

## Softwaretechnik – Praktikum 2006

### Entwurfsbeschreibung

#### 1. Allgemeines

##### a) Ziel des Produktes

Die Firma "SoftConsult" berät Firmen bei Zuschnitt und Einführung der betrieblichen IT-Infrastruktur. Über diese Infrastruktur müssen insbesondere die in der Firma vorhandenen Informationen und Wissensbausteine verwaltet und weiterentwickelt werden. Dafür haben sich in den letzten Jahren Konzepte des "Semantischen Web" durchgesetzt.

Die Firma "SoftConsult" hat in einem Vorprojekt einen Prototyp entwickelt, der als Plugin InstanceXL für Protégé eine solche Konzeptualisierung unterstützt, indem mit diesem Werkzeug Ontologien modelliert, Wissensbasen verwaltet und Ergebnisse von Suchanfragen verschiedener Art in Tabellenform ausgegeben werden können.

Die prototypische Implementierung ist als Einzelplatzapplikation ausgelegt und für den Einsatz in einer verteilten Umgebung nicht geeignet. Andererseits stellt Protégé gewisse Grundfunktionalitäten für einen Client-Server-Betrieb zur Verfügung.

Ziel ist es, die prototypische Implementierung zu einer Client-Server-Variante umzubauen. Der Server soll auf der Basis von Protege 3.1 die Wissensbasis verwalten. Als Client soll ein Web-Browser verwendet werden, dessen Server-Anbindung über Portlets und die Portlet-Engine Jetspeed-2 realisiert ist.

##### b) Systemvoraussetzungen

Um die Applikation nutzen zu können, wird die Rahmenapplikation Protégé in der Version 3.1 benötigt. Protégé selbst benötigt eine JVM2 in der Version 1.4 oder höher. Eine JVM ist für alle gängigen Betriebssysteme verfügbar.

Darüber hinaus muss die Möglichkeit existieren entweder auf ein internes Netzwerk (beim Einsatz in der Firma am Arbeitsplatzrechner z.Bsp.) oder auf das Internet zuzugreifen (wenn die Applikation z.Bsp. auch von zu Hause aus erreichbar sein soll).

Für die Darstellung wird ein Web-Browser benötigt.

#### 2. Produktübersicht

##### a) Grundlegende Funktionen des Programms

###### - Nutzerregistrierung und Verwaltung

Um die Applikation nutzen zu können muss man sich über das Web-Portal als registrierter Nutzer einloggen. Die Registrierung und der Login können auf der Startseite vorgenommen werden. Die so erstellten Nutzeraccounts werden von einem Administrator verwaltet, der dabei die Rechte der Nutzer bestimmt und die Möglichkeit hat Accounts zu löschen.

###### - Anzeige und Auswahl von Ontologien

Dem Nutzer werden die verwalteten Ontologien angezeigt und er kann sich die gewünschte auswählen.

## Softwaretechnik – Praktikum 2006

### Entwurfsbeschreibung

- Anzeige und Auswahl von Klassen  
Dem Nutzer werden die Klassen der ausgewählten Ontologie angezeigt und er kann sich die gewünschte Klasse auswählen.
- Anzeige und Editieren von Instanzen  
Dem Nutzer werden die Instanzen ausgewählter Klasse angezeigt und er kann die einzelnen Instanzeigenschaften je nach Nutzerrechten editieren.

#### **b) Rollenverteilung**

In einem System, wo Daten verwaltet, abgerufen und geändert werden können ist es sinnvoll Zugriffsrechte verschiedener Stufe einzurichten. Unmittelbar nach der Registrierung hat der Nutzer nur Leserechte. Der Administrator kann diese Zugriffsrechte für jeden Nutzer einsehen und verändern. Zur Verfügung stehen:

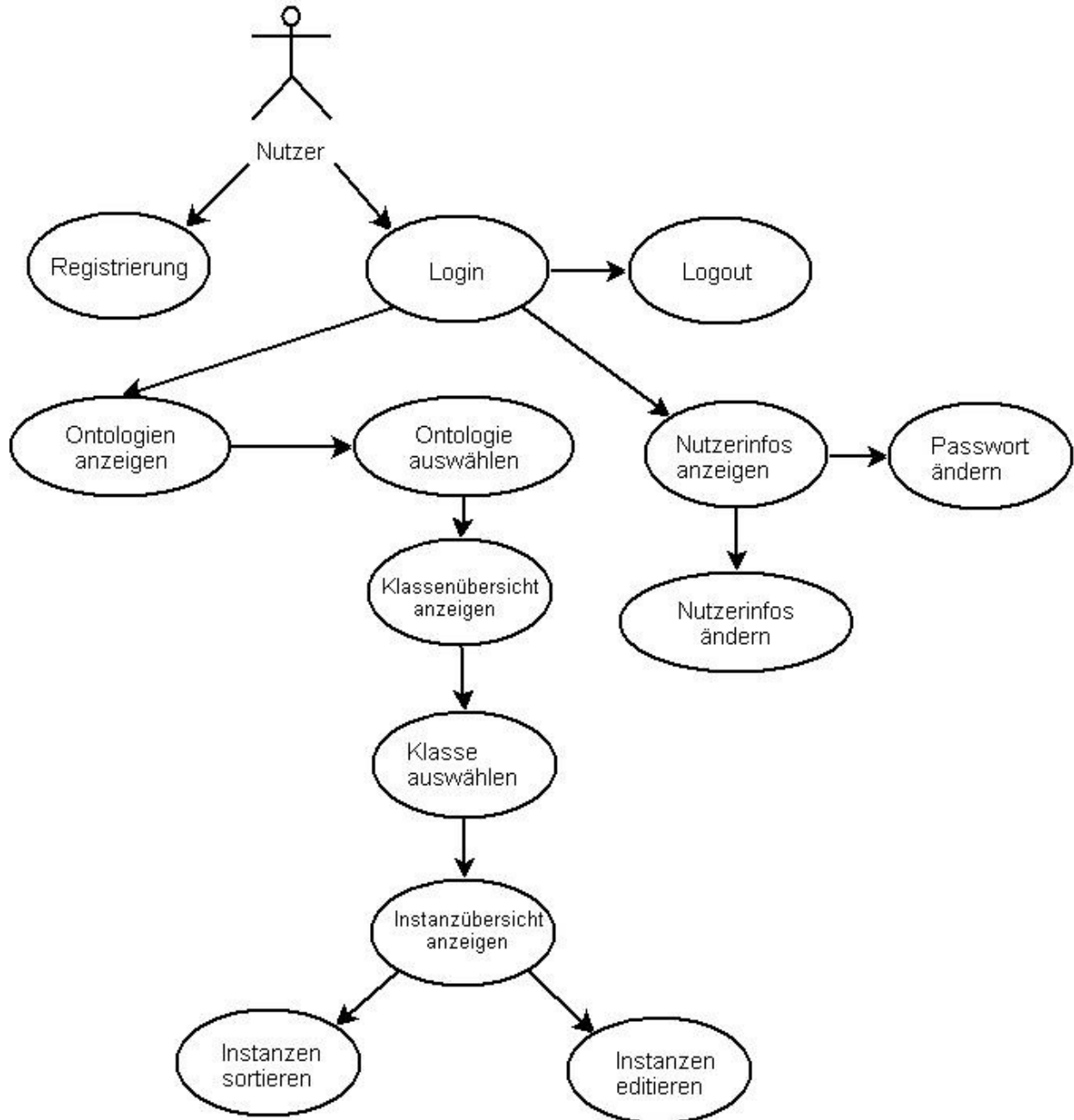
1. Leserechte, der Nutzer ist nur berechtigt Instanzen einzusehen, aber nicht zu ändern.
2. Editierrechte, der Nutzer darf Instanzen einsehen und ändern.
3. Administrator, der Administrator hat uneingeschränkte Zugriffsrechte, er kann alle Funktionen der Applikation nutzen.

Diese Rollenverteilung wird über die durch Jetspeed-2 gegebenen Rollen realisiert. Für die Verwaltung einer Ontologie ist dies so ausreichend und in vorerst so angedacht. Es können auch Wissensbasen mit mehreren Ontologien verwaltet werden. Dafür müsste das Rechtesystem überarbeitet und erweitert werden.

Es ist sinnvoll die Rolle des Administrators auf mehrere Personen aufzuteilen. So wird es einen Administrator für den öffentlichen Bereich geben, der sich um die Nutzerregistrierung und Rechteverwaltung kümmert und einen für den internen Bereich, der für die Kontrolle der Wissensbasen verantwortlich ist. Bei einer Vielzahl von großen Wissensbasen ist es darüber hinaus sinnvoll Administratoren für jede Wissensbasis einzusetzen.

## Softwaretechnik – Praktikum 2006 Entwurfsbeschreibung

### c) Use-Case-Diagramm



## Softwaretechnik – Praktikum 2006

### Entwurfsbeschreibung

### 3. Grundsätzliche Designentscheidungen

#### a) MVC – Modell

Der Begriff Model-View-Controller (MVC); Modell-Präsentation-Steuerung (MPS); bezeichnet ein Architekturmuster zur Aufteilung von Softwaresystemen in die drei Einheiten Model, View und Controller.

Ziel eines solchen Modells ist ein flexibles Programmdesign, um u.a. eine spätere Änderung oder Erweiterung einfach zu halten und die Wiederverwendbarkeit der einzelnen Komponenten zu ermöglichen. Außerdem sorgt das Modell bei großen Anwendungen für eine gewisse Übersicht und Ordnung durch Reduzierung der Komplexität. Gleichzeitig bringt die Trennung auch eine Rollenverteilung bei der Implementierung mit sich. Die Aufgabenverteilung bei der Implementierung kann somit den Fähigkeiten der Gruppenmitglieder angepasst werden:

- Controller - Der Programmierer erstellt die nötige Geschäftslogik (implementiert Algorithmen, stellt fertig aufbereitete Daten bereit, etc.)
- View - Ein Gestalter oder GUI-Designer erstellt das grafische Erscheinungsbild, die Anordnung, Größe und Ausrichtung der Portlets
- Modell - Datenbankexperten kümmern sich um die optimale Datenverwaltung, Datenbankdesign usw.

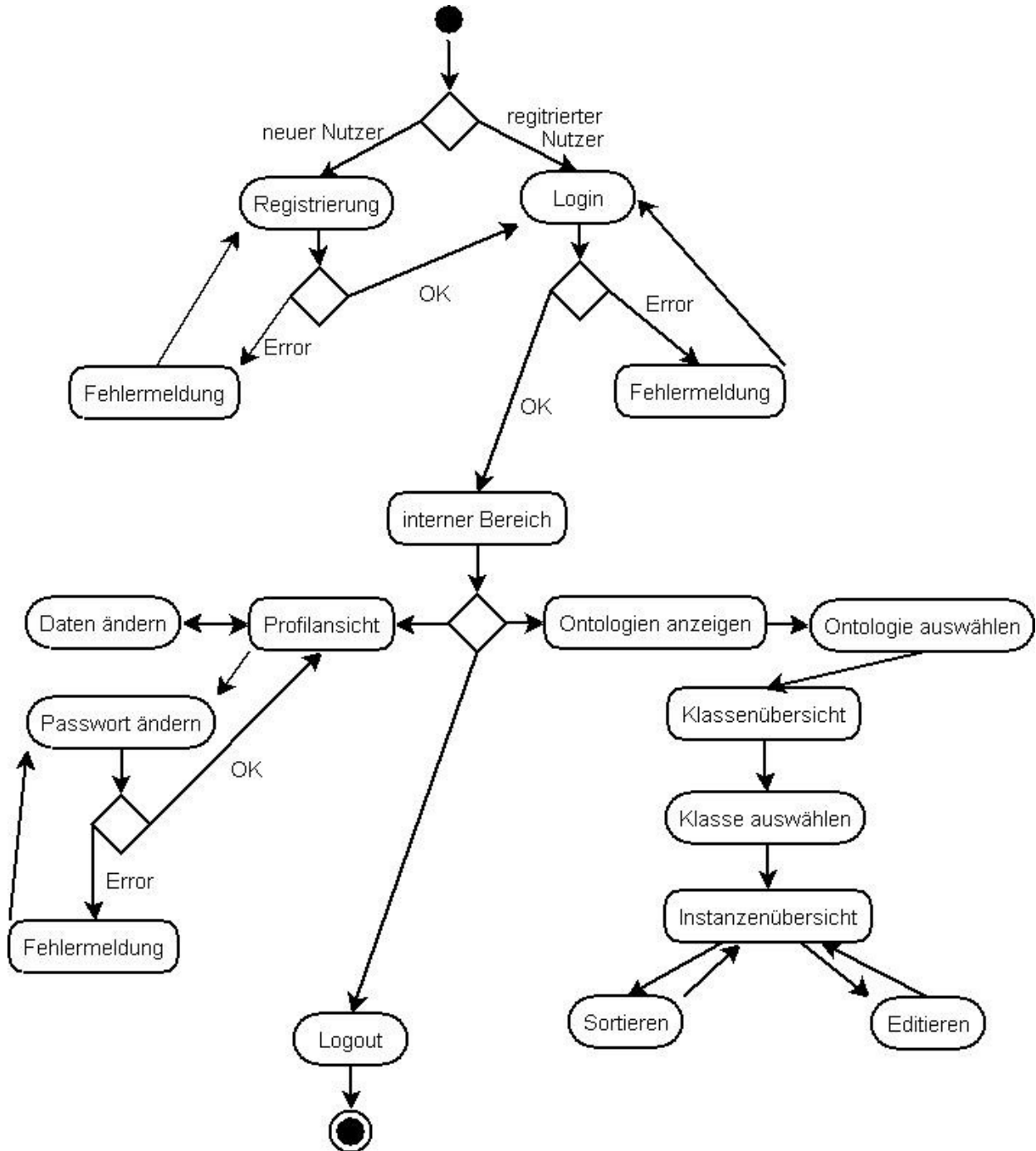
#### b) Schichten des MVC-Modells

Model	Das Datenmodell enthält die dauerhaften (persistenten) Daten der Anwendung. Das Model hat lesenden Zugriff auf diverse Backend-Speicher wie zum Beispiel Datenbanken. Das Model kennt weder die View noch den Controller, es weiß also gar nicht, wie, ob und wie oft es dargestellt und verändert wird. Änderungen im Model werden allerdings über einen Update-Mechanismus bekannt gegeben, indem ein Event ausgelöst wird. Dazu muß sich zumindest die View als abhängiges Objekt am Model registrieren, um über Datenänderungen informiert zu werden
View	Die Darstellungsschicht präsentiert die Daten in der Regel - jedoch nicht notwendigerweise - zwecks Anzeige. Die Programmlogik sollte aus dem View entfernt werden. Der View kennt das Model und ist dort registriert, um sich selbständig aktualisieren zu können.
Controller	Der Controller verwaltet die Sichten, nimmt von ihnen Benutzeraktionen entgegen, wertet diese aus und hat schreibenden Zugriff auf das Modell. Er enthält die Intelligenz und steuert den Ablauf (engl. Workflow) der Anwendung.  Das MVC-Muster trifft keine Aussage über die Positionierung der Geschäftslogik innerhalb der MVC-Klassen. Diese kann je nach Anwendungsfall besser im Controller aufgehoben sein oder besser in das Modell verlagert werden (z.B. wenn es mehrere Controller gibt).

## Softwaretechnik – Praktikum 2006 Entwurfsbeschreibung

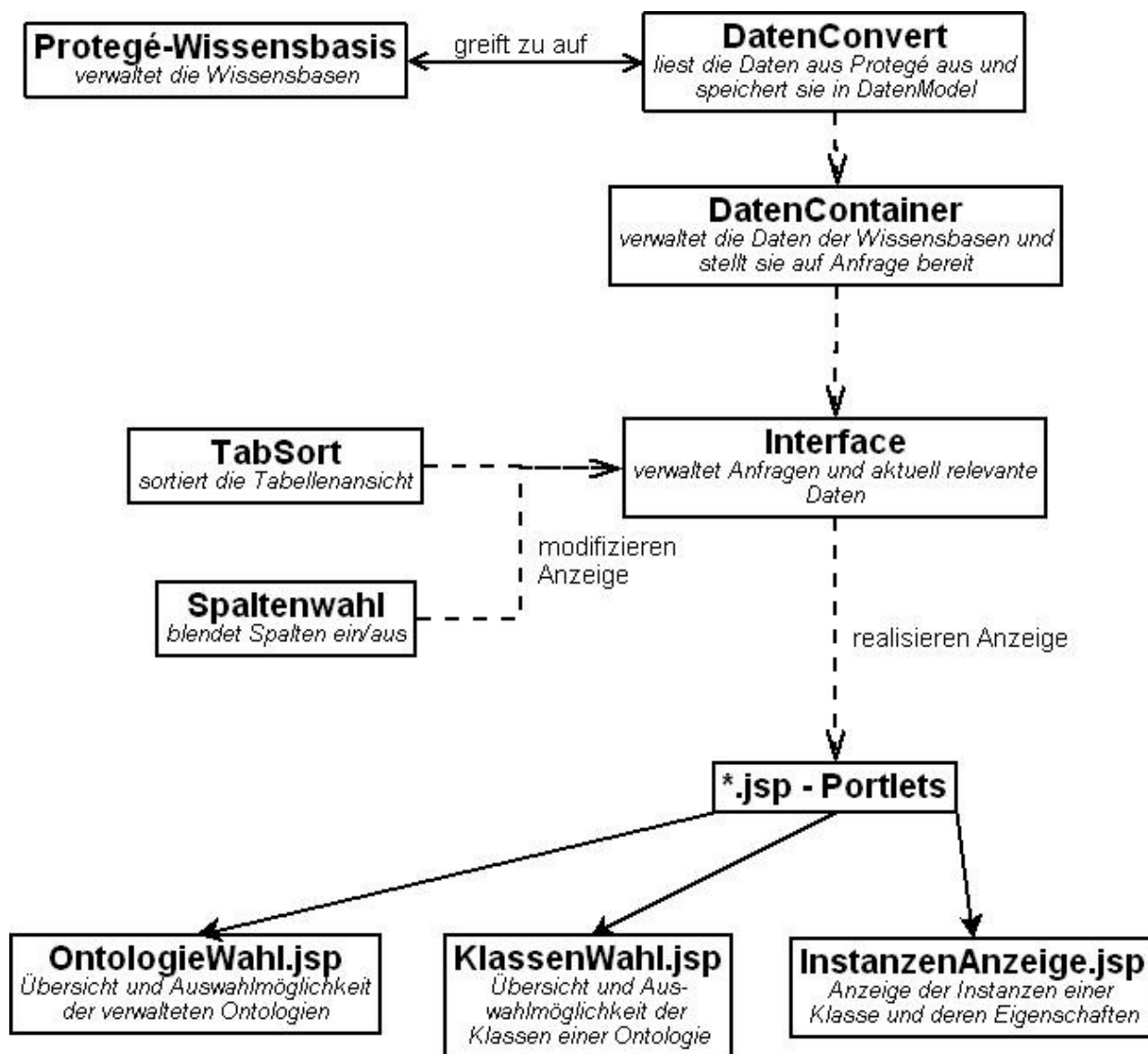
### 4. Grundsätzliche Struktur- und Entwurfsprinzipien

#### a) Aktivitätsdiagramm



## Softwaretechnik – Praktikum 2006 Entwurfsbeschreibung

### b) Klassendiagramm



## Softwaretechnik – Praktikum 2006 Entwurfsbeschreibung

### c) Paketübersicht

