

SymposiumPlanner-2011: Querying Two Virtual Organization Committees

Zhili Zhao¹, Adrian Paschke¹, Chaudhry Usman Ali², and Harold Boley^{2,3}

¹ Computer Science Department, Freie Universität Berlin, Germany
`{zhili.zhao,paschke}@inf.fu-berlin.de`,

² Faculty of Computer Science, University of New Brunswick, Canada
`maniali@gmail.com`,

³ Institute for Information Technology, National Research Council Canada
`harold.boleymail@nrc.gc.ca`

Abstract. SymposiumPlanner-2011 is the newest in a series of Rule Responder instantiations for the Questions&Answers (Q&A) sections of the official websites of the RuleML Symposia. It supports committee members via personal agents based on member-encoded rule knowledge. The personal agents are invoked by an organizational agent which selects them using a responsibility assignment matrix. This paper describes SymposiumPlanner-2011, which goes beyond the previous instantiations by providing a more powerful user interface and reducing redundancy in the internal conference data repositories.

1 Introduction

SymposiumPlanner⁴ is a series of Rule Responder[1, 5] instantiations for the Questions&Answers (Q&A) sections of the official websites of the RuleML Symposia since 2007. SymposiumPlanner utilizes an Organizational Agent (OA) to handle the filtering and delegation of incoming queries. Each committee chair has a Personal Agent (PA) that acts in a rule-based manner on behalf of the chair. This PA manages the chair's personal information, such as a FOAF(Friend of a Friend⁵)-like profile containing a layer of facts about the committee member as well as FOAF-extending rules. These facts and rules allow PAs to automatically respond to requests concerning the RuleML Symposia. Query responsibility for the organization is managed through a responsibility assignment matrix, which defines the classes of queries PAs of chairs are responsible for. External Agents (EAs) constitute the public interface to the OA of a Symposium's virtual organization, through which enquiry users can send queries and receive answers.

While the instantiations from 2007 to 2009 employed the same Rule Responder infrastructure, the 2010 implementation of SymposiumPlanner additionally incorporated the EMERALD⁶ framework, which was built on top of the JADE⁷

⁴ <http://ruleml.org/SymposiumPlanner/>

⁵ <http://www.foaf-project.org/>

⁶ <http://lps.csd.auth.gr/systems/emerald/>

⁷ <http://jade.tilab.com/>

multi-agent system. This project successfully bridged EMERALD and Rule Responder, and added new functionalities such as the mapping of paper topics to Symposium tracks.

For the two installments of the 5th International RuleML Symposium⁸, "RuleML 2011 - Europe" and "RuleML 2011 - America", our SymposiumPlanner-2011 employs two Sub-Organizational Agents (Sub-OAs) and an overarching Super-Organizational Agent (Super-OA). Along with the personal agents, these agents handle other issues associated with Super-OA-to-Sub-OA delegation, intelligent decision-tree-like Q&A, e.g. about where to submit theory papers, etc. We upgraded the communication middleware and rule agents of SymposiumPlanner to their latest versions and complement the user client to issue queries via a controlled natural language.

The rest of the paper is structured as follows: Section 2 introduces the SymposiumPlanner-2011 system and highlights its novel features. Section 3 describes the implementation of SymposiumPlanner-2011 in detail. Based on our experience and analysis, we close with some suggestions about SymposiumPlanner's future development.

2 SymposiumPlanner-2011 Use Case Description

The SymposiumPlanner-2011 agents support the organizing committee members in their organization tasks, such as: helping the program chair to monitor and possibly move important dates, finding contact information about selected chairs of the symposium, answering questions of participants about the conference, etc. The organization of the symposium committee is modeled by a coordinator, who is responsible for communicating with the executors which complete tasks on behalf of the symposium chairs. In SymposiumPlanner, the coordinator is implemented as an organizational agent (OA), which knows the responsibilities and the roles of each executor (Symposium chair). The OA manages the executors, which are implemented as personal agents (PAs).

The 5th International Symposium on Rules 2011 has two installments "RuleML 2011 - Europe" (IJCAI 2011) and "RuleML 2011 - America" (BRF 2011). To better manage disparate symposiums, SymposiumPlanner-2011 provides three Organizational Agents (OAs) to model the overall RuleML-2011 organization and its two sub-committees of the two Symposium installments:

- The Super-OA delivers and filters queries and requested tasks to the Sub-OAs, single point of entry for that specific instalment of RuleML 2011.
- The Sub-OAs manage the organization committee members of the two RuleML-2011 installments. They filter, decide and delegate incoming queries to responsible personal agents in the appropriate sub-organization.

An ontological RAM represents the roles and responsibilities of the Sub-OAs and the personal agents in the virtual organization of RuleML Symposium 2011.

⁸ <http://2011.ruleml.org>

Table 1 gives a RAM fragment of the committee chairs of RuleML Symposium 2011. Negotiation and distributed coordination protocols are applied to manage and communicate with the organizations' agents and external agents.

Table 1. Responsibility Assignment Matrix: Each committee chair has different roles, such as: responsible, supportive, consulted, informed, for a particular task.

	General Chair	Program Chair	Publicity Chair	...
Symposium	responsible	consulted	supportive	...
Website	accountable	responsible
Sponsoring	informed, signs	verifies	responsible	...
Submission	informed	responsible
...

The personal agents are self-autonomous agents and have their own rule-based decision and behavioral logic on top of their personal information sources, Web services, vocabularies/ontologies and knowledge structures. The rules are executed by different platform-specific rule engines which communicate via an Enterprise Service Bus (ESB) using standardized Reaction RuleML messages (event messages).

Besides consulting knowledge from the internal data repositories, which was very common in previous instantiations, the SymposiumPlanner-2011 rule agents access external data sources to reduce redundancy in the conference data via query languages such as SQL, SPARQL, etc. External data sources such as calendars, vocabulary definitions, databases, web pages, metadata sources, personal data are dynamically queried at runtime and used as facts in the internal knowledge base of an agent. For instance, in SymposiumPlanner-2011, the rule agents access data coming from the Semantic Web Dog Food RDF database, which contains information about the RuleML-2011 proceedings such as authors, papers, reviewers.

As a Web-based Q&A application, the previous instantiations of SymposiumPlanner allowed users to issue queries via selection of the queries in adaptable Web form menus. The potential queries needed to be defined before delivering the system to users. It becomes a real burden when there are many kinds of queries available. For the purpose of resolving this problem, SymposiumPlanner 2011 provides a translator service, which can automatically translate public interface descriptions of function rules to Reaction RuleML messages. Meanwhile, a controlled English has a restricted syntax and a restricted semantics described by a small set of construction and interpretation rules. It is not difficult for users to learn it. SymposiumPlanner 2011 also strives for allowing users to issue the queries in controlled English and then translate them into the standardized Reaction RuleML messages.

3 SymposiumPlanner-2011

SymposiumPlanner-2011 provides three Organizational Agents (OAs) to model the overall RuleML-2011 organization and its two sub-committees of the two Symposium installments. The general architecture of SymposiumPlanner 2011 is shown in Figure 1. In SymposiumPlanner-2011, we use the latest Prova 3.1.3 rule engine for the OAs, which is now an OSGi bundle and can run in an OSGi container with just a few dependencies. It packs a lot of features, some of them are inspired by the latest developments in modern computational logic, functional programming, distributed systems, and event driven architectures, some of them completely new and original. Some of very unique features include: reaction groups for event processing, guards and guarded cut, dynamic branches in workflows, and etc. Meanwhile, we utilize latest Mule ESB 3.1, which includes major improvements to Cloud Connect, including custom schemas for each connector, much simpler invocation of connectors, a new polling mechanism, message enrichment capabilities, and a simple yet powerful logging facility. In what follows, we will detail the implementation of the SymposiumPlanner-2011 system.

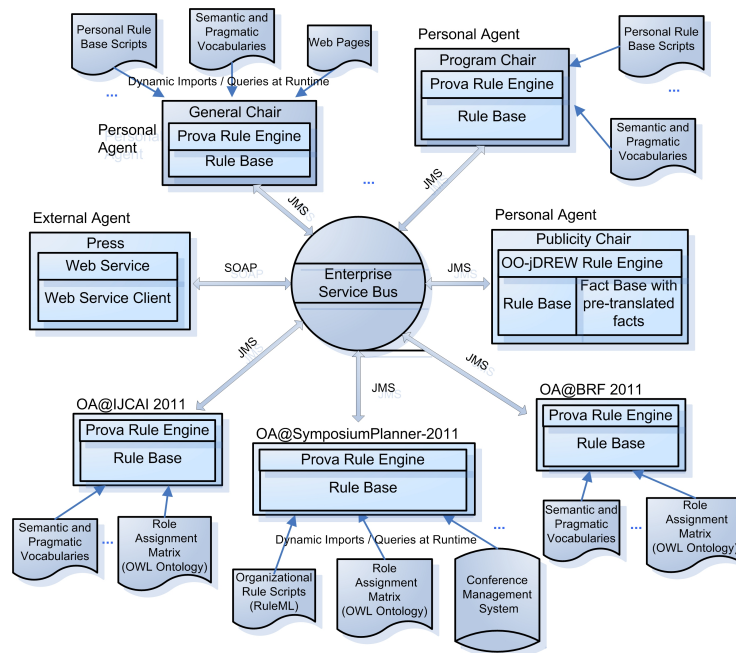


Fig. 1. SymposiumPlanner 2011 Architecture: Each rule engine is implemented as a Web-based service consisting of a set of internal or external knowledge sources. Reaction RuleML (interchange language) messages are transported via the ESB to the appropriate agent with different transport protocols.

3.1 Enterprise Service Bus

Mule open-source ESB⁹ allows deploying the 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. Mule provides a distributable object broker which follows the Staged Event Driven Architecture (SEDA)¹⁰ pattern to manage all sorts of service components. This design decouples event and thread scheduling from application logic and avoids the high overhead associated with thread-based concurrency models[1].

The distributed agent services which at their core run the rule engines are installed as Mule components which listen at configured endpoints, e.g., JMS message endpoints, HTTP ports, SOAP server/client addresses or JDBC database interfaces. Reaction RuleML¹¹ is used as a common platform independent rule interchange format between the agents (and possibly other rule execution / inference services). The translator services are used to translate inbound and outbound messages from platform-independent Reaction RuleML and/or controlled natural language into the platform-specific rule engines execution syntaxes and vice versa.

3.2 Prova Rule Engine

Generally speaking, each agent service might run one or more arbitrary rule engines. Here we only describe the rule engine Prova¹² which is used for implementing the complex logic of the organizational agents. 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 [6].

```
File Input / Output
... , fopen(File,Reader), ...
XML (DOM)
document(DomTree,DocumentReader) :- XML(DocumentReader),...
SQL
... ,sql_select(DB,cla,[pdb_id,"1alx"],[px,Domain]).
RDF
... ,rdf(http://..., "rdfs",Subject,"rdf_type","gene1_Gene"),...
XQuery
... , XQuery = 'for $name in StatisticsURL//Author[0]/@name/text()
return $name', xquery_select(XQuery,name(ExpertName)),...
SPARQL
... ,sparql_select(SparqlQuery,...
```

Prova has its separation of logic, data access, and computation and its tight integration of Java and Semantic Web technologies. Due to the natural integration of Prova with Java, it offers an economic and compact way of specifying

⁹ <http://www.mulesoft.org>

¹⁰ <http://www.eecs.harvard.edu/~mdw/proj/seda/>

¹¹ <http://reaction.ruleml.org>

¹² <http://prova.ws>

agents' behavior while allowing for efficient Java-based extensions to improve performance of critical operations. The main language constructs of messaging reaction rules in Prova are: `sendMsg`, `rcvMsg` and `rcvMult`. For instance, the following query retrieves the tracks of the symposium RuleML 2011@IJCAI.

```
getTracks(XID,Track):-
    % look-up responsible agent (Program Chair) from RAM
    assigned(XID,Agent,ruleml2011ATijcai_ProgramChair,ruleml2011ATijcai_responsible),
    % send the query to personal agent
    sendMsg(XID,esb,Agent, "query", getTrack(Track)),
    % receive answers multiple times
    rcvMult(XID,esb,Agent, "answer", substitutions(Track)).
```

3.3 Reaction RuleML Rule Interchange Format

Reaction RuleML acts as an interchange language between distributed rule agents in SymposiumPlanner. It is a general, practical, compact and user-friendly XML-serialized sub-language of RuleML for the family of reaction rules and incorporates various kinds of production, action, reaction, and knowledge representation temporal/event/action logic rules as well as (complex) event/action messages into the native RuleML syntax using a system of step-wise extensions [4]. For the communication between distributed rule-based (agent) systems Reaction RuleML provides a general message syntax as follows:

```
<Message directive="<!-- pragmatic context -->">
  <oid> <!-- conversation ID--> </oid>
  <protocol> <!-- transport protocol --> </protocol>
  <sender> <!-- sender agent/service --> </sender>
  <receiver> <!-- receiver agent/service --> </receiver>
  <content> <!-- message payload --> </content>
</Message>
```

Distributed rule agents can be programmed by the proprietary languages and engines. Reaction RuleML provides a translator service framework which translates the rule messages from and to Reaction RuleML, controlled natural language and platform-specific rule languages, such as Prova, Drools, POSL, EMERALD, etc. For example, the query in Attempto Controlled English (ACE) "What is the contact-information of the general-chair-of-RuleML-2011-IJCAI?" can be translated into a Reaction RuleML query message, in order to get the contact information of the RuleML 2011@IJCAI general chair.

```
<RuleML>
  <Message mode="outbound" directive="query-sync">
    <oid><Ind>SymposiumPlannerSystem</Ind></oid>
    <protocol><Ind>esb</Ind></protocol>
    <sender><Ind>User</Ind></sender>
    <content>
      <Atom>
        <Rel>getContact</Rel>
        <Ind>ruleml2011ATijcai_GeneralChair</Ind>
        <Var>Contact</Var>
      </Atom>
    </content>
  </Message>
</RuleML>
```

This Reaction RuleML message is then translated into a Prova message query which is executed in the engine:

```
[SymposiumPlannerSystem,esb,User,query-sync,[getContact,ruleml2011ATijcai_GeneralChair>Contact]].
```

3.4 SymposiumPlanner User Client

The SymposiumPlanner user client supports two solutions to construct queries. The straightforward solution uses an XML based file, which describes publicly accessible rule functions together with their mode and type declarations, to create HTML forms. After user initializes the parameters, the translator service combines the structure of function rules and the parameters values to create the standardized Reaction RuleML message.

SymposiumPlanner user client also allows issuing queries in a controlled natural language (Attempto Controlled English [2]) via a web browser. The ACE2RML translator forwards the text to the Attempto Parsing Engine (APE), which translates the text into a discourse representation structure (DRS) and/or advices to correct malformed input. The DRS gives a logical/structural representation of the text which will be fed to an XML parser and to be translated into a domain specific Reaction RuleML representation of the query. For example, the query follows ACE format: "Who are the authors of 'Rule-based Distributed and Agent Systems' "? is translated by the APE into the following DRS:

```
<DRS domain="">
  <Question>
    <DRS domain="A B C">
      <query obj="A" question="who" sentid="1" tokid="1"/>
      <relation obj1="C" rel="of" obj2="string('Rule-based Distributed and Agent Systems')"
        sentid="1" tokid="5"/>
      <object ref="C" noun="author" struct="countable" unit="na" numrel="geq" num="2"
        sentid="1" tokid="4"/>
      <predicate ref="B" verb="be" subj="A" obj="C" sentid="1" tokid="2"/>
    </DRS>
  </Question>
</DRS>
```

By parsing the DRS and applying the domain-specific rules which map named objects to constants and predicate relations to atomic predicates. The core element of the result is shown as follows:

```
<Atom>
  <Rel>getAuthorsOfPaper</Rel>
  <Ind>'Rule-based Distributed and Agent Systems'</Ind>
  <Var>author</Var>
</Atom>
```

4 Conclusion and Future Work

The communication middleware Mule uses SEDA, which decomposes the processes of Q&A in SymposiumPlanner with event-driven stages connected by explicit queues. These queues enable Mule to decouple the receiver of a message from the other steps in processing the message. That means the operations of SymposiumPlanner lie mostly on the SEDA processing mechanism and its performance. As a well developed framework, the performance of SEDA has been evaluated by many efforts. In [7, 3], the authors shown that SEDA maximizes throughput and exhibits higher performance and more robust behavior under load than traditional service designs. Accordingly, these efforts can reflect the

performance of SymposiumPlanner from the other side, that is, SymposiumPlanner can process users' queries reasonably and prevent resources from being overcommitted when demand exceeds agent capacity.

In future, while there is always a potential need to add more human support to the system in order to enhance flexibility, we will also aim at achieving more efficiency in terms of responses from agents as they automate redundant tasks that human users can avoid and be able to respond to increasingly complex queries. This may also lead to the need for peer to peer communication between PAs in order to help each other in answering queries posed by external agents. In such a scenario we can see PAs using FOAF-like profiles to advertise their capability of solving any query and cooperating with other PAs and the OA to achieve a collaborative environment of query response. In the case of a complex query, the query can be decomposed into parts, the OA can then delegate parts of the decomposed query to relevant PAs to solve individually. The OA can then eventually assemble the responses from PAs into a complete solution and send it back to the External Agent. In view of the aforementioned and proposed developments in our effort to build a framework to assist human organizations, we can see the SymposiumPlanner truly provides the basis for our gradual transformation of workplaces into an efficient and productive environment.

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>
3. Lucian, M.: Building a scalable enterprise applications using asynchronous io and seda model. Tech. rep., TheServerSide.com (2008), <http://www.theserverside.com/news/1363672/Building-a-Scalable-Enterprise-Applications-Using-Asynchronous-IO-and-SEDA-Model>
4. Paschke, A., Kozlenkov, A., Boley, H.: A homogenous reaction rules language for complex event processing. In: *International Workshop on Event Drive Architecture for Complex Event Process (2007)*, <http://ibis.in.tum.de/staff/paschke/>
5. Paschke, A., Boley, H., Kozlenkov, A., Craig, B.: Rule responder: Ruleml-based agents for distributed collaboration on the pragmatic web. In: *Proceedings of the 2nd international conference on Pragmatic web*. pp. 17–28. ICPW '07, ACM, New York, NY, USA (2007), <http://doi.acm.org/10.1145/1324237.1324240>
6. Paschke, A., Kozlenkov, A.: A rule-based middleware for business process execution. In: *Multikonferenz Wirtschaftsinformatik (2008)*
7. Welsh, M., Culler, D., Brewer, E.: Seda: an architecture for well-conditioned, scalable internet services. *SIGOPS Oper. Syst. Rev.* 35, 230–243 (October 2001), <http://doi.acm.org/10.1145/502059.502057>