen aktuellen relationalen Datenbanksystemen kompatibel ist.

### **LMS/CMS:**

Ein LMS (Learning Management System) ist ein Softwaresystem, das 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 Texten und Multimedia-Dokumenten ermöglicht und organisiert.

### **Servlet/JSP:**

Als Servlets bezeichnet man Java-Klassen, deren Instanzen innerhalb eines J2EE Application Servers Anfragen von Clients entgegen nehmen und beantworten. Der Inhalt der Antworten kann dabei dynamisch, also im Moment der Anfrage, erstellt werden und muss nicht bereits statisch für den Webserver verfügbar sein. Servlets stellen somit das Java-Pendant zu CGI-Skripten oder anderen Konzepten dar, mit denen dynamisch Web-Inhalte erstellt werden können.

JSP (Java Server Pages) ist eine Technologie die zur Erzeugung von HTML- und XML-Ausgaben eines Webserver dient. Durch JSP ist es möglich, Java-Code oder spezielle JSP-Aktionen in statischen Inhalt einzubetten. Dadurch ist es möglich, die Logik unabhängig vom Design zu implementieren.

### **Apache/Tomcat:**

Apache Tomcat stellt eine Umgebung zur Ausführung von Java-Code auf Webservern bereit. Dabei handelt es sich um einen in Java geschriebenen Servlet-Container, der mithilfe des JSP-Compilers Jasper auch Java Server Pages in Servlets übersetzen und ausführen kann. Dazu kommt ein kompletter HTTP-Server. Der HTTP-Server des Tomcat wird vor allem zur Entwicklung eingesetzt, während in Produktion zumeist ein Apache Web-Server vor den Tomcat geschaltet wird.

### **SCORM/IMS Content Packaging/IMS QTI:**

SCORM (Sharable Content Object Reference Model) umfasst eine Sammlung von Standards und Spezifikationen, um einfache Austauschbarkeit, einen allgemeinen Zugriff und Wiederverwendbarkeit in verschiedenen Umgebungen von web-basierten Lerninhalten zu ermöglichen. Es ist das Referenz-Modell für austauschbare elektronische Lerninhalte der Advanced Distributed Learning Initiative.

IMS Content Packaging und IMS Question & Test Interoperability (QTI) sind weitere E-Learning Standards, die unter anderem die Struktur von Online-Lernmaterialien (IMS Content Packaging) und standardisierte Datenformate für diverse Online-Tests festlegen.

### **Shibboleth:**

Shibboleth ist ein Verfahren zur verteilten Authentifizierung und Autorisierung für Webanwendungen und Webservices. Das Konzept von Shibboleth sieht vor, dass sich der Benutzer nur einmal authentifizieren muss, um ortsunabhängig auf Dienste oder lizenzierte Inhalte verschiedener Anbieter zugreifen zu können.

## 2. Konzepte

### **Open Source**

OLAT ist eine Open Source Lernplattform und steht unter einer von der Open Source Initiative (OSI) anerkannten Lizenz (Apache 2.0) zur Verfügung. Die Verfügbarkeit des Quelltextes ist dabei fast deckungsgleich mit sogenannter freier Software ist, d.h. der Quelltext ist auch offen für Bearbeitung und Weiterverbreitung.

### **OLAT Schichten**

OLAT kann in drei Schichten unterteilt werden.

- **Applikationschicht:**

OLAT besteht aus verschiedenen Hauptfunktionen, die im Graphical User Interface (GUI) oft auch direkt als sogenannte Sites sichtbar sind. Sites können per Konfiguration hinzugeschaltet werden. So lässt sich OLAT einfach um ganze Funktionsbereiche erweitern. Die Funktionsbereiche sind in der Regel in sich geschlossen und bieten Schnittstellen für den gegenseitigen Gebrauch von Funktionen an. Die wichtigsten Funktionsbereiche sind das Home, das Lernressourcen Repository, das Kurssystem, das IMS-QTI Testsystem, die Gruppenverwaltung und Benutzer- wie Systemadministration.

- **Serviceschicht:**

Die Serviceschicht stellt der OLAT Applikation wesentliche Dienste und Komponenten kontextfrei zur Verfügung, das bedeutet, dass die Implementierung der einzelnen Dienste und Komponenten nicht für ein spezielles Einsatzgebiet sondern generisch umgesetzt sind. Alle Module und Dienste greifen auf ein zentrales Benutzer- und Sicherheitsframework zu.

- **Integrationschicht:**

Die Anbindung an das Dateisystem, das Datensystem oder an andere Dienste erfolgt in der Integrationsschicht. In der Regel wird hier ein Dienst in einer abstrahierten Form an die Serviceschicht weitergegeben, so dass die konkrete Implementierung ohne Änderung an der Applikation oder dem Service verändert werden kann.

### **Modularisierung**

Durch die Zerlegung eines umfangreichen Softwareprojektes in viele kleinere Einheiten, sog. Module, wird die Austauschbarkeit einzelner Bausteine ermöglicht. Dadurch werden folgende Vorteile erlangt:

- Einfache Aufteilung der Entwicklung auf verschiedene Gruppen
- Wiederverwendung schon vorhandener Module
- Vereinfachung von Tests
- Vereinfachung von Planungs-, Definitions- und Entwurfsphase

### **JAVA**

Java ist eine objektorientierte Programmiersprache, welche von Sun Microsystems entwickelt wurde. Java wurde mit dem Ziel entwickelt, plattformunabhängig zu sein. Dies wird realisiert, indem der Quellcode zunächst in Bytecode übersetzt wird. Dieser wird nun von der für das jeweilige System verfügbaren Java-Laufzeitumgebung ausgeführt. Der

wichtigste Bestandteil dieser Umgebung ist die Java Virtual Machine (Java-VM), welche den Bytecode interpretiert.

OLAT setzt auf der Java 2 Enterprise Edition (J2EE) auf und basiert auf einer Servlet-Architektur.

### Client/Server-Konzept

Das Client-Server-Modell ist eine Möglichkeit Aufgaben im Netzwerk zu verteilen. Dabei ist der Server ein Programm, welches einen oder mehrere Dienste anbietet, welche von einem anderen Programm, dem Client, genutzt werden können. Dabei verhält sich der Server passiv und wartet auf Anfragen, während der Client aktiv Dienste des Servers abfragt um benötigte Informationen zu erhalten. Die Regeln dieser Kommunikation, wie Format und Art der Informationen, werden durch Protokolle festgelegt.

OLAT nutzt dieses Konzept.

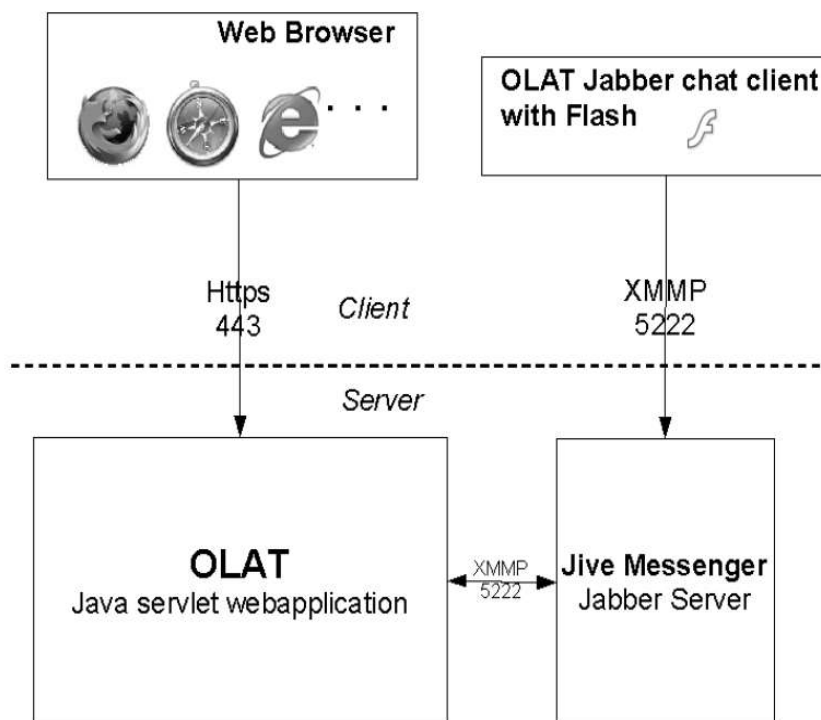


Abbildung 1: Client-Server-Konzept von OLAT

Hier ist der Client der Browser bzw. ein Jabber-Client. Mittels Http oder Https resp. XMMP werden Dienste auf dem Server abgefragt, wobei der Browser generierte HTML-Seiten bekommt, der Jabber-Client Textnachrichten.

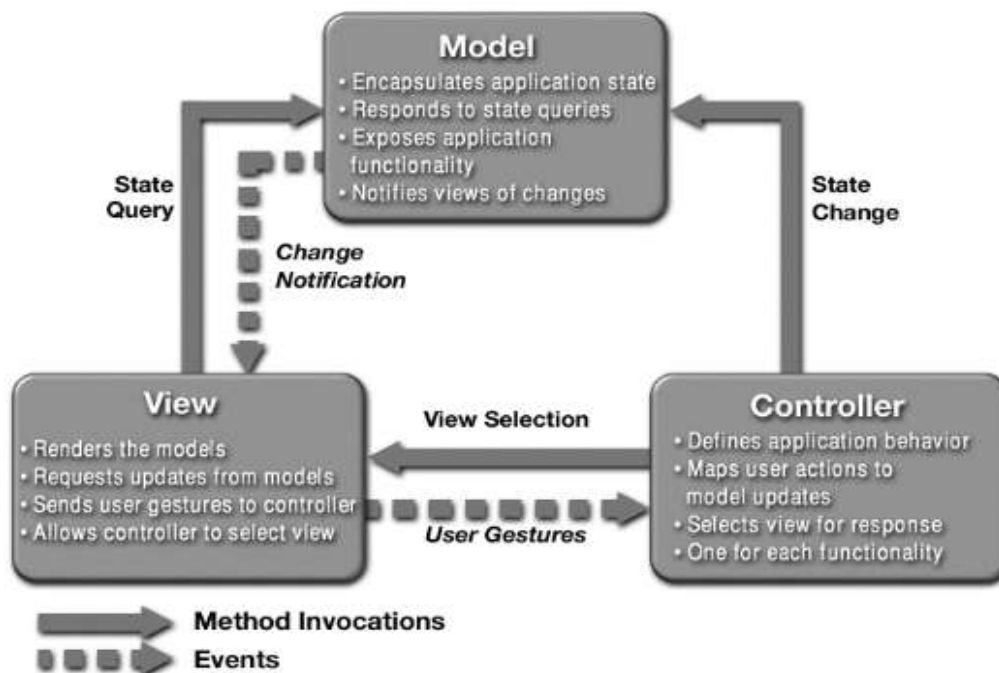
Durch dieses Konzept wird eine hohe Datenunabhängigkeit erreicht. Der Client muss sich nicht um Berechnungen oder Bearbeitung von Daten kümmern, er bekommt schon für ihn zugeschnittene Daten vom Server.

## MVC-Architektur

MVC steht für Modell-View-Controller. Bei diesem Konzept setzt man auf eine strenge Trennung zwischen Datenmanipulation (Modell), Anzeige bzw. Ausgabe (View) und Eingabe (Controller).

- **Modell:** Das Modell beinhaltet sämtliche Daten. Diese werden mittels Schnittstellen zur Verfügung gestellt. Außerdem beinhaltet das Modell diverse Methoden, die zur Manipulation dieser Daten erforderlich sind.
- **View:** Der View ist für die Darstellung der Daten zuständig, welche vom Modell angefordert werden, ist aber nicht für die Berechnung dieser zuständig.
- **Controller:** Der Controller nimmt Benutzereingaben im View auf und ermittelt deren Bedeutung. Anschließend werden eventuelle Daten dem Modell zur Berechnung übergeben bzw. das Modell aufgefordert Berechnungen auszuführen.

Dieses Konzept ermöglicht ein hohes Maß an Übersicht und Wiederverwendbarkeit sowie eine leichte Erweiterung. So müssen z.B. bei einer Überarbeitung des GUI keine Änderungen am Controller bzw. Modell vorgenommen werden.



### 3. Beschreibung der Rahmenapplikation - OLAT

OLAT ist ein an der Universität Zürich zum campusweiten Einsatz im heterogenen Hochschul Umfeld entwickeltes, webbasiertes Learning Management System (LMS), welches verschiedene E-Learning Standards wie SCORM oder das IMS QTI-Format unterstützt und somit eine hohe Interoperabilität mit anderen LMS ermöglicht. Gleichzeitig besitzt OLAT Funktionen eines Content Management Systems, die unter anderem das gemeinsame, ortsunabhängige Arbeiten ermöglichen.

OLAT beinhaltet ein flexibles Kurssystem, welches eine hierarchische Kursstruktur nutzt um beliebige didaktische Konzepte abzubilden. Weiter werden sowohl kursinterne als auch kursübergreifende Funktionen des Campusmanagements, wie die Verwaltung von Lernressourcen (Katalogisierung, Editoren für Test, etc.), bereitgestellt. Kursen werden dabei verschiedene Bausteine (etwa PDF-Dokumente oder Tests) zugeordnet, die einer beliebigen hierarchischen Struktur unterliegen dürfen. Gleichzeitig können diese Bausteine noch mit Zugangsbeschränkungen versehen werden. So ist es möglich einen Test nur für einen festgelegten Zeitraum zu Bearbeitung freizugeben oder das Anzeigen von Folgeseiten nur zu erlauben, wenn ein Test erfolgreich absolviert wurde. Weiter ist es natürlich auch möglich die Kursbausteine nur für gezielte Gruppen freizugeben. In OLAT gibt es Lerngruppen, Arbeitsgruppen und Rechtegruppen. Lerngruppen dienen dem gemeinsamen Arbeiten, wobei bei den Mitgliedern der Gruppe zwischen Betreuern und Teilnehmern unterschieden wird und den Betreuern automatisch alle Funktionen zur Gruppenverwaltung zur Verfügung stehen. Arbeitsgruppen können dabei von jedem registrierten Benutzer angelegt werden und dienen dem kollaborativen Arbeiten ohne Betreuer. Die Rechtegruppen dienen schließlich dazu, ausgewählten Personen selektiven Zugang zu Kurswerkzeugen zu geben ohne diesen die umfassenden Rechte eines Kursadministrators einräumen zu müssen. Um eine gezielte Zuweisung von Rechten zu gewährleisten bedient sich OLAT des folgenden Rechtskonzepts.

Nach außen hin werden die Rechte über Rollen vergeben. Es gibt dabei folgende Rollen:

<b>Rolle</b>	<b>Wichtigste Rechte</b>
Gast	Anonyme Benutzer mit stark eingeschränkten Rechten
Benutzer	Benutzeroberfläche kann angepasst werden In Kursen darf als Teilnehmer gestartet werden
Autor	Lernressourcen herstellen und editieren
Gruppenverwalter	Verwalten von kursübergreifenden Lern- und Rechtegruppen
Benutzerverwalter	Benutzer erstellen/löschen sowie Benutzern Rollen zuweisen
Administrator	Berechtigt zu administrativen Tätigkeiten im gesamten System

Über eine Rolle darf man dann ggf. weitere Rechte vergeben, so darf etwa ein Autor eine Lernressource erzeugen und dann eine Rechtegruppen erzeugen deren Mitglieder die Ressource editieren dürfen.

Systemintern wird dabei ein Policy-Konzept verfolgt. Es gibt Identitäten (Benutzer), Gruppen mit Identitäten darin (so genannte Securitygroups), Rechte, Ressourcen und Policies. Identitäten gehören dabei zu einer oder mehreren Gruppen. Eine Policy besteht dann aus einer Gruppe, einer Ressource und einem Recht und die Policy bedeutet dann, dass jede Identität dieser Gruppe das referenzierte Recht auf die referenzierte Ressource besitzt. Rechte sind dabei stets „positiv“, d.h. sie erlauben etwas zu tun. Auch die Systemrollen sind somit nichts anderes als diese Securitygroups, wie etwa die Gruppe der Gruppenverwalter. Im Gegensatz dazu steht das Businessgroup-Konzept. Hierbei liegt eine Schablone für

Gruppenimplementation vor und die Arbeits-, Lern- und Rechtegruppen sind nichts anderes als konfigurierte Ausprägungen dieser generischen Gruppenimplementation. Sie sind also unabhängig von der Implementation der Securitygroups.

OLAT wurde komplett in der Programmiersprache Java umgesetzt, so dass der Betrieb unter verschiedenen Betriebssystemen, wie etwa Windows, Linux oder MacOSX, problemlos möglich ist. Diese weit gefächerte Nutzbarkeit von OLAT wurde noch weiter unterstützt, indem bei der Umsetzung auf ein virtuelles Filesystem und eine Hibernate-Integrationsschicht für Datenbanken gesetzt wurde. Damit ist es möglich verschiedene Datenbankmanagementsysteme einzusetzen, wie etwa MySQL oder Oracle, und diese auch bei Bedarf mit wenig Aufwand durch leistungsfähigere zu ersetzen.

Die Architektur von OLAT ist dabei hochgradig modular und leicht zu erweitern. Verschiedene sogenannte „Extension-Points“ erlauben es, OLAT nach kundenspezifischen Bedürfnissen anzupassen, ohne Änderungen am Grundsystem vornehmen zu müssen. Um diese Extensions zu Olat hinzuzufügen muss man eine Zeile in der `olat_extensions.xml` einfügen und seine Extension als `.jar` Paket in den Ordner `WEB-INF/lib` einfügen. Um die Programmierung von Extensions zu vereinfachen gibt es fünf Extension-Points um folgende Funktionen zu gewährleisten:

- Der Pfad im Olat ist bekannt, mit der die Extension assoziiert ist.
- Es können Links definiert werden und deren Funktionen.
- Der Extension kann eine CSS-Datei zugewiesen werden.
- Die Extension kann zusätzliche Hibernate Pfade anlegen.
- Es können neue Olat-Seiten erstellt werden.

Die unter Linux standardmäßig unter `/usr/local/olat/` befindliche OLAT-Installation beinhaltet im Allgemeinen die Verzeichnisse `bin` (mit Hilfsskripten), `conf` (mit Konfigurationsvorlagen), `database` (mit Datenbankvorlagen), `doc` (mit der Dokumentation), `monitoring` (mit Vorlagen für das externe MRTG Monitoring) und `webapp` (mit der eigentlichen OLAT Web-Applikation). Letzteres Verzeichnis enthält neben Bibliotheken (im Unterverzeichnis `WEB-INF/lib`) und dem Quellcode (im Unterverzeichnis `WEB-INF/src`) auch die kompilierten Klassen (meist im Unterverzeichnis `WEB-INF/classes`). Diese Verzeichnisse werden für die Weiterentwicklung von OLAT benötigt.

Für den Betrieb einer OLAT-Instanz werden die Verzeichnisse unter `olatdata` benötigt. Das Verzeichnis `bcroot` enthält dabei Laufzeitdaten beispielsweise zu Kursen (`course`), Gruppenordnern (`cts`), Benutzerordner (`homes`), etc. Im Verzeichnis `oldadata` finden sich neben dem `bcroot`-Verzeichnis u.a. noch Dateien zu Kalendern (`calendars`), Logs (`logs`) und der Systemüberwachung (`monitoring`).