Ontology Updating Driven by Events

Frederik Hogenboom, Jordy Sangers, and Flavius Frasincar
fhogenboom@ese.eur.nl, jordysangers@hotmail.com, and frasincar@ese.eur.nl

Econometric Institute
Erasmus University Rotterdam
PO Box 1738, NL-3000 DR
Rotterdam, the Netherlands

In most modern Web applications, domain specific knowledge is represented by means of one or more Web ontologies, containing concepts and relations, represented using triples consisting of a subject, a predicate, and an object. Such structured data facilitates data understandability, and thus the interoperability between different computer systems.

Due to society's non-static nature, knowledge reflecting the real world requires regular updates. Traditional data sources like relational databases have mechanisms for automatic updates. However, a principled way of automatic Web ontology updating does not yet exist. This forces domain experts to manually update ontologies, which is a tedious, repetitive, error-prone, and time-consuming activity.

The event-triggered Ontology Update Language (OUL) [1] alleviates the process of manual Web ontology updating by providing a means to define sets of SPARQL/Update rules and is based on the automatic update mechanism from active databases: SQL-triggers. Using an Event-Condition-Action model, a list of ontology update actions are performed, triggered by event occurrences through so-called changehandlers.

However, this method does not support a fully automated ontology update mechanism. Therefore, in our recent efforts [2], we have extended OUL to OULx with language features, i.e., prefixes and negation, and various update execution mechanisms. For example, adding to a product ontology an item that was not yet present could be accomplished using the following OULx changehandler:

```
CREATE CHANGEHANDLER addProductHandler
PREFIX kb: <http://www.hermes.com/
            knowledgebase.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/
            22-rdf-syntax-ns#>
FOR    add(?company kb:hasProduct ?product)
AS IF (contains(?company rdf:type kb:Company)
 and !(contains(?product rdf:type kb:Product)))
THEN  insert data{?product rdf:type kb:Product};
      applyRequest;
```

where lines containing `PREFIX` define prefixes that stand for various namespaces, and the negation operator is represented by the exclamation symbol (`!`).

For update execution mechanisms, in OULx, we incorporated immediate updating, as opposed to the default deferred updating. Also, we added an internal triggering mechanism for changehandlers called updates chaining, allowing for automatic event triggering based on event actions. Furthermore, we included support for looping, enabling repetitive treatment of an event. Last, we added the option to execute all matching changehandlers for an event, instead of just the first matching handler. As a proof-of-concept, we implemented the language and its execution models, which we made available at `http://people.few.eur.nl/fhogenboom/oulx.html`.

In his talk, Frederik Hogenboom, PhD student at the Erasmus University Rotterdam, will be focusing on several aspects of OULx. First, the specifications of the existing OUL language are discussed. Second, the extensions implemented in OULx are presented. Third, a use case-based evaluation of the implemented execution models is given. The work presented here follows from a paper to be published in the proceedings of the Thirteenth International Conference on Web Information System Engineering (WISE 2012) [2].

References

[1] Uta Lösch, Sebastian Rudolph, Denny Vrandečić, and Rudi Studer. Tempus Fugit. In *6th European Semantic Web Conference on The Semantic Web: Research and Applications (ESWC 2009)*, pages 278–292. Springer-Verlag, 2009.

[2] Jordy Sangers, Frederik Hogenboom, and Flavius Frasincar. Event-Driven Ontology Updating. In X.S. Wang et al., editor, *Thirteenth International Conference on Web Information System Engineering (WISE 2012)*, volume 7651 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2012.