

Semantic Web Technologies II

SS 2009

25.05.2009

Übung: Ontologiemodellierung in OWL

Dr. Sudhir Agarwal

Dr. Stephan Grimm

Dr. Peter Haase

PD Dr. Pascal Hitzler

Denny Vrandečić

Übersicht

- Einführung in Protégé 4
- Umsetzung DL nach Protégé
- Modellierungsaufgaben

Übersicht

- Einführung in Protégé 4
- Umsetzung DL nach Protégé
- Modellierungsaufgaben

- Protégé 3.x
 - Entwicklungsumgebung für Ontologien in Frames, RDF(S) und OWL 1
- Protégé 4
 - Neueste Variante der Ontologieentwicklungsumgebung mit Unterstützung von OWL 2 (beta-Stage)
 - Download:
<http://protege.stanford.edu/download/download.html>
- Grundlegende Funktionalität
 - Demonstration am Werkzeug

Übersicht

- Einführung in Protégé 4
- Umsetzung DL nach Protégé
- Modellierungsaufgaben

Klassensicht und Komplexe Axiome

- Nützliche Äquivalenzen zur Umformung
 - $C \sqsubseteq D \Leftrightarrow \top \sqsubseteq \neg C \sqcup D$
 - $C \sqsubseteq D \sqcap E \Leftrightarrow C \sqsubseteq D, C \sqsubseteq E$
 - $C \sqcup D \sqsubseteq E \Leftrightarrow C \sqsubseteq E, D \sqsubseteq E$
 - *DeMorgan laws*
- Beispiel
 - $Student \sqcap \exists holds.\{BachelorDegree\} \sqsubseteq Graduate$
 - $\top \sqsubseteq \neg(Student \sqcap \exists holds.\{BachelorDegree\} \sqcup Graduate)$
 - $\top \sqsubseteq \neg Student \sqcup \forall holds.\neg\{BachelorDegree\} \sqcup Graduate$
 - $Student \sqsubseteq \forall holds.\neg\{BachelorDegree\} \sqcup Graduate$

Namenskonventionen

- Klassen
 - Substantive (nouns)
 - Singular-Formulierung
 - Große Anfangsbuchstaben (capitalised)
 - Bsp: *IntermediateCourse*
- Rollen (Properties)
 - Verben (verbs)
 - Konjugierte Form, evtl. Modifikatoren
 - Kleine Anfangsbuchstaben (wie Methodennamen in Java)
 - Bsp: *isEnrolledIn*
- Individuen
 - Eigennamen
 - Bsp: *JohnDoe*

Manchester Syntax

DL notation

Manchester OWL syntax

\top

owl:Thing

\perp

owl:Nothing

$\neg C$

not C

$C \sqcap D$

C and D

$C \sqcup D$

C or D

$\exists R.C$

R some C

$\forall R.C$

R only C

$C \sqsubseteq D$

C SubClassOf D

$\mathcal{O} \models a$

ontology \mathcal{O} entails axiom a

Inferenzprobleme in Protégé

- **Ontology Consistency**
 - Reasoner exception if inconsistent
- **Ontology Coherency**
 - Inconsistent classes are indicated after classifying
- **Classification**
 - Separate inferred class hierarchy
- **Instance Retrieval**
 - DL Query tab – asking for instances of a class expression
- **Subsumption**
 - DL Query tab – asking for sub-/super classes

Übersicht

- Einführung in Protégé 4
- Umsetzung DL nach Protégé
- Modellierungsaufgaben

Incoherent Ontologies I

- Model the following
 - „graduate students as well as undergraduate students are both students“
 - „a graduate student is not an undergraduate student“
 - „an intermediate student is both graduate and undergraduate“
- Check for coherency
- Add an instance of intermediate student and see what happens

Incoherent Ontologies II

- Model the following
 - „students and staff exclude each other“
 - „staff are exactly those employed at some university“
 - „a PhD student is a graduate student employed at a university“
- Check for coherency
- Weaken the equivalence for staff to subsumption
 - why is the class PhDStudent no longer unsatisfiable?
 - think about models

General Inclusion Axioms

- Model the following as GCI
 - „staff who lectures some course is a professor“
 - „Bill is staff and lectures the statistics course“
- Add as a condition of class *Staff*
 - $Staff \sqsubseteq \forall lectures. \neg Course \sqcup Professor$
 - $Staff(Bill), Course(Statistics), lectures(Bill, Statistics)$
- Check whether Bill is among the Professors

Covering Axioms

- Model the following
 - „susan is a student but no undergraduate student“
- Check if susan is concluded to be a graduate student
- Fix the ontology appropriately, if not
 - Add the axiom $Student \sqsubseteq Graduate \sqcup Undergraduate$

Universal Quantification Trap

- Modell the following
 - „professors are staff and teach some course“
 - „any student is enrolled in some course“
 - „intermediate courses are exactly those courses in which, in any case, both graduate and undergraduate students are enrolled“ <use an inverse property>
 - „lazy professors are professors who teach only intermediate and also only non-intermediate courses“
- why is the concept for lazy professor satisfiable?
- fix this in the ontology
 - Add „Any professor teaches some course“

Open-World Reasoning

- Model the following
 - „John teaches Susan“
- Recall that
 - $KB \models Graduate(Susan)$
- Check the following
 - „Does John teach only graduate students?“
- Close off the role teaches for John
 - Add „ $(\leq 1 \text{ teaches})(John)$ “

Individual (in-)equality

- „Concrete degrees can be bachelor, master and PhD“
 - $Degree \equiv \{BachelorDegree, MasterDegree, PhDDegree\}$
- „Professor John holds all a bachelor, a master and a PhD degree“
 - $Professor(John), holds(John, BachelorDegree), holds(John, MasterDegree), holds(John, PhDDegree)$
- „Does John hold more than two degrees?“
 - $KB \models (\geq 2 \text{ holds})(John) ?$
 - only if the degrees are declared to be different individuals

Modular Design by Import

- Model part-of relationships (in a separate ontology)
 - *partOf, hasPart, Part, Aggregate*
- Import part-of ontology for modelling the following
 - „a university has faculties as parts“
 - „a faculty is part of a university and has departments as part“
 - „a department is part of a faculty and has professors as part“
 - „a professor is part of a department“
- Ask the following
 - „is a professor part of some university?“
 - „does a faculty have parts?“
 - „does a professor have parts himself?“
- Instantiate the parts and do some querying

Local vs. Global Domain/Range

- Add range *degree* to the role *holds*
- Model that
 - „a professor holds some certificate, while certificates are distinct from degrees“
- See what happens
- Introduce two subproperties of holds
 - one for degree and one for certificate
 - specify their ranges
 - ask „who holds a degree as well as a certificate“

Reasoning by Cases

- model the following
 - Mary is an undergraduate student enrolled in the Logics course
 - Susan is enrolled in the Statistics course
 - Peter is a student enrolled in both Logics and Statistics <notice that it is not known whether he is a graduate or undergraduate student>
 - John is a professor who teaches both Logics and Statistics
- check whether any of Logics or Statistics can be concluded to be an intermediate course
- check if John teaches an intermediate course
- think about it

Excercise

- Play around with the tool a bit further