Outline

- Rule Responder Overview
- Agents
  - Personal / Organizational / External
- Infrastructure
  - Reaction RuleML Messages
  - Message Performatives
  - Agent Communication Protocols
  - Mule ESB (Communication Middleware)
- Rule Engines (for Realizing Agents)
  - Prova
  - OO jDREW
- Symposium Planner Use Case
  - Query Delegation/Answering
  - Shared Knowledge between Pas
  - Ontology Description
- Future Work and 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 **virtual organizations**.
- Members of each virtual organization are assisted by **semi-automated rule-based agents**, which use rules to describe the **decision** and **behavioral** 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 (decision)
  - Reaction RuleML: Interaction (behavior)

- Implemented on top of a Mule-based Service Oriented Architecture (SOA) as an Enterprise Service Bus (ESB)
Personal Agents

- A personal agent assists a **person**
  – sometimes several – of an organization, (semi-autonomously) acting on their behalf
- It contains a FOAF*-like **fact** profile plus FOAF-extending **rules** to encode ‘routine’ knowledge of its human owner

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

* To be brief, the term ‘rule’ encompasses ‘fact’ (which is a rule without premise)
External Agents

- External agents exchange messages with (the public interface of) organizational agents, sending queries (requests), receiving answers (results), or interchanging complete rule sets
- End users, as external agents, employ a Web (HTTP) interface of Rule Responder (currently an API-like browser interface)
- Support for simultaneous external agents:
  - Currently, end users (B2C)
  - Ultimately, other organizations (B2B)
Rule Responder as a Multi-Agent Infrastructure

- Realizes Virtual Organizations in which a central OA mediates between EAs and PAs
  - Built on top of the Mule ESB
- Each OA is realized with an instance of a Rule Engine
- Each PA is realized with a Servlet using a Rule Engine – sometimes several
- Combines ideas of multi-agent systems, distributed rule management systems, as well as service-oriented and event-driven architectures
Two Simple Rule Responder Virtual Organizations on ESB Infrastructure
Translation Between PAs' Native Languages and OA's Interchange Language

- Each rule engine can use its own rule language.
- Agents require an interchange language so they can communicate with each other.
- Rule Responder uses RuleML as its interchange language.
- Translations between the interchange language and the PA languages are done by the PAs.
Reaction RuleML

- Reaction RuleML is a branch of the RuleML family that supports actions and events.
- When two agents want to communicate, each others’ Reaction RuleML messages are sent through the ESB.
- The ESB carries RuleML queries (requests), answers (results), and rule bases to/from agents.
Example Reaction RuleML Message

```xml
<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"
xmlns:ruleml2007="http://ibis.in.tum.de/projects/paw#">
  <Message mode="outbound" directive="query-sync">
    <oid> <Ind>RuleML-2009</Ind> </oid>
    <protocol> <Ind>esb</Ind> </protocol>
    <sender> <Ind>User</Ind> </sender>
    <content>
      <Atom>
        <Rel>getContact</Rel>
        <Ind>ruleml2009_PanelChair</Ind>
        <Ind>update</Ind>
        <Var>Contact</Var>
      </Atom>
    </content>
  </Message>
</RuleML>
```
Message Performatives

- The attribute `directive="..."` specifies the pragmatic performative
  - Message exchange/interaction protocols
- Rule Responder Performatives
  - In tradition of KQML and FIPA-ACL
  - Currently implemented: Query and Answer
  - Retract and Update requests planned in collaboration with RIF-PRD
Agent Communication Protocols

WSDL-like communication protocols

- **In-Only**
  - Message is sent from agent$_1$ to agent$_2$; then agent$_2$ executes performative

- **Request-Response**
  - Performs above In-Only; then agent$_2$ sends response to agent$_1$

- **Request-Response-Acknowledge**
  - Performs Request-Response; then agent$_1$ sends an acknowledgement to agent$_2$

- **Workflows**
  - Generalizes the above protocols to allow other compositions of message interchange between agents
Communication Middleware

- **Mule** Enterprise Service Bus (ESB)
  - Mule* is used to create communication endpoints 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)
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: [http://www.jdrew.org/oojdrew/demo/translator](http://www.jdrew.org/oojdrew/demo/translator))
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 since 2008)
    - General Chair, Program Chair, Panel Chair, Publicity Chair, etc.
Star-Like Rule Responder Architecture

EA: External Agent
OA: Organizational Agent
Spoke — PA: Personal Agent

Hub — General Chair

Liaison Chair

Program Chair

Liaison Chair

Publicity Chair

Program Chair

Publicity Chair

General Chair

Expert

Expert

Expert
Online Use Case Demo

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

- RuleML-2007/2008/2009 Symposia: 

- Organizational agent: 
  Supporting Symposium as a whole

- Personal agents: 
  Supporting all Chairs
Query Delegation

- Query delegation to personal agents is done by the organizational agent.
- Tasks for the symposium organization are managed via a role assignment matrix.
- Is defined here by an OWL Lite Ontology (alternatives: RDFS, RuleML, ...).
- Assigns (meta)topics to agents within the virtual organization: ... see next slide ...
Multiple Query Answers by PAs

- Some queries have more than one answer

- The PA will send the answers one at a time to the OA
  - interleaving backtracking and transmission

- When the PA finds no more answers, it sends an end-of-transmission message
Knowledge Shared Between Personal Agents

- Rules can be shared among personal agents
- Rules that apply to all PAs can be lifted to the OA level
- ... see next slide ...
% Sample Prova rule stored in the OA:

getContact(XID,Topic,Request,Contact) :-

% Retrieve the responsible PA (Agent) for the Topic
assigned(XID,Agent,Topic,ruleml2009_responsible),

% Send the query to the PA
sendMsg(XID,esb,Agent,"query",
     person(Role,Name,Title,Email,Telephone) ),

% Receive the answer(s)
rcvMult(XID,esb,Agent,"answer",Contact).
Personal General Chair Agent
Knowledge Base: Fact

% Sample FOAF-like fact used by the OA rule:
% Example fact stored in the General Chair’s PA

person(
  symposiumChair[ ruleML_2009, general ],
  foafname[ firstName[ Adrian ], lastName[ Paschke ]],
  foafTitle[ title[ Dr ]],
  foafMbox[
    email[ adrianDOTpaschkeATbiotecDOTtuDASHdresdenDOTde ],
    exPhones[ telephoneNumbers[ office[ 4935146340074 ]]]].

% Sample query in RuleML syntax:

... see next slide ...
Sample Message to Organizational Agent

<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"
xmlns:ruleml2007="http://ibis.in.tum.de/projects/paw#">
  <Message mode="outbound" directive="query-sync">
    <oid>
      <Ind>RuleML-2009</Ind>
    </oid>
    <protocol>
      <Ind>esb</Ind>
    </protocol>
    <sender>
      <Ind>User</Ind>
    </sender>
    <content>
      <Atom>
        <Rel>getContact</Rel>
        <Ind>ruleml2009_GeneralChair</Ind>
        <Ind>update</Ind>
        <Var>Contact</Var>
      </Atom>
    </content>
  </Message>
</RuleML>

Query Selection:  General Chair Contact
Architecture - Execution
Architecture - Execution

Use Case 4

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)

Organizational Agent 4
- RDF BibTex
- RDF vCard

Organizational Agent 5
- Rule Engine (Prova)
- Knowledge Base

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

Use Case 4

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

Use Case 5

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
- RDF BibTex
- RDF vCard

Organizational Agent 5
- Rule Engine (Prova)
  - Knowledge Base
Use this text form to send a query in Reaction RuleML format to the RuleML-2009 Organizational Agent:

```xml
<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/RuleML/0.2/ruleml2009.xsd"
xmllns:ruleml2009="http://ibis.in.tum.de/projects/paw#">
<Message mode="outbound" directive="query-sync">
  <oid>
    <Ind>RuleML-2009</Ind>
  </oid>
  <protocol>
    <Ind>esb</Ind>
  </protocol>
  <sender>
    <Ind>User</Ind>
  </sender>
  <content>
    <Atom>
      <Rel>getContact</Rel>
      <Ind>ruleml2009_GeneralChair</Ind>
    </Atom>
  </content>
</Message>
</RuleML>
```

Query Selection
The drop-down boxes show sample queries you -- as an External Agent -- can send to the RuleML-2009 Organizational Agent. These examples can also act as initial templates that you can edit to create your own queries.
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>

Online

Query Selection: Publicity Chair Sponsoring

English Description:
Mark from JBoss would like to sponsor RuleML-2009 with $500. What level, benefits, and deadline results will this provide, and what kind of action should be taken?
<Atom>
  <Rel>sponsor</Rel>
  <Expr>
    <Fun>contact</Fun>
    <Ind>Mark</Ind>
    <Ind>JBoss</Ind>
  </Expr>
  <Ind type="integer">500</Ind>
  <Expr>
    <Fun>results</Fun>
    <Ind>bronze</Ind>
  </Expr>
  <Expr>
    <Fun>benefits</Fun>
    <Expr>
      <Fun>logo</Fun>
      <Expr>
        <Fun>on</Fun>
        <Ind>site</Ind>
      </Expr>
    </Expr>
  </Expr>
  <Expr>
    <Fun>acknowledgement</Fun>
    <Expr>
      <Fun>in</Fun>
      <Ind>proceedings</Ind>
    </Expr>
  </Expr>
</Atom>
Personal Publicity Chair Agent
Knowledge Base: Rule

% Rule stored in the Publicity Chair’s PA

sponsor(contact[?Name,?Organization],
  ?Amount:integer,
  results[?Level,?Benefits,?DeadlineResults],
  performative[?Action]) :-

  requestSponsoringLevel(?Amount:integer,?Level),
  requestBenefits(?Level,?Benefits),
  checkDeadline(?DeadlineResults),
  checkAction(?Action,?Level,?Amount:integer).

Orange: Query other rules
requestSponsoringLevel(?Amount:integer,?Level),
% Satisfied by rule:
requestSponsoringLevel(?Amount:integer,?Level) :-
    sponsoringLevel(rank0, 
    ?Level, 
    under[us$[?UnderBronzeAmount:integer]],
    lessThan(?Amount:integer,?UnderBronzeAmount:integer)).

requestBenefits(?Level,?Benefits),
% Satisfied by rule:
requestBenefits(?Level,?Benefits) :-
    benefits(?Level,?Benefits).
Personal Publicity Chair Agent
Knowledge Base: 3rd & 4th Rule Premise

checkDeadline(?DeadlineResults),
% Satisfied by rule:
checkDeadline(passed[deadline]):-
date(?X:integer),
deadline(sponsoring,?D:integer),
greaterThan(?X:integer,?D:integer).

checkAction(?Action,?Level,?Amount:integer).
% Satisfied by rule:
checkAction(?Action,?Level,?Amount:integer) :-
actionPerformed(?Action,?Level,?Amount:integer).

What happens if we now provide a $5000 sponsorship?
... see next slide ...
Sample Message to Publicity Chair Agent (II)

- `<content>`
- `<Atom>`
  - `<Rel>sponsor</Rel>`
  - `<Expr>`
    - `<Fun>contact</Fun>`
    - `<Ind>Mary</Ind>`
    - `<Ind>Super</Ind>`
  - `</Expr>`
  - `<Ind type="integer">5000</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>`

Online


Query Selection: Publicity Chair Sponsoring (edit)

**English Description:**

Mark from JBoss would like to sponsor RuleML-2009 with $5000. What level, benefits, and deadline results will this provide, and what kind of action should be taken?
- <Atom>
  - <Rel> sponsor </Rel>
- <Expr>
  <Fun> contact </Fun>
  <Ind> Mark </Ind>
  <Ind> JBoss </Ind>
</Expr>
<Ind type="integer"> 5000 </Ind>
- <Expr>
  <Fun> results </Fun>
  <Ind> platinum </Ind>
- <Expr>
  <Fun> benefits </Fun>
  - <Expr>
    <Fun> logo </Fun>
    - <Expr>
      <Fun> on </Fun>
      <Ind> site </Ind>
  </Expr>
</Expr>
- <Expr>
  <Fun> acknowledgement </Fun>
  - <Expr>
    <Fun> in </Fun>
    <Ind> proceedings </Ind>
  </Expr>
</Expr>
- <Expr>
  <Fun> option </Fun>
  - <Expr>
    <Fun> sponsor </Fun>
    <Ind> student </Ind>
  </Expr>
</Expr>
- <Expr>
  <Fun> free </Fun>
  <Ind> registration </Ind>
- <Expr>
  <Fun> amount </Fun>
  <Ind> 2 </Ind>
</Expr>
Conclusion (I)

- Rule Responder was implemented & tested for various use cases (http://responder.ruleml.org) and deployed for RuleML-2008/2009 Q&A
- Its organizational agents delegate external queries to topic-assigned personal agents
- It couples rule engines OO jDREW & Prova (& Euler) via Mule middleware and RuleML 0.91 XML interchange format
Conclusion (II)

- Without a Reaction Rule Dialect, RIF could not be used for behavioral Responder logic.
- Current system is reusable on all levels: Symposium Planner, Rule Responder, POSL, RuleML, OO jDREW, Prova, Mule.
- RuleML Techn. Group with Adrian Paschke, Alexander Kozlenkov and Nick Bassiliades.
- Integrated another ‘partner engine’, Euler, for recent use cases, e.g. in WellnessRules.
Future Work (I)

- Communication between Personal Agent and Expert Owner
  - The PA may need to interact with its expert owner
  - The intended formal interaction between PAs and their owners could use email (SMTP)
  - The initial interaction language for these emails could be Reaction RuleML

- Query Decomposition
  - Each subquery of a rule can be decomposed for different PAs, followed by answer integration
Future Work (II)

- Centralized, Distributed-Hierarchical (here), and Distributed-Networked (future) Query Answering
- Centralized: Avoids Communication Overhead
- Distributed: Fault Tolerance
  - Alternative agents when an agent becomes defunct
  - Hierarchical: OA still a bottleneck
- From Centralized to Distributed Knowledge Maintenance
  - Easier to keep Distributed Rules up-to-date