

# Deploying a Distributed Symposium Planner Through Rule Responder

Benjamin Larry Craig (ben.craig AT unb DOT ca)

Faculty of Computer Science, University of New Brunswick  
Fredericton New Brunswick, Canada

**Abstract.** Rule Responder is an intelligent multi-agent infrastructure for collaborative teams and virtual communities. It supports rule-based collaboration between the distributed members of such a virtual organization. Members of a virtual organization are represented as semi-automated rule-based agents which utilize rules to describe the behavioral and decision logic. The system uses a RuleML subset as its Rule Markup Language, based on logic and XML. Rule Responder is implemented as a Web-based service architecture on top of the Mule Enterprise Service Bus. The paper describes a deployed question-answerer about the RuleML-2008 Symposium as an instantiation of a symposium planner based on Rule Responder.

## 1 Introduction

Rule Responder is a service-oriented architecture that can be used by virtual organizations for automated rule-based collaboration. Distributed users (humans or agents) can interact with Rule Responder for query answering or for negotiation and coordination. Rule Responder agents will process events, queries, and requests according to their rule-based decision and behavioral logic. It can also delegate subtasks to other agents, collect partial answers, and send the completed answer(s) back to the requester. The communication language used between the architectural components of Rule Responder (e.g., external, personal, and organizational agents) is Reaction RuleML.<sup>1</sup> A virtual organization contains, as members, a community of independent and often distributed suborganizations, teams, or individual agents. An example of a virtual organization is the organizing committee of the RuleML-2008 Symposium, which is assisted by our symposium planner, deployed on the meeting page<sup>2</sup> under “Rule Responder Q&A”.

## 2 Rule Responder as a Rule-Based Multi-Agent Infrastructure

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. The semi-autonomous PAs and OAs are implemented by (an instance of) a rule engine each, which acts as the inference and execution environment for the rule-based decision and behavioral logic of that semi-autonomous agent. The rule-based PAs represent, as their ‘dynamic profiles’, all of the participating human members of the virtual organization modeled by Rule Responder. An OA constitutes an intelligent filtering and dispatching system, using a rule engine execution environment for either blocking incoming messages, answer queries, or selectively delegating them to other agents. The communication middleware is an Enterprise Service

<sup>1</sup> <http://ibis.in.tum.de/research/ReactionRuleML/>

<sup>2</sup> <http://2008.ruleml.org/>

Bus (ESB) supporting various transmission protocols (e.g., JMS, HTTP, SOAP). The EAs can interact with the Rule Responder-enabled virtual organization via its public communication interface (e.g., an HTTP endpoint interface to an OA as the “single point of entry”). The current development API of Rule Responder uses a Web browser (Web form) for human-machine communication. The ESB implementation for Rule Responder is Mule, an efficient open source communication middleware.

### 3 Deploying the RuleML-2008 Symposium Planner

One group of use cases created to demonstrate Rule Responder is the organization of meetings such as the RuleML Symposium series, which is an example of a virtual organization that requires online collaboration within a team. Rule Responder started to support the organizing committee of the RuleML-2007 Symposium<sup>3</sup> and was further developed to assist the RuleML-2008 Symposium. The RuleML-2008<sup>4</sup> use case consists of fully functional knowledge bases for personal agents. This use case strives for embodying responsibility assignment, automating first-level contacts for information regarding the symposium, helping the publicity chair<sup>5</sup> with sponsoring correspondence, helping the panel chair<sup>6</sup> with managing panel participants, and the liaison chair<sup>7</sup> with coordinating organization partners. The planner could also aid with other issues associated with the organization of a meeting, including presentation scheduling, room allocation, and special event planning.

The RuleML-2008 use case utilizes a single organizational agent<sup>8</sup> to handle the filtering and delegation of incoming queries. Each committee chair has a personal agent that acts in a rule-governed manner on behalf of the committee member. Each agent manages personal information, such as a FOAF-like profile containing a layer of facts about the committee member as well as FOAF-extending rules. These rules allow the PA to automatically respond to requests concerning the RuleML-2008 Symposium. Task responsibility for the organization is currently managed through a responsibility matrix<sup>9</sup>, which defines the tasks committee members are responsible for. The matrix and the roles assigned within the virtual organization are defined by an OWL (Ontology Web Language) Lite Ontology. The Pellet reasoner is used to infer subclasses and properties from the ontology.

External agents and the RuleML-2008 agents can communicate by sending messages that transport queries, answers, or complete rule sets through the public interface of the OA (e.g., an EA can use an HTTP port to which `post` and `get` requests can be sent from a Web form). The standard protocol for intra-transport of Reaction RuleML messages between Rule Responder agents is JMS. HTTP SOAP is used for communication with external agents, such as Web services or HTTP clients.

<sup>3</sup> <http://ibis.in.tum.de/projects/paw/ruleml-2007/>

<sup>4</sup> <http://www.ruleml.org/RuleML-2008/RuleResponder/>

<sup>5</sup> <http://www.jdrew.org/ojdrew/ruleSetsRuleML-2008/publicityChairRuleML-2008.posl>

<sup>6</sup> <http://www.jdrew.org/ojdrew/ruleSetsRuleML-2008/panelChairRuleML-2008.posl>

<sup>7</sup> <http://www.jdrew.org/ojdrew/ruleSetsRuleML-2008/liaisonChairRuleML-2008.posl>

<sup>8</sup> <http://www.ruleml.org/RuleML-2008/RuleResponder/RuleML-2008-Responder.prova>

<sup>9</sup> <http://www.ruleml.org/RuleML-2008/RuleResponder/RuleML-2008.owl>

## 4 Interaction Between Organizational and Personal Agents

Organizational agents are used to describe the organization as a whole; for example, an OA contains a knowledge base that describes the organization's policies, regulations, and opportunities. This knowledge base contains condition/action/event rules as well as derivation rules. The OA of the RuleML-2008 Symposium can answer queries like this: "Who is the contact responsible for the symposium's panel discussion?" When a RuleML-formalized version of this query is received by the OA, this agent must first determine who the correct contact person is for the panel discussion. When the correct contact person for the panel discussion has been selected, the OA delegates the query to that committee member's personal agent. The PA will then respond with the member's name and contact method (e.g., email or telephone number, depending on contact preferences in their FOAF-like profile). Alternatively, if that contact person was on vacation or currently busy, then the PA would respond back to the OA that the contact person is unavailable. If the first-line contact person cannot be reached, then the OA will use the responsibility matrix (i.e., which committee members are responsible for certain tasks, and what members can fill their role if they are unavailable) to try to contact the next PA. This is one way that Rule Responder can act in an automatic process by chaining subqueries that find the best contact person at the time the original query is posed.

The personal agents used by Rule Responder contain FOAF-extending profiles for each person of the organizational team. Beyond FOAF-like facts, person-centric rules are used. All clauses (facts and rules) are serialized in Naf Hornlog RuleML, the RuleML sublanguage for Horn logic (allowing complex terms) enriched by Naf (Negation as failure). These FOAF-extending profiles have access to RDF (BibTeX, vCard, iCard, Dublin Core) and RDFS/OWL (role and responsibility models). The RuleML-2008 Symposium use case assists each organization committee member by an implemented personal agent. So the panel chair, general chair, publicity chair, etc. each have their own PA. Each PA contains a knowledge base that represents its chair's responsibilities to answer corresponding queries. For example, the query "What benefits would I receive for sponsoring the symposium with 500 dollars as opposed to 1000 dollars" will be delegated to the publicity chair's agent because it deals with sponsoring for the symposium.

## 5 Conclusion

Rule Responder has been used to implement a number of use cases, including the RuleML-2007/2008 symposium organization and the W3C Health and Life Science (HCLS). The middleware used by Rule Responder allows the simultaneous deployment of these use cases. The ESB provides the communication backbone to synchronously or asynchronously interchange messages between multiple agents. RuleML is a descriptive rule interchange language that so far was able to implement all logical structures that were necessary for Rule Responder. Rule Responder is an open source project, using the open source rule engines Prova<sup>10</sup> and OO jDREW<sup>11</sup>. The Rule Responder Technical Group<sup>12</sup> is open to extend the number of rule engines.

<sup>10</sup> <http://www.prova.ws/>

<sup>11</sup> <http://www.jdrew.org/ojdrew/>

<sup>12</sup> <http://responder.ruleml.org/>