Zusammenfassung des Vortrages

“ Semantic Web (Ontologien und Werkzeuge) **”**

Artem Khvat, 27.04.2005
khvat_a@informatik.haw-hamburg.de

1.Geschichte der Ontologien.

Ontologien haben ihre weite Verbreitung in den unterschiedlichsten Bereichen der Informatik gefunden. Heute findet man schon viele Applikationen, bei denen die Ontologien eine zentrale Rolle bei der Lösung dieser Probleme spielen. Meist sind das Applikationen, die sich mit knowledge Management, natürlichen Sprachen, e-commerce und Semantik Web beschäftigen.

Aber bevor wir uns den Implementierung Syntax und anderen Feinheiten der Ontological Ingineering widmen, möchte ich kurz die Geschichte und die Entstehung dieser Wissenschaft erläutern. Dabei werden die Hintergründe der Ontologien deutlich gemacht, was für spätere Anwendung in der Praxis von sehr großem Nutzen sein kann.

Was ist etwas? Was heißt „da sein“, wie kann man die Bedeutung von irgendetwas ausdrücken und vermitteln? Was passier mit Dingen, wenn sie sich ändern oder geändert werden? Wie beschreibe ich die Welt? Existieren die Dinge außerhalb meines Bewusstseins? Seit 1000 Jahren versucht man diese Fragen zu beantworten und während des Suchprozesses sind die klassischen Ontologien entstanden.

Grundsätzlich teilen sich die Ontologien in zwei Aspekte:

- „Essence“ – was etwas ist (Gamrad,1999)
- „Existence“- wie wird dieses etwas in der realen Welt präsentiert.

Ein Beispiel dazu wäre das Wesen Zentaur:

Essence – Ein Wesen halb Mensch, halb Pferd.
Existence – Ø.

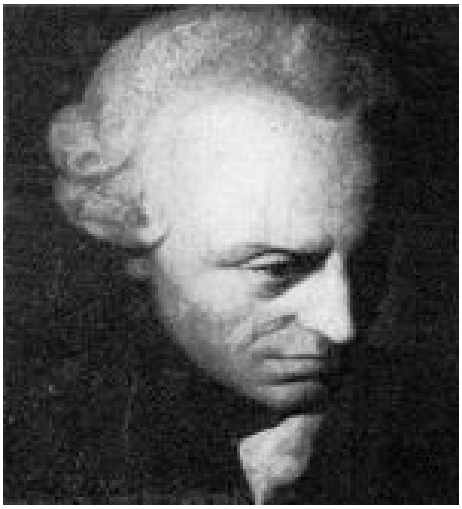
Die ersten bekannten Ansätze zur Lösung der globalen Fragen der Weltbeschreibung und deren Essence und Existence kamen von einem griechischen Philosophen namens Parmenides, aus der Zeit etwa 5. - 4. Jhr. v. Chr. Parmenides hat den ersten Schritt auf den Weg zu dieser Problemlösung gemacht. Seine Idee war, dass die Dinge unabhängig von unserer Betrachtung existieren. Diesen Ansatz nutzte später ein anderer griechische Philosoph namens Aristotel (384-322 v. Chr). Aristotel hat die Idee von Parmenides erweitert und eine neue Wissenschaft mit dem Namen „Metaphysik“ erschaffen. Nach Aristotel wurde die Welt in Kategorien eingeteilt, die dem Betrachter ermöglichen sollten, alles in der Welt zu beschreiben. Es wurde zwischen folgenden Kategorien unterschieden:

- Substanz
- Qualität
- Quantität
- Relation
- Action
- Platz
- Zeit
- Neigung

Als Beispiel nach Aristoteles :

- Computer ist auf dem Tisch (Kategorie „ Platz „).
- Computer ist schwarz (Kategorie „Qualität“).

Den nächsten Schritt in Bezug auf die Weltbeschreibung und Klassifikation wurde von einem bekannten deutschen Philosophen Immanuel Kant (1724-1804) gemacht. Er hat die Kategorie Modell der Welt von Aristoteles erweitert und präzisiert, die Hauptüberlegung bei der Entwicklung seines Frameworks war - „ Welche Strukturen benutzt Verstand, um Realität wahrzunehmen “?



Kants Framework :

- ***Quantität***
  - Einheit
  - Vielheit
  - Allheit
- ***Qualität***
  - Realität
  - Negation
  - Limitation
- ***Modalität***
  - Möglichkeit – Unmöglichkeit
  - Dasein – Nichtsein
  - Notwendigkeit – Zufälligkeit
- ***Relation***
  - Inhärenz und Subsistenz
  - Kausalität und Dependenz
  - Gemeinschaft

Als Beispiel dafür kann die Person Artem Khvat dienen, nach Kant ist Artem Khvat:

- ***Quantität***
  - Einheit
- ***Qualität***
  - Realität
- ***Modalität***
  - Dasein
- ***Relation***
  - Ø.

Der letzte Schritt zum „*Ontological Engineering*“ und deren Basisidee wurde dann 1939 von Jose Ortega y Gasset (1883 – 1955) gemacht. Nach Ortega ist die Welt von dem Beobachter abhängig. Als Begründung sagte er:

„ It is nonsense to have a real thing which is always the same independently from where it is perceived“.

Genau diese Idee spiegelt die Grundlage der heutigen „*Ontological Engineering“ wieder:* die Informations-Systeme werden nicht für die exakte Weltabbildung

konstruiert, sondern für die effektive Realisierung der Aufgaben (***Gomez-Perez 2003).***
Als Fazit zu diesem Abschnitt:

- Ontologien dienen für die strukturierte Darstellung des Realitätswissens.
- Ein Artefakt der realen Welt kann von mehreren Ontologien aus verschiedenen Sichten beschrieben werden.
- Das ist gut so …

2.Motivation der Ontologien heute

Grundsätzliche Motivation für den Ansatz der Ontologien heute:

- Digital gespeicherte Informationen existieren in großen Mengen.
- Problem: Fehlertoleranter Zugriff.
- Effiziente Suche.
- Individuelle Filterung.
- Lücke zwischen Bedeutung und Speicherung von Informationen.

3.Typen der Ontologien

Man unterscheidet zwischen folgenden Typen der Ontologien:

- Ontologien für die Präsentation des Wissens.
- Allgemeine Ontologien.
- „Top-level "oder „Upper level" Ontology.
- Domain Ontologien.
- Aufgabenbezogene Ontologien.
- Domain-Aufgabenbezogene Ontologien.
- Methode Ontologien.
- Applikation Ontologien.

Allgemeine Ontologien:

- repräsentieren allgemeine Widerverwendbarkeit des Wissens innerhalb eines Domains.
- Beinhalten Vokabular der in Verbindung zur Zeit, Ort, Funktion, Komponenten etc. steht.

Beispiel: *Standart Units Ontology (Gomez-Perez 2003 Ontological Engin.)* ***Beschreibung der Zeiteinheit "Minute"****:*

(***defin -frame*** *Minute)*
: own - slot
(Documentation " Time Unit")
(Instance-Of Unit-of-Measure))
: axiom - def
((Quantity.Dimension Minute Time-Dimension)))

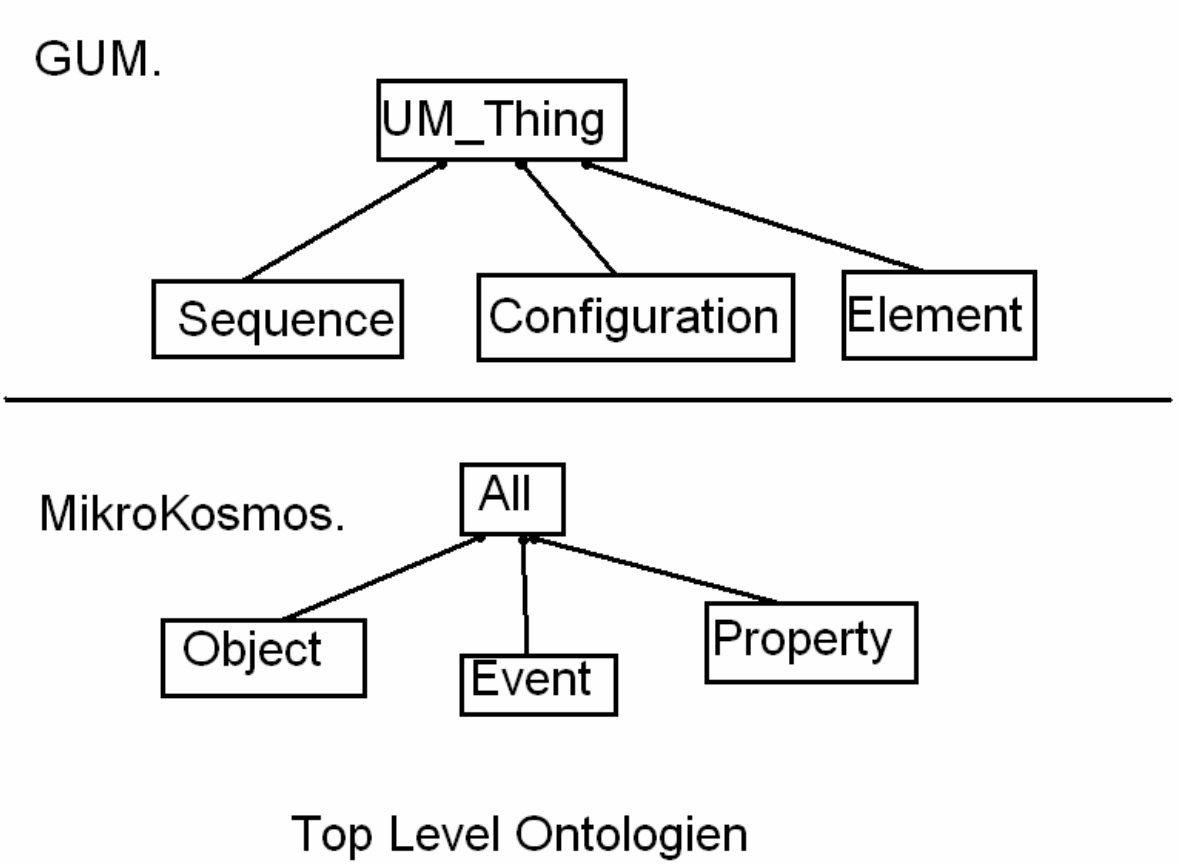
(***define - frame*** *Second-of-Time*
: own - slots
((Documentation „ The SI standard unit of time")
(Instance-Of Si-Unit Unit-of-Measure)
(Quantity.Dimension Time-Dimension))
: axiom - def

((=Minute (60 Second-Of-Time))))*

Top-level“ Ontologien

- beschreiben sehr allgemeine Konzepte.
- liefern Regeln für das Linken der Root-Termen einer Ontologie.
- oft widerverwendbar.

Beispiel sind zwei bekante Top Level Ontologien GUM (Beschreibung der natürlichen Sprache) und MikroKosmos (Beschreibung des Universums)



Top Level Ontologien

Domain Ontologien

- sind widerverwendbar innerhalb eines Domains.
- liefern Vokabular über Konzepte, Relationen und Aktivitäten innerhalb eines bestimmten Domains.
- basieren oft auf „Top-level“ Ontologien.

Aufgabebezogene Ontologien:

- Liefern Vokabular, das für die Lösung des Problems benutzt wird, welches der Domain gehören wird, oder auch nicht.

Domain- Aufgabebezogene Ontologien:

- sind Aufgabebezogene Ontologien, die widerverwendbar innerhalb einer Domain sind, und nicht Domainübergreifend.
- sind applikationsunabhängig.
- Beispiel Travel-Domain: next city, previos city.

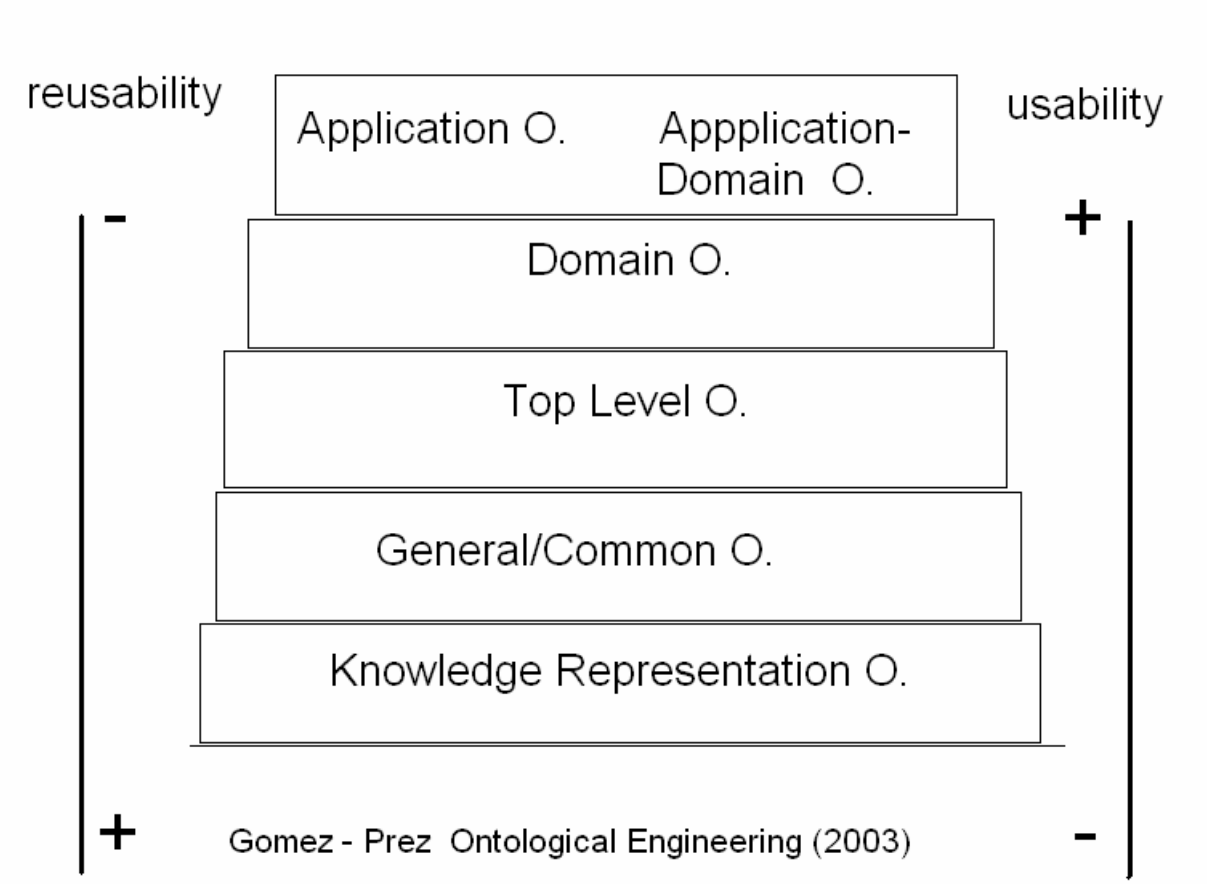
Methode Ontologien:

- Eine Ontologie über die Terminplanung, mittels der Aufgabeaufspaltung würde dieser Kategorie angehören.

Applikation Ontologien:

- applikationsabhängige Ontologien.
- enthalten alle Definitionen, die notwendig für die Modellierung des Wissens bestimmter Applikation sind.
- Beispiel: Ontologie für die Reisebüros mit Reisen nur nach Kanada.

Zwei Hauptkriterien sind über die Art der Ontologien für ein bestimmtes Problem entscheidend. Einsetzbar sind Nutzbarkeit und Widerverwendbarkeit. Die nächste Abbildung ordnet die Ontologien Anhand dieser Kriterien:



Wie man hier deutlich sieht, steht die Nutzbarkeit der Ontologie in direktem Widerspruch zu ihrer Widerverwendbarkeit, und umgekehrt. Am Schluss, ist es dem Entwickler selbst überlassen, welche Art der Ontologie für die Lösung eines konkreten Problems zu wählen ist.

5.Ontologiesprachen

Ontologiesprachen werden durch folgende Merkmale gekennzeichnet:

- Eine Ontologiesprache bezieht sich in aller Regel auf Konzepte (Klassen, Entitäten, ...).
- Eigenschaften von Konzepten (Slots, Attribute, ...) und Relationen zwischen Konzepten.
- (Assoziationen) und zusätzliche Sprachmittel für Einschränkungen.
- Größe Bandbreite verschiedener Ontologiesprachen:
  - Einfach (nur Konzepte und Taxonomie RDF)
  - Frame-basiert (Konzepte plus Konzepteigenschaften RDF(S))
  - Logik-basiert (z. B. Ontolingua, DAML+OIL, ...)
- Ontologien werden oftmals durch Diagramme ausgedrückt (meistens ist es nicht möglich dabei alles auszudrücken).
- Entity-Relationship Schemata und UML Klassendiagramme können als Ontologiesprachen verstanden werden.

5.1.RDF *„Resource Description Framework“*

Ziele:

- Entwurf zukunftssicherer Anwendungen im Hinblick auf die sich entwickelnden Schemata.
- Automatisiertes Ranking von Ressourcen.
- Entwicklung maschinellverarbeitbarer Semantiken.
- Interoperabilität von Metadaten.

RDF-Syntax

Die Bestandteile der RDF-Syntax sind Ressourcen, Eigenschaften und Aussagen.

Ressourcen: sind mittels RDF beschriebene Objekte. Ressourcen werden eindeutig mittels eines URIs3 und eineroptionalen *Anchor-ID* spezifiziert. Da es prinzipiell möglich ist, für beliebige Objekte URIs zu definieren, beschränkt sich die Anwendbarkeit von RDF nicht nur auf Webdokumente und andere Ressourcen im Netz, sondern kann nach Belieben auf die physische Welt ausgedehnt werden.
(http://www.ibr.cs.tu-bs.de/lehre/ws0304/svs/work/rdf_paper_final.pdf)

Eigenschaft: unter einer Eigenschaft wird ein Aspekt, ein Charakteristikum oder eine Relation, welche zum Beschreiben einer Ressource verwendet wird, verstanden. Jede Eigenschaft hat eine bestimmte Bedeutung, welche die erlaubten Werte, die Art von Ressourcen, welche sie beschreiben kann, und ihre Beziehung zu anderen Eigenschaften definiert.
(http://www.ibr.cs.tu-bs.de/lehre/ws0304/svs/work/rdf_paper_final.pdf)

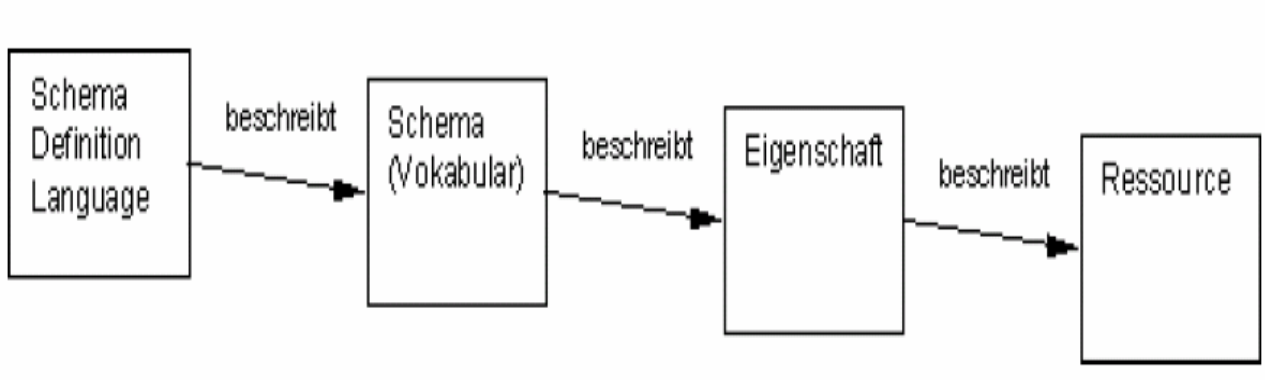
Ein RDF-Aussage wird durch ein Tripel aus einer Ressource, einer Eigenschaft und dem Wert dieser Eigenschaft für die jeweilige Ressource spezifiziert. Diese drei Teile einer Aussage werden als

Subjekt (die Ressource)
Prädikat (die Eigenschaft)
Objekt (der Wert der Eigenschaft)

bezeichnet. Als Objekt sind sowohl Strings, andere einfache Datentypen, oder auch andere Ressourcen möglich, welche über einen URI spezifiziert werden.
(http://www.ibr.cs.tu-bs.de/lehre/ws0304/svs/work/rdf_paper_final.pdf)

5.2. RDF(S) (http://www.w3.org/TR/rdf-schema)

- RDF(s)-Schema beschreibt Eigenschaften.
- ähnelt stark dem hierarchischen Klassensystem objektorientierter Sprachen.

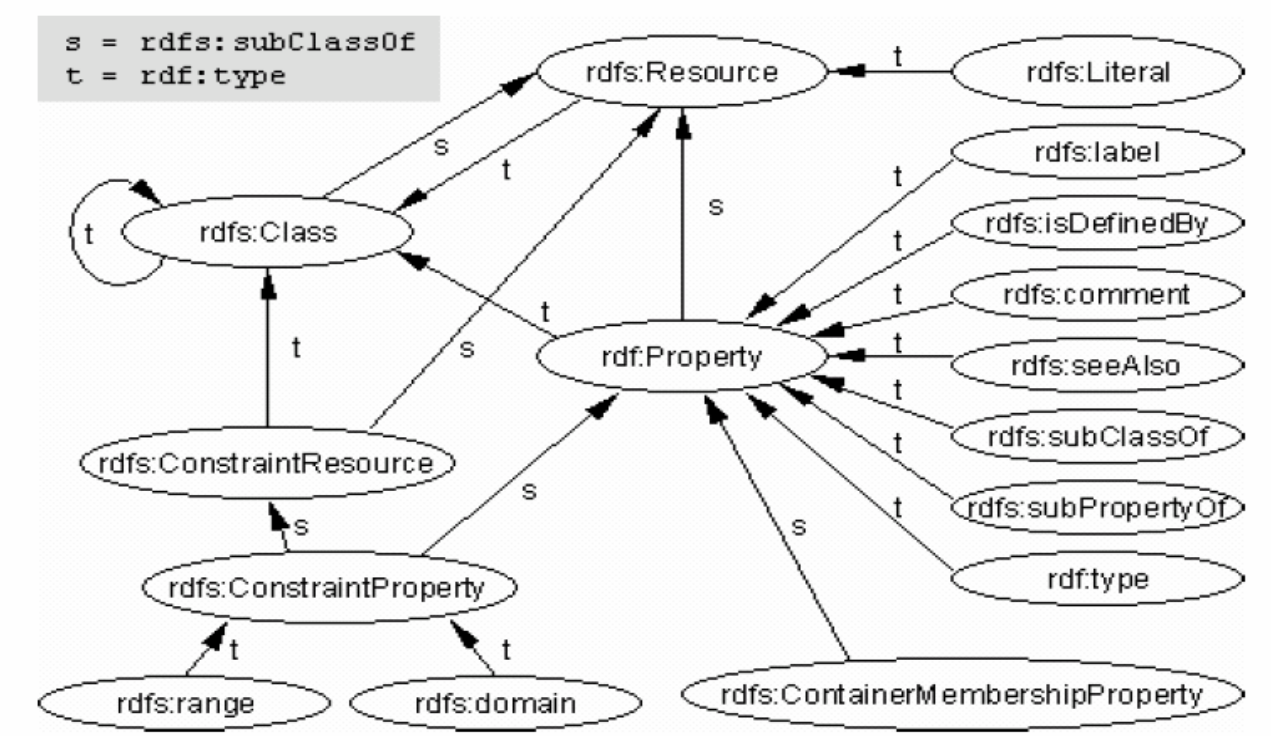


Besteht im Wesentlichen aus Klassen, Eigenschaften und Ressourcen, die auf RDF-Elementen basieren.

- *rdfs:Resource:* Alles, was mit RDF beschrieben wird, ist eine Ressource und somit eine Instanz dieser Klasse.
- *rdf:Property:* Stellt die Klasse der Eigenschaften dar und repräsentiert eine Untermenge der Ressourcen.

In der nächsten Abbildung wird der genauere Zusammenhang zwischen RDF und RDF(S) dargestellt.

Klassenhierarchie von RDF-Schema (Gomez-Perez 2003 Ontological Engineering)



5.3.DAML+OIL

DAML+OIL ist keine komplette Neuentwicklung, sondern baut (wie DAML) auf RDF und RDF/S auf. Damit sollte erreicht werden, dass RDF-basierende Anwendungen mit DAML+OIL- ausgezeichneten Seiten so wenig Probleme wie möglich bekämen. Die Syntax von DAML+OIL ist daher selbst in RDF/S formuliert.

5.4.OWL (Web Ontology Language)

- Erweiterung von DAIML+OIL.
- ähnelt DAML+OIL sehr stark.
- Einige Konstrukte umbenannt.
- Neu Konstrukte für die Beschreibung der Web-Services eingefügt.

6.Mein Beitrag für das Project :

Mein Beitrag für dieses Project wäre die Erstellung von halb-automatischer Kommunikation zwischen den einzelnen Artefakten des Systems, unter Verwendung von Ontologien. Dabei müssen die Kommunikationspartner nicht unbedingt das volle Wissen über die Ontologie der anderen haben. Es wäre dann genau meine Aufgabe, diesem Abstimmungs-Verfahren die Semantische Bedeutung der jeweiligen Ontologien zu realisieren. Ein Szenario wird in folgendem Bild Präsentiert: Kommunikationspartner 1 will in die Stadt, es wird eine Anfrage mit der Ontologie 1 an Partner 2 und 3 geschickt. Diese nehmen den Auftrag entgegen, holen die entsprechende Semantik über das Ziel und die Art der Fortbewegung aus der Ontologie des Senders und schicken ihre Antwort zurück. Der Empfänger kriegt wiederum die Ontologie des Partners, entschlüsselt dann die Semantik und handelt weiter.

7.Literatur :

- **http://www.semantic-web.at**
- **http://www.w3.org/DesignIssues/Semantic.html**
- **http://www.semanticweb.org**
- **Asucion Gomez-Perez Mariano Fernadez-Lopez, Oscar Corcho Ontological Engineering Springer Verlag 2003.**
- **http://www.w3.org/TR/rdf-schema/)**
- **http://www.w3.org/2001/sw/WebOnt/**
- **http://www.ibr.cs.tu-bs.de/lehre/ws0304/svs/work/rdf_paper_final.pdf**