RuleML Query Answering with Personal OO jDREW Agents in Rule Responder

Benjamin Craig
Harold Boley
Fredericton, NB
National Research Council - IIT
May 15, 2008
Outline

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

- 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 the virtual organization are assisted by semi-automated rule-based agents, which use rules to describe the behavioral and decision logic.
Overview of Rule Responder (II)

- Uses languages and engines of the RuleML family for rule serialization, based on logic and XML:
  - Hornlog RuleML: Reasoning
  - Reaction RuleML: Interaction
- Implemented on top of a Mule-based Service Oriented Architecture (SOA)
Personal Agents

- A personal agent assists a single person of an organization, (semi-autonomously) acting on his/her behalf.
- It contains a FOAF*-like fact profile plus FOAF-extending rules to encode some of the knowledge of its human owner.

* The Friend of a Friend (FOAF) project: http://www.foaf-project.org
Organizational Agents

- An organizational agent represents goals and strategies shared by each member of the organization.
- It contains rule sets that describe the policies, regulations, opportunities, etc. of its organization.
External Agents

- External agents communicate with the public interface of organizational agents, exchanging messages that transport queries, answers, or complete rule sets.

- End users, as external agents, employ a Web (HTTP) interface of Rule Responder (currently an API-like browser interface).

- Support for multiple external agents (end users) at the same time.
Architecture - Overview

Use Case 4 Use Case 5

Browser (external Agent)
- Web Browser
- HTTP Web Form
- Reaction RuleML Message

Personal Agent 1
- Rule Engine (Prova)
- Knowledge Base (dynamic access to external data sources)

Personal Agent 2
- Rule Engine (OO jDrew)
- Knowledge Base (with translated and replicated facts)

Organizational Agent 4
- Rule Engine (Prova)
- RDF BibTex
- RDF vCard

Organizational Agent 5
- Rule Engine (Prova)
- Knowledge Base
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.
- It implements Hornlog RuleML for agent reasoning (Horn logic rules).
- **Supports rules in two formats:**
  - **POSL:** Positional Slotted presentation syntax
  - **RuleML:** XML interchange syntax (can be generated from POSL)
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](http://mulesource.com)
Reaction RuleML

- Reaction RuleML is a branch of the RuleML family that supports actions and events.
- When two agents need to communicate, each others’ Reaction RuleML messages are sent through the ESB.
Use Case: Symposium Planner

- **RuleML-20xy Symposium**
  - An organizational agent acts as the single point of entry to the symposium
    - Assists with planning, preparing, and running the symposium
  - Personal agents support chairs of the symposium
    - Program Chair, Panel Chair, Publicity Chair, General Chair, etc.
Online Use Case Demo

- Rule Responder: [http://responder.ruleml.org](http://responder.ruleml.org)

- Personal agents: Supporting Panel and Publicity Chairs
- Organizational agent: Supporting Symposium as a whole
% Sample FOAF-extending rule in POSL syntax:
    mailphone(?person, ?email, ?telephone),
    role(?person, ?role),
    title(?person, ?title).

% Sample FOAF-like facts used by the above rule:
mailphone(John, john@email.com, 1-555-555-5555).
role(John, Panel Chair).
title(John, PHD).
% Sample Prova-like rule in POSL syntax:
getContact(?conference_part, ?info, ?contact) :-
Sample Message to Organizational Agent

```
<Message mode="outbound" directive="query-sync">
  <oid><Ind>RuleML-2007</Ind></oid>
  <protocol><Ind>esb</Ind></protocol>
  <sender><Ind>user</Ind></sender>
  <content>
    <Atom>
      <Rel>getContact</Rel>
      <Ind>ruleml2007_Challenge</Ind>
      <Ind>update</Ind>
      <Var>Contact</Var>
    </Atom>
  </content>
</Message>
```

Architecture - Execution

Use Case 4  Use Case 5

Browser (external Agent)
- Web Browser
  - HTTP Web Form
  - Reaction RuleML Message

Personal Agent 2
- Rule Engine (OO jDrew)
  - Knowledge Base
    (with translated and replicated facts)

Personal Agent 1
- Rule Engine (Prova)
  - Knowledge Base
    (dynamic access to external data sources)

Organizational Agent 4
- Rule Engine (Prova)
  - Knowledge Base

Organizational Agent 5
- Rule Engine (Prova)
  - Knowledge Base
Architecture - Execution
Architecture - Execution
RuleML-2007 Rule Responder

Use this text form to send a query in Reaction RuleML in format to the RuleML-2007 Responder:

```xml
<xml:ruleml2007="http://ibis.in.tum.de/projects/paw/#">
  <Message mode="outbound" directive="query">
    <oid></oid>
    <Ind>RuleML-2007</Ind>
    <protocol>
      <Ind>emb</Ind>
    </protocol>
    <sender>
      <Ind>user</Ind>
      <content>
        <Atom>
          <Rel>getContact</Rel>
          <Ind>ruleml2007.Challenge</Ind>
          <Ind>update</Ind>
          <Var>Contact</Var>
        </Atom>
      </content>
    </sender>
  </Message>
</xml:ruleml2007>
```

Send

Description:

RuleML-2007 Responder Use Case

Rule Interface Descriptions (Signatures)

(you might copy and paste the examples in the Rule Responder form):

- `performative[Performative][example]`
- `interface[Query, Description][example]`
- `agent[Agent][example]`
- `topic[Topic][example]`
- `rule[Rule][example]`
- `assignd[Agent, Topic, Role][example]`
- `getContact[Topic, Task, ContactInfo][example]`
- `permits[Author, submitting[Author, Submission]][example]`
- `submitted[Submission][example]`
- `accepted[Submission][example]`
<?xml version="1.0" encoding="UTF-8" ?>
<RuleML xmlns="http://www.ruleml.org/0.91/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.ruleml.org/0.91/xsd http://ibis.in.tum.de/research/ReactionRuleML/0.2/rr.xsd">
  <Message mode="outbound" directive="answer">
    <oid>
      <Ind>RuleResponder@iitfrdextdev01.iit-iti.priv41</Ind>
    </oid>
    <protocol>
      <Ind>esb</Ind>
    </protocol>
    <sender>
      <Ind>RuleResponder</Ind>
    </sender>
    <content>
      <Atom>
        <Rel>getContact</Rel>
        <Ind>ruleml2007_Challenge</Ind>
        <Ind>update</Ind>
        <Expr>
          <Fun>person</Fun>
          <Ind>johnAtemailDotcom</Ind>
          <Ind>PHD</Ind>
          <Ind>PanelChair</Ind>
          <Ind>John</Ind>
          <Ind>155555555555</Ind>
        </Expr>
      </Atom>
    </content>
  </Message>
</RuleML>
Sample Message to Publicity Chair Agent (I)

<content>
<Atom>
  <Rel>sponsor</Rel>
  <Expr>
    <Fun>contact</Fun>
    <Ind>Mark</Ind>
    <Ind>JBoss</Ind>
  </Expr>
  <Ind type="integer">500</Ind>
  <Expr>
    <Fun>results</Fun>
    <Var>Level</Var>
    <Var>Benefits</Var>
    <Var>DeadlineResults</Var>
  </Expr>
  <Expr>
    <Fun>performative</Fun>
    <Var>Action</Var>
  </Expr>
</Atom>
</content>
<atom>
  <sponsor>
    <contact>
      Mark
    </contact>
    <JBoss>
    </JBoss>
  </sponsor>
  <amount>500</amount>
  <results>
    <bronze/>
  </results>
  <benefits/>
  <logo/>
  <site/>
  <acknowledgement/>
  <in>
    <proceedings/>
  </in>
  <ongoing>
    <deadline/>
  </ongoing>
  <performative>
    <email/>
  </performative>
</atom>
Sample Message to Publicity Chair Agent (II)

`<content>
  <Atom>
    <Rel> sponsor </Rel>
    <Expr>
      <Fun> contact </Fun>
      <Ind> Mark </Ind>
      <Ind> JBoss </Ind>
    </Expr>
    <Ind type="integer"> 5000 </Ind>
  </Expr>
  <Var> results </Var>
  <Var> Level </Var>
  <Var> Benefits </Var>
  <Var> DeadlineResults </Var>
</content>`
- <Expr>
  <Fun>in</Fun>
  <Ind>proceedings</Ind>
</Expr>
- <Expr>
  <Fun>option</Fun>
  <Var>Benefits</Var>
  <Ind>demo</Ind>
</Expr>
- <Expr>
  <Fun>name</Fun>
  <Ind>all</Ind>
  <Expr>
    <Fun>advance</Fun>
    <Ind>publicity</Ind>
  </Expr>
</ Expr>
- <Expr>
  <Fun>distribution</Fun>
  <Expr>
    <Fun>brochures</Fun>
    <Expr>
      <Fun>all</Fun>
      <Ind>participants</Ind>
    </Expr>
  </Expr>
</ Expr>
- <Expr>
  <Fun>free</Fun>
  <Var>Benefits</Var>
  <Ind>registration</Ind>
</Expr>
- <Expr>
  <Fun>amount</Fun>
  <Ind>2</Ind>
</Expr>
- <Expr>
  <Fun>onGoing</Fun>
  <Ind>deadline</Ind>
</Expr>
- <Expr>
  <Fun>performative</Fun>
  <Ind>phone</Ind>
</Expr>
  <Fun>logo</Fun>
</Atom>
Conclusion

- Rule Responder can be used to implement a wide range of use cases that require a semi-automated decision layer.
- The Mule middleware of Rule Responder allows platform-independent deployment of multiple running use cases simultaneously.
- The system is reusable on all levels: Symposium Planner, Rule Responder, POSL, RuleML, OO jDREW, Prova, Mule.