

Entwurfsbeschreibung des eigenen Projektes

Version 1

1. Allgemeines
2. Produktübersicht
3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem
4. Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete

1. Allgemeines

Ziel des Projektes ist die Darstellung und Manipulation von Wissensbasen mit Hilfe der Portaltechnologie. Über einen Browser erfolgt der Zugriff auf Klassen und Instanzen mit deren Properties der Wissensbasis. Die folgenden Funktionen werden dabei unterstützt: Klassenhierarchieanzeige, Instanztabelle, Suchen, Sortieren, Filtern und Editieren. Zum Einsatz kommt das Portal Jetspeed-2, welches eine gewisse Funktionalität mitbringt. Die folgenden Portalfunktionen werden verwendet: Einloggen/Ausloggen, Registrieren, Nutzerverwaltung, Rechteverwaltung, Passwort ändern, Sprache wählen, etc.

Es werden Nutzer, Editor und Administrator unterschieden. Je nach zugeordneten Rechten können diese User verschiedene Funktionen ausführen. Je nach Funktion ändert sich auch das Erscheinungsbild des Portals. (Die Funktion (Portlet) Rechte verwalten steht beispielsweise nur dem Administrator zur Verfügung.)

Ein Portlet ist dabei eine Erweiterung des Servlets, so wie der PortletContainer (bspw. Pluto) eine Erweiterung des Servletcontainers darstellt (bspw. Tomcat). Portlets bilden auf der Clientseite eine einfach zu benutzende Oberfläche innerhalb des Browsers (Fenster mit Schaltflächen zum Maximieren, Minimieren, Editieren, Hilfe). Intern, also auf Serverseite, kann nun eine beliebige Anwendung liegen, die ihre Darstellung auf das Portlet weiterleitet. Sie entsprechen somit einer Sicht im Rahmen des Model View Controller-Konzeptes (MVC).

Systemvoraussetzungen:

Auf der Serverseite bestehen folgende Anforderungen:

- Java 1.4.2_02 or higher
- JetSpeed-2
- Servlet Engine (Tomcat 5.5.8 or higher)
- Zugriff auf die Wissensbasis

Browser auf Clientseite (mit Netzwerkanschluss)

Verwendete Libraries, die Speziell für die Portlets und die Wissensbasis benötigt werden:

Portlet-API (portlet-api-1.0.jar)

KnowledgeBase (protege.jar) bzw.

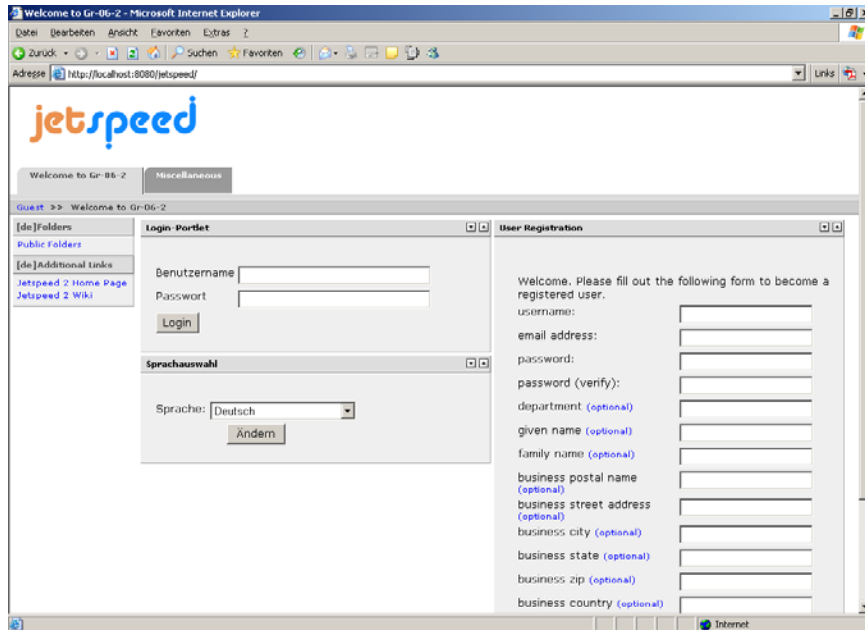
JenaOWLKnowledgeBase (jena.jar)

Die java-Klasse und die Deployment-Deskriptoren werden in ein WAR gepackt. Um die Anwendung im Portal verfügbar zu machen, genügt es, die WAR-Datei in das Verzeichnis *Jetspeed2.0/webapps/jetspeed/WEB-INF/deploy/* zu kopieren. Danach steht das Portlet beim Konfigurieren von Portalseiten zur Verfügung und kann eingefügt werden.

2. Produktübersicht

Im Rahmen dieses Projektes kommt Jetspeed-2 als erweiterbares Open Source Enterprise Information Portal zum Einsatz. Dekoration und Layout werden bei Jetspeed-2 über Velocity Templates oder JSPs gesteuert und sind über diese konfigurierbar. Verschiedene Arten der Dekoration können parallel definiert werden.

Aufbau des Portals (Startseite für nicht eingeloggten Benutzer):



Zentralen Komponenten bzw. Dienste des Portals sind:

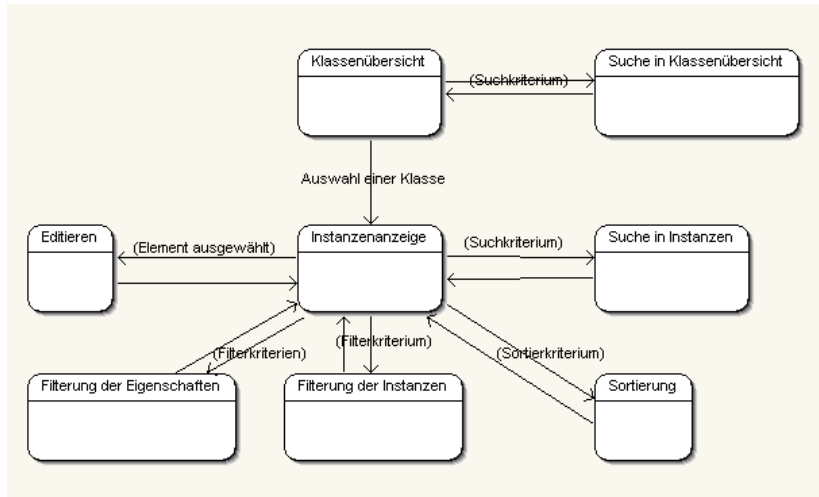
Nutzerregistrierung: Ein Nutzer registriert sich mit Username und Passwort am Portal und wird der Gruppe Gast zugeordnet.

Login: Ein Nutzer meldet sich am Portal an.

Rollenkonfiguration: Ein eingeloggtter Nutzer kann sich die Konfiguration seiner Rolle ansehen.

Ein Portlet ist dabei eine Erweiterung des Servlets, so wie der PortletContainer (bspw. Pluto) eine Erweiterung des Servletcontainers darstellt (bspw. Tomcat). Portlets bilden auf der Clientseite eine einfach zu benutzende Oberfläche innerhalb des Browsers (Fenster mit Schaltflächen zum Maximieren, Minimieren, Editieren, Hilfe). Intern, also auf Serverseite, kann nun eine beliebige Anwendung liegen, die ihre Darstellung auf das Portlet weiterleitet. Sie entsprechen somit einer Sicht im Rahmen des Model View Controller-Konzeptes (MVC).

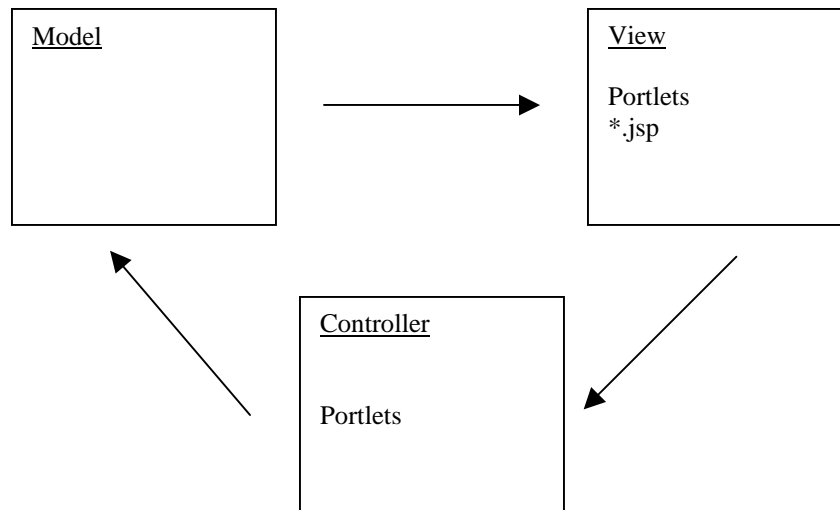
Funktionen der Wissensbasis:



Die Implementierung der Funktionen zur Darstellung und Manipulation der Wissensbasis werden im Folgenden dargestellt.

3. Grundsätzliche Struktur- und Entwurfsprinzipien für das Gesamtsystem

Innerhalb des Projektes wird das Model View Controller-Konzept umgesetzt.



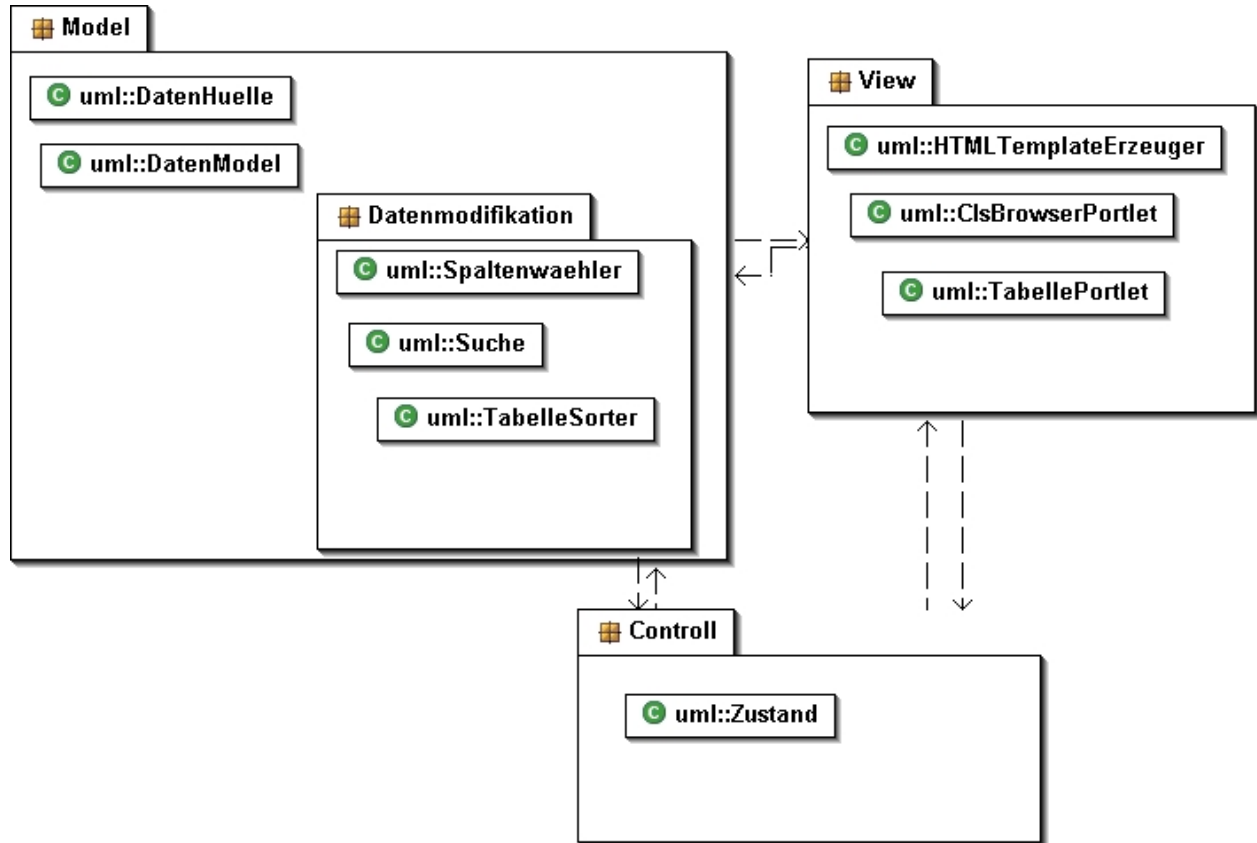
Model: Das Model enthält die Daten sowie die Logik zur Datenhaltung und Operationen auf den Daten

View: Der View präsentiert die Daten.

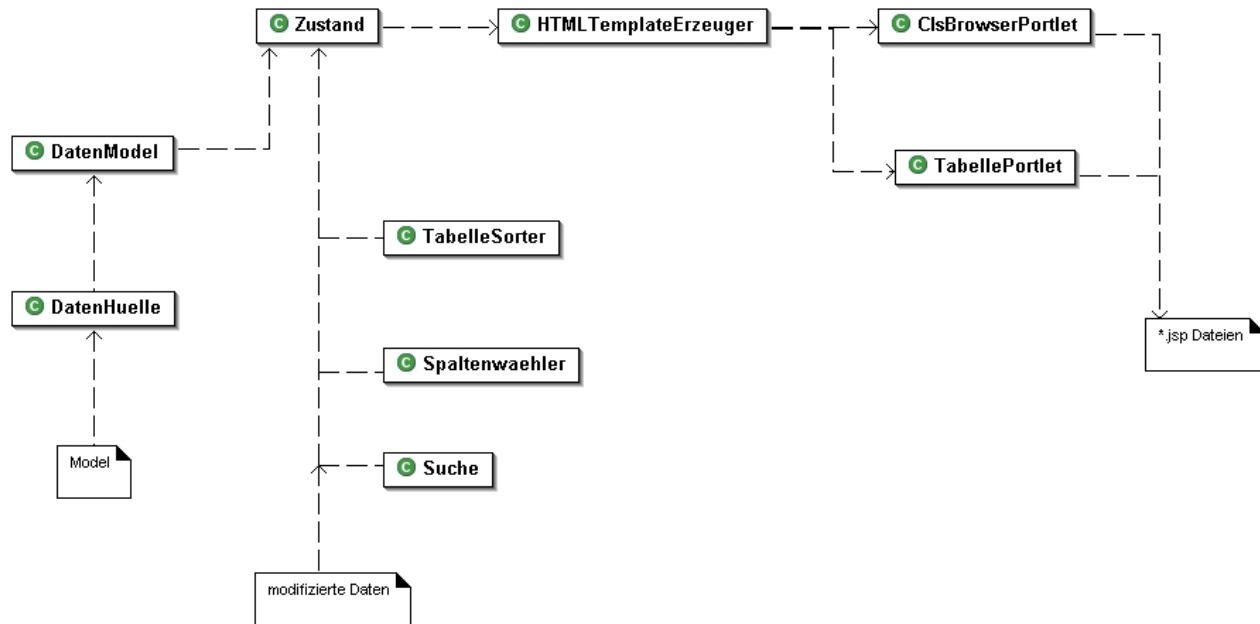
Die Aufgabe des Views übernehmen die Portlets in Verbindung mit dem Portal. Zum Erzeugen der Darstellung werden gsp Dateien verwendet.

Controller: Der Controller ist für die Steuerung des Ablaufes verantwortlich. Als Controller, agieren die Portlets, deren Methoden (processAction() und render(), destroy()) die Anzeige der aktuellen Daten gewährleisten, indem auf eine Benutzeraktion reagiert wird.

Die folgende Abbildung zeigt die Paketstruktur sowie die enthaltenen Klassen:



4. Grundsätzliche Struktur- und Entwurfsprinzipien der einzelnen Pakete



Beschreibung zu den Klassen:

- **Controll:**

Zustand : enthält aktuell relevante Daten, die gerade zur Anzeige, Sortierung, Suche usw. zur Verfügung gestellt werden

- **Model:**

DatenModell: dient als Modell für die Daten der Wissensbasis. Sie hält die aktuell relevanten Daten als Attribute und stellt diese bei Bedarf der Tabelle zur Verfügung.

DatenHuelle: übernimmt die Verbindung von Tabellenmodell und Wissensbasis. Dabei realisiert sie die Abstraktion und Transformation von der konkrete Klasse der Wissensbasis und die Vorformatierung der Daten entsprechend der Implementation des Tabellenmodells.

- **Datenmodifikation:**

TabelleSorter: dient als Dekor für die Tabelle. Sie übernimmt die Sortierung und sorgt für die korrekte Weiterleitung von Aktion der Benutzer in der Tabelle.

Spaltenwaehler : dient als Dekor für die Tabelle. Sie übernimmt das Ein- bzw. Ausblenden von Tabellenspalten.

Suche : dient zum Durchsuchen der Daten (dabei sowohl der Klassen als auch der Instanzen).

Es gibt die Möglichkeit bei der Datenmodifikation, die Daten zur Darstellung zum Beispiel bei der Sortierung oder beim Spalten-ein oder -ausblenden per jsp-Skripten auf dem Client zu modifizieren. Die interne Datenhaltung wird dabei nicht verändert.

- **View:**

HTMLTemplateErzeuger : dient als aktuelle Ausgabe des html Code.

ClsBrowserPortlet und *TabellePortlet*: dient als eine HTML-Code Weiterleitung zwischen HTMLTemplateErzeuger und jsp-Dateien zur Darstellung von Klassen in Baumstruktur und Instanzen in Tabellen.