

## Übung „Netzbasierende Informationssysteme“ WS 2008/2009

---

### Übungsblatt 5

Ausgabe am 2.12.2008

Abgabe bis spätestens 16.12.2008, 16.00 Uhr

#### Aufgabe 17: Semantic Web Metadata (2 Punkte)

1. Take a look at the vCard metadata standard <http://www.imc.org/pdi/vcardoverview.html> and the W3C vCard RDF representation: <http://www.w3.org/TR/vcard-rdf>.
2. Write your own vCard using RDF/XML.
3. Take a look at the FOAF metadata proposal <http://www.foaf-project.org/>. Take a look at <http://planetrdf.com/bloggers.rdf>.
4. Create your own FOAF profile, using e.g. the foaf-a-matic tool <http://www.ldodds.com/foaf/foaf-a-matic>.
5. Extend your RDF/XML FOAF profile, so that it also contains the vCard properties TITLE, ROLE and BDAY.

#### Aufgabe 18: OWL (3 Punkte)

Conceive a Family Ontology with classes for: person, male, female, father, mother, son, daughter, children, parent. And the properties(relationships): married\_to, parent\_of, children\_of, sibling\_of the ontology axiomatization should contain the following axioms:

- $(\text{male} \cap \text{female}) = \emptyset$
- $\text{married\_to}(x; y) \Leftrightarrow \text{married\_to}(y; x)$
- $\text{sibling\_to}(x; y) \Leftrightarrow \text{sibling\_to}(y; x)$
- $\text{parent\_of}(x; y) \Leftrightarrow \text{children\_of}(y; x)$

1. Implement the ontology using OWL.
2. Add several instance examples to your ontology

#### Aufgabe 19: Industrial Use Case: Project Management (5 Punkte)

Project management methodologies such as a responsibility assignment matrix are used to describe roles of project members and responsibilities of them. You as a Semantic Web specialist are requested to implement the responsibility assignment matrix of the RuleML-2008 event organizing committee which is a typical project team.

1. Familiarize with the concepts of a responsibility assignment matrix at <http://www.pmi.org/> and at [http://en.wikipedia.org/wiki/Responsibility\\_assignment\\_matrix](http://en.wikipedia.org/wiki/Responsibility_assignment_matrix)
2. Take a look at the ACM conference committee role descriptions at [http://www.acm.org/signs/volunteer\\_resources/conference\\_manual/2-1-2job](http://www.acm.org/signs/volunteer_resources/conference_manual/2-1-2job)
3. Now model the responsibility assignment matrix for the RuleML-2008 organizing committee <http://2008.ruleml.org/oc.php> in OWL.

Note: Some of the RuleML-2008 roles do not directly map to the ACM roles. Use your imagination! Hint: You might take a look at the RuleML-2007 RAM ontology at <http://ibis.in.tum.de/projects/paw/ruleml-2007/RuleML-2007.owl>.

#### Aufgabe 20: Industrial Use Case: Discounted Pricing Policy (5 Punkte)

You are requested to implement the rule-based pricing policy for the Berlin Zoo. Base entrance price for adults is 8 Euro, and for children 4 Euro.

1. Write a rule program in Prolog syntax (<http://prova.ws>; ISO Prolog syntax) to compute the entrance price. The program should implement the following rules:
  - Couples receives a discount of 5% on the entrance price.
  - A parent with two child receive receives a discount of 10% on the entrance price of each.

2. Discuss: Where do you see the benefits of rules over ontologies?
3. Serialize the rules of your Prova program in Reaction RuleML  
(<http://ibis.in.tum.de/research/ReactionRuleML>, i.e. write your rules in a Reaction RuleML document. You might take a look at the Derivation RuleML example on the Reaction RuleML website.
4. Validate your RuleML document. You might use the online W3C XML schema validator.  
[http://www.w3.org/2001/03/webdata/xsv\\_to\\_check\\_your\\_schema](http://www.w3.org/2001/03/webdata/xsv_to_check_your_schema).

#### Aufgabe 21: Semantic Web Programming (15 Punkte)

Familiarise yourself with Prova: Download and install

1. Download the zipped Prova workspace from the lecture's website and import it to your eclipse repository.

Remark: If you don't want to use the pre-built Zip file - to import the latest full distribution of Prova from the Prova subversion version control system at Sourceforge.net into your Eclipse project using SVN and install it using Maven2. The Subversion connection string (not an HTTP access) is  
<https://mandarax.svn.sourceforge.net/svnroot/mandarax/prova>. (The latest Prova 2.0 RC2 Update 8 pre-built for Java 6.0 can be also downloaded at <http://prova.ws/etc/prova-2.0-SNAPSHOT.jar>)

2. ( To compile it you would need to install the Maven2 Eclipse plug-in of the Maven Software Project Management Tool
  - a. Maven in 5 minutes  
<http://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>
  - b. Maven Getting Started  
<http://maven.apache.org/guides/getting-started/index.html>
  - c. Guide to using Eclipse with Maven 2  
<http://maven.apache.org/guides/mini/guide-ide-eclipse.html>

However, there is already a pre-built snapshot of Prova available in the \target folder in your zipped distribution.)

3. Read the Prova 2.0 User's guide [http://prova.ws/etc/provauserguide\\_2\\_0.pdf](http://prova.ws/etc/provauserguide_2_0.pdf) and take a look at the Prova website <http://prova.ws/> and the Prova slides on the NBI lecture website.
4. Please go to the folder (. \rules and . \rules\prova-examples) in the prova project. Run some of the examples using the prova.bat file (or prova.sh script under Linux) under the folder rules (on the console write `prova ./prova-examples/[filename:prova]`).

#### Rule Programming (5 Punkte)

5. Add some facts (some persons which are children, parents etc.) and some queries, to ask for the entrance price of certain persons, to your Prova rule program which you have written in exercise 20.
6. Run your rule program. Which results do you get?

#### SPARQL Programming (5 Punkte)

7. Take a look at . \rules\prova-examples\ex059.prova
8. Write a Prova program which we executes a SPARQL query to your RDF/XML FOAF profile from exercise 17 and returns the first name, last name, birthday and your friends.  
Note: You can publish you RDF FOAF profile locally using your Apache/Tomcat.
9. Run the Prova program. Which results do you get?

#### OWL Programming (5 Punkte)

10. Take a look at the rule example . \rules\examples\function\_tests\dl\_typing\typedOWL.prova and the Wine ontology . \rules\examples\function\_tests\dl\_typing\WineProjectOWL.owl.
11. Run the typeOWL.prova program. Which results do you get? Briefly explain why you get these results?
12. Briefly explain how you could use this approach to apply the family ontology which you have written in exercise 18 to the rule program which you have written in exercise 20. Give some concrete rule code examples/snippets to exemplify your description. Where do you see the benefits of this ontology/rule combination?

Kontextrelevante Online-Informationen finden sie z.B. unter:

Prova	<a href="http://prova.ws">http://prova.ws</a>
Reaction RuleML	<a href="http://ibis.in.tum.de/research/ReactionRuleML">http://ibis.in.tum.de/research/ReactionRuleML</a>
RDF, RDFS, OWL, SPARQL	W3C Webseiten

Viel Erfolg!