
Recherchebericht OLAT

Gruppe SWP08-7

Inhaltsverzeichnis

1	Begriffe	2
1.1	OLAT	2
1.2	LMS	2
1.3	Java	2
1.4	Shibboleth	2
1.5	MVC	2
1.6	Servlets	2
1.7	Extension / Extension-Points	2
1.8	Portlets / Portal	3
1.9	Gruppen / Gruppentypen	3
1.10	XML	3
2	Konzepte	3
2.1	Schichtenmodelle	3
2.2	Servertechnologien	4
2.2.1	Client-Server-Modell	4
2.2.2	Java / JEE	4
2.2.3	Servlets	4
2.2.4	Tomcat	5
2.3	Standards	5
2.3.1	IMS	5
	IMS Content Packaging	5
	IMS Question Test Interoperability	5
2.3.2	SCORM	5
2.4	Hibernate	6
2.5	Open Source	6
2.5.1	Extreme Programming	6
3	Beschreibung der zu studierenden Applikation	6
3.1	Erweiterbarkeit	6
3.2	Verzeichnisstruktur von OLAT	7
3.3	Rechtesystem von OLAT	8
3.4	Funktionalitäten	10
3.4.1	Privater Bereich	10
3.4.2	kollaborativer Bereich	11
3.4.3	öffentlicher Bereich	11
3.5	Modifikationen der OLAT-Installation	11
3.5.1	Entfernen des Tabs "Lernressourcen"	11
3.5.2	Entfernen der Demokurse	11

Quellen:

- de.wikipedia.org - die freie Enzyklopädie
- www.frentix.com - OLAT Dokumentation
- www.olat.org - Quellen

1 Begriffe

1.1 OLAT

OLAT steht für Online Learning And Training und ist ein Open Source Lernmanagement-System (LMS), das an der Universität Zürich mit dem Ziel entwickelt wurde, Vorlesungen und Übungen über das Internet anbieten zu können, wobei der Fokus auf dem Aufbau und der Durchführung von Lehrveranstaltungen liegt.

1.2 LMS

Ein Lernmanagementsystem (LMS), auch Learning Management System oder Lernplattform genannt, bildet in der Regel den technischen Kern einer komplexen webbasierten E-Learning Infrastruktur. Es handelt sich dabei um eine auf einem Webserver installierte Software, die das Bereitstellen und die Nutzung von Lerninhalten unterstützt und Instrumente für das kooperative Arbeiten und eine Nutzerverwaltung bereitstellt.

1.3 Java

Java ist eine objektorientierte Programmiersprache. Java benötigt eine Laufzeitumgebung (Java Virtual Machine oder JVM), die vom Benutzer installiert werden muss. Somit kann in *Java* plattformunabhängig programmiert werden.

Zunächst werden vom Programmierer Quelldateien (Source-Code) erstellt. Diese werden von einem Java-Compiler in Bytecode übersetzt und in einer class-Datei gespeichert. Diese class-Datei wird dann von der Java - Laufzeitumgebung ausgeführt.

1.4 Shibboleth

Shibboleth ist ein webbasiertes System zur Authentifizierung und Autorisierung für Webanwendungen und Webservices.

1.5 MVC

Model-View-Controller (MVC) bezeichnet ein Architekturmuster zur Aufteilung von Softwaresystemen in die drei Einheiten: Datenmodell (Model), Präsentation (View) und Programmsteuerung (Controller). Ziel des Musters ist ein flexibles Programmdesign, das u. a. eine spätere Änderung oder Erweiterung erleichtern und eine Wiederverwendbarkeit der einzelnen Komponenten ermöglichen soll.

1.6 Servlets

Als *Servlets* bezeichnet man Java-Klassen, deren Instanzen innerhalb eines JEE Applicationsservers Anfragen von Clients entgegen nehmen und beantworten.

1.7 Extension / Extension-Points

Verschiedene so genannte "*Extension-Points*" erlauben das *LMS* nach kundenspezifischen Bedürfnissen zu erweitern, ohne dabei Änderungen am Grundsystem vornehmen zu können.

Dies erleichtert die Installation von Updates und schützt somit getätigte Investitionen. Beispiele für solche Erweiterungen wären ein zusätzlicher Navigationspunkt in der Hauptnavigation oder ein neuer Kursbaustein. Die Erweiterungen können Layoutelemente und Übersetzungen mit sich bringen oder auf bestehende zurückgreifen. Eine *Extension* soll eine Erweiterung der vorliegenden Rahmenapplikation *OLAT* beschreiben, mittels derer neue Funktionalitäten hinzugefügt werden können.

1.8 Portlets / Portal

Portlets sind beliebig kombinierbare Komponenten einer Benutzeroberfläche, die von einem *Portalserver* angezeigt und verwaltet werden. Sie erzeugen Fragmente von HTML-Code und fügen sich in einer Portalseite ein. Typischerweise besteht eine *Portalseite* aus vielen, nicht-überlappenden Portlet-Fenstern (“Kacheln”), in denen jeweils ein *Portlet* ausgeführt wird. Beispiele für Portlets sind E-Mail, Wetterbericht, Diskussionsforen oder Nachrichten.

1.9 Gruppen / Gruppentypen

Gruppentypen:

Lerngruppen Kursübergreifend oder Kursintern Zusammenfassen von Lerngruppen in Lernbereiche
Zwei Rollen: Teilnehmer und Betreuer

Rechtegruppen Kursübergreifend oder Kursintern Zuweisen von Rechten innerhalb eines Kurses

Arbeitsgruppen Von jedem Benutzer erstellbar für private Zwecke
Zwei Rollen: Teilnehmer und Administratoren

1.10 XML

XML (Extensible Markup Language) ist eine Auszeichnungssprache zur Beschreibung und Strukturierung von Daten in einer Textdatei. Die offiziellen Spezifikationen der Sprache werden vom World Wide Web Consortium (W3C) herausgegeben und definieren *XML* als eine Metasprache durch die weitere anwendungsspezifische Sprachen erstellt werden können (z.B. RSS, SVG und XHTML). Der Inhalt einer *XML-Datei* wird hierarchisch strukturiert. Die einzelnen Formatabschnitte werden durch Tags definiert, wie dies etwa von HTML (selbst eine Auszeichnungssprache) bekannt ist.

2 Konzepte

2.1 Schichtenmodelle

Schichtenmodelle unterteilen ein beliebiges System hierarchisch in verschiedene Stufen wobei das Abstraktionsniveau in vertikaler Richtung steigt. Für die Kommunikation zwischen verschiedenen Stufen, sowie auch innerhalb einer Stufe werden Schnittstellen definiert. Mit Hilfe eines solchen Modells lassen sich neben Abstraktion von Systemspezifika auch gut

Rechteverwaltungen einrichten. Elemente die sich auf der selben Stufe befinden sind im allgemeinen gleichgestellt und ihnen stehen die selben Schnittstellen zur Verfügung.

Vgl. [Abbildung 1] - OLAT-Schichtenmodell

2.2 Servertechnologien

2.2.1 Client-Server-Modell

Das *Client-Server-Modell* beschreibt die Verteilung von Aufgaben innerhalb eines Netzwerkes. Hierbei bietet ein *Server* einem *Client* einen Dienst an, wobei der Server bestimmt welche Daten ausgetauscht werden. Hierbei verhält sich der Server passiv, wartet auf Anforderung vom Client (aktiv).

Beispiele für Server sind *Mail-Server*, *Web-Server*, *Applikations-Server*, *Datenbank-Server*. Diese können sowohl auf demselben, aber auch auf verschiedenen Computern ausgeführt werden.

2.2.2 Java / JEE

Java ist eine objektorientierte Programmiersprache. Wichtiger Aspekt ist hier, dass Java in einer Virtuellen Maschine (*JVM - Java Virtual Machine*), teil der *Java Runtime Environment*, ausgeführt wird. Dies gewährt Plattformunabhängigkeit.

Die Java Platform Enterprise Edition *JEE 5* dient als Spezifikation einer transaktionsbasierten Ausführung. Als Laufzeitumgebung dient hier ein *Java EE Application Server* welcher

- Sicherheit
- Transaktionsmanagement
- Namens- und Verzeichnisdienste
- Interoperabilität
- Management des Lebenszyklus (*Object-Life-Cycle Management*)
- Unterstützung für die Installation (*Deployment*, z.B. durchgeführt von *ANT*)

als Funktionalitäten zur Verfügung stellt.

Der Clientseitige Zugriff auf Anwendungen und Inhalte des JEE Application Servers erfolgt meistens über einen Browser, aber auch Client-Applications (*CORBA*, *Java-Application*, *WebServices*) sind möglich.

2.2.3 Servlets

Der Begriff "Servlet" ist ein Neologismus (*Wortneubildung*) aus "Server" und "Applet" und kann knapp als ein "serverseitiges Applet" umschrieben werden.

Es handelt sich dabei um Java-Klassen die Anfragen von Clients entgegennehmen und beantworten. Die Antworten werden hierbei meistens dynamisch erstellt.

Typisches Ablaufdiagramm einer Web-Anwendung mit Servlet siehe [Abbildung 2]

2.2.4 Tomcat

Apache Tomcat ist ein in Java geschriebener Servlet-Container (*Java EE Application Server*). Durch die Java-Implementierung wird Plattformunabhängigkeit gewährt - Tomcat ist der meistgenutzte Servlet-Container.

Da der Tomcat die offizielle Referenz-Implementierung des Servlets-Containers darstellt, ist garantiert, dass er sich konform zur JSP- und Servlet Spezifikation verhält.

2.3 Standards

2.3.1 IMS

IMS Content Packaging

IMS Content Packaging ist eine Beschreibung eines auf XML basierenden Datenformats wodurch Online-Lernmaterialien sowie deren Struktur für *eLearning-Systeme* vereinheitlicht werden sollen. Ziel ist die Interoperabilität zwischen unterschiedlichen Lernsystemen um digitale Lernmaterialien auf verschiedenen Systemen nutzen zu können.

Ein Paket wird durch ein *Manifest* auf XML-Basis vom Aufbau sowie Inhaltlich beschrieben, die Lerninhalte selbst unterliegen hierbei keiner Beschränkung im Datenformat, so können XHTML, HTML, XML, DOC, JPG uvm. verwendet werden.

Sowohl einzelne Lernmaterialien, sowie eine Sammlung von Inhalten (z.B. für Vorlesungsreihen oder Tutorials) können im Datenformat IMS Content Packing enthalten sein.

IMS Question Test Interoperability

Beim *IMS Question Test Interoperability (IMS QTI)* handelt es sich um ein vom *IMS Global Learning Consortium* festgelegtes Datenformat für die Erstellung von Online-Fragen und Tests *Quizzes, Multiple-Choice-Tests*.

2.3.2 SCORM

Das *Sharable Content Object Reference Model (SCORM)* ist eine Sammlung von Standards und Spezifikationen um Austauschbarkeit, Zugriff und Wiederverwendbarkeit von Lerninhalten zu ermöglichen.

Im *SCORM* beschreibt das *Content Aggregation Model (CAM)* die verwendbaren Ressourcen sowie die Möglichkeit von Zusammenfassung sowie Strukturierung von Lernpaketen. Die *Run-Time Environment (RTE)* bildet die Schnittstelle zwischen Lern-System und *CAM*. Zudem wird die Verwendbarkeit von Benutzerdaten wie z.B. zur Fortschrittsspeicherung definiert. *Sequencing and Navigation (SN)* beschreibt in seiner Spezifikation die Reihenfolge von Lerninhalten über Aktivitätsbäume.

2.4 Hibernate

Hibernate ist ein Open-Source Persistenz Framework für Java, welches es ermöglicht den Zustand eines Objektes in einer relationalen Datenbank zu speichern. Entsprechend kann aus den Datensätzen wieder ein Objekt erzeugt werden (*O-R-Mapping, ORM*).

Hibernate ermöglicht die Nutzung fast aller aktuellen relationalen Datenbanksystem (*Oracle, MySQL, PostgreSQL, Sybase, SAP DB vvm.*) und stellt mit der *Hibernate Query Language (HQL)* (SQL-ähnlich) eine Anbindungs-Schnittstelle zur Verfügung.

Auf der Anwendungsebene kann Hibernate in Java-Applications und Servlet-Engines benutzt sowie in Applicationserver integriert werden.

2.5 Open Source

Open-Source bezeichnet Software welche zusammen mit ihrem Quellcode angeboten wird. Hintergrund ist, dass es dem jeweiligen Nutzen offensteht diese weiterzuentwickeln oder auf etwaige Fehler zu prüfen. Oftmals wird diese Software kostenfrei angeboten und versteht sich als Gegenpol zur "etablierten" kommerziellen Software-Welt. Bekannte Open-Source-Projekte sind *Linux, Apache oder PHP*.

2.5.1 Extreme Programming

Extreme-Programming ist eine Methode der Software-Entwicklung die das Lösen der Aufgabe in den Vordergrund stellt und an Stelle formaler Definitionen zwischen Kunden und Entwickler viele Zwischenschritte definiert an denen in direkter Interaktion der Fortschritt des Projekts bewertet wird und der weitere Fortgang festgelegt wird.

Insbesondere im kommerziellen *Open-Source*-Umfeld bietet sich eine solche Herangehensweise an, da durch eine rege Beteiligung externer Entwickler eine starre Struktur schwer zu vermitteln ist. Ein weiterer interessanter Aspekt ist, dass nahegelegt wird auch innerhalb des Entwicklungs-Teams starre Strukturen aufzuweichen, um so eine breitere Wissensbasis zu haben, die Engpässe bei Ausfällen von Personen mindert und auch im Entwicklungsprozess zu weniger Fehlern führen kann.

3 Beschreibung der zu studierenden Applikation

3.1 Erweiterbarkeit

Erweiterungen werden im Ordner WEB-INF/lib abgelegt und in der *olat extensions.xml* definiert. Geladen werden sie vom OLAT-Extension-Manager (*org.olat.core.extensions.ExtManager*). Identifiziert werden die Erweiterungen über eine ID sowie über ihre Klasse (btw. Package-Namen).

- Interface:** org.olat.extensions.globalmapper
Erweiterungspunkt: org.olat.dispatcher.DispatcherAction
Klassen: org.olat.dispatcher.DispatcherAction
Beschreibung: Die Erweiterung kann als Mapper bezeichnet werden und kennt den path mit welchem er assoziiert wurde.
- Interface:** org.olat.extensions.action.ActionExtensionorg.olat.extensions.globalmapper
Erweiterungspunkt: org.olat.home.HomeMainController
Klassen: org.olat.dispatcher.DispatcherAction
Beschreibung: Die Erweiterung bietet einen Link mit Text an und eine Beschreibung und definiert was beim anklicken des Links geschehen soll.
- Interface:** org.olat.extensions.css.CSSIncluder
Erweiterungspunkt: org.olat.gui.components.Window
Klassen: org.olat.gui.css.CSSGenerator
Beschreibung: neue CSS-Stylesheets
- Interface:** org.olat.extensions.hibernate.HibernateConfigurator
Erweiterungspunkt: org.olat.persistence.DB
Klassen: org.olat.persistence.DB
Beschreibung: Die Erweiterung kann zusätzlich hibernate mappings enthalten. Es sei bemerkt dass sich das erstellen von Tabellen als schwierig erweist.
- Interface:** org.olat.extensions.sitescreator.SitesCreator
Erweiterungspunkt: org.olat.gui.control.generic.dtabs.DTabs
Klassen: org.olat.FullChiefController
Beschreibung: neue Seiten, muss SitesCreator enthalten der Liste von Site-Definition Objekten enthält.

3.2 Verzeichnisstruktur von OLAT

Für den Betrieb von OLAT sind die Pfade unterhalb von “webapp” relevant. Diese entsprechen der Servlet-Spezifikation von JEE, was bedeutet: die Dateien unter WEB-INF sind “persönlich” und nicht für den Client erreichbar, unterhalb von WEB-INF/classes befinden sich die Klassen-Archive, sortiert nach Klassennamen. Die Dateien für die Entwicklung, sprich die Java-Quellcode Dateien, befinden sich im src Verzeichniss. Dieses und auch patchesSrc werden für den Betrieb einer OLAT-Instanz nicht benötigt.

bin	Shell-Skripte für Linux/MacOSX
conf	Konfigurations-Dateien für Apache und IM
database	Setup-Dateien für die Datenbank und Dateien zur Hilfe beim Upgrade von älteren OLAT Versionen
dist	OLAT Lizenzdateien und ähnliches
doc	Dokumentation
htdocs	schönere Fehlercode-Seiten für Apache (bspw. 404)
interfaces	Beispieldateien für Flash-Integration in OLAT QTI
monitoring	Setup-Dateien und Shell-Skripte für das MRTG Überwachungstool
temp	Temporäre Dateien
webapp	Basis-Pfad für das Servlet
+ content	Layout-Dateien für VelocityContainer
+ examples	Beispieldateien (bspw. der Demokurs)
+ help	Online-Hilfe
+ i18n	Zeichenketten für die verschiedenen Sprachen
+ static	Statische Dateien (bspw. Multimedia-Daten)
+ WEB-INF	privater Teil des Servlets / Konfigurationsdateien von OLAT
++ classes	Java-Klassen von OLAT
++ lib	Java-Bibliotheken von OLAT
++ patchesSrc	Quellcode von Bibliotheken die verändert wurden
++ src	Java-Quellcode und Hibernate, sowie log4j

3.3 Rechtesystem 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 Rechengruppen 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 additiv und positiv:

additiv 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 (*Tabelle o_gp_business*) und eine **SecurityGroup** enthält User und wird z.B. einer Policy angehängt für die Rechteverwaltung. (*Tabelle o_bs_secgroup*)

3.4 Funktionalitäten

3.4.1 Privater Bereich

Administration von Gruppen und Benutzern Berechtigte Personen können Gruppen und Benutzern entsprechende Rechte zuweisen.

Systemadministration:

Systeminformationen Administratoren können technische Informationen zum System abrufen

Quotaverwaltung jedem Ordner im Olat kann ein Speicherlimit zugewiesen werden

Übersetzungstool Olat-Texte können einfach verändert werden

Benachrichtigungen versenden Benachrichtigungen über neue Ereignisse

Volltextsuche liefert aktuelle Parameter zum System

Überwachung überwachen von kritischen Werten im System, z.b. Anzahl angemeldeter Benutzer

Profil alle Angaben zu einem Benutzer

Einstellungen jeder Benutzer kann eigene Einstellungen vornehmen

Passwort jeder Benutzer kann ein eigenes Passwort wählen

Portal:

Benachrichtigung wenn Neuigkeiten für den Benutzer vorliegen werden sie sofort im Portal angezeigt

Kalender jeder Benutzer hat einen persönlichen Kalender

Bookmarks Links um Lernressourcen und Kurse schnell zu erreichen

Notizen jeder Benutzer kann für sich eigene Notizen zu einem Kurs anlegen

Leistungsnachweise Überblick über seine eigenen Prüfungs- oder ähnliche Leistungen

3.4.2 kollaborativer Bereich

Lerngruppen verbinden Teilnehmer eines Kurses mit bestimmten Mitgliedern eines Kurses

Rechtegruppen geben in Kursen entsprechenden Benutzern, entsprechende Rechte

Arbeitsgruppen können von jeden Benutzer erstellt werden, um gemeinsam an Projekten zu arbeiten

Gruppenkalender organisiert gemeinsame Termine einer Lern- oder Arbeitsgruppe

Email jeder Benutzer einer Lern- oder Arbeitsgruppe kann jedem anderen oder dem Betreuer eine E-Mail schreiben

Mitgliederliste zeigt alle Teilnehmer einer Gruppen

Ordner dient zum Austausch von Dateien zwischen Gruppenmitgliedern

Forum dient zum Informationsaustausch zwischen Gruppenmitgliedern

Wiki Tool um gemeinsames Wissen auszutauschen

Glossar es werden Begriffe erklärt die missverstanden werden können

Chat (optional) synchrone Kommunikation zwischen Benutzern

3.4.3 öffentlicher Bereich

Visitenkarte persönliche Daten die von allen Benutzern eingesehen werden können

Benutzersuche suche nach anderen Benutzer und ihrem Profil

Volltextsuche suche nach Kursen oder Lerninhalten, sowie Einträgen in Foren und Ordner, wenn Berechtigungen entstehen

Chat man kann mit allen angemeldeten Benutzern chatten, Mitglieder eigener Lern- oder Arbeitsgruppen werden in separaten Menü angezeigt

3.5 Modifikationen der OLAT-Installation

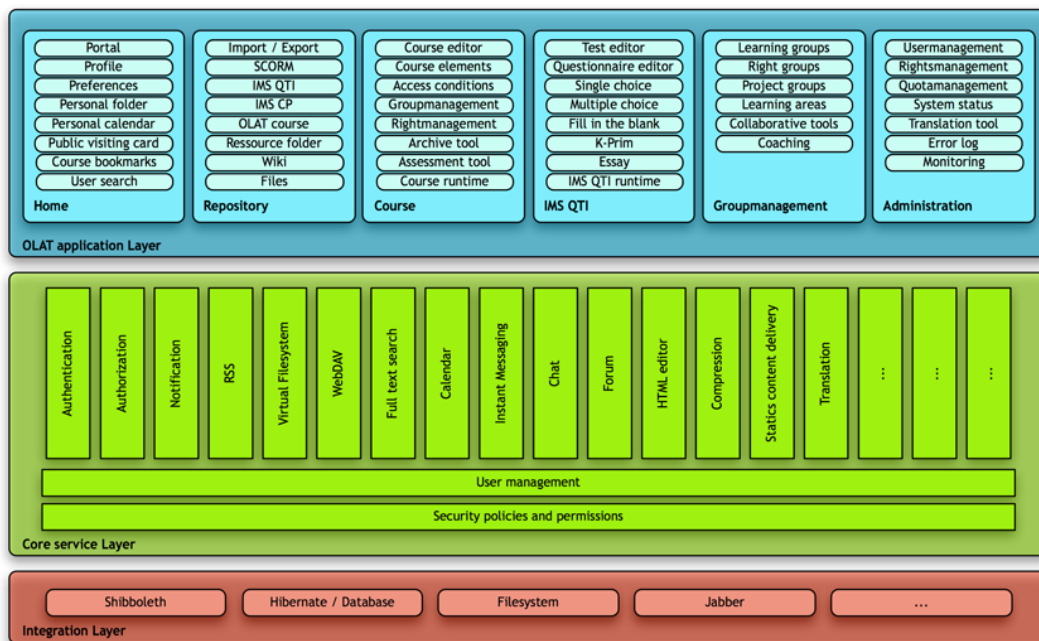
3.5.1 Entfernen des Tabs "Lernressourcen"

In der Datei *olat_extensions.xml* werden in der Bean `<bean id="olatsites" ... >` die Navigations-Tabs geladen. Kommentiert man dort die Bean mit `id="olatsites_repository"` aus, so wird "Lernressourcen" aus der Webdarstellung entfernt.

3.5.2 Entfernen der Demokurse

In der Datei *olat_config.xml* wird ein Modul `<module classname="org.olat.course.CourseModule">` definiert, unter diesem findet sich der Eintrag `<DeployCourseExports enabled="true">`. Setzt man dieses auf "false" oder kommentiert die Nachfolgend angegebenen Demo-Kurse aus und "builded" und "deployed" das OLAT-System, so werden die Demo-Kurse nicht mitinstalliert.

OLAT 5 System architecture



2006 © by good solutions

Abbildung 1: OLAT-Schichtenmodell

Servlet

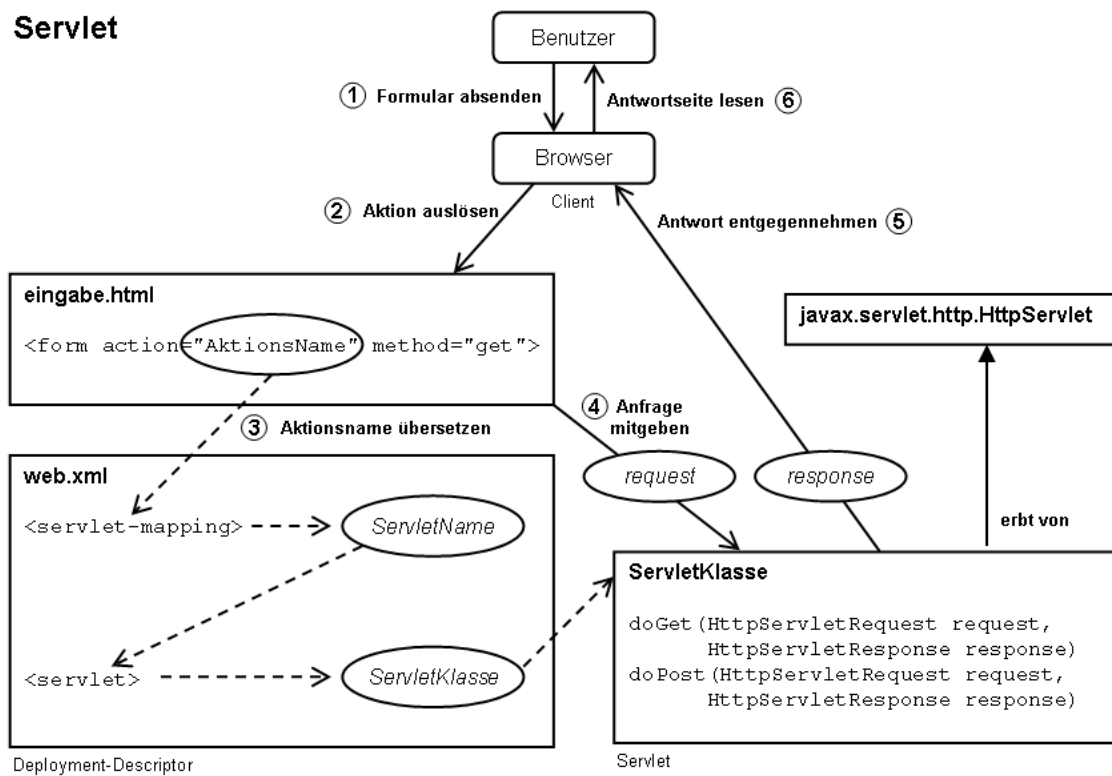


Abbildung 2: Ablaufdiagramm einer Web-Anwendung mit Servlet