

Softwaretechnik-Praktikum SS 2005 Aufgabenblatt 2

Recherchebericht

1. Begriffe

Im Folgenden werden projektrelevante Begriffe zusammengefasst, welche später in ein Glossar aufgenommen werden sollen. Diese Übersicht erhebt noch keinen Anspruch auf Vollständigkeit.

Protégé: ¹⁾	Editor zum Erstellen, Navigieren und Verarbeiten von Ontologien
Semantic Web: ¹⁾	Erweiterung des World Wide Web (WWW) durch maschinenlesbare logisch / semantisch modellierte Daten, z.B. auf Basis einer Ontologie, erlaubt hohen Grad an automatischer Verarbeitung
Ontologie: ¹⁾	formal definiertes System von Dingen und/oder Konzepten und Relationen zwischen diesen Dingen
Wissensbasis: ¹⁾	Grundlage von Informationen zur Weiterverarbeitung für Programme
Plugin: ¹⁾	Software, die über eine definierte Schnittstelle auf die Architektur einer Applikation aufsetzt und deren Möglichkeiten erweitert
RDF: ¹⁾	Resource Description Framework, Auszeichnungssprache für Metadaten basierend auf Tripel (Subjekt, Prädikat, Objekt)
OWL: ¹⁾	Web Ontology Language, Spezifikation des World Wide Web Consortium um Ontologien anhand formaler Beschreibungssprache zu erstellen, publizieren und verteilen
XML: ¹⁾	Extensible Markup Language, Metasprache für das Definieren von Dokumententypen
Klasse: ²⁾	Struktur eines Objektes in der Ontologie
Slot: ²⁾	Eigenschaft eines Objektes in der Ontologie
Instanz: ²⁾	repräsentiert Objekt in der Ontologie
Facetten: ²⁾	Eigenschaften eines Slots
Form: ²⁾	Eingabemaske und gleichzeitig die Ausgabeformatierung für die Instanzen einer Klasse
TabWidget: ²⁾	Protégé-Plugin um einfache Anfragen an Wissensbasis zu stellen oder auf API der Wissensbasis zuzugreifen
SlotWidget: ²⁾	Protégé-Plugin zur grafischen Repräsentation
Backend Plugin: ²⁾	Protégé-Plugin zur Speicherung/Ablegen von Daten der Wissensbasis in Filesystem oder Datenbank ohne grafische Oberfläche

Begründungen für die Aufnahme in die Liste:

¹⁾ Terminologie zur Problemstellung von „SoftCosult“

²⁾ Fachbegriffe der Rahmenapplikation

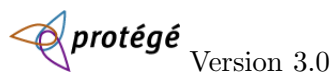
2. Konzepte

Folgende grundlegende Konzepte sollten bekannt sein und werden vermutlich für ein erfolgreiches Abschließen des Projekts benötigt.

- Plugin-Konzept** Hierbei handelt es sich um ein gängiges Konzept zur Erweiterung von Softwareprodukten durch neue Funktionalität. Dazu muß die Rahmenapplikation, die erweitert werden soll, eine (im besten Fall dokumentierte) Schnittstelle bieten, die es ermöglicht Software zu entwickeln, die bestimmte Dienste der Rahmenapplikation nutzt oder neue Dienste über die definierte Schnittstelle bereitstellt. Beispiele für Plugins sind z.B. erweiterungsfähige Webbrowser oder Eclipse. Für unsere Applikation wird Protégés AbstractTabWidget als Schnittstelle implementiert.
- XML** Die Extensible Markup Language, abgekürzt XML, ist ein Standard zur Erstellung maschinen- und menschenlesbarer Dokumente in Form einer Baumstruktur. XML definiert dabei die Regeln für den Aufbau solcher Dokumente. Für einen konkreten Anwendungsfall müssen die Details der jeweiligen Dokumente spezifiziert werden. Dies betrifft insbesondere die Festlegung der Strukturelemente und ihre Anordnung innerhalb des Dokumentenbaums. XML ist insbesondere Wichtig zum Parsen / Erstellen von Dateien im RDF / OWL-Format. Diese bilden die konzeptionelle Grundlage zur Speicherung und Distribution von Ontologien und Meta-Daten für das Semantic Web. Für unsere Applikation wird es erforderlich sein, Daten nach RDF zu konvertieren.
- RDF** Das Resource Description Framework (RDF) ist eine Spezifikation für ein Modell zur Repräsentation von Metadaten, die erstmals 1999 vom World Wide Web Consortium vorgelegt wurde. Am weitesten verbreitet ist die Serialisierung von RDF in XML. In Verbindung mit RDF-Schema und der Web Ontology Language (OWL) dient RDF als grundlegendes Format zur Repräsentation von Ontologien.
- OWL** Die Web Ontology Language (OWL) ist eine Spezifikation des W3C, um Ontologien anhand einer formalen Beschreibungssprache erstellen, publizieren und verteilen zu können. Dabei werden Terme einer Domäne und deren Beziehungen formal so beschrieben, dass die Semantik automatisch verarbeitet werden kann. OWL ist daher ein wesentlicher Bestandteil des Semantic Webs.
- Introspektion** Unter Introspektion (auch bekannt als Reflexion) versteht man die Möglichkeit eines Programms, Erkenntnisse seiner eigenen Struktur zu erkennen. Introspektion/Reflexion ermöglicht das Abfragen von Informationen über Klassen oder deren Instanzen zur Laufzeit. Bei einer Methode sind das u.a. deren Sichtbarkeit, die Art des Rückgabewertes oder die Art der Übergabeparameter. Die Umsetzung ist dabei sprachspezifisch realisiert. Introspektion wird ein wichtiges Konzept zur Erstellung der Instanzentabelle einer Klasse in unserer Applikation sein.

Client-Server	Das Client-Server-Konzept ermöglicht das Bereitstellen von zentralen Diensten, die von verschiedenen Benutzern an verschiedenen Orten (auch gleichzeitig) in Anspruch genommen werden können. Der Server stellt einen solchen Dienst bereit, während die Clients diesen Dienst benutzen. Die Programmlogik liegt dabei auf dem Server. Häufige Anwendungsbeispiele sind das Bereitstellen von Dateien an einem zentralen Ort (Fileserver) oder aber auch das Internet. Das Client-Server-Konzept wird eventuell in einer späteren Version unserer Applikation eingesetzt um einen Zugriff auf die Wissensbasis im Internet zu realisieren.
Standalone-Application	In einer Standalone-Application liegt die Programmlogik und Datenhaltung, sowie die Benutzeroberfläche im Gegensatz zum Client-Server-Konzept auf einem lokalen Rechner. Hierbei ist es nicht möglich, daß mehrere Benutzer gleichzeitig mit der Applikation arbeiten können. Unsere Applikation wird in der ersten Stufe dieses Konzept verwenden.
MVC	Beim MVC-Modell handelt es sich um ein Entwurfsmuster zur Trennung bestimmter Programmeigenschaften. Die Idee des Musters ist die Trennung des Programms in die drei Einheiten Datenmodell (Model), Präsentation (View) und Programmsteuerung (Controller). Ziel des 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.
Rollenkonzept	Unter Rollenkonzept versteht man das Verteilen bestimmter Aufgaben oder Benutzerrechte auf bestimmte Personen oder Personengruppen. Die Rollen und Rechte werden dabei entsprechend der Qualifikation und des Status der Person oder Personengruppe erteilt. Ein gängiges Beispiel findet man in einem Redaktionssystem, hier darf ein Benutzer nur Artikel lesen, während ein Autor berechtigt ist, neue Artikel zu verfassen. In der Realen Welt ist der Benutzer entsprechend ein Interessent, der Informationen sucht, während der Autor ein Mitarbeiter der informationsgebenden Gesellschaft ist. Für eine Webapplikation ist dieses Konzept von hoher Bedeutung.

3. Beschreibung der Rahmenapplikation



Quelle:

- <http://protege.stanford.edu>

Leistungsparameter:

- Ontologie-Editor
- Editor zum Erstellen, Navigieren und Verarbeiten von Ontologien
- Open-Source
- erweiterbar durch Plugins

- Unterstützung vielfältiger Darstellungsformate, z.B.
 - OWL
 - RDF
 - XML
 - UML
- Multiuserfähigkeit (Standalone als auch Server)

Architektur:

- Java-Plattform
- Swing-GUI

Einsatzgebiet:

- Wissensrepräsentation

3.1 Verbale Beschreibung der Leistungsmerkmale

Protégé 3.0 ist ein Editor zum Erstellen, Navigieren und Verarbeiten von Ontologien, der von der Stanford Universität (<http://protege.stanford.edu>) entwickelt und publiziert wurde. Ursprünglich wurde Protégé zur Wissensrepräsentation in der Medizininformatik eingesetzt. Mittlerweile wird es auch in anderen Bereichen eingesetzt.

Protégé 3.0 ist eine OpenSource Software basierend auf der Java-Plattform und einer Swing-GUI zur Gestaltung der grafischen Oberfläche. Des weiteren lässt sich Protégé mit Plugins in Form von zu implementierenden Java-Interfaces erweitern und anpassen. Durch schon existierende Plugins ist auch eine Unterstützung vielfältiger Darstellungsformate von Ontologien wie z.B. OWL, RDF und UML möglich.

In Protégé werden die Konzepte aus dem zu modellierenden Weltausschnitt in Klassen dargestellt. Slots stellen dazu die Attribute der Klassen dar und die Facetten deren Eigenschaften.

Protégé ist seit der Version 2.0 mutliuser-fähig. Hierfür wird ein Protégé-Server benötigt auf dem die Wissensbasis liegt. Die Protégé-Clients führen einen Abgleich in Echtzeit durch und können so immer an der aktuellsten Wissensbasis arbeiten.

3.2 Beschreibung der fundamentalen Anwendungsfälle

Geschäftsprozess:	Ontologieeditierung: Von der Erstellung bis zur Bearbeitung
Akteur:	Autor
Beschreibung:	Der Autor legt eine Ontologie mit Klassen, Slots, Relationen in Protégé an und kann diese später bearbeiten.
<hr/>	
Geschäftsprozess:	Wissensbasispflege - Von der Eingabe bis zum Editieren
Akteur:	Autor
Beschreibung:	Der Autor gibt Wissen in Form von Instanzen ein und kann sie ändern.
<hr/>	
Geschäftsprozess:	Abfrage
Akteur:	User
Beschreibung:	Der User führt eine Abfrage (Query) auf einer vorhandenen Wissensbasis / Ontologie durch.
<hr/>	
Geschäftsprozess:	Ontologienavigation
Akteur:	User
Beschreibung:	Der User kann die Ontologie mit Hilfe von Links / Relationen traversieren.

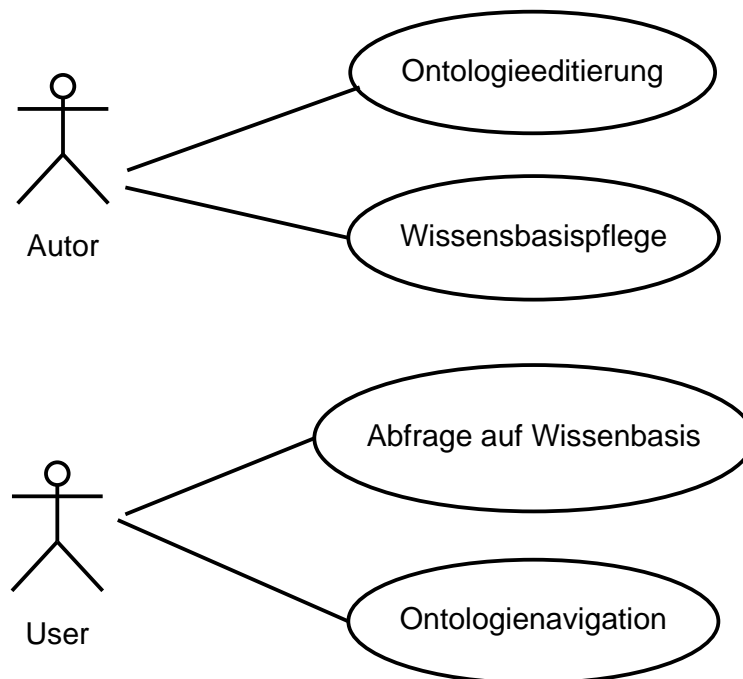


Abbildung 1: Grafische Darstellung der fundamentalen use-cases.

3.3 Darstellung äußerlich sichtbarer Aspekte

Nachdem man Protégé 3.0 (Basic Installation) gestartet hat, öffnet sich ein Dialog-Fenster. Hier kann man wählen, ob man ein neues Projekt starten oder ein vorhandenes öffnen möchte. Sowohl bei einem neuen Projekt als auch bei einem vorhandenen öffnet sich die gesamte Protégé-Oberfläche. Am oberen Bildschirmrand findet man die Haupttaskleiste mit den Menüpunkten File, Edit, Project, Window und Help. Des weiteren findet man standardmäßig fünf Registerkarten mit der Beschriftung Class, Slots, Forms, Instances und Queries.

Wenn man ein neues Projekt gestartet hat, muss man immer zuerst Klassen anlegen um später Instanzen hinzufügen und Slots anlegen zu können. Wenn sowohl Klassen, Instanzen als auch Slots angelegt sind, kann man über Queries Anfragen starten, die durch das betätigen des „Find“-Buttons ausgeführt werden. Ergebnisse werden auf der rechten Seite auf der Query-Registerkarte angezeigt. Jederzeit kann das Projekt gespeichert und/oder beendet werden. Auch das Löschen einzelner Komponenten ist jederzeit möglich.

Beim Start eines vorhandenen Projektes kommt es darauf an, welche Komponenten schon vorhanden sind, um die verschiedenen Handlungen durchzuführen. Sind schon Klassen vorhanden, kann natürlich sofort eine Instanz und die dazugehörigen Slots angelegt werden. Sind schon Instanzen vorhanden, ist das Anlegen von Slots sofort möglich. Das Abfragen per Queries ist immer erst möglich, wenn Klassen, Instanzen und Slots angelegt wurden bzw. vorhanden sind.

Das entstandene Projekt lässt sich nicht nur im eigenen Protégé-Format speichern, sondern auch als HTML-Datei exportieren.

Nachfolgend ein grafischer Ablaufplan, der die wichtigsten Schritte im Umgang mit der Rahmenapplikation beinhaltet.

