Master Thesis Preparation

Dominic Bosch
Departement Mathematics and Computer Science
University of Basel

July 14, 2013

Abstract. This document subsumes the main findings from the preparation work for the masters thesis.

1 Introduction

The web over time changed its role. In the first stage, it was a passive document storage driven in client-server mode. Using web 2.0 technology in the next phase more and more desk top computer usage went into the web. Today web clouds are proliferating. We anticipate a next change in the near future: The live web or reactive web.
The goal of this project is to enhance the web with rule-based event-condition-action (ECA) mechanisms such that the web turns itself into an reactive entity. After an analysis of current approaches, a generalized ECA language together with a rule engine should be provided. For test and demonstration purposes, usage scenarios should be specified and new web services should be derived. As a case study, the rule engine should be integrated with the ProBindr software, which is an advanced collaboration platform based on the shelf, binder, and register notion.

2 Related Work

In [8] the authors supplied general descriptions and classifications of different research efforts in terms of events, rules and reactiveness. Particularly of interest is their identification and summarization of existing research:

- Event/Action Logics, Transition Logics and Process Calculi: Events/Actions transit states and effect the lifetime of changeable properties (fluents). Used in [1] to specify complex actions, or in to model the communication behaviour of inbound and outbound message links in rules.
- Dynamic/Update/Transition Logics:
- Production Rule Systems:
- Active Databases and ECA Rule Systems:
- Rule-Based Complex Event Processing and Event Notification Systems: In such approaches the communication is often eased by using a middelware such as service buses. The upcoming paradigms of service-oriented architecture (SOA) and event-driven architecture (EDA) such systems allow for the reaction to the fashionable com-

plex events. The applied reactive rules are executed to a certain context,

Language XChange uses Xcerpt(?) to express web queries. MARS was postulated in 2008 (not available?). In contrast to standard ECA rules, which typically only have one global state, messaging reaction rules maintain a local conversation state that refelects the process exe3cution state. This supports supports the performing of different activities within process instances managed in simultaneous conversation branches. CEP provides enhanced situation awareness.

2.1 Rule Engines

2.1.1 Kinetics Rule Engine

2.1.2 Prova

2.1.3 OO jDrew

2.2 Rule Languages

2.2.1 Kinetics Rule Language

2.2.2 RuleML

RuleML [2] is a rule specification standard to express both forward and backward rules for derivation, reaction, rewriting, messaging, verification and transformation. The building blocks of RuleML are predicates, derivation rules, facts, queries, integrity constraints and transformation rules.

The Rule Markup Initiative [3].

2.2.3 Reaction RuleML

Reaction RuleML [10] extends RuleML to allow reaction rules and complex event/action messages, e.g. for complex events processing (CEP). It adds various kinds of production, action, reaction and knowledge representation (KR) temporal/event/action logic rules, as well as (complex) event/action messages. It consists of one general reaction rule form that can be specialized, e.g. into production rules, trigger rules, ECA rules or messaging rules. Three different execution styles (active, messaging, reasoning). Messages define inbound or outbound event messages and are used to interchange events and rule bases. A reaction rule can be globally or locally nested within other reaction or derivation rules. RuleML Interface Description Language (RuleML IDL) is a sublanguage of Reaction RuleML and allows the description of public rule functions.

2.2.4 JSON Rules

2.3 Frameworks

2.3.1 Kynetx

A framework presented in [12]

2.3.2 Rule Responder

Rule Responder [9] is a project to extend the Semantic Web towards a Pragmatic Web infrastructure for collaborative human-computer networks, which they call an architecture of a Pragmatic Agent Web (PAW). It supports the formation of virtual groupings and allows semi-automated agents with their individual contexts, decisions and actions. The authors postulate agents empowered with automatic rule-driven data transformation, decision derivation from existing knowledge and reaction according to changed situations or occurred events. The work done in this project concentrates on a layer on top of a rule engine and language,

and thus allows for a combination of arbitrary rule-based systems via their framework. This is achieved through the usage of general messge oriented communication interfaces and a platform-independent rule interchange format.
The authors of Rule Responder built their reference system on top of the Mule [5] open-source Enterprise Service Bus (ESB) which acts as a communication middleware. The decision to use Mule was made because it goes beyond the typical definition of an ESB by providing a distributable object broker to manage all sorts of service components. Each agent runs its own rbitrary rule engine. For demonstartion purposes Prova and OO jDrew were used to demonstrate the rule interchange between different rule engines.
An investigated use case for Rule Responder was a symposium organization as a virtual organization. Documentation and source files were accessible via [7].

2.4 Towards ECA Mashups

In [6], the founders of JSON Rules [4] describe a lightweight architecture that allows to react and proact on behalf of events in the ontology of web browsers.

3 Use Case Study

In order to verify some of the identified related work, use cases around the succcessor of useKit [11], ProBinder [?], have been developed and investigated.

4 Conclusion

There has been a lot of efforts to

5 Future Work

References

[1] Erik Behrends, Oliver Fritzen, Wolfgang May, and Franz Schenk. Embedding Event Algebras and Process for ECA Rules for the Semantic Web. *Fundam. Inf.*, 82(3):237–263, August 2008.

[2] H. Boley. The Rule-ML Family of Web Rule Languages, 2006.

[3] H. Boley and S. Tabet. The Rule Markup Initiative. `http://ruleml.org`. Accessed: 2013-07-07.

[4] A. Giurca and E. Pascalau. JSON Rules, 2008.

[5] R. Mason. muleESB. `http://www.mulesoft.org`. Accessed: 2013-07-07.

[6] E. Pascalau and A. Giurca. A Lightweight Architecture of an ECA Rule Engine for Web Browsers, 2009.

[7] A. Paschke. Rule Responder - Rule-based Semantic Agent Architecture. `http://www.corporate-semantic-web.de/rule-responder.html`. Accessed: 2013-07-14.

[8] A. Paschke and H. Boley. Rules Capturing Events and Reactivity, 2009.

[9] A. Paschke, H. Boley, B. Craig, and A. Kozlenkov. Rule Responder: RuleML-Based Agents for Distributed Collaboration on the Pragmatic Web, 2007.

[10] A. Paschke, H. Boley, Z. Zhao, K. Teymourian, and T. Athan. Reaction RuleML

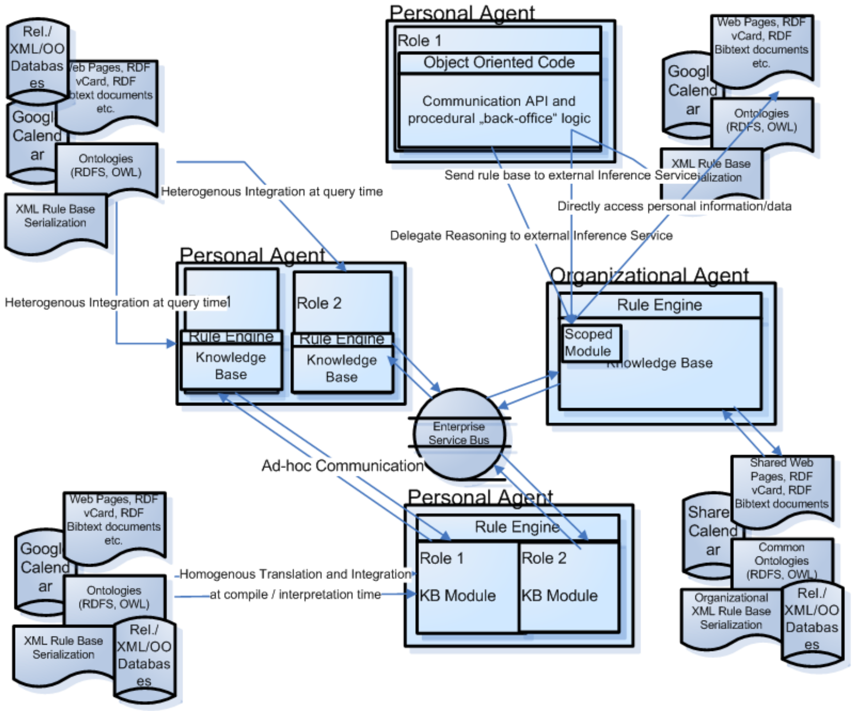
Personal Agent
Role 1
Object Oriented Code
Communication API and procedural „back-office“ logic
Rel./ XML/OO Databases
Web Pages, RDF vCard, RDF Bibtext documents etc.
Google Calendar
Ontologies (RDFS, OWL)
XML Rule Base Serialization
Heterogenous Integration at query time
Send rule base to external Inference Service
Directly access personal information/data
Delegate Reasoning to external Inference Service
Personal Agent
Role 2
Rule Engine
Knowledge Base
Heterogenous Integration at query time
Organizational Agent
Rule Engine
Scoped Module
Knowledge Base
Enterprise Service Bus
Ad-hoc Communication
Personal Agent
Rule Engine
Role 1
Role 2
KB Module
KB Module
Homogenous Translation and Integration
at compile / interpretation time
Shared Web Pages, RDF vCard, RDF Bibtext documents
Shared Calendar
Common Ontologies (RDFS, OWL)
Organizational XML Rule Base Serialization
Rel./ XML/OO Databases

Figure 1: Rule Responder Architecture, taken from [9]

1.0: Standardized Semantic Reaction Rules, 2012.

[11] S. Rizzotti and H. Burkhart. useKit - Lightweight Mashups for the Personalized Web, 2010.

[12] P. Windley. *The Live Web: Building Event-Based Connections in the Cloud*. Cengage Learning PTR, 2011.