

Pflichtenheft

Inhaltsverzeichnis

- 1. Zielbestimmung**
 - 1.1. Musskriterien**
 - 1.2. Wunschkriterien**
 - 1.3. Abgrenzungskriterien**
- 2. Produkteinsatz**
 - 2.1. Anwendungsbereiche**
 - 2.2. Zielgruppen**
 - 2.3. Betriebsbedingungen**
- 3. Produktübersicht**
- 4. Produktfunktionen**
- 5. Produktdaten**
- 6. Produktleistungen**
- 7. Qualitätsanforderungen**
- 8. Benutzungsoberfläche**
- 9. Nichtfunktionale Anforderungen**
- 10. Technische Produktumgebung**
 - 10.1. Software**
 - 10.2. Hardware**
 - 10.3. Orgware**
 - 10.4. Produkt-Schnittstellen**
- 11. Spezielle Anforderungen an die Entwicklungsumgebung**
- 12. Gliederung in Teilprodukte**
- 13. Ergänzungen**

1. Zielbestimmung

Die Firma "SoftConsult" verwaltet die vorhandenen Informationen und Wissensbausteine des Unternehmens über ein, in einem Vorprojekt entwickelten Prototyp Plugin InstanceXL für Protegé. Dieser soll zu einer Client-Server-Variante ausgebaut werden.

(Thema: Semantic Web)

1.1. Musskriterien

Ermöglicht wird der Zugriff auf die Wissensbasis über einen Browser. Die Darstellung von Klassen und Instanzen erfolgt in einer Hierarchiedarstellung (Baum) bzw. einer Tabelle der Instanzen. Innerhalb dieser Darstellungen ist das Suchen, Sortieren, Filtern und Editieren möglich.

Der Server soll auf der Basis von Protegé 3.1 die Wissensbasis verwalten. Als Client wird ein Web-Browser verwendet werden, dessen Server-Anbindung über Portlets und die Portlet-Engine Jetspeed-2 realisiert ist.

1.2. Wunschkriterien

- Möglichkeit der Auswahl einer Wissensbasis (Funktion: Öffnen).
- Editieren in den Klassen und der Ontologie.
- Druckfunktion
- Datenexport

1.3. Abgrenzungskriterien

-

2. Produkteinsatz

Der Einsatz ist bei der Firma "SoftConsult" vorgesehen. Ermöglicht wird die Verwaltung der Wissensbasis.

2.1. Anwendungsbereiche

Zum Einsatz kommt die Lösung in einer verteilten Umgebung, um damit die Wissensbasis betrachten und ggf. editieren zu können. Im Client-Server-Betrieb sind gewisse Grundfunktionalitäten von Protegé bzw. des Plugin InstanceXL verfügbar.

2.2. Zielgruppen

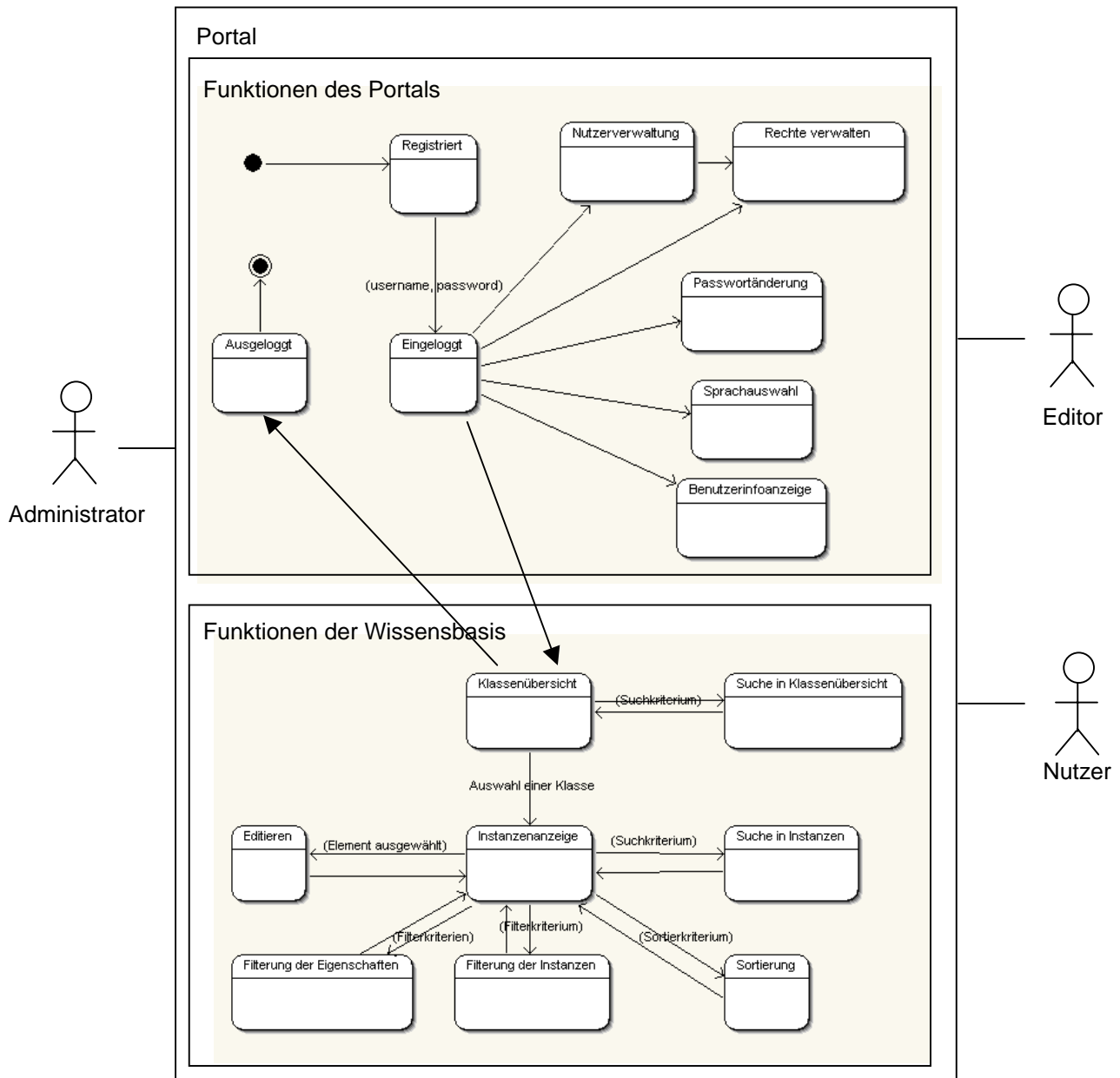
Zum Einsatz kommen wird die Entwicklung bei den Mitarbeitern der Firma "SoftConsult". Zukünftig ist auch eine Auslieferung an die Kunden von "SoftConsult" denkbar.

2.3. Betriebsbedingungen

Mit Hilfe eines Browsers kann der registrierte User (Mitarbeiter) auf das Portal zugreifen, welches die Funktionalität bereitstellt. Diese werden mittels Portlets realisiert.

Die Anwendung wird in einer Büroumgebung eingesetzt.

3. Produktübersicht



Den Mitarbeitern stehen nach Login am Server, je nach zugeordneten Rechten verschiedene Funktionalitäten der bestehenden Implementierung weiterhin zur Verfügung. Über einen Web-Browser können Nutzer lesend auf die Daten der Wissensbasis zugreifen. Editoren können neue Datensätze (Instanzen) anlegen, sowie vorhandene editieren. Der Administrator hat die Möglichkeit die Rechte der Benutzeraccounts.

4. Produktfunktionen

4.1. Funktionen des Portals

/F10/ Anwendungsfall: Registrieren

Akteur: Nutzer, Editor

Beschreibung: Bevor das Einloggen und Nutzen des Portals möglich ist, ist die Registrierung nötig. Dabei wird der Username und ein Passwort spezifiziert. Der User (Nutzer, Editor, Administrator) wird einer Gruppe zugeordnet, um später die Rechteverwaltung zu gewährleisten.

Vorbedingung: -

Auslösendes Ereignis: -

Nachbedingung Erfolg: Ein registrierter User hat die Möglichkeit sich einzuloggen.

Nachbedingung Fehlschlag: Bei fehlerhafter Registrierung erscheint eine Fehlermeldung.

/F20/ Anwendungsfall: Einloggen

Akteur: Nutzer, Editor, Administrator

Beschreibung: Ein User kann sich am Portal (Zugriff über Browser) einloggen (mit Username und Passwort) und die seinen Rechten entsprechenden Funktionen nutzen. Bei einem fehlerhaften Einlogversuch erscheint eine Fehlermeldung.

Vorbedingung: Voraussetzung dafür ist, dass der User am Portal registriert ist.

Auslösendes Ereignis: -

Nachbedingung Erfolg: Im internen Bereich sind die Funktionen der Wissensbasis nutzbar. Es ist die Klassenübersicht sichtbar.

Nachbedingung Fehlschlag: Fehlermeldung

/F30/ Anwendungsfall: Ausloggen

Akteur: Nutzer, Editor, Administrator

Beschreibung: Ein eingeloggter User meldet sich durch Klicken des Logout-Buttons vom Portal ab.

Vorbedingung: Voraussetzung dafür ist, dass der User am Portal eingeloggt ist.

Auslösendes Ereignis: Klick auf Logout-Button

Nachbedingung Erfolg: -

Nachbedingung Fehlschlag: -

/F40/ (FL100) Anwendungsfall: Rechte verwalten

Akteur: Administrator

Beschreibung: Die Administration legt die Zugriffsrechte für die Datensätze fest und legt dafür Benutzer mit entsprechenden Rechten an. Dabei kann ein Editor in den Instanzen lesen und editieren, ein Nutzer hat ausschließlich Leserechte auf die Wissensbasis.

(Grundsätzlich sollen keine Einschränkungen bezüglich bestimmter Klassen oder Instanzen bestehen, d.h. alle Nutzer können alle Klassen und Instanzen sehen.)

Die ausschließlichen Leserechte eines Nutzers beziehen sich nicht auf die Funktionen des Portals (speziell /F50/, /F60/).

Vorbedingung: -

Auslösendes Ereignis: -

Nachbedingung Erfolg: Ein User kann die seinen Rechten entsprechenden Funktionen nutzen.

Nachbedingung Fehlschlag: -

/F50/ (/LF110/) Anwendungsfall: Passwort ändern

Akteur: Nutzer, Editor, Administrator

Beschreibung: Ein User kann sein Passwort ändern. Dabei ist die doppelte Eingabe des neuen Passwortes nötig.

Vorbedingung: Voraussetzung ist, dass der Nutzer eingeloggt ist.

Auslösendes Ereignis: Klick auf Button

Nachbedingung Erfolg: -

Nachbedingung Fehlschlag: Es erscheint eine Fehlermeldung bei fehlerhafter Eingabe (beispielsweise wenn die zweite Eingabe des Passwortes von der ersten abweicht).

/F60/ (/LF120/) Anwendungsfall: Sprache wählen

Akteur: Nutzer, Editor, Administrator

Beschreibung: Auswahl der Sprache der Applikation durch Auswahl aus einer Liste.

Vorbedingung: Voraussetzung ist, dass der Nutzer eingeloggt ist.

Auslösendes Ereignis: Klick auf Button

Nachbedingung Erfolg: Neu laden

Nachbedingung Fehlschlag: -

/F70/ (/LF130/) Anwendungsfall: Nutzerverwaltung

Akteur: Administrator

Beschreibung: Löschen, Anlegen und sperren von Benutzern.

Vorbedingung: -

Auslösendes Ereignis: -

Nachbedingung Erfolg: Nach Anlegen eines neuen Nutzers schließt sich die Rechteverwaltung für den entsprechenden User an.

Nachbedingung Fehlschlag: -

/F80/ (/LF140/) Anwendungsfall: Benutzerinformationen anzeigen

Akteur: Nutzer, Editor, Administrator

Beschreibung: Anzeige aller Informationen (Username, Passwort, ...) des eingeloggten Users.

Nach Anlegen eines neuen Nutzers schließt sich die Rechteverwaltung für den entsprechenden User an.

Vorbedingung: User ist eingeloggt.

Auslösendes Ereignis: Klick auf Button

Nachbedingung Erfolg: Anzeige der Informationen

Nachbedingung Fehlschlag: Fehlermeldung

4.2. Funktionen der Wissensbasis

/F100/ (/LF40/) Anwendungsfall: Klassenübersicht anzeigen

Akteur: Nutzer, Editor, Administrator

Beschreibung: Die vorhandenen Klassen der Wissensbasis werden in Tabellenform angezeigt. Durch Auswahl einer Klasse gelang man zur Detailansicht (Instanzen anzeigen).

Dabei ist lediglich der Zugriff auf eine Default-Wissensbasis möglich. Die Auswahl der Wissensbasis durch den User ist nicht vorgesehen.

Vorbedingung: Eingeloggter User

Auslösendes Ereignis: -

Nachbedingung Erfolg: -

Nachbedingung Fehlschlag: -

/F110/ (/LF50/) Anwendungsfall: Instanzen anzeigen

Akteur: Nutzer, Editor, Administrator

Beschreibung: Nach Auswahl einer Klasse erfolgt eine tabellarische Ansicht der Instanzen und Properties. Dabei repräsentiert eine Zeile jeweils eine Instanz, in den Spalten werden die Eigenschaften angezeigt.

Vorbedingung: Auswahl einer Klasse in der Klassenübersicht

Auslösendes Ereignis: Klick auf ausgewählte Klasse

Nachbedingung Erfolg: Anzeige der Instanzen in Tabellenform

Nachbedingung Fehlschlag: Fehlermeldung

/F120/ (/LF60/) Anwendungsfall: Suchen in der Klassenübersicht

Akteur: Nutzer, Editor, Administrator

Beschreibung: In der Klassenübersicht kann nach einem von User eingegebenen Suchbegriff gesucht werden. Der Suchbegriff (sofern gefunden) wird durch farbige Markierung hervorgehoben.

Vorbedingung: Eingeloggter User

Auslösendes Ereignis: Eingabe Suchbegriff, Klick auf Button

Nachbedingung Erfolg: Farbliche Hervorhebung in der Klassenübersicht

Nachbedingung Fehlschlag: Meldung

/F130/ Anwendungsfall: Suchen in den Instanzen

Akteur: Nutzer, Editor, Administrator

Beschreibung: In der Übersicht der Instanzen kann nach einem vom User eingegebenen Suchbegriff gesucht werden. Instanzen, die den Suchbegriff enthalten werden in einer Tabelle (analog zur Instanzanzeige /F110/) angezeigt.

Vorbedingung: Instanzenansicht

Auslösendes Ereignis: Eingabe Suchbegriff, Klick auf Button

Nachbedingung Erfolg: Neue Tabellenansicht, welche die entsprechenden Suchbegriffe enthält.

Nachbedingung Fehlschlag: Meldung

/F140/ (/LF70/) Anwendungsfall: Tabelle sortieren

Akteur: Nutzer, Editor, Administrator

Beschreibung: Die Tabelle der Instanzen kann pro Eigenschaft (Spalte) sortiert werden.

Es ist jeweils nur ein aktiver Filter zur Zeit möglich. Dabei ist die Sortierung von Einträgen des Typs int, float, String und boolean möglich (lexikographische Sortierung).

Vorbedingung: Instanzenanzeige

Auslösendes Ereignis: Klick auf Button (evt. Spaltenüberschrift)

Nachbedingung Erfolg: Aktualisierte Anzeige der Instanzentabelle

Nachbedingung Fehlschlag: -

/F150/ (/LF80/) Anwendungsfall: Filtern der Eigenschaften

Akteur: Nutzer, Editor, Administrator

Beschreibung: Die Tabelle der Instanzen kann pro Eigenschaft (Spalte) gefiltert werden.

Nach Auswahl der entsprechenden Eigenschaften wird die Tabellenansicht aktualisiert.

Vorbedingung: Instanzenanzeige

Auslösendes Ereignis: Auswahl entsprechender Spalten

Nachbedingung Erfolg: Aktualisierte Anzeige der Instanzentabelle

Nachbedingung Fehlschlag: -

/F160/ Anwendungsfall: Filtern der Instanzen

Akteur: Nutzer, Editor, Administrator

Beschreibung: Die Tabelle der Instanzen kann nach Instanzen (Zeilen) gefiltert werden.

Nach Auswahl des entsprechenden Filterwunsches wird die Tabellenansicht aktualisiert.

Als Filterkriterium ist die Eigenschaft (Spalte) und das gesuchte Filterkriterium anzugeben.

Vorbedingung: Instanzenanzeige

Auslösendes Ereignis: Auswahl Eigenschaft und Spezifizierung des Filterkriteriums

Nachbedingung Erfolg: Aktualisierte Anzeige der Instanzentabelle

Nachbedingung Fehlschlag: Meldung

/F170/ (/LF90/) Anwendungsfall: Instanzen editieren

Akteur: Editor

Beschreibung: Der Editor kann Änderungen (inkl. Löschen und neue Instanzen anlegen)

an den Instanzen im Editiermodus vornehmen. Es ist sicherstellen dass kein zweiter

Zugriff auf den Datensatz zugelassen wird. Nach Änderung wird diese in die

Wissensbasis zurückgeschrieben.

Vorbedingung: Instanzenanzeige

Auslösendes Ereignis: Editiermodus wählen

Nachbedingung Erfolg: Editieren im Editiermodus und zurückschreiben in die

Wissensbasis

Nachbedingung Fehlschlag: -

Die Funktionen /F90/-/F170/ (Funktionen der Wissensbasis) sind ausschließlich dem eingeloggten User zugänglich.

5. Produktdaten

/D10/ Max 5000 Nutzer haben Zugriff auf das Portal.

/D20/ Max 100 Editoren haben Zugriff auf das Portal.

/D30/ Die Wissensbasis enthält max. 5000 Klassen.

/D40/ Eine Klasse enthält max. 5000 Instanzen (Tabellenzeilen). Die Anzeige erfolgt ggf. in mehreren Seiten.

/D50/ Maximale Anzahl der darstellbaren Eigenschaften (Tabellenspalten): 20. Diese unterliegen jeweils den in der Ontologie spezifizierten Restriktionen (Wertebereiche, Datentypen etc.)

6. Produktleistungen

/L10/ Die Funktion /F20/, /F30/, /F50/, /F60/, /F80/ und /F100/ bis /F170/ sollte nicht länger als 20 Sekunde benötigen, idealer Weise aber in ca. 3 Sekunden abgeschlossen sein. (Die Beeinflussbarkeit ist aufgrund der Client-Server-Architektur nicht gegeben, da die Zugriffszeit von der Geschwindigkeit des Netzwerkanschlusses abhängt.)

/L20/ Es können sich max. 200 Nutzern gleichzeitig am Portal einloggen.

7. Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität				
Angemessenheit		x		
Richtigkeit		x		
Interoperabilität		x		
Ordnungsmäßigkeit		x		
Sicherheit		x		
Zuverlässigkeit				
Reife		x		
Fehlertoleranz		x		
Wiederherstellbarkeit		x		
Benutzbarkeit				
Verständlichkeit		x		
Erlernbarkeit		x		
Bedienbarkeit	x			
Effizienz				
Zeitverhalten		x		
Verbrauchsverhalten		x		
Änderbarkeit				
Analysierbarkeit			x	
Modifizierbarkeit		x		
Stabilität			x	
Prüfbarkeit			x	
Übertragbarkeit				
Anpassbarkeit		x		
Installierbarkeit			x	
Konformität		x		
Austauschbarkeit			x	

8. Benutzungsoberfläche

/B10/ Die Benutzeroberfläche ist ein Portal. Die einzelnen Funktionalitäten (bes. die Klassenansicht und die Instanzenansicht) werden mir Hilfe von Portlets realisiert. Der Zugriff erfolgt über einen Browser.

/B20/ Standardmäßige Webbrowser werden unterstützt.

/B30/ Die Benutzung erfolgt hauptsächlich durch Mausbedienung.

9. Nichtfunktionale Anforderungen

-

10. Technische Produktumgebung

10.1. Software

Auf der Serverseite bestehen folgende Anforderungen:

- Java 1.4.2_02 or higher
- JetSpeed-2
- Servlet Engine (Tomcat 5.5.8 or higher)

Auf der Seite der Clients ist ein Browser mit Zugriff auf das Portal nötig.

10.2. Hardware

- Server: - auf welchem JetSpeed und die Wissensbasis (Protege 3.1) laufen
- Client: - mit Grafikbildschirm für die Darstellung und einem Browser für den Zugriff auf das Portal
- Anschluss an ein Netzwerk (lokal bzw. Internet)

10.3. Orgware

Nötig ist ein Netzwerkanschluss mit Verbindung zum lokalen Netzwerk bzw. zum Internet (je nach Erfordernis).

10.4. Produkt-Schnittstellen

Die Interaktion mit dem Benutzer erfolgt über das Portal welches mit JetSpeed-2 realisiert wird und über einen Browser zugänglich ist.

Die Schnittstelle zur Wissensdatenbank wird über die Java-basierte API von Protege realisiert.

11. Spezielle Anforderungen an die Entwicklungsumgebung

-

12. Gliederung in Teilprodukte

Die Funktionen lassen sich in solche, die durch das Portal zur Verfügung gestellt werden (/F10/ - /F80/) und solche, die sich auf die Wissensbasis (/F100/ - /F170/) beziehen unterscheiden. Somit ist eine Implementierung unabhängig voneinander möglich.

13. Ergänzungen

Erweiterungen der Funktionalität in bezug auf die Umsetzung von Protege-Funktionalitäten können in einem weiteren Entwicklungsschritt folgen.

Glossar

Administrator	Verantwortlicher für die Rechte- und Rollenverwaltung
API	Application Programming Interface: Schnittstelle zur Anwendungsprogrammierung
Browser	Programm zum Betrachten von Seiten im Internet.
Client	Ein Client ist eine Anwendung, die den Dienst (den der Client nicht selbst bereit stellt) eines Servers in Anspruch nimmt (Client-Server-Prinzip). Der Client stellt die Benutzerschnittstelle der Anwendung dar. Ein Webbrowser ist wahrscheinlich das bekannteste Beispiel eines Clients. Er stellt eine Anfrage an den Web-Server und gibt dessen Antwort aus. Bei der Datenübertragung ist der Client für die Kontaktaufnahme verantwortlich. Eine Netzverbindung ist demnach erst zum Zeitpunkt der Kontaktaufnahme nötig.
Daten(-sätze)	Bezeichnet die Klassen bzw. Instanzen mit ihren Eigenschaften in der Wissensbasis
Editor	User, mit dem Recht, in den Klassen und Instanzen Änderungen (einschließlich löschen und anlegen) vorzunehmen (einschließlich Leserechte)
Instanz	Repräsentiert in Protege ein Objekt in der Ontologie
JetSpeed-2	Jetspeed ist ein Open Source Enterprise Information Portal auf Basis des JSR 168-Standards das unter Verwendung von Java und XML programmiert wurde. Die einzelnen Seiten des Portals sind aus Portlets aufgebaut. Dies ist die neueste Version von Jetspeed, es beinhaltet Pluto und unterstützt somit die Java-Portlet-Spezifikation.
Klasse	Repräsentiert in Protege die Struktur eines Objektes in der Ontologie
Nutzer	User mit ausschließlich lesenden Rechten auf der Wissensbasis
Ontologie	Unter einer Ontologie versteht man in der Informatik im Bereich der Wissensrepräsentation ein formal definiertes System von Begriffen und/oder Konzepten und Relationen zwischen diesen Begriffen. Zusätzlich enthalten Ontologien zumindest implizit Regeln.
Plugin	Als Plugins bezeichnet man kleine Zusatzprogramme, die den Funktionsumfang eines Basisprogramms erweitern.
Plugin InstanceXL	Plugin für Protege, welches die Individuen einer Klasse in Tabellenform anzeigt sowie Such-, Sortier- und Filterfunktion bietet. Der Einsatz erfolgt als Einzellplatzanwendung.
Portal	Ein Portal ist eine Applikation, welche basierend auf Webtechnologien einen zentralen Zugriff auf personalisierte Inhalte sowie bedarfsgerecht auf Prozesse bereitstellt. Charakterisierend für Portale ist die Verknüpfung und der Datenaustausch zwischen heterogenen Anwendungen über eine Portalplattform. Eine manuelle Anmeldung an den in das Portal integrierten Anwendungen ist durch Single-Sign-On

	nicht mehr notwendig, es gibt einen zentralen Zugriff über eine homogene Benutzungsoberfläche. Innerhalb des Portals können einzelne Portlets Zugang zu unterschiedlichen Informationen und Services bieten.
Portlet	Ein Portlet ist 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 <i>Model View Controller</i> -Konzeptes (MVC).
Properties/Eigenschaften	Die zu einer Instanz verfügbaren Informationen in der Wissensbasis
Protegé	Editor zum Erstellen, Navigieren und Verarbeiten von Ontologien
Rolle	Zusammenfassung von Funktionen, die ein Nutzer (bzw. eine Nutzergruppe) ausführen darf, dem diese Rolle zugeteilt wurde.
Semantic Web	Das Semantische Web stellt eine Erweiterung des World Wide Web (WWW) dar. Daten können in maschinenlesbarer Form zur Verfügung gestellt werden. Der Inhalt von Dokumenten wird (über Metadaten) formal spezifiziert, wodurch intelligentere und effizientere Zugriffe auf Daten ermöglicht werden.S
Server	Der Server ist ein Programm, welches nach der Kontaktaufnahmen durch einen Client mit diesem Nachrichten austauscht. Das verwendete Protokoll bestimmt dabei Regeln und Format des Nachrichtenaustausches. (Beispiele für Protokolle: http, ftp, nfs)
Servlet	Als Servlets bezeichnet man Java-Klassen, deren Instanzen innerhalb eines Webservers Anfragen von Clients entgegen nehmen und beantworten. Solche Klassen müssen immer die Schnittstelle „javax.servlet.Servlet“ oder eine davon abgeleitete (normalerweise „javax.servlet.http.HttpServlet“) implementieren. Der Inhalt der Antworten kann dabei dynamisch, also im Moment der Anfrage, erstellt werden und muss nicht bereits statisch (etwa in Form einer HTML-Seite) für den Webserver verfügbar sein. Servlets stellen somit das Java-Pendant zu CGI-Skripten oder anderen Konzepten, mit denen dynamisch Web-Inhalte erstellt werden können (PHP etc.), dar.

Tomcat	Apache Tomcat stellt eine Umgebung zur Ausführung von Java-Code auf Webservern bereit, die im Rahmen des Jakarta-Projekts der Apache Software Foundation entwickelt wird. Es handelt sich um einen in Java geschriebenen Servlet-Container, der mithilfe des JSP-Compilers Jasper auch JavaServer Pages in Servlets übersetzen und ausführen kann. Dazu kommt ein kompletter HTTP-Server.
User	Wird hier zusammenfassend für die Begriffe Nutzer, Editor, Administrator verwendet
Wissensbasis	Bezeichnet (in Abgrenzung zur Ontologie) die Daten