
Bridging Axioms

SoSe 06

18.404 Semantic Integration
vorgetragen von Zehra Öztürk
am 02. Mai 2006

Literatur

Dejing Dou, Drew McDermott & Peishen Qi: Ontology Translation on the Semantic Web. *LNCS Journal of Data Semantics*, Volume II, LNCS 3360, 2004, 35-57.

Dejing Dou, Drew McDermott & Peishen Qi: Ontology translation by ontology merging and automated reasoning. In *Proc. EKAW Workshop on Ontologies for Multi-Agent Systems*, 2002, 3-18.

Dejing Dou and Drew McDermott: Deriving Axioms Across Ontologies. In *Proc. Int'l joint conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, 2006.

Inhalt

1. Ontology Translation
 - 1.1. Ontologieunterschiede in ähnlichen Domänen
 - 1.2. Ontology Translation vs. Ontology Mapping
2. Lösungsvorschlag
 - 2.1. OntoMerge
 - 2.2. Ontology Translation in 3 Schritten
 - 2.2.1. Web-PDDL
 - 2.2.2. OntoEngine
 - 2.3. Bridging Axioms
3. Ontology Translation Probleme
 - 3.1. Ontology Translation für Datenmengen
 - 3.2. Erzeugung von Ontologie-Erweiterungen
 - 3.3. Anfragen über verschiedene Ontologien
4. Zusammenfassung

1. Ontology Translation

- Ontologie: formale Spezifikation des Vokabulars und der Beziehung zwischen Konzepten, die im Semantic Web genutzt werden
- es werden immer mehr Ontologien entwickelt, die ähnliche Domänen beschreiben
- die geteilte Natur des Web erlaubt es, webbasierten Agenten verschiedene Ontologien zu nutzen
 - ⇒ die syntaktischen und semantischen Unterschiede zwischen den Ontologien zu überbrücken ist eines der schwierigsten Probleme für den Agenten

1.1. Ontologieunterschiede in ähnlichen Domänen

- die Unterschiede beinhalten syntaktische und semantische Unterschiede
- Semantische Unterschiede können durch viele verschiedene Faktoren auftreten
 - unterschiedliche taxonomische Strukturen von Konzepten
 - auch wenn Konzepte von 2 Ontologien den gleichen Klassen- oder Eigenschaftsnamen haben, können sie unterschiedliche Bedeutungen haben

1.2. Ontology Translation vs. Ontology Mapping

- Ontology Mapping ist der Prozess des Findens von Korrespondenzen zwischen den Konzepten von Ontologien
 - die Abbildungen sollten durch Regeln ausgedrückt werden, welche erklären wie die Konzepte korrespondieren
- Ontology Translation muss die Abbildungen zwischen zwei Ontologien kennen, bevor Regeln erstellt werden können, die diese beschreiben
 - die Abbildungen werden durch Experten generiert

2. Lösungsvorschlag

- Ontology Translation wird benötigt bei
 - Übersetzungen von Datenmengen
 - Generierung von Erweiterungen der Ontologie
 - Anfragen, die über verschiedene Ontologien hinausgehen
- Vorgehen: Ontology Translation durch Ontology Merging and Automated Reasoning
 - **OntoMerge**: Onlinesystem welches dieses Vorgehen implementiert

2.1. OntoMerge

- **OntoMerge**: implementiert Ontologie Translation mit Ein- und Ausgaben in DAML+OIL oder anderen Websprachen
 - Fokus der Methode: formale Inferenz von Fakten, die in einer Ontologie ausgedrückt sind zu den Fakten einer anderen
 - **Merge** von 2 Ontologien: Vereinigung der Terme (Konzepte) und der Axiome, die diese definieren
 - **XML namespaces** werden verwendet
 - **Bridging axioms** werden hinzugefügt, um Konzepte von 2 Ontologien in Relation zueinander zu setzen

2.2. Ontology Translation in 3 Schritten

- Herangehensweise für diese Lösung, durch Teilung der Ontology Translation in 3 Schritte:
 - **syntaktische Übersetzung** von einer Websprache in eine interne Repräsentationsform
 - interne Repräsentation: **Web-PDDL**
 - **semantische Übersetzung** durch Inferenz unter Nutzung der internen Notation
 - Ontologieübersetzung: **OntoEngine**
 - **syntaktische Übersetzung** aus der internen Repräsentationsform in die Ziel-Websprache

2.2.1. Web-PDDL

- Web-PDDL ist eine first-order Sprache mit LISP-ähnlicher Sytax

```
(define (<entity> <name>)  
      <clauses>)
```

- Variante von PDDL
 - erweitert um XML namespaces
 - um Konflikte bei den Namen zu vermeiden
 - und einer flexibleren Notation für Axiome
 - Definition von Domänen und den Entitäten innerhalb der Domänen kann besser erfolgen

2.2.1. Web-PDDL: Beispiel

Beispiel aus der yale_bib Ontologie in Web-PDDL:

```
(define (domain yale_bib-ont)
  (:extends (uri http://www.w3.org/2000/01/rdf-schema# :prefix
            rdfs))
  (:types Publication - Obj
           Article Book Incollection Inproceedings - Publication
           Literal - @rdfs:Literal)
  (:predicates (author p - Publication a - Literal)
               ...))
```

2.2.2. OntoEngine (1)

- die semantische Übersetzung ist implementiert unter Nutzung einer inference machine, der OntoEngine
 - berechnet Annahmen und Anfragen in Web-PDDL Syntax
 - läuft entweder auf data-driven (Vorwärtsverkettung) or demand-driven (Rückwärtsverkettung) Art
- wenn eine Eingabe zum Übersetzen erhalten wird, braucht die OntoEngine eine Merged-Ontologie
- daraufhin werden die Datenmengen (Fakten) geladen und mit den Bridging Axiomen Vorwärtsverkettung durchgeführt

2.2.2. OntoEngine (2)

- durch die Verkettung mit spezifizierten Richtungen anstatt mit vollständiger Resolution ist die OntoEngine nicht vollständig (im logischen Sinne)
 - in diesem Fall wird aber die Effizienz bevorzugt, denn:
Auch ein logisch vollständiger Theorembeweiser könnte scheitern, bei dem Versuch eine vollständige Übersetzung zu generieren, da die Quell-Ontologie und die Ziel-Ontologie sich evtl. nicht überlappen

2.3. Bridging Axioms (1)

- Beschreibung von Abbildungen zwischen Ontologien
- innerhalb der Merged Ontology muss es allen relevanten Symbolen erlaubt sein zu interagieren
 - dafür werden semantische Relationen zwischen den korrespondierenden Symbolen erstellt, die durch Axiome ausgedrückt werden können, die Bridging Axiome
- Axiome beim Ontology Merging können simple und komplizierte Mappings zwischen Konzepten der Quell- und Zielontologien darstellen

2.3. Bridging Axioms (2)

- die simplen semantischen Mappings enthalten z.B. “subClassOf“, “subPropertyOf“ oder “equivalent“ Relationen
- da die Typen nicht Objekte sind, sind Ausdrücke wie $(= T_1 T_2)$ nicht zulässig
- aufgrund dessen muss ein Pseudo-Prädikat **T->** verwendet werden, und Bridging Axiome, um die Relation für die äquivalenten Typen herzustellen
- Beispiel hierfür in Web-PDDL:

```
(:axioms
  (T-> @yale_bib:Book Book)
  (T-> @cmu_bib:Book Book)
  ...)
```

2.3. Bridging Axioms: Beispiel

```
(forall (a - Article t1 - String)
  (iff (@yale_bib:booktitle a t1) (booktitle a t1)))
```

```
(forall (a - @yale_bib:Inproceedings t1 - String)
  (iff (booktitle a t1)
    (exists (p - Proceedings)
      (and (contain p a)
           (@cmu_bib.inProceedings a p)
           (@cmu_bib:booktitle p t1))))))
```

3. Ontology Translation Probleme

- Ontology Translation für Datenmengen
 - Problem beim Austausch von Datenmengen, da unterschiedliche Ontologien verwendet werden, um sie zu beschreiben
- Erzeugung von Ontologieerweiterungen
 - Problem der Generierung von korrespondierenden Subontologieerweiterungen
- Anfragen über verschiedene Ontologien
 - Problem bei der Anfragebeantworten mit multiplen Wissensbasen, die unterschiedliche Ontologien haben

3.1. Ontology Translation für Datenmengen (1)

- **Datenmengen** sind Mengen von Fakten, die in einer Ontologie ausgedrückt sind
- bei einer gegebenen Menge von Fakten in einem Vokabular (Quelle) wird die größtmögliche Menge von Konsequenzen in einem anderen Vokabular (Ziel) inferiert
 - ⇒ dies kann in 2 Phasen aufgeteilt werden:
 - Inferenz
 - Projektion

3.1. Ontology Translation für Datenmengen (2)

- Schritt 1: Inferenz: arbeitet in einer Merged-Ontologie, die alle Symbole und Axiome von beiden, Quelle und Ziel, kombiniert und Inferenzen von den Quellfakten darstellt
 - **Merged-Ontologien** enthalten Symbole, Fakten und Bridging Axiome
 - ⇒ müssen noch von Experten erstellt werden
 - ⇒ die Merged-Ontologie ist eine, mit welcher man einen Merge mit anderen Ontologien durchführen kann
- Schritt 2: Projektion: behalte alle Folgerungen, die nur in dem Zielvokabular ausgedrückt sind zurück

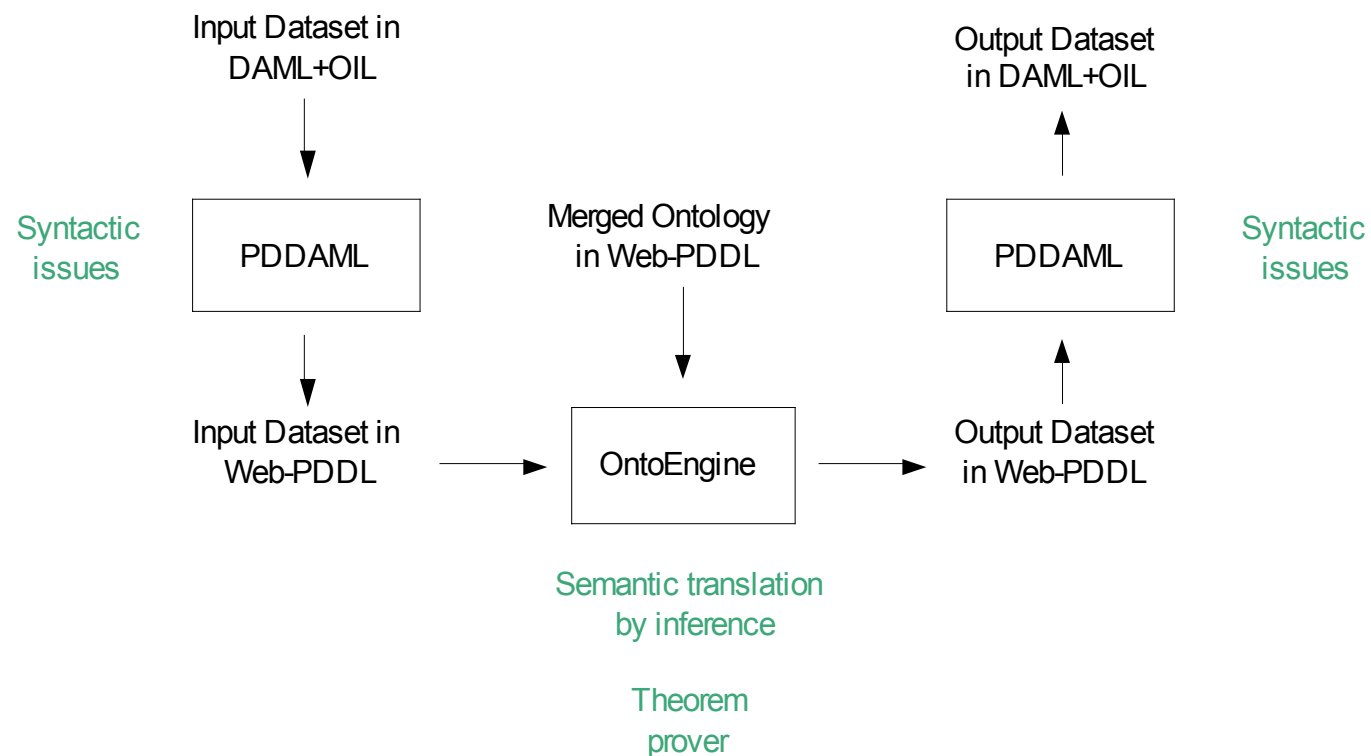
3.1. Ontology Translation für Datenmengen (3)

- **Skolemterme** werden verwendet, wenn die Übersetzung es erfordert, über ein Objekt zu sprechen, welches nicht mit einem existierenden Objekt identifiziert werden kann
- Termgenerierungsfunktionen sind mit eingefügt
 - erlauben feinere Kontrolle über die Generierung als die Skolemisierung
- für die Übersetzung wird ein Theorembeweiser verwendet

3.1. Ontology Translation für Datenmengen (4)

- die Inferenzen fokussieren auf:
 - Vorwärtsverkettung von der Quelle zum Ziel
 - Rückwärtsverkettung von Anfragen in einer Ontologie zu Datenmengen in einer anderen
 - Einführung von Skolemtermen und Termgenerierungsfunktionen
 - Nutzung von Äquivalenzen, um existierende konstante Terme durch Skolemterme zu ersetzen
- der hier verwendete Theorembeweiser **OntoEngine** ist auf diese Arten von Inferenzen spezialisiert

3.1. Ontology Translation für Datenmengen (5)



3.2. Erzeugung von Ontologie-Erweiterungen (1)

- gegeben sind zwei in Relation zueinander stehende Ontologien O_1 und O_2 , sowie eine Erweiterung O_{1s} von O_1
- wie kann die korrespondierende Erweiterung O_{2s} generiert werden?
- manuelle Erzeugung von Subontologien im Web nicht möglich
 - Tools werden benötigt, da die Anzahl der Subontologien viel größer ist im Web

3.2. Erzeugung von Ontologie-Erweiterungen (2)

- Szenarien:
 - Ontologieexperten haben Subontologien der existierenden Ontologien bereits erstellt und wollen die korrespondierenden Subontologien von Ontologien, die zu diesen in Relation stehen, generieren
 - wenn die Beziehung zwischen diesen Ontologien bekannt ist, können einige Ontology Translation Tools diesen Prozess automatisieren

3.2. Erzeugung von Ontologie-Erweiterungen (3)

- Szenarien:
 - Ontologieexperten müssen oft bereits existierende Ontologien updaten, wenn neues Wissen oder neue Anforderungen auftreten
 - muss manuell erfolgen
 - wie könnte man aber die Subontologien updaten?
 - wenn die Beziehungen zwischen den alten und den neuen Ontologien bekannt sind, können die neuen Subontologien automatisch generiert werden

3.2. Erzeugung von Ontologie-Erweiterungen (4)

- erfolgt ähnlich wie bei dem Ontology Translation mit Datenmengen
 - statt Datenmengen als Eingaben werden Subontologien verwendet
 - statt einer Menge von Fakten wird eine Menge von Subeigenschaftsdefinitionen eingegeben

3.3. Anfragen über verschiedene Ontologien (1)

- Wissen, welches verwendet werden kann, um Anfragen zu beantworten, kann aus verschiedenen Wissensbasen stammen, welche unterschiedliche Ontologien verwenden können
- Agent muss nicht spezifizieren, welche Wissensbasis seine Anfrage beantworten kann und auch nicht welche Ontologie durch sie verwendet wird
- für diese Fälle wurde ein Reasoner mit Rückwärtsverkettung in die OntoEngine eingebaut, welche demand-driven läuft

3.3. Anfragen über verschiedene Ontologien (2)

- die PDDAML wurde erweitert, um mit der Syntaxübersetzung von einer DQL-Anfrage und Web-PDDL umgehen zu können
- Tools für Anfragenauswahl und Anfragenreformulierung wurden in OntoMerge eingebaut
- eine eingegebene Anfrage kann eine Konjunktion von Subanfragen sein und jede von ihnen kann von unterschiedlichen Wissensbasen beantwortet werden

4. Zusammenfassung

- Ontology Translation umfasst schwierige Aufgabenbereiche
- Ontology Translation wird verwendet zum Übersetzen von Datenmengen, bei der Generierung von Ontologieerweiterungen und bei Anfragen, die über verschiedene Ontologien hinausgehen
- Ontology Translation kann man sich als Terme von Ontology Merging vorstellen
- wenn alle Ontologien, Datenmengen und Anfragen in Termen einer internen Repräsentation ausgedrückt werden können, kann semantische Übersetzung durch Reasoning implementiert werden
- das erforderliche Schließen kann als simple first-order Inferenz ausgedrückt werden