

Deploying a Distributed Symposium Planner Through Rule Responder

**Benjamin Craig
Harold Boley**

Institute for Information Technology
National Research Council, Canada
Fredericton, NB, Canada

RuleML-2008 Challenge
Orlando Florida
October 30-31, 2008

Outline

- Rule Responder Overview
- Symposium Planner Use Case
 - Agents
 - Personal / Organizational / External
 - Rule Engines (for Realizing Agents)
 - Prova
 - OO jDREW
 - Communication Middleware (for Connecting Agents)
 - Mule ESB
 - Reaction RuleML Messages
- Online Demo
- Conclusion

Overview of Rule Responder

- Rule Responder is an experimental multi-agent system for collaborative teams and virtual communities on the Web
- Supports rule-based collaboration between the distributed members of such a virtual organization
- Members of each virtual organization are assisted by semi-automated rule-based agents, which use rules to describe the decision and behavioral logic
- Implemented on top of a Mule-based Service Oriented Architecture (SOA)

Use Case: Symposium Planner

- RuleML-20xy Symposia
 - An organizational agent acts as the single point of entry to **assist** with the symposium:
 - Currently, query answering about the symposium
 - Ultimately, preparing and running the symposium
 - Personal agents have supported symposium chairs since 2007 (deployed as Q&A in 2008)
 - General Chair, Program Chair, Panel Chair, Publicity Chair, etc.

Organizational Agents

- The organizational agent represents the goals and strategies shared by each committee chair
- It contains rule sets that describe the policies and regulations of the RuleML Symposium
- Delegates incoming queries to the chair's PAs

Personal Agents

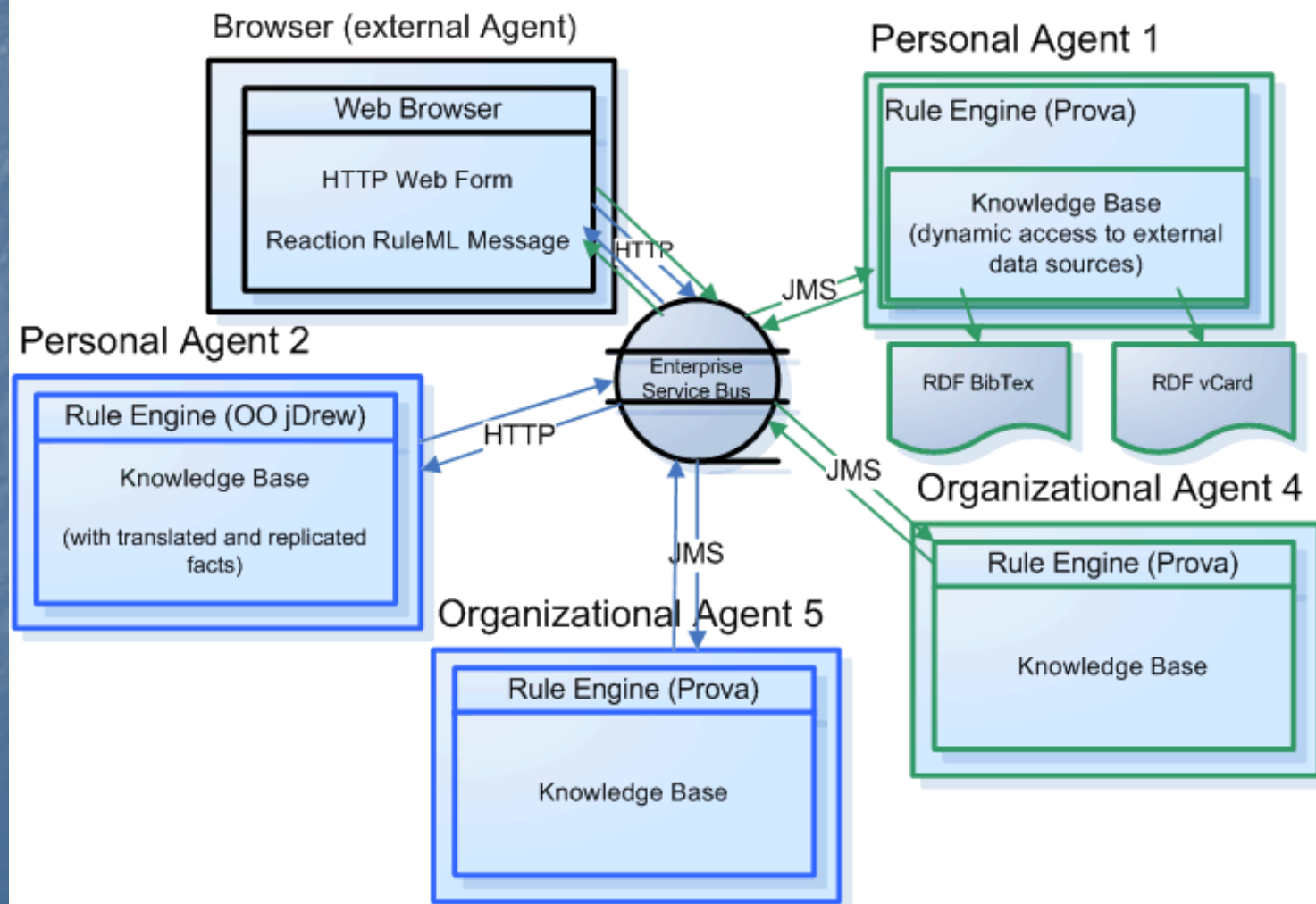
- A personal agent assists a single chair of the symposium, (semi-autonomously) acting on his/her behalf
- Each personal agent contains a rule-base FOAF-like profile
- It contains a FOAF*-like **fact** profile plus FOAF-extending **rules** to encode selected knowledge of its human owner

External Agents

- External agents exchange messages with the RuleML-2008 OA.
 - They submit queries and receive answers
- End users, as external agents, interact with the OA using a Web (HTTP) interface to the Symposium Planner
- Support for simultaneous external agents
 - Many EAs can communicate with the OA

Infrastructure - Overview

Use Case 4 Use Case 5



Reaction RuleML

- Reaction RuleML is a branch of the RuleML family that supports actions and events
- When an external agent submits a query to the Symposium planner a Reaction RuleML message must be used
- In general, when any two agents communicate, Reaction RuleML messages are sent through the ESB
 - Our ESB implementation is MULE

Communication Middleware

- **Mule Enterprise Service Bus (ESB)**
 - Mule* is used to create communication end points at each personal and organizational agent of Rule Responder
 - Mule supports various transport protocols (e.g. HTTP, JMS, SOAP)
 - Rule Responder currently uses HTTP and JMS as transport protocols

* Mule – The open source SOA infrastructure:
<http://mulesource.com>

Current Rule Engines

- Prova: Prolog + Java
- OO jDREW: Object Oriented
java Deductive Reasoning Engine for the Web

Prova

- Prova is mainly used to realize the organizational agents of Rule Responder
- It implements Reaction RuleML for agent interaction (event-condition-action rules)

OO jDREW

- OO jDREW is used to realize the personal agents of Rule Responder
 - Deployed as Java Servlets
- It implements Hornlog RuleML for agent reasoning (Horn logic rules)

Online Use Case Demo

- Rule Responder:
<http://responder.ruleml.org>
- RuleML-2007/RuleML-2008 Symposia:
<http://ibis.in.tum.de/projects/paw/ruleml-2007>
<http://www.ruleml.org/RuleML-2008/RuleResponder/>
- Personal agents:
Supporting all Chairs
- Organizational agent:
Supporting Symposium as a whole



Online

Conclusion

- Rule Responder was implemented & tested for various use cases (<http://responder.ruleml.org>) and deployed for RuleML-2008 [Q&A](#)
- Its organizational agents delegate external queries to topic-assigned personal agents
- It couples rule engines [OO jDREW](#) & [Prova](#) via Mule middleware and [RuleML 0.91](#) XML interchange format