

Softwaretechnik-Praktikum SS 2005

GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

Recherchebericht

Der Recherchebericht erläutert themenrelevante **Begriffe** und **Konzepte**, die im Rahmen der Projektaufgabe bedeutend sind. Weiterhin wird die **Rahmenapplikation** Protégé vorgestellt. (Da unser Team Verstärkung bekommen hat, folgt zum Schluß noch eine aktualisierte Rollenverteilung.)

1. Begriffe

Im folgenden ist die erste Version des Glossar gezeigt. Er enthält projektspezifische und mißverständliche Begriffe. Alle Begriffe wurden in den Glossar aufgenommen, um Klarheit in der Kommunikation innerhalb der Gruppe und mit dem Kunden zu schaffen.

Glossar

- **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 intelligenter und effizientere Zugriffe auf Daten ermöglicht werden.
(Dieser Begriff wurde in den Glossar aufgenommen, da er als aktuelles Schlagwort gewissen Unklarheiten oder Mißdeutungen unterliegt.)
- **Ontologie:** Im Rahmen der KI-Forschung Modellierung von Domänen der realen Welt mit dem Ziel eines strukturierten und fundierten Aufbaus von Wissensbasen, damit rechnergestützt maximales Wissen generiert werden kann. Ontologien ermöglichen die Festlegung eines einheitlichen Vokabulars und die Konzeptualisierung komplexen Wissens.
(Dieser Begriff wurde in den Glossar aufgenommen, um ihn gegenüber dem Begriff der Ontologie (Seinslehre) in der Philosophie abzugrenzen.)
- **Ontologiesprache:** Ontologien können mit Hilfe verschiedener (maschinenlesbarer) Sprachen abgebildet werden. Hauptsächlich handelt es sich hierbei um XML, RDF und OWL, sowie deren Unterkategorien und Erweiterungen.
(Dieser Begriff wurde in den Glossar aufgenommen, da die Standardisierung im Rahmen des Semantic Web eine entscheidende Rolle spielt, und diese Vereinheitlichung über klar spezifizierte Sprachen erst ermöglicht wird.)
 - **XML** (eXtensible Markup Language): ist ein Standard zur Erstellung maschinen- und menschenlesbarer Dokumente in Form einer Baumstruktur.
 - **XML Schema** : XML Schema ist eine Empfehlung des W3C zum Definieren von XML-Dokumentstrukturen.
 - **OWL** (Web Ontology Language): OWL ist eine Spezifikation des W3C, um Ontologien anhand einer formalen Beschreibungssprache erstellen, publizieren und verteilen zu können.
 - **OWL Lite:** Die "Light-Version" wurde mit dem Ziel geschaffen, eine einfach zu implementierende Untermenge der Sprache zur Verfügung zu stellen. Sie dient vor allem zum Erschaffen einfacher Taxonomien und leicht axiomatisierter Ontologien.
 - **OWL DL** (Description Logic): Dies ist die Ebene, deren Semantik noch am ehesten an DAML+OIL (ein Vorgänger von OWL) heranreicht. DL steht für Beschreibungslogik, einer entscheidbaren Untermenge der Prädikatenlogik erster Stufe.
 - **OWL Full:** OWL Full besteht aus den selben Sprachkonstrukten wie OWL DL, verzichtet aber auf die dort vorhandenen Einschränkungen. Dadurch

Softwaretechnik-Praktikum SS 2005

GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

sind die Ontologien unentscheidbar, können dafür aber prädikatenlogische Ausdrücke höheren Grades ermöglichen.

- **RDF** (Resource Description Framework): RDF ist eine Spezifikation für ein Modell zur Repräsentation von Metadaten, die erstmals 1999 vom World Wide Web Consortium vorgelegt wurde.
- **RDFS** (RDF Schema): RDFS ist ein Vokabular zur Formulierung Ontologien in RDF.
- **Metadaten:** Als Metadaten oder Metainformationen bezeichnet man allgemein Daten, die Informationen über andere Daten enthalten.
(Dieser Begriff wurde in den Glossar aufgenommen, da er im Bereich des Semantic Web eine wesentliche Rolle einnimmt.)
- **Plugin:** Als Plugins bezeichnet man kleine Zusatzprogramme, die den Funktionsumfang eines Basisprogramms erweitern.
(Dieser Begriff wurde in den Glossar aufgenommen, das es sich bei dem zu erstellenden Produkt um ein Plugin handelt.)

2. Konzepte

(a) Ontologien

- **Überblick**

Ontologien stellen ein Vokabular zur Verfügung, welches zur Modellierung einer Wissensdomäne eingesetzt werden kann. Dabei können Objekte mit ihren Eigenschaften und ihren Beziehungen untereinander dargestellt werden.

- **Vorteile**

- Domänenwissen wird für Mensch und Maschine lesbar festgehalten
- Domänenwissen kann - einmal modelliert - beliebig oft und in verschiedenen Kontexten wiederverwendet werden
- Während der Modellierung gibt es eine eindeutige Trennung von operationalem Wissen und Domänenwissen
- Domänenwissen ist analysierbar

- **Woraus setzt sich eine Ontologie im Allgemeinen zusammen?**

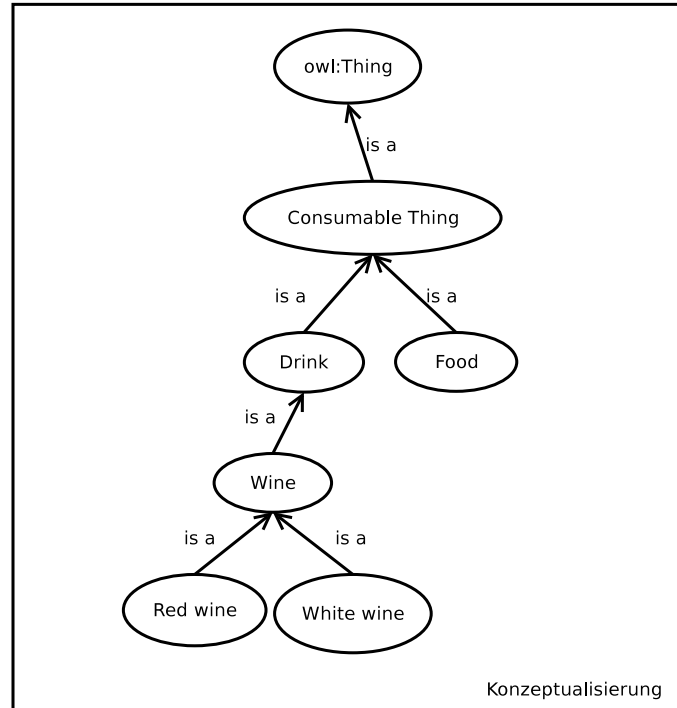
- Konzepte (concepts): Allgemein: Begriffe, die zu einer Ontologie zusammengeführt werden sollen.
- Konzepthierarchie (concept hierarchy): Anordnung der Begriffe.
- Konzepteigenschaften (concept properties): Eigenschaften, die das jeweilige Konzept/Begriff besitzen kann.
- Wertebereiche der Konzepteigenschaften (facets): Beschränkungen, die auf den Konzepteigenschaften liegen.
- Instanzen (instances): Ausprägungen eines Konzeptes (einer Klasse).

Softwaretechnik-Praktikum SS 2005

GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

Folgende Abbildung zeigt einen Ausschnitt aus einer Beispielontologie, die sich vornehmlich mit Speisen und Getränken befaßt:



- **Welche Schritte sind zur Erstellung einer Ontologie notwendig?**

- Die Anwendung der Ontologie und die Domäne müssen bestimmt werden.
- Bestehende Ontologien können - falls vorhanden - genutzt oder eingebunden werden.
- Die wichtigen Bestandteile der Ontologie werden gesammelt.
- Es werden die Konzepte und die Konzepthierarchie festgelegt.
- Es werden die Konzepteigenschaften definiert.
- Die Wertebereiche der Konzepteigenschaften werden festgelegt.
- Die Instanzen werden erzeugt.

Das Erstellen einer Ontologie ist stets ein *iterativer* Prozess, d.h. es können mehrere Schritte in dieser Abfolge mehrfach durchlaufen werden. Eine festgelegtes Rezept für die perfekte Ontologie gibt es nicht. Das Gelingen hängt auch von weiteren Fragestellungen ab, z.B. welchen Grad von Erweiterbarkeit soll die Ontologie haben oder wieviel Domänenwissen steht zum Zeitpunkt der Modellierung zur Verfügung.

- **Anwendung**

Ontologien können in verschiedenen Bereichen angewendet werden:

- Neutral Authoring: Anwendungsunabhängige Beschreibung von Informationen
- Informationszugang ohne Grenzen: Ontologien können als Wissensvermittler zwischen Mensch und Maschine dienen
- Bibliothekswissenschaft, Information Retrieval und Erweiterung herkömmlicher Suchmaschinen
- Im Rahmen von KI-Anwendungen (Künstliche Intelligenz)
- Verarbeitung von natürlicher Sprache
- Semantic Web

Softwaretechnik-Praktikum SS 2005

GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

(b) Semantic Web**• Historie und Motivation**

Die Bezeichnung 'Semantic Web' ist einem gewissen medialen Hype ausgesetzt gewesen - unter anderem wurde und wird vom Semantic Web als dem Nachfolger des herkömmlichen World Wide Web (WWW) gesprochen. Dies ist für sich genommen nicht richtig. Es handelt sich, falls man sich explizit auf das WWW bezieht, eher um eine Bereicherung.

Unter Semantic Web versteht man eine Zusammenfassung von Bemühungen, Spezifikationen und Technologien, die die Möglichkeiten der Wissensmodellierung und der Mensch-Maschine-Interaktion erforschen und befördern. Der Begriff selbst geht auf einen Gründervater des WWW - Tim Berners-Lee - zurück. Die Vision ist, daß Computer den Menschen verstehen lernen (und weniger umgekehrt).

Eine Initiative des W3C (WWW Consortium) beschäftigt sich seit 2001 mit der Formulierung der Ziele und der Etablierung von Standards, die das Semantic Web betreffen (<http://www.w3.org/2001/sw>).

Die entscheidende Motivation ist der Mangel an strukturierter Information im globalen Netzwerk. Zwar sind theoretisch Millionen von Ressourcen (Texte, Applikationen, Multimediale Inhalte, Datenbanken, etc.) verfügbar, doch läßt sich dieser unerschöpfliche Datenvorrat (und auch Wissensvorrat) meist nur syntaktisch mit Hilfe von Suchmaschinen durchsuchen. Inhaltliche Aspekte können mit einer herkömmlichen Suchmaschine nicht erfaßt werden. Die Vision des Semantic Web ist die Suche nach Ressource über inhaltliche Aspekte. Suchanfrage werden effektiver und genauer. Das automatische Ausschöpfen von Wissen und das Extrahieren von Informationen wird möglich und kann automatisiert werden. Allerdings müssen dafür die jeweiligen Ressourcen entsprechend gekennzeichnet, bzw. mit Metainformationen versehen werden. Diese Einschränkung ist der Preis für die tatsächlich vielversprechende Vision des Semantic Web.

• Realisierung

Das Semantic Web basiert auf einer Reihe von Komponenten und Ansätzen:

- Sprachen, die Metainformationen über eine Ressource in maschinenlesbarer Form bereitstellen (z.B. RDF)
- Sprachen, die Wissen - also Begriffe und Beziehungen - in maschinenlesbarer Form bereitstellen (z.B. OWL)
- Wissensbasen. Modelliertes Wissen einer bestimmten Domäne (z.B. Taxonomie der Pflanzen, Komponenten eines Autos eines bestimmten Fabrikates)
- Applikationen zur Erstellung und Bearbeitung von Wissensbasen (z.B. Ontologie-Editoren (Protégé, etc.))
- Applikationen zur Ausschöpfung von Wissensbasen und für das Automated Reasoning (z.B. RACER)
- Initiativen zur Standardisierung von Vorgehensweisen (z.B. W3C)

Die Zusammenführung dieser und anderer Techniken und Technologien des Semantic Web wird letztendlich über den Erfolg dieses Konzeptes entscheiden.

(c) RDF und OWL**• RDF (Resource Description Framework)**

Mit Hilfe von RDF können Informationen über (Web-)Ressourcen repräsentiert werden. Insbesondere können Metadaten über Webseiten in RDF abgelegt

Softwaretechnik-Praktikum SS 2005

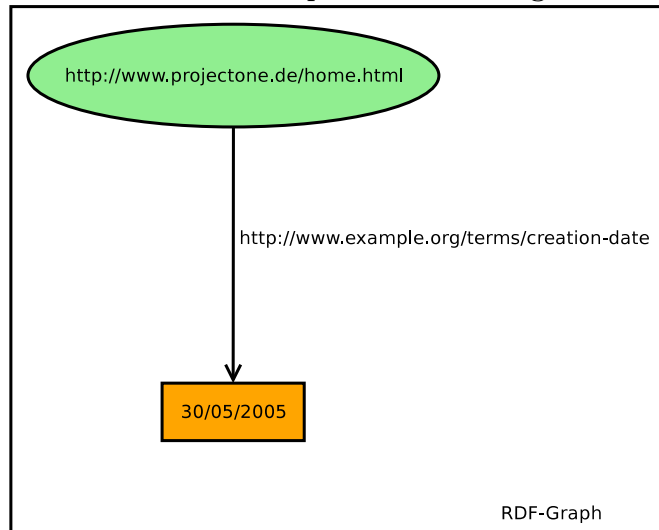
GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

werden, wie z.B. Autor der Seite, Sprache, Datum usw. Applikationen können auf diese Informationen zugreifen und sie beispielsweise bei Suchanfragen berücksichtigen. RDF basiert dabei auf der Idee, Webressourcen über einen URI (Unified Resource Identifier) zu identifizieren.

RDF-Daten können als Graph veranschaulicht werden oder in XML-Syntax gespeichert werden.

Ein einfacher RDF-Graph könnte wie folgt aussehen:



Ein einfacher RDF/XML Ausschnitt einer Webseite könnte folgendermaßen aussehen:

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/20-rdf-syntax-ns#"
xmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#">
<contact:Person rdf:about="www.projectone.de/team/KS/contact#me">
<contact:fullName>Karla Smith</contact:fullName>
<contact:mailbox rdf:resource="mailto:ks@our-project.net" />
</contact:Person>
</rdf:RDF>
```

RDF Statements bestehen aus einem Triplet, wobei ein Triplet (ähnlich einem natürlichsprachlichem Ausdruck) aus Subjekt, Prädikat und Objekt besteht. Beispielsweise könnte man sagen: "Das *Erstellungsdatum* von *http://www.projectone.de* ist der *30.05.2005*". Oder anders formuliert: Die Webseite (*http://...*) hat eine Eigenschaft (*Erstellungsdatum*), deren Wert (*30.05.2005*) bekannt ist.

Subjekte und Prädikate müssen als URI notiert werden. Objekte können als Literale oder als URI notiert werden.

Mit Hilfe von RDF können also Ressourcen beschrieben werden (auch RDF Statements selbst können als Ressourcen verwendet werden), und liefern somit eine erste Beschreibungsebene des Semantic Web.

- **OWL (Web Ontology Language)**

OWL ist eine Sprache, die auf die inhaltliche Verarbeitung von Informationen unterstützt. OWL kennt drei Sprachebenen, die in ihrer Ausdrucksstärke verschieden sind: OWL Lite, OWL DL, OWL Full.

OWL ermöglicht die Festlegung von Vokabular und Beziehungen. Beides zusammen wird auch als Ontologie bezeichnet.

OWL beginnt, wo die Möglichkeiten von RDF enden. OWL erweitert die Möglichkeiten, Klassen, Eigenschaften und ihre Beziehungen darzustellen.

Softwaretechnik-Praktikum SS 2005

GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

(d) Plugins**• Was sind Plugins?**

Plugins sind Softwaremodule die die Funktionalität einer Basisapplikation erweitern. Plugins bieten folgende Vorteile:

- Plugins können meist einfach in die Basisapplikation eingebunden werden
- Plugins können die Funktionalität einer Basisapplikation für konkrete Anforderungen erweitern
- Plugins können auch ohne ein umfassendes Kenntnis der Implementierung der Basisapplikation entwickelt werden
- Die meisten Applikationen erlauben das Laden von Plugins zur Laufzeit, wodurch das Leistungsverhalten der Applikation erhalten bleibt

• Plugins für Protégé

Die Protégé-Architektur ermöglicht es, die Vorteile von Plugins in vollem Umfang zu nutzen. Plugins werden von Protégé genutzt, um die vielfältigen Aufgaben, die im Rahmen der Erstellung einer Ontologie erforderlich sind, zu modularisieren. So stehen neben den Standard-Plugins für das Editieren von Klassen und Instanzen eine Reihe verschiedener Erweiterungen zu Verfügung, unter anderem folgende:

- Jambalaya: Graphische Darstellung der Hierarchie einer Ontologie
- OWL-Wizard: Unterstützung für die Erstellung von OWL-Ontologien
- PAL (Protégé Axiom Language): Festlegen von Beschränkungen (constraints) über einer Ontologie
- Prompt: Verwaltung von Ontologien
- PSM Librarian: Möglichkeiten zur Wiederverwendung von Ontologien und Standardisierung von Modellierungsansätzen
- RDF Backend: Unterstützung für das RDF-Format
- String Search: Suche nach Zeichenketten
- XML: Unterstützung für das XML-Format

3. Beschreibung der Rahmenapplikation (Protégé)**(a) Background und Leistungsmerkmale**

Protégé ist ein frei verfügbarer Ontologie-Editor. Er wurde und wird an der Stanford University School of Medicine von den Stanford Medical Informatics entwickelt. Die Entwicklung des Projekts wird von staatlichen Einrichtungen und Unternehmen unterstützt, unter anderem von der Defense Advanced Research Projects Agency (DARPA) und DaimlerChrysler.

Mit Protégé können Ontologien erstellt und bearbeitet werden. Ontologien haben im Zuge der Entwicklung und Erforschung des Semantic Web einen Aufschwung erfahren. Sie stellen formale Konzepte zur Modellierung von Wissen dar.

Die Architektur von Protégé folgt dem Plugin-Konzept. Die Grundfunktionalität wird von einer Basisversion zur Verfügung gestellt, weitere Features können über Plugins eingebunden werden. Es stehen bereits eine Reihe von Plugins für verschiedene Aufgaben zur Verfügung, so für den Import und Export verschiedener Dateiformate, für die graphische Darstellung oder Visualisierung von Daten, Editor- und Spracherweiterungen.

Protégé ist in Java geschrieben und ist somit auf allen Plattformen, für die eine Java Virtual Machine verfügbar ist, einsetzbar.

Softwaretechnik-Praktikum SS 2005

GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

Ein ursprüngliches Einsatzgebiet von Protégé ist das Einsatzgebiet aller Ontologie-Editor: Die Erstellung von Ontologien. Diese selbst stellen keinen Wert für sich dar, sondern gewinnen erst im Einsatz in verschiedenen Domänen an Bedeutung. Ist einmal eine konsistente Modellierung von Wissen einer Domäne vorhanden, können Daten genauer, aber auch assoziativer gefunden. Eingesetzt werden kann Protégé auch für die Überprüfung bereits vorhandener Modelle, die Veranschaulichung von Ontologien und für den Import und Export von Daten.

Protégé kann als Stand-alone-Applikation oder als Server eingesetzt werden. Letztere Anwendung eröffnet die Möglichkeit, Wissensbasen verteilt einzusetzen, was besonders für Unternehmen oder Einrichtungen, die über ein Netzwerk verfügen, interessante Optionen eröffnet.

(b) Use Cases (Anwendungsfälle)

Die folgenden Anwendungsfälle beschreiben einige Funktionen von Protégé, die dem Benutzer zu Verfügung stehen:

Anwendungsfall	Ontologie erstellen
Akteur	Nutzer
Beschreibung	Eine neue Ontologie konzipieren und erstellen
Ergebnis	Ontologie erstellt

Anwendungsfall	Klassenhierarchie bearbeiten
Akteur	Nutzer
Beschreibung	Klassen in die Hierarchie einfügen oder löschen
Ergebnis	Klassenhierarchie verändert

Anwendungsfall	Klasseneigenschaften bearbeiten
Akteur	Nutzer
Beschreibung	Eigenschaften einer Klasse modifizieren
Ergebnis	Klasseneigenschaften verändert

Anwendungsfall	Instanzen bearbeiten
Akteur	Nutzer
Beschreibung	Instanzen hinzufügen, löschen oder modifizieren
Ergebnis	Instanz gelöscht, bearbeitet oder modifiziert

Anwendungsfall	Suchanfrage stellen
Akteur	Nutzer
Beschreibung	Suchanfrage formulieren und ausführen
Ergebnis	Wissensbasis durchsucht, Suchergebnisse

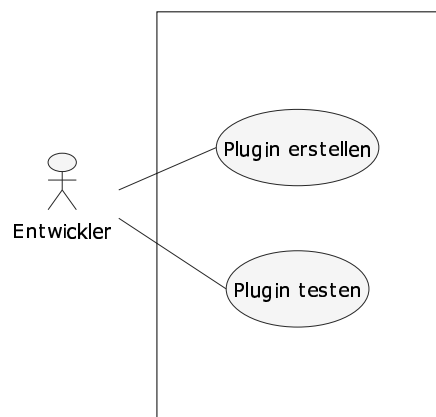
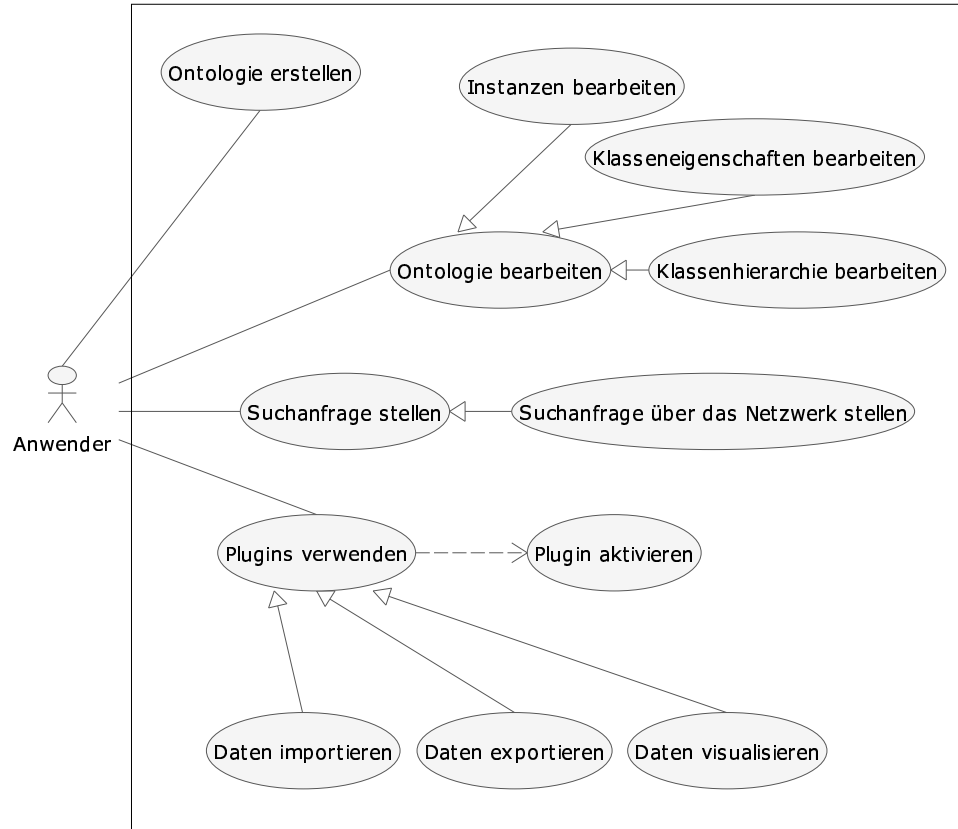
Anwendungsfall	Plugins verwenden
Akteur	Nutzer
Beschreibung	Plugin wählen und Plugin-Funktionalität nutzen
Ergebnis	Jeweilige Plugin-Funktionen ausgeführt

Softwaretechnik-Praktikum SS 2005

GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

Folgende UML-Diagramme zeigen Anwendungsfälle für Nutzer und Entwickler:



Softwaretechnik-Praktikum SS 2005

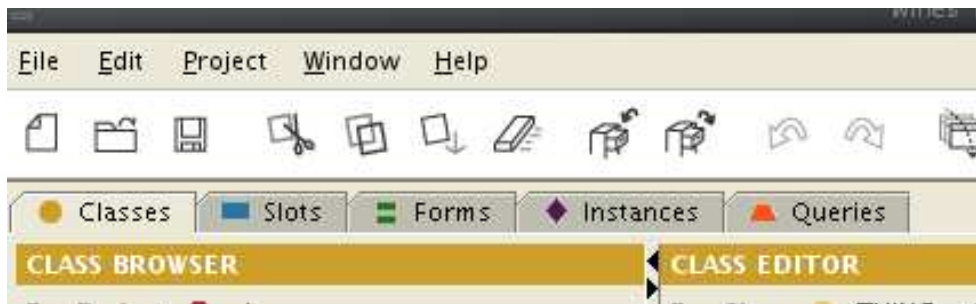
GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

(c) Sichtbare Aspekte

Man kann mit Protégé neue Ontologien erstellen oder vorhandenen Daten öffnen bzw. importieren. Ein Dialogfenster beim Start von Protégé lässt den Nutzer zwischen diesen beiden Optionen wählen.

Nach dieser Wahl wird bereits ein Teil der Plugin-Struktur von Protégé deutlich. Man hat als Default-Einstellung fünf Tabs zur Auswahl, die unterhalb der Menüleiste und der Iconbar angezeigt werden:



Das Classes-Tab zeigt die Klassenhierarchie an. Unter dem Slots-Tab finden sich die in der Ontologie definierten Eigenschaften, ebenfalls hierarchisch geordnet. Mit Hilfe des Forms-Tab können Eingabemasken vom Benutzer festgelegt werden. Das Instances-Tab zeigt die zu einer Klasse zugehörigen Instanzen an. Mit Hilfe des Query-Tabs kann man die Wissensbasis durchsuchen.

Softwaretechnik-Praktikum SS 2005

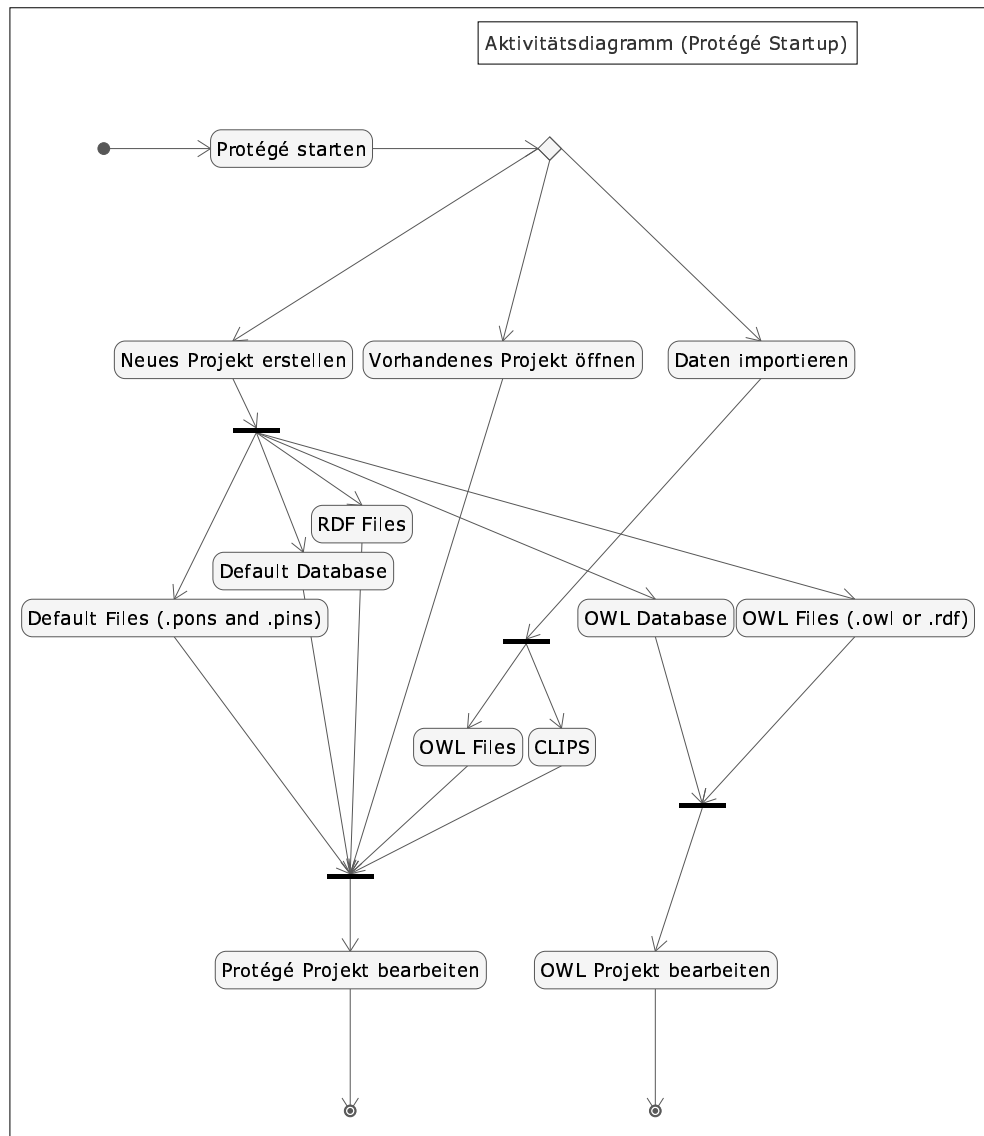
GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

(d) **Sichtbare Aspekte in UML Notation**

Folgende Diagramme stellen funktionale Aspekte von Protégé in UML-Notation dar.

Mögliche Aktivitäten nach dem Start von Protégé:



Softwaretechnik-Praktikum SS 2005

GR-1

Projektleiter: Adrian Kiess | Dokument erstellt von Bai Wang, Guang Yu Wang, Martin Czygan

4. Im Praktikum zu besetzende Rollen**Team:** Bai Wang, Guang Yu Wang, Adrian Kiess, Martin Czygan, Tobias Mern

Projektleiter	Adrian Kiess
Modellierung	Martin Czygan
Dokumentation/QS	Tobias Mern
Recherche	Bai Wang
technischer Assistent	Adrian Kiess
Test	Bai Wang, Guang Yu Wang
Implementierung	Martin Czygan