Faculty of Computer Science

University of New Brunswick

Thesis Proposal for MCS Degree Program

Design and Implementation of a Distributed Query System
Supporting Symposium Organization Committees

By

Usman Ali Chaudhry

Supervisors: Dr. Harold Boley, Dr. Weichang Du

September 9, 2011
1 Introduction

Personal software agents are an exciting application of organizational multi-agent systems. Recent research has been concerned with developing personal agents for a diverse range of domains, including emergency response, military teams, office environments, factory floors and even outer space [OSS09]. The representation and reasoning of organizational structures is currently one of the most active areas of agent organization research.

When studying symposium sites online, it is obvious that symposium organization involves a multitude of procedures, consuming much effort and time of the organizing body. Individual tasks which might not seem laborious at first, bundled together, form a formidable burden for any OC to overcome. Symposium organization typically involves submission reviewing, organization partner coordination, sponsoring correspondence, panel participants management, etc. Some of these tasks can be (partially) automated. Possible OC support includes:

- Answering question by (potential) participants about the conference such as important dates, paper topics, chairs, program schedule, etc.
- Submission reviewing.
- Coordinating chair responsibilities.
- Helping the program chair to monitor and if necessary move important dates.
- Helping the liaison chair with special events by symposium partners.
- Helping the publicity chair with sponsoring correspondence
- Helping the panel chair in managing panel participants.
Earlier conference support systems include EasyChair\(^1\) and WitanWeb\(^2\), with a focus on submission reviewing, as well as the SymposiumPlanner-2007 through -2011 series of Rule Responder instantiations\(^3\), which focuses on question answering as part of the official websites of the RuleML Symposia. The SymposiumPlanner instantiations span various implementations from initial versions in 2007, 2008, and 2009, to the 2010 instantiation based on Emerald (hence Jade) [KOBB11], to the 2011 double-instantiation using the latest versions of Mule and Prova as well as a more user friendly interface involving Suborganizational Agents for covering both sites.

Our improved SymposiumPlanner design and implementation will be explored in a use case, the RuleML-2012 Symposium. Its support will build on the earlier SymposiumPlanners and an extensive literature survey, distilling design principles and implementation techniques for future SymposiumPlanners from the lessons learned so far, as started in [ZPAB11].

## 2 Background & Related Work

Rule Responder\(^4\) is a service-oriented middleware tool that can be used by virtual organizations for automated rule-based collaboration [PKB10]. Distributed users (humans or agents) can interact with Rule Responder by query answering conversations [CB08] based on an Enterprise Service Bus (ESB). Rule Responder’s architecture realizes a system of personal agents (PAs) and organizational agents (OAs), accessed by external agents (EAs), on top of an Enterprise Service Bus (ESB) communication middleware [CB08]. The rule-based PAs represent, as their ‘dynamic profiles’, all of the participating human mem-

---

\(^1\)http://www.easychair.org/
\(^2\)http://witanweb.ca/cascon2010/WitanWebFAQ.jsp
\(^3\)http://ruleml.org/SymposiumPlanner
\(^4\)http://responder.ruleml.org
bers of the virtual organization modelled by the Rule Responder. An OA constitutes, an intelligent query dispatching system, using a rule engine execution environment for selectively delegating queries to one of the PAs.

2.1 Approaches to Knowledge Representation

In Rule Responder we make use of two paradigms of knowledge representation: Ontologies and rules. The OA uses a Responsibility Assignment Matrix to delegate incoming queries to particular PAs based on their responsibilities. This is achieved through the use of an ontology (OWL Lite). The Organizational Agent is implemented using the Prova Semantic Web rule engine. The Organizational Agent uses reaction rules that describe its policies, regulations, opportunities, etc. Each Personal Agent uses derivation rules to answer queries relevant to
the supported chair’s role. Hence, our framework uses a combination of rules and ontologies.

2.2 Hierarchical Distributed Architecture

A Star topology (hierarchy) constitutes simple but effective distributed topology. In our SymposiumPlanner, the star topology connects the specialized PAs ‘spokes’ to the centralized OA ‘hub’. A star topology is partially fault tolerant: If a spoke is broken then it will not affect the rest of the spokes.\(^5\) However, if the hub is broken then the system ceases to function.

2.3 Organizational Agent

An Organizational Agent is used to describe the goals of the symposium as a whole and contains, a knowledge base that describes the symposium’s policies, regulations, and opportunities. This knowledge base contains condition/action/event rules as well as derivation rules. An OA manages its local PAs, providing control of their life cycles and ensuring overall goals and policies of the organization and its semiotic structures [BP11].

2.4 Personal Agents

In the SymposiumPlanner system, each organization committee chair is designed as a Personal Agent, which contains a knowledge base that represents its chair’s responsibilities to answer corresponding queries. Personal agents are chair’s roles in the symposium organization. Personal Agents contain FOAF-extending profiles for each person of the organizational team.

\(^5\)http://learn-networking.com/network-design/a-guide-to-network-topology
2.5 Communication Among Agents

A lot of the early ESB products had a history in the enterprise application integration (EAI) market. We can identify two major differences between enterprise service bus (ESB) and EAI products. The first is the change from the hub-and-spoke model in EAI products to a bus-based model in ESB products. The hub-and-spoke model is a centralized architecture, where all data exchange is processed by a hub. The bus model uses a distributed architecture, in which the ESB functionality can be implemented by several physically separated functions. A second difference between EAI and ESB products is the use of open standards.

Mule open-source ESB allows the deployment of rule-based agents on the Mule object broker and supports the communication in this rule-based agent processing network via a multitude of transport protocols. Several agent services which at their core run a rule engine are installed as Mule components which listen at configured endpoints, e.g., JMS message endpoints, HTTP ports, SOAP server/client addresses or JDBC database interface. The large variety of transport protocols
provided by Mule can be used to transport the messages to the registered endpoints or external applications/tools. Usually, JMS is used for the internal communication between distributed agent instances, while HTTP and SOAP is used to access external Web services. The use of Mule allows architectural flexibility by decoupling the functional components of Rule Responder from the communication components [Cha04].

2.6 Interchange Languages and Rule Engines

For SymposiumPlanner System, we use Reaction RuleML as our interchange language between agents. Reaction RuleML\(^6\) is a general, practical, compact and user-friendly XML-serialized sublanguage of RuleML. The SymposiumPlanner System uses two representative rule engines, namely Prova\(^7\) and OO jDREW\(^8\). Prova is mostly used to process reaction rules. OO jDREW, is used to process derivation rules. There exists a bi-directional translator between RuleML/XML and POSL\(^9\)(human-readable, Prolog like syntax). The POSL language is faster to write and easier to read than any XML syntax.

\(^6\)http://reaction.ruleml.org
\(^7\)http://prova.ws
\(^8\)http://ruleml.org/oojdrew/
\(^9\)http://ruleml.org/submission/ruleml-shortation.html
3 Thesis Objectives

When studying symposium sites online, it’s obvious that symposium organization involves a multitude of procedures, consuming much effort and time of the organizing body. Individual tasks which might not seem laborious at first, bundled together, form a formidable challenge for any OC to tackle. Symposium organization typically involves submission reviewing, organization partner coordination, sponsoring correspondence, panel participants management, etc. Our overarching objective is to design, implement and evaluate a distributed query system supporting symposium organization committees. This is achieved by the following specific objectives.

1. Our system will distill design principles and implement techniques for SymposiumPlanner-2012 from the lessons learnt so far.

2. Our system will provide Personal Agents which can assist the users and participants of a virtual organization. The Personal Agents are semi-autonomous in nature; they can provide help in dealing with routine tasks. The Personal Agents use knowledge bases that encode knowledge of each human chair.

3. Our system will allow different rule engines to execute global and local rulebases.

4. Our system will allow rules as well as queries and their answers to be transmitted over an Enterprise Service Bus (ESB).

5. As a demonstration of our methodology, our deployed system will support the Question&Answer (Q&A) parts of the official websites for the RuleML Symposia.

6. In order to evaluate our system, as we will solicit feedback from the real-world OC. This practise can result in continuous revisions
of the knowledge bases of the Personal Agents, the Organizational Agent and perhaps even the system architecture.

4 Proposed Work

The main approach of this thesis is as follows: We will build a new Rule Responder instantiation, SymposiumPlanner-2012 for symposium planning.

- Architecture Design
  - In our system the agent communication will be achieved in a star-like (hierarchical) way (see Figure 3). In particular, when an Enquiry User asks a question to the organization via the External Agent, it does not know any Personal Agent such as the one that ultimately might answer the question. The query will be forwarded to the Organizational Agent, which will then delegate it to an
appropriate Personal Agent.

- **Ontology Design and Rules**
  - Our system will employ both ontologies and rulebases. The use of an ontology for the Responsibility Assignment Matrix (OWL Lite) and rulebases in the Organizational Agent and Personal Agents, gives the framework a hybrid character with both forms of knowledge representation.

- **Implementation of the System:**

  1. We will use the Mule open-source ESB to handle message-based interactions between the rule engines PROVA and OO jDrew. JMS (Java Message Service) will be used for internal communication between distributed agent instances while the HTTP and SOAP will be used to access external Web services.

  2. Our system will include the Prova rule engine to execute the Knowledge Bases (KBs) rules for the Organizational Agent. Reaction RuleML will be used by PROVA and act as an interchange language between distributed rule agents in the SymposiumPlanner.

  3. A single Organizational Agent will be implemented in PROVA to deliver and filter queries to the Personal Agents. The selection logic for the dissemination of queries to PAs will be described by Responsibility Assignment Matrix implemented using an OWL Lite Ontology.

  4. PAs represent a committee member including the following ones:
     - General Chair
     - Program Chair
     - Publicity Chair
     - Liaison Chair
The Positional-Slotted Language (POSL), will be used to build various profiles of Personal Agents. Personal Agents for each OC member will be implemented in OO jDREW having their own rule-based decision logic on top of their personal information sources. The OO jDREW rule engine will be used by each of the Personal Agents.

5. An External Agent will be implemented to provide a user the Web interface to interact with the system and use queries to find information regarding the Symposium.

6. Our system will integrate existing factual information from the hosting conference on the Internet, this will help us avoid redundancy in the knowledge bases of the SymposiumPlanner agents.

• SymposiumPlanner-2012 Use Case

In the development of the SymposiumPlanner-2012 use case we will be taking a proactive approach. We are designing, implementing, and deploying the system well in advance to the actual August 2012 symposium collocated with ECAI-2012, so that OC roles will be well-prepared, based on the 2007-2011 experience. ‘Corporate’ know-how of the RuleML directors, will be available to the thesis developer. The human RuleML-2012 chairs can then fully benefit from the prepared PA knowledge bases, building on and modifying/Extending the encoded knowledge. This will also help the OC chairs to select and possibly change roles in case they fit into another role in a more efficient manner. This gradual adoption will serve to evaluate and tune the system before it is utilized it for participant query answering.

Our Web interface should allow participants to issue queries via web forms, which will generate both RuleML/XML and controlled
natural language. By reusing and integrating existing factual information about hosting ECAI-2012 conference on the Internet, we envision to avoid redundancy in the knowledge bases of the SymposiumPlanners agents\(^{10}\). Since, there will be only one symposium site in 2012, we will need to reunify the 2-SubOrganizational Agents’ business logic of SymposiumPlanner 2011 into one Organizational Agent.

In order to evaluate the SymposiumPlanner-2012 system, we will solicit feedback from the emerging RuleML-2012 OC, which will lead to continuous revisions of the knowledge bases, and perhaps the SymposiumPlanner-2012 system architecture as an instantiation of Rule Responder. This will then allow further improved designs and implementations for the SymposiumPlanner systems of 2013 and beyond.

One of the main advantages of SymposiumPlanner is that it answers user’s queries and reduces user’s burden of finding the interested information by themselves. The queries include the information about the Symposia. For its usability, the SymposiumPlanner user client provides an interface to distributed Personal Agents, allowing users to query the available interfaces, describe and submit the queries, and retrieve the answers from a standard Web browser for the External Agent.

SymposiumPlanner user client allows users to query the SymposiumPlanner agents via the SymposiumPlanner interface [FKS06].

---

\(^{10}\text{There are large semantic data repositories on the Internet, such as: DBpedia (http://de.dbpedia.org), Freebase (http://www.freebase.com), YAGO (http://www.mpi-inf.mpg.de/yago-naga/yago), and Semantic Web Dog Food (http://data.semanticweb.org).}\)
• Evaluation of the System

In order to evaluate our system, implementation will be subjected to testing in the following scheme,

– Mule ESB Testing
  * Level One: Unit Testing

JUnit is a well-known test framework and is the industry standard to implement unit tests for Java applications. For message flows developed with Mule, we can unit test the system with various inputs. A sample output from a JUnit test is as follows,

```java
import org.mule.umo.*;
import org.mule.tck.FunctionalTestCase;
import org.mule.extras.client.MuleClient;
import org.mule.providers.NullPayload;

public class MuleServerTest extends FunctionalTestCase {
  public MuleServerTest() {
    super();
  }

  protected String getConfigResources() {
    return "mule-config.xml";
  }

  public String getName() {
    return "Mule Server Test";
  }

  public static void testTestCase() throws Exception {
    MuleClient client = new MuleClient();
    UMOMessage message = client.send("vm://test","hello", null);
    assertNotNull(message);
    assertFalse(message.getPayload() instanceof NullPayload);
    System.out.println("Got Response: " + message.getPayloadAsString());
  }
}
```

Testing: MuleServerTest(MuleServerTest)=
log4j:WARN No appenders could be found for logger (org.mule.config.MuleDtdResolver).
log4j:WARN Please initialize the log4j system properly.

Got Response: OK - Completed Event: hello

* Level Two: Functional Testing

This testing involves testing the transformation scripts (XSL) to check if the scripts used in the system produce expected results for known inputs. This can be easily achieved by defining a set of typical acceptable messages, applying the
desired transformation and comparing the outcome with what was expected.

* Level Three: Integration Testing

This testing involves deploying the whole configuration in a testing environment that is as close to the production one as possible and exercising the ESB with different happy and failing messaging scenarios. Since, Mule supports heterogeneous inbound endpoints for example,

<http:inbound-endpoint address="http://${http.host}:${BRF2011_PORT}"/>
<http:inbound-endpoint address="http://${http.host}:${IJCAI2011_PORT}"/>

It can be tested through it’s responses for the correctness of the integration of endpoints to see, whether they match the input or not.

– Correctness of Responses: The system can be tested with various inputs to check whether the query is delegated to the right Personal Agent and the response(s) are as expected.

- The Rule Responder Benchmark and Testing tool can be used to test the response time of the various use cases of Rule Responder and provide the user with the ability to gather execution time data for specific instantiation queries of their choice\(^\text{11}\).

- While developing the infrastructure for SymposiumPlanner-2012, we will solicit feedback from the OC for evaluating the system including aspects like knowledge bases and architecture. The evaluation process will also include testing from end user perspective to gauge the completeness, correctness and consistency of the system responses.

\(^{11}\text{https://mandarax.svn.sourceforge.net/svnroot/mandarax/RRBenchmarkingAndTesting/}\)
5 Schedule

<table>
<thead>
<tr>
<th>Time Schedule</th>
<th>2011, April - August</th>
</tr>
</thead>
<tbody>
<tr>
<td>Literature Study and Background Research</td>
<td>2011, August</td>
</tr>
<tr>
<td>Proposal Submission</td>
<td>2011, August</td>
</tr>
<tr>
<td>SymposiumPlanner Design</td>
<td>2011, September - Oct</td>
</tr>
<tr>
<td>- Ontology Design</td>
<td>2011, September</td>
</tr>
<tr>
<td>- System Design</td>
<td>2011, September - Oct</td>
</tr>
<tr>
<td>SymposiumPlanner Implementation</td>
<td>2011, Oct - Jan</td>
</tr>
<tr>
<td>- PAs Implementation</td>
<td>2011, Oct - Nov</td>
</tr>
<tr>
<td>- OA Implementation</td>
<td>2011, Nov - Dec</td>
</tr>
<tr>
<td>- EA and interface Implementation</td>
<td>2011, Dec - Jan</td>
</tr>
<tr>
<td>Testing and Evaluation</td>
<td>2012, Feb - March</td>
</tr>
<tr>
<td>Thesis Defense</td>
<td>2012, June</td>
</tr>
</tbody>
</table>

References


[OSS09] Steven Okamoto, Katia Sycara, and Paul Scerri. Personal assistants for human organizations. In Virginia Dignum,


