

Aufgabenblatt 2

- Rahmenapplikation, Themenverwandte Projekte, projektrelevante Begriffe –

A. Beschreibung der Rahmenapplikation

In Bezug auf die Aufgabenstellung wurde zunächst die Rahmenapplikation untersucht, wobei auf folgendes geachtet wurde:

1. Quellen
2. Konzepte/Architektur
3. Funktionalitätsmerkmale
 - (1) Verbale Beschreibung
 - (2) Use-case (Anwendungsfall)
 - (3) Use- Case- Diagramm
4. Leistungsparameter
5. Einsatzgebiet
6. Innere Logik

A.1 Quellen zum Uebman 0.65

- Designbeschreibung der Gruppe Ueb20 aus dem Softwaretechnik- Praktikum 2003, die die Grundlagen (1. Version) des Uebmanagers geschrieben haben
 - "<http://pcai003.informatik.uni-leipzig.de/swp-03/Ergebnisse/ueb20/final/Designbeschreibung.pdf>"
- Java-Doc des Übmanagers v 0.65 von
 - "<http://sourceforge.net/projects/uebman/>"
- Herr Riechert, Institut für Informatik, ABT. „Anwendungsspezifische Informationssysteme“

A.2 Konzepte/ Architektur des Uebman 0.65

Im Folgenden werden alle verwendeten Konzepte für Uebman 0.65 betrachtet und kurz erläutert.

Die grundlegenden Konzepte, die hinter dem Uebman stecken, sind das **Struts-Framework** und **Java-Servlets**. Mit den beiden Konzepten wird eine Trennung in html-Frontend und Java-Backend vollzogen, da es sich um eine Online-Plattform handelt.

Weiterhin ist die gesamte Software auf einer **Client-Server Architektur** aufgesetzt, um die Bedienung, Installation und die technischen Anforderungen auf einem einfachen Niveau zu halten. Es wurde auf eine sehr modular gehaltene Architektur Wert gelegt, um die einzelnen Schichten so weit wie möglich unabhängig voneinander operieren zu lassen und somit eine einfache Erweiterbarkeit zu ermöglichen.

Sämtliche Daten werden mittels **des Sicherheitskonzeptes** behandelt und versendet.

A.2.1 Client-Server- Konzept

Server

Der Server setzt seinerseits auf dem Konzept der Java-Servlet-Technologie auf, mit den Vorteilen:

1. Zum einen muss der Nutzer nicht die gesamten Daten speichern, sondern dies wird durch den Server erledigt, zugleich steigt die Sicherheit, die in einem vernetzten System mit vertraulichen Daten zu einer wichtigen Komponente avanciert.
2. Des Weiteren steht im Hintergrund eine Form von Datenbank, die alle Daten in Form von XML-Dateien speichert und verwaltet.

Client

Der Client unterstützt das html-Frontend, also wird nur ein Java-fähiger Browser benötigt, um mit dem Verwaltungssystem zu arbeiten, z.B. mit Mozilla, Netscape, Opera.

A.2.2 Sicherheitskonzept

Die vertraulichen Daten, wie Passwörter, werden jedoch nicht offen übertragen, sondern, genau wie ihre Speicherung, mittels einer Verschlüsselung nach dem 1992 von Ron Rivest entwickelten MD5-Algorithmus.

Die Daten der Nutzer werden mittels eines Webservices zwischen verschiedenen Instanzen des Uebman transferiert.

A.2.3 Struts- Framework

Struts ist ein sehr erfolgreiches Framework, welches der elementaren und schnellen Entwicklung von einfachen, aber auch komplexen html-Frontend Applikationen dient. Seine modulare Struktur und problemlose Erweiterbarkeit machen es möglich, es für Anwendungen in nahezu allen Themengebieten als Grundlage zu nutzen. Struts ist ein Open-Source Projekt, welches von IBM und Sun gleichermaßen stark unterstützt wird und die MVC (Model-View-Controller) Architektur realisiert.

A.2.4. Java- Servlets

Die Java Servlet Technologie beinhaltet Funktionen, um Webseiten in automatisierter Form zu erstellen und bestimmte Inhalte wieder zu geben. Die automatisierte Verarbeitung von Informationen und die Bereitstellung der Funktionalität sind entscheidende Hilfsmittel, die eine enorme Zeiteinsparung mit sich bringen.

A.3 Funktionalitätsmerkmale

Der Uebman ist eine Verwaltungssoftware, die fast ausschließlich für Verwaltungsarbeiten bezüglich von Lehrveranstaltungen genutzt wird. Mit ihm arbeiten können:

1. Studenten
2. Dozenten
3. Tutoren
4. Administrator.

Dem jeweiligen Akteur werden mittels Rechten rollenspezifische Tätigkeiten ermöglicht.

A.3.1 Verbale Beschreibung

Mit dem Uebman ist es möglich, verschiedenste Verwaltungsaufgaben innerhalb einer Veranstaltung zu erledigen, wie z.B.:

1. Administration und Konfiguration des Uebman durch einen Administrator
2. Registrierung für den Uebman (einmalig, unter Angabe relevanter Angaben), womit automatisch eine Einschreibung zur entsprechenden Lehrveranstaltung erfolgt und das Arbeiten innerhalb des Uebman gewährleistet wird
3. Verwaltung von Gruppen (durch Tutor/Dozent)
4. Einschreiben in Gruppe (durch Student)
5. Bereitstellung von Übungsaufgaben, Information für Studenten / Lehrkräfte in einer News-Rubrik oder in dem eigens dafür vorhandenen Forum
6. Herunterladen/Hochladen von Materialien (Übungen, Lösungen, Skripte, korrigierte Lösungen mit Fehlerhinweisen etc.) durch Studenten, Dozenten, Administrator und Tutor
7. Eintragung von Punkten zur Benotung der Aufgaben
8. Anmeldung zu Klausuren/Prüfungen.

Eine wesentliche Einschränkung des Uebman ist, dass immer nur eine Veranstaltung pro Uebmanagerinstanz verwaltet werden kann.

A.3.2 Use-Case

Anwendungsfall: Administration
Akteur: Administrator
Beschreibung: Einrichten des Uebmanagers und Wartung
Ergebnis: Uebmanager ist für eine Veranstaltung konfiguriert

Anwendungsfall: Registrierung
Akteur: Student, Dozent, Tutor
Beschreibung: Akteur registriert sich um, Login-Daten zu erhalten und Zugriff zu bekommen
Ergebnis: Akteur kann sich einloggen

Anwendungsfall: Übungsgruppe erstellen
Akteur: Dozent
Beschreibung: Dozent erstellt Übungsgruppen, in die sich die Studenten eintragen können
Ergebnis: Übungsgruppe angelegt

Anwendungsfall: Einschreibung
Akteur: Student
Beschreibung: Student schreibt sich in eine Übungsgruppe ein
Ergebnis: Student ist in einer Übungsgruppe

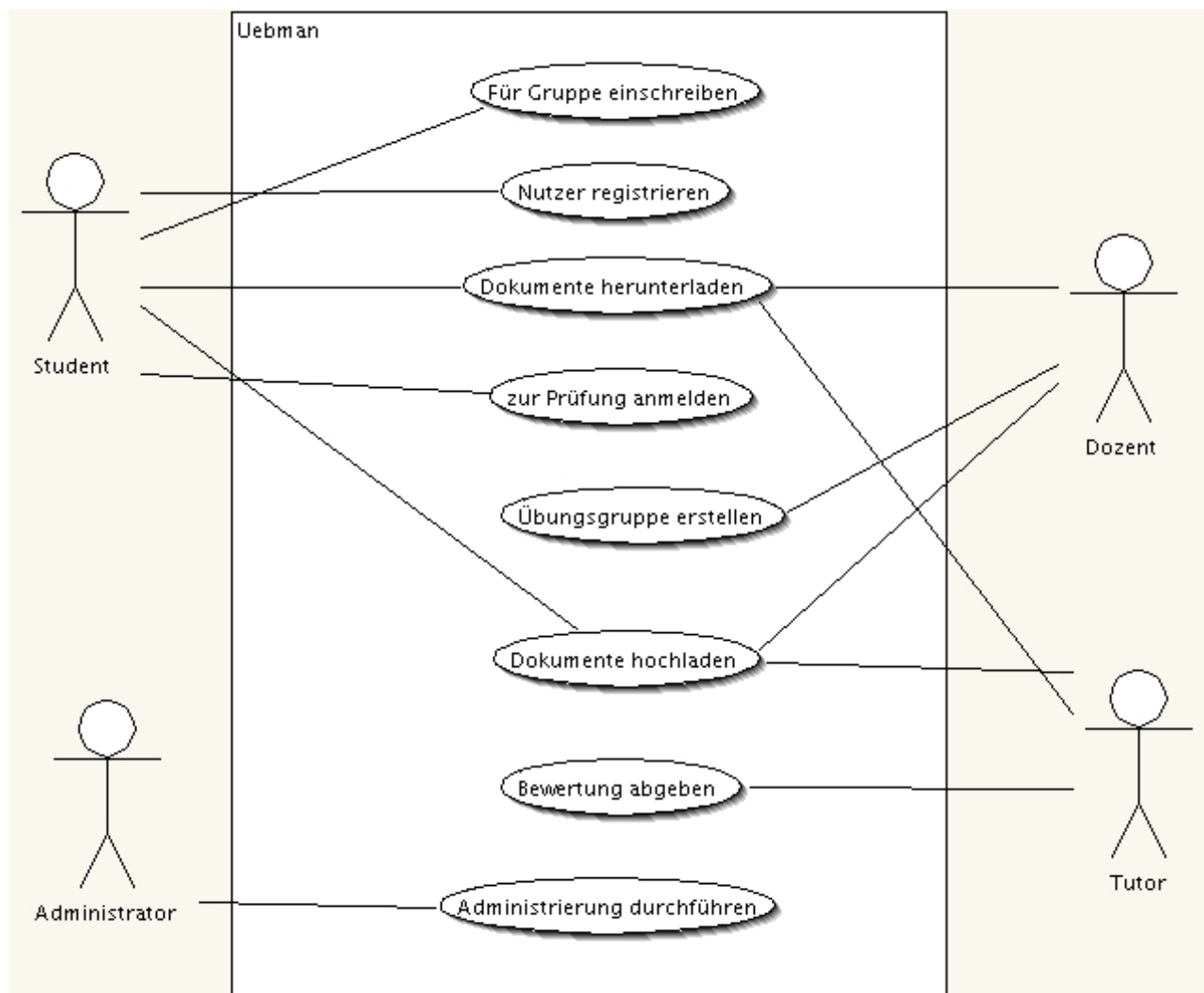
Anwendungsfall: Download
Akteur: Dozent, Student, Administrator, Tutor
Beschreibung: Materialien, Aufgaben und Lösungen können heruntergeladen werden
Ergebnis: Aufgaben, Materialien und Lösungen stehen für weitere Verarbeitung zur Verfügung

Anwendungsfall: Upload
Akteur: Dozent, Tutor, Student
Beschreibung: Lösungen, Aufgaben und Materialien können hochgeladen werden
Ergebnis: Lösungen, Aufgaben und Materialien stehen zum Download und darauf folgenden Arbeiten zur Verfügung

Anwendungsfall: Bewertung abgeben
Akteur: Tutor
Beschreibung: Tutoren vergeben Punkte an Studenten für deren Lösungen der Aufgaben
Ergebnis: Student hat Punktebewertung

Anwendungsfall: Prüfungsanmeldung
Akteur: Student
Beschreibung: Student kann sich für eine Klausur / Prüfung anmelden
Ergebnis: Student ist eingeschrieben und kann an Klausur teilnehmen

A.3.3 Use- Case- Diagramm



A.5 Leistungsmerkmale

Der Uebmanager ist bisher nicht auf seine Leistungsfähigkeit getestet worden. Jedoch traten bis zum jetzigen Zeitpunkt keine auffallenden Einschränkungen in der Arbeit mit dem Uebman auf, auch nicht in beispielhaften Belastungssituationen, z.B. bei einer Online-Klausur am Institut für Pädagogik an der Universität Leipzig, bei der in etwa 40 bis 60 Studenten gleichzeitig mit dem Uebman arbeiteten.

A.6 Einsatzgebiet

Das Einsatzgebiet des Uebman in der Version 0.65 ist derzeit nur der Lehrstuhl für "Anwendungsspezifische Informationssysteme" an der Universität Leipzig, für die Verwaltung der Vorlesung "Softwaretechnik" und dem anschließenden "Softwaretechnik-Praktikum".

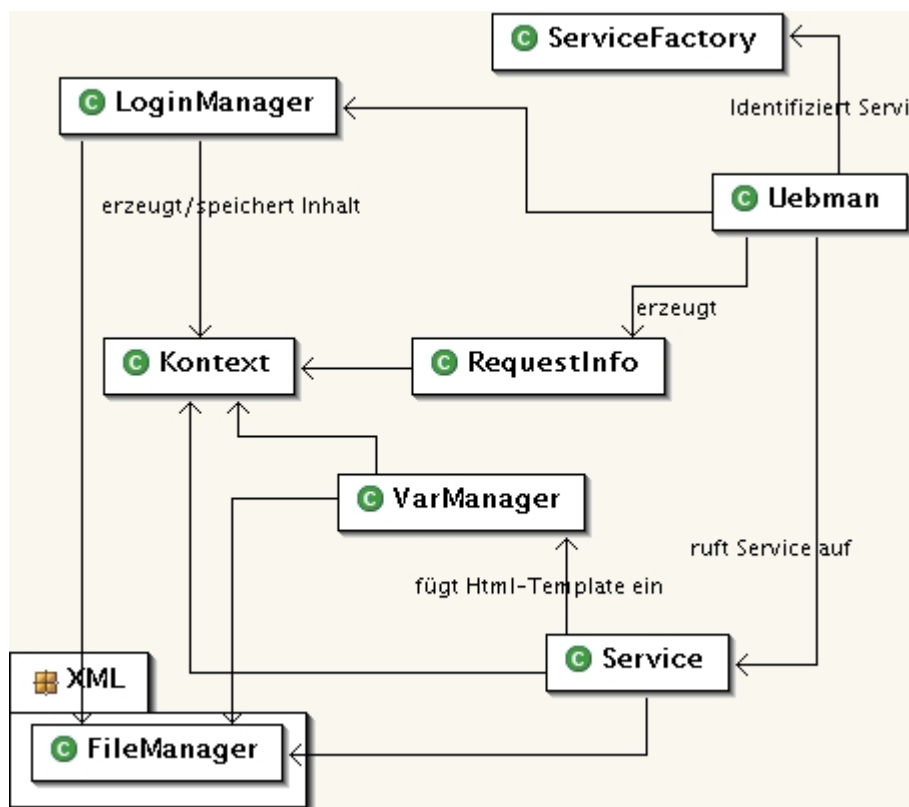
Es wurde ebenfalls beschlossen, den Uebman als Open-Source – Projekt unter der Adresse

- "<http://sourceforge.net/projects/uebman/>"

zur freien Verfügung zu stellen.

Eine neuere Version wird am Institut für Pädagogik der Universität Leipzig eingesetzt, diese Version steht jedoch noch nicht als Open-Source zur Verfügung.

A.7 Äußere Aspekte der inneren Logik



In diesem UML-Diagramm sollen die wichtigsten, sichtbaren Aspekte dargestellt werden. Hierbei ist das Uebman-Servlet die zentrale Instanz.

Der LoginManager verwaltet die Anmeldung und Registrierung von Nutzer-Accounts und bezieht/speichert die benötigten Daten, wie auch Services und VarManager, (z.B. Login mit Matrikelnummer und Paßwort) aus/in XML-Dateien.

Die ServiceFactory ordnet einer Anfrage den passenden Service zu und ruft ihn auf. Services sind zum Beispiel individuelle Index-Seiten (Student | Tutor ...), hoch-/herunterladen von Dateien, etc. Der FileManager sorgt für die sichere Verwaltung der zum Download bereitgestellten Dateien und Abgrenzung von sensiblen Daten in den XML-Dateien.

B. Themenverwandte Projekte und deren Konzepte

Im Rahmen der Recherche wurden themenverwandte Anwendungen im Internet gesucht, um mögliche Funktionen, Umsetzungsmöglichkeiten etc. herauszufiltern, die bei der Entwicklung der Kooperationsplattform mit integriert werden können. Dabei haben wir uns auf zwei Anwendungsmodelle konzentriert.

1. HIS GX(**H**ochschul-**I**nformations-**S**ystem)
2. TEMPLUS 5.0 (**T**eaching **M**anagement **P**latform for **U**niversitie**S**)

B.1 HIS GX

Alle Informationen bezüglich HIS GX sind zu finden unter:

- „<http://www.his.de>“

HIS GX(Hochschul-Informationen-System) ist eine kommerzielle Software der Firma HIS GmbH zur Hochschulverwaltung, welche aus mehreren Modulen bezüglich der Aufgabenbereiche besteht. Das unserer Aufgabenstellung entsprechende Modul ist HISLSF, für den Aufgabenbereich **L**ehre- **S**tudium- **F**orschung.

Das Modul LSF bietet Schnittstellen auf XML-Basis, um Daten innerhalb der einzelnen Module von HIS-GX zu übertragen. Ebenso können Fremdsysteme mit Hilfe dieser Schnittstellen angebunden werden. Diese Eigenschaft gestattet es, den Funktionsumfang auf die Bedürfnisse der jeweiligen Hochschule abzustimmen und bei Bedarf eine spätere Funktionserweiterung vor Ort durchzuführen, indem neue Module nachinstalliert werden.

Das Modul LSF bietet **Funktionen** für:

1. Erfassung und Präsentation von Lehrveranstaltungen, Forschungsprojekten und den damit verbundenen Ressourcen (Einrichtungen, Personen, Räume)
2. Das Anmelden bzw. Einschreiben zu einer Veranstaltung (insbesondere auch teilnahmebeschränkte Veranstaltungen)
3. LSF dient als Studieninformations-, Studienberatungs- und Planungssystem, für verschiedene Nutzerkreise (Studierende, Lehrpersonal, Administratoren, Raumverwalter)

Funktion 1 und 2 werden nur in geringem Umfang vom derzeitigen Uebman unterstützt und sollen in der zu entwickelnden Kooperationsplattform herausgearbeitet werden. Die Funktion 3 könnte perspektivisch umgesetzt werden.

LSF ist genau wie die zukünftige Kooperationsplattform eine rein webbasierte Anwendung, in der sämtliche Nutzer- und Administrationsfunktionen über einen Web-Browser zugänglich sind.

Die wichtigsten **Konzepte** von HIS GX (HISLSF im speziellen) sind:

1. Modularisierung
 - HIS umfasst integrierte, separat einsetzbare Module mit Schnittstellen, die auch durch Fremdsysteme ansprechbar sind (**webbasierte Selbstbedienungsfunktion**)
2. Redundanzfreie Datenerfassung
 - Ist ein wesentliches Konzept, um eine komfortablere Bedienung und vor allem eine fehlerfreie Datenpflege zu ermöglichen.

3. Intuitive Benutzbarkeit
 - Durch logische Menüführung und Verwendung von Windows-Style-Elementen
4. Verwaltung verschiedener Nutzergruppen
 - Verschiedene Nutzergruppen mit entsprechenden Rechten, die nur im Rahmen dieser agieren.
5. Client-Server/ Datenbank
 - Auf eine zentrale Datenbank kann mittels Internet/ Intranet zugegriffen werden. Dies ermöglicht erst die redundanzfreie Datenerfassung und auch die webbasierte Selbstbedienungsfunktion.

Das Konzept 5 wird bereits von dem vorhandenen Uebman unterstützt und weiterführend von der Kooperationsplattform genutzt. Die Konzepte 1- 4 werden bei der Projektdurchführung, in Bezug auf Realisierung der Kooperationsplattform, mit integriert und im Wesentlichen verarbeitet.

B.2 TEMPLUS 5.0

Alle Informationen bezüglich TEMPLUS 5.0 sind zu finden unter:

- „<http://templus.cs.uni-dortmund.de/>“

Ziele des TEMPLUS-Projektes sind die Bereitstellung und der Betrieb einer internet-basierten Plattform, die es den unterschiedlichen universitären Nutzergruppen (Student, Dozent, Organisator, Tutor) ermöglicht, effizient zu kooperieren, und ihre Arbeitsabläufe optimal unterstützt.

Das Tool bietet **Funktionen** zur Verwaltung von Lehrveranstaltungen, Seminaren und Übungen:

1. Übungsgruppeneinteilung
2. Studierendendaten anzeigen
3. Mailverteilung für Lehrveranstaltung und Übungsgruppen
4. Lehrveranstaltungsdaten anzeigen (Links, Material etc.)
5. Anzeigen von Punkten, Anwesenheit, Mitarbeit
6. Angaben über Scheinerhalt.

Alle die genannten Funktionen sollen im wesentlichen Bestandteil unserer Kooperationsplattform werden. Die Umsetzung der Funktionen kann von daher als Hauptziel für unser Projekt betrachtet werden.

Die wichtigsten **Konzepte** von TEMPLUS sind:

1. Modularisierung
 - Zum Betrieb gehören das Benutzer- und Rollenmanagement
 - Dienste erfordern einen SSL fähigen Browser
 - Zur Darstellung werden Frames benutzt.
2. Redundanzfreie Datenerfassung
 - Austausch von Daten und Funktionalitäten zwischen den einzelnen Diensten, daher keine doppelte Datenhaltung
3. Intuitive Benutzbarkeit
 - Leichte Bedienbarkeit, Personalisierbarkeit
 - Personalisierte Navigation und Benutzerführung

4. Verwaltung verschiedener Nutzergruppen
 - Rollen- und profilspezifische Zugangskontrolle und Nutzerführung
5. Client-Server/ Datenbank
 - Architektur wird realisiert als Server-Farm mit Lastverteilungs-Komponente bestehend aus 4 Schichten (Web Server (1), Load-Balancing (2), Application Server (3), Database Server (4))
 - o Schicht (1) und (3) beliebig skalierbar, von daher Ausfallsicherheit und kurze Zugriffszeiten
 - o Datenzugriff auf Schicht (1) nur extern und Firewall → Datensicherheit
 - o Schicht (3) und (4) im internen Netzwerk → kein direkter Zugriff von außerhalb
 - o Schicht (2) gewährleistet eine lastabhängige Verteilung der Anfragen auf die Application Server in Schicht (3).
6. Sicherheit
 - Zugriffssicherheit und Datenschutz durch komplexe Checkmechanismen
 - Caching, d.h. häufig benötigten Informationen werden in speziellen Caches, auf die alle Dienste Zugriff haben, abgelegt.

Das Konzept 4 wird bereits vom Uebman unterstützt und auf die Kooperationsplattform übertragen. Das Konzept 5 ist für unser Projekt nicht relevant, da wir die gegebene Server- und Datenbanksysteme übernehmen. Die Konzepte 3 und 6 werden wir als Basiskonzepte verwenden und darauf aufbauend unsere Benutzerarchitektur gestalten.

C. Projektrelevante Begriffe

<u>Begriff</u>	<u>Erläuterung</u>	<u>Warum</u>
Kooperationsplattform	Im Rahmen unseres Projektes zu gestaltende Lehrveranstaltungsverwaltung	Abgleich des Verständnisses über den Auftrag
Caching	Zwischenspeichern von Daten	Fachbegriff
Account	Gespeicherte Nutzerdaten	Fachbegriff
Use-case	Konkretisierung einzelner Funktionen	Fachbegriff
Korrektur	Anmerkungen bei unzureichender / falsche Lösung	Hier nur Anmerkung, keine Ausbesserung, Verwechslungsgefahr
Service	Dienst zur Erfüllung einer Aufgabe / Funktion	Fachbegriff
Servlet	Webserver-seitige Erweiterung zur Erzeugung dynamischer Webseiten-Inhalte	Fachbegriff
Login / Logout	Anmelden / Abmelden bei einer Online-Plattform	Fachbegriff
Session	Arbeitszyklus bei Login in Online-Plattform	Fachbegriff
Übungsgruppe	Seminar zum Thema einer Veranstaltung	Verwechslungsgefahr mit Aufbaukurs
Forum	Online-Diskussions-Plattform	Fachbegriff
Kontakt	Information, um bestimmte Person zu erreichen (E-Mail, Tel.nr., etc)	Fachbegriff
Rolle	Je nach Einstufung durch Rechte für Uebman/Kooperationsplattform handelt es sich um Student, Dozent,	Fachbegriff

	Tutor oder Administrator	
Struts- Framework	Ein Hilfsmittel zum leichteren Erstellen von Online-Applikationen	Fachbegriff
Konzepte	Softwaretechnische Grundlagen zur Erstellung einer Anwendung (z.B. Servlet, Struts Framework)	Fachbegriff
Frontend	Client	Fachbegriff
Backend	Server	Fachbegriff
XML	Datenspeicherung / -Verwaltung	Fachbegriff
Open- Source	Frei für jedermann erhältlich und veränderbar	Fachbegriff
Verwaltungssoftware	Stellt Funktionen für vorher definiertes Einatzgebiet zur Verfügung	Fachbegriff
Instanz	Kopie des Uebman für eine bestimmte Veranstaltung konfiguriert	Fachbegriff
Download	Dateien/Verzeichnisse von einem Webserver laden	Fachbegriff
Upload	Dateien/Verzeichnisse auf einen Webserver laden	Fachbegriff
Module	Bausteine	Fachbegriff