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

August 15, 2011

1 Introduction

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

Rule Responder[1] is a service-oriented middleware tool that can be used by virtual organizations for automated rule-based collaboration. Distributed users (humans or agents) can interact with Rule Responder by query answering conversations based on an Enterprise Service Bus (ESB). Rule Responder agents will process events, queries and requests according to their rule-based decision and behavioural logic. It can also delegate subtasks to other agents, collect answers, and send the validated answer(s) back to the requester. Since the Rule Responder framework has been conceived, many instantiations of it have been developed in areas as diverse as Service Level Management, Business Process Management, Symposium Planning, and Health Care systems. The main benefits from such use are,

- Alleviating the burden of repetitive tasks.

- Automation of rule based organization processes.

This thesis will design and implement, in Rule Responder, a distributed query system supporting symposium organization committees. When studying symposium sites online, it can be observed that symposium organization involves a multitude of procedures, consuming a lot of energy and time of the organizing body. Individual tasks which might not seem laborious at first, bundled together, form a formidable

[1] http://responder.ruleml.org

challenge for any organizing committee (OC) to tackle with. Symposium organization typically involves submission reviewing, organization partner coordination, sponsoring correspondence, panel participants management, etc. Our envisioned OC support includes:

- Coordinating chair responsibilities
- Finding contact information about selected chairs of the symposium.
- Helping the program and track chairs with mapping planned paper.
- Helping the program chair to monitor and possibly move important dates.
- Helping the liaison chair with special events by symposium partners.
- Helping the panel chair with managing panel participants.
- Helping the publicity chair with sponsoring correspondence
- Answering questions of participants about the conference such as important dates, topics addresses, program schedule etc.

Earlier conference support systems include EasyChair[2] and WitanWeb[3], with a focus on submission reviewing, as well as the SymposiumPlanner-2007 through -2011 series of Rule Responder instantiations[4], with a focus on Question Answering (QA) 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), to the 2011 double-instantiation using the latest in Mule and Prova with a more user friendly interface involving three Organizational Agents instead of just one for covering both sites.

[2] http://www.easychair.org/
[3] http://witanweb.ca/cascon2010/WitanWebFAQ.jsp
[4] http://ruleml.org/SymposiumPlanner

Our SymposiumPlanner design and implementation for RuleML-2012 will build on these earlier instantiations and an extensive literature survey, distilling design principles and implementation techniques for future SymposiumPlanners from the lessons learned so far, as started in [Zhao Paschke 2011].

In the development of SymposiumPlanner-2012 we will be taking a proactive approach. We are designing, implementing, and deploying the system well in advance to the actual August 2012 event, so that OC roles will be well-prepared with the 2007-2011 experience and corporate know-how of the RuleML directors, available to the thesis developer. RuleML-2012 can then fully benefit from the prepared personal agents knowledge bases, building on and modifying/extending the encoded knowledge. This will also help human OC members to select and possible change roles should they observe that they fit into another role in a more efficient manner. This will also serve as a suitable way of evaluating the system before the actual symposium committee utilizes it for its own purpose.

For SymposiumPlanner-2012 we will strive for increased automation of processes as well as better support for human operators to add flexibility to the overall system. Through extensive literature review, all the lessons learnt so far will be incorporated into the design and implementation of a new OC. Our Web interface should allow users to issue formal queries via web forms, which will generate both RuleML/XML and controlled natural language. By reusing and integrating existing factual information on the Internet, we plan to avoid redundancy in the knowledge bases of the SymposiumPlanners agents[5]. Since there will be only one symposium site in 2012, we will need to reunify the business logic into one Organizational Agent.

In order to evaluate the SymposiumPlanner-2012 system, we will

[5]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).

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.

2 Background

Rule Responder‘s architecture realizes a system of personal agents (PAs) and organization agents (OAs), accessed by external agents (EAs), on top of an Enterprise Service Bus (ESB) communication middleware. The rule-based PAs represent, as their dynamic profiles, all of the participating human members of the virtual organization modelled by the Rule Responder. An OA on the other hand constitutes an intelligent filtering and dispatching system, using a rule engine execution environment for either blocking incoming queries or selectively delegating them to other agents.

2.1 Approaches to Knowledge Representation

In terms of knowledge representation, we have two paradigms in practise, as in the use the ontologies and rules. We make use of both in our framework, the Organizational agent uses an Responsibility Assignment Matrix to assign queries to particular Personal Agents based on their responsibilities to answer certain type of queries. This is achieved through the use of an ontology (OWL Lite). This Responsibility Assignment Matrix can be used by the Rule Responder agent via querying it with the Semantic Web built-ins of Prova, binding the respective roles and responsibilities to typed variables in the agent‘s rule logic. The Organizational Agent which requires high levels of expressiveness to represent the logic of cognitive agents, are implemented

using the Prova Semantic Web Rule engine. The Organizational Agent uses a rulebase that describes its policies, regulations, opportunities, etc. Hence, our framework uses a combination of rules and ontologies. Each Personal Agent has a knowledge base containing the responsibilities of the position in order to answer queries relevant to the chair's role.

2.2 Hierarchical Distributed Architecture

A hierarchical distributed architecture employs a client-server-like topology with communication restricted to 'top-down' querying from the server to the clients and, conversely, 'bottom-up' query answering,as well as a single point of contact. Star topologies are the most commonly used topologies. The star topology connects all nodes through a centralized hub. This is also commonly known as a client-server-like architecture. Where the server is the hub and all the clients are the outside spokes. A star topology is fault tolerant because if a spoke is broken then it will not affect the rest of the spokes.[6]

2.3 Organizational Agent

An organizational agent is used to describe the goals shared by its 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 Personal Agents (PAs), providing control of their life cycles and ensuring overall goals and policies of the organization and its semiotic structures [1].

[6] http://learn-networking.com/network-design/a-guide-to-network-topology

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 chairs' roles in the symposium organization. Personal Agents contain FOAF-extending profiles for each person of the organizational team.

Query delegation is done by the organizational agent, but the personal agents can help the OA in this responsibility. The task responsibility in SymposiumPlanner is managed through a Responsibility Assignment Matrix, which defines the tasks that committee members are responsible for. The matrix, defined by an OWL Lite Ontology, assigns roles to topics within the virtual organization. Should there be still no unique PA to delegate a query to, the OA needs to make a heuristic delegation decision and send the query to the PA that most likely would be able to answer the query.

2.5 Communication Among Agents

To seamlessly handle message-based interactions between the Rule Responder agents/services and other agents/services using disparate complex event processing (CEP) technologies, transports, and protocols, the Mule open-source ESB[7] is used in Rule Responder as the communication middleware. This ESB allows deploying the rule-based agents as highly distributed rule inference services installed as Web-based endpoints on the Mule object broker and supports the communication in this rule-based agent processing network via a multitude of transport protocols. That is, the ESB provides a highly scalable and flexible application messaging framework to communicate synchronously or asynchronously amongst the ESB-local agents and with

[7] http://www.mulesoft.org/

agents/services on the Web.

Distributed agent services, which at their core run a rule engine, are deployed as Mule components which listen at configured endpoints, e.g., JMS message endpoints, HTTP ports, SOAP server/client addresses or JDBC database interfaces, etc.

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.

2.6 Interchange Languages and Rule Engines

For SymposiumPlanner System we use Reaction RuleML as our interchange language between agents. Reaction RuleML[8] is a general, practical, compact and user-friendly XML-serialized sublanguage of RuleML for the family of reaction rules. It incorporates various kinds of production, action, reaction, and KR temporal/event/action logic rules as well as (complex) event/action messages into the native RuleML syntax using a system of step-wise extensions.

The system uses two representative rule engines, namely Prova[9] and OO jDREW[10]. Prova follows the spirit and design of the recent W3C Semantic Web initiative and combines declarative rules, ontologies and inference with dynamic object-oriented Java API calls and access to external data sources via query languages such as SQL, SPARQL and XQuery.

In the case of OO jDREW, there exists a bi-directional translator between RuleML and POSL[11](RuleML human readable, Prolog like syntax). The POSL language is faster to write and easier to read

[8] http://reaction.ruleml.org
[9] http://prova.ws
[10] http://ruleml.org/oojdrew/
[11] http://ruleml.org/submission/ruleml-shortation.html

than any XML syntax. OO jDREW is an object oriented extension of jDREW(Java Deductive Reasoning Engine for the Web) and it works as a java reasoning engine for executing RuleML rules as well as the Object Oriented extensions.

2.7 SymposiumPlanner User Client

One of the main advantages of SymposiumPlanner is that it answers user's queries promptly 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.

SymposiumPlanner user client allows users to query to the SymposiumPlanner agents via the SymposiumPlanner interface [2].

3 Thesis Objectives

Our overarching objective is to design, implement and evaluate a distributed query system supporting symposium organization committees. This is achieved by the following concrete objectives.

1. In our system the agent communication will be achieved in a hierarchical way, the agent-to-agent communication is channelled through the Organizational Agent. In particular, when an External Agent asks a question to the organization, 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.
2. Our system will provide Personal Agents which can assist the local entities of a virtual organization. Often these are humans but it

can also be services or applications. The Personal Agents are semi-autonomous in nature as we do not think it is possible to replace an actual human in the organization, however they can provide a helping hand in dealing with certain tasks that have a certain degree of redundancy in terms of information.

3. Our system will benefit from the Personal Agent knowledge bases, building on and modifying the encoded knowledge. This opportunity can be utilized to pre-structure the knowledge bases for the Personal Agents.

4. Our system will be able to be deployed on a local machine as well as the Internet. The Symposium planning websites online can host the system to assist users find information by issuing queries to the system. The system is developed locally and hence can be deployed in a local environment as well depending on needs of an organization. However, since our implementation is mostly dealing with Symposium Organizing bodies, it is more useful online.

5. Our system will support the Questions&Answers (Q&A) parts of the official websites for the RuleML Symposia. It will embody responsibility assignment, automated first-level contacts for information regarding the symposium, helping as well in sponsoring correspondence.

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

7. Since, the system will be ready before the actual Symposium dates. We plan to have ample time for the OC (Organizing Committee) roles to be well prepared from the experience gained from 2007-2011 and corporate know-how of the RuleML directors. The OC members will be able to select or possibly change their roles, should

they observe that they fit in some other role in a more efficient manner.

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

9. Our system will reunify the business logic into one Organizational Agent instead of three in the previous configuration of 2011.

10. Our system will involve development of virtual RuleML 2012 OC. This implementation will also help in evaluating our system, as we will solicit feedback from the real world OC who it is designed to assist. This practise can result in continuous revisions of the knowledge bases of the Personal Agents, the Organizational Agent and perhaps even the system architecture. These improvements can provide the ground for further improvements in future designs and implementations of the SymposiumPlanner.

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.

1. Our system will employ both ontologies and rulebases for functioning. The use of ontology in the responsibility assignment matrix (OWL Lite) and rulebases in the Organizational Agent giving the framework a hybrid of both forms of knowledge representation. Responsibility Assignment Matrix will be used by the Organizational Agent to delegate queries to the Personal Agents.

2. Our system will include the Prova rule engine to execute the Knowledge Bases (KBs) rules for the Organizational Agent.

3. The OO jDREW rule engine will be used by each of the Personal Agents representing a committee member including the following ones:

   (a) General Chair
   (b) Program Chair
   (c) Publicity Chair
   (d) Liaison Chair
   (e) Panel Chair

   The Positional-Slotted Language (POSL), will be used to build various profiles of Personal Agents.

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

5. Personal Agents for each OC member will be implemented having their own rule-based decision and behavioural logic on top of their personal information sources.

6. We will use the Mule open-source ESB to handle message-based interactions between the agents/services and other agents/services using disparate complex event processing technologies, transports and protocols. JMS (Java Message Service) will used for internal communication between distributed agent instances while the HTTP and SOAP will be used to access external Web services.

7. Reaction RuleML will act as an interchange language between distributed rule agents in the SymposiumPlanner.

8. SymposiumPlanner User client will be implemented to provide user the Web interface to interact with the system and use queries to find information regarding the Symposium.

9. 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. 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[12].

[12] https://mandarax.svn.sourceforge.net/svnroot/mandarax/RRBenchmarkingAndTesting/

5 Schedule

Time Schedule	
Literature Study and Background Research	2011, April - August
Proposal Submission	2011, August
SymposiumPlanner Design	2011, August - September
SymposiumPlanner Implementation	2011, September - January
Testing and Evaluation	2012, Jan - Feb
Thesis Report writing	Feb, 2012 - March, 2012
Thesis Defense	2012, April

References

[1] Boley, H., Paschke, A.: Rule responder agents framework and instantiations. In: Eli, A., Kon, M., Orgun, M. (eds.) Semantic Agent Systems, Studies in Computational Intelligence, vol. 344, pp. 3–23. Springer Berlin / Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-18308-9_1

[2] Fuchs, N.E., Kaljurand, K., Schneider, G.: Attempto controlled english meets the challenges of knowledge representation, reasoning, interoperability and user interfaces. In: Sutcliffe, G., Goebel, R. (eds.) FLAIRS Conference. pp. 664–669. AAAI Press (2006), http://dblp.uni-trier.de/db/conf/flairs/flairs2006.html#FuchsKS06