



OULx: Event-Driven Ontology Updates

Introduction

The Ontology Update Language (**OUL**) alleviates the process of manual updating by defining sets of SPARQL/Update rules. It is based on an automatic update mechanism and operates using the **Event-Condition-Action** model, where event occurrences trigger actions through **handlers** and preconditions are assessed. However, OUL only supports **deferred** execution of the **first** matching handler, and hence there is no support for a fully automated update mechanism.

Language

In our extended language, **OULx**, we additionally allow for the use of various operators in the update rule syntax:

- **Negation**: expresses the non-existence of specific information in an ontology.
- **Prefixes**: creates shorthands for commonly used namespaces in rules.

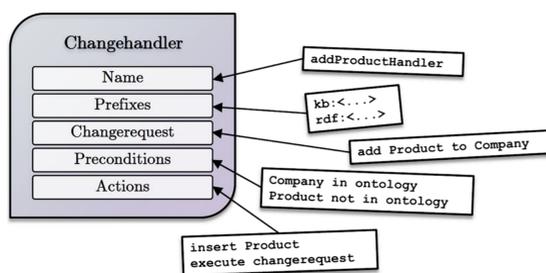


Fig. 1 A schematic representation of a handler.

Also, we extend OUL in such a way, that the following execution models are available:

- **Immediate** updating: executes updates immediately when triggered by event occurrences (instead of **deferred** updating, which stores actions first).
- Executing **all** matching handlers: considers every specified set of update actions (instead of merely the **first** match).
- Update **chaining**: allows actions performed by handlers to trigger new events that are subsequently handled by other handlers.
- Update **looping**: re-firing events until no matching handlers are found.

These execution models can also be combined into more complex configurations. We implemented the language and its execution models in the Hermes News Portal (**HNP**), an ontology-based news personalization service. A stand-alone version is available at <http://people.few.eur.nl/fhogenboom/oulx.html>.

Conclusions

A preliminary analysis of the characteristics of the various execution models shows that although termination is not always guaranteed, the proposed extensions are viable, provided that technical experts who are accustomed to the update language work together with experts of the knowledge domain.

Acknowledgement

This poster was partially supported by the Dutch Organization for Scientific Research (NWO) Physical Sciences Free Competition Project 612.001.009: Financial Events Recognition in News for Algorithmic Trading (FERNAT) and the Dutch national program COMMIT.

Contact

Frederik Hogenboom
 Postal: Erasmus University Rotterdam
 P.O. Box 1738, 3000 DR Rotterdam
 Phone: +31 (0)10 408 1262
 E-Mail: fhogenboom@ese.eur.nl
 Web: <http://people.few.eur.nl/fhogenboom/>

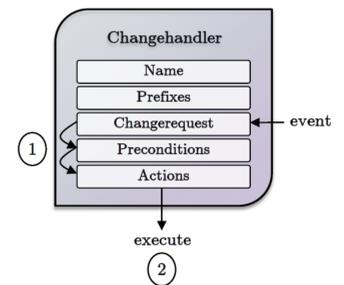


Fig. 2 Immediate execution.

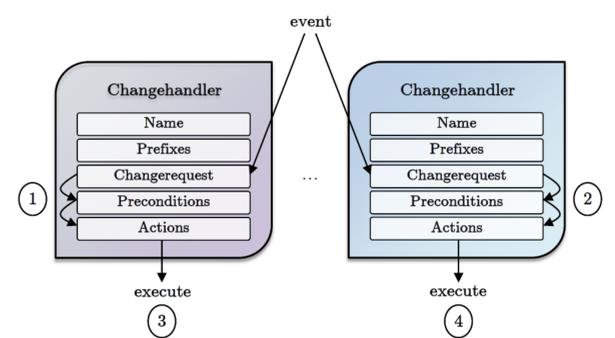


Fig. 3 Execution of all matching handlers.

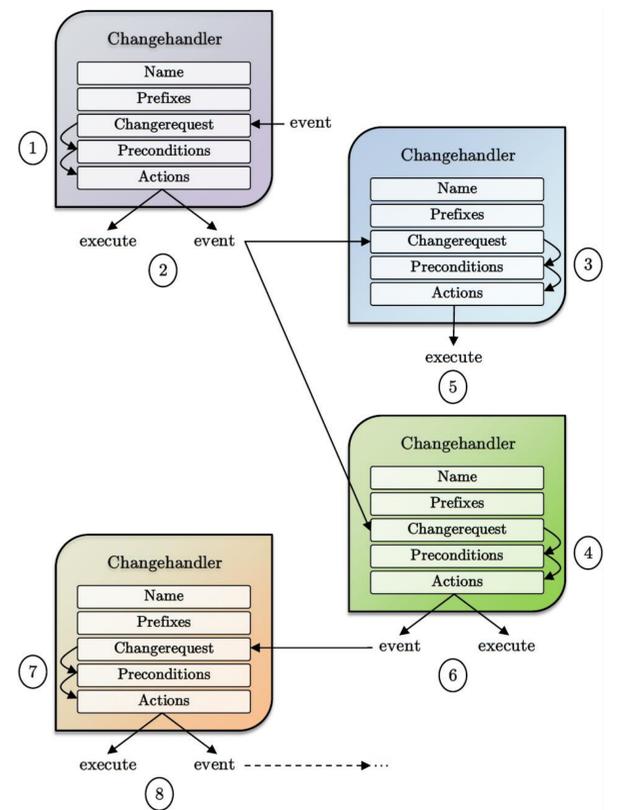


Fig. 4 Update chaining.

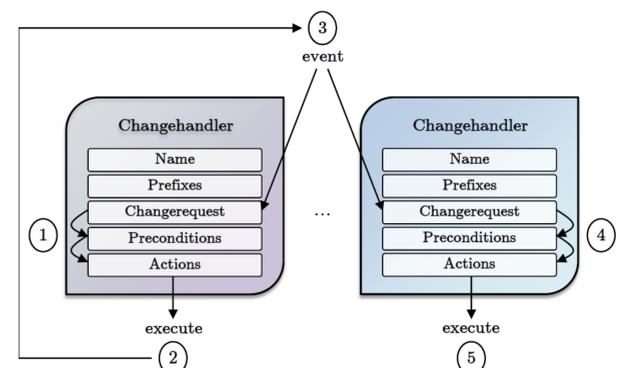


Fig. 5 Update looping.

