1 Introduction

We have all heard about a “science fiction” vision comprising agents running around the Web performing complex actions for normal users. Although, many people think that such a dream is far from reality, yet James Hendler\(^1\) thinks that such a concept is quite achievable in the Semantic Web. The Semantic Web will not primarily consist of neat ontologies, rather it will be a complex web of ontologies with pointers to other ontologies. The present form of the Web which is mainly in the form of websites can be modeled based on the ontological information of the terms and concepts used in the Web. Service advertisements of the ontologies will be able to find matches for semantic links. James Hendler elucidates his concept of the future Web with an example of a ship stuck in a storm and how the captain manages to save the day using his web device to find relevant information regarding the storm systems using a web agent and is also able to call the coast guard to save his crew. The integration of agent technology and ontologies might significantly affect the use of Web services and the ability to extend programs to perform tasks for users more efficiently \[Hen01\].

In this paper, we shall attempt to use extensive literary survey to understand the extent to which agent oriented thinking and technology are helping realize the dream of agents and communities of agents interacting with one another to help human users as envisioned in \[Hen01\]. We shall see, how agents are interacting in different multi-agent systems? and how multi-agent systems are interacting with other multi-agent systems to form true virtual communities that are slowly becoming part of organizations to alleviate tasks for real human users.

2 Background

The key technology behind the Semantic Web is the Resource Description Framework (RDF), which can be seen both as a graphical and an object oriented knowledge representational model. Currently web is composed mainly of documents. There is a middle way between formalism of ontologies and information of documents. This is called Linked Data \(^2\). The Semantic Web in Linked Data format is a pragmatic way of representing knowledge on the Web because the object-oriented model is simple to understand. Whenever we talk about agents, we must

\(^1\) http://www.cs.rpi.edu/ hendler/
\(^2\) http://www.ibm.com/developerworks/web/library/wa-intelligentage/
bring the concept of an organization into our discussion. So, let us define an organization, “An organization provides a framework for activity and interaction through the definition of roles, behavioral expectations and authority relationships (e.g. control) [FGM03].” In the virtual world, “Virtual Organizations are a set of individuals and institutions that need to coordinate resources and services across institutional boundaries”. In [CAJB09] a modeling system called Virtual Organization Model (VOM) for virtual organizations is proposed. The four main dimensions of VOM are:

- The structural dimension that describes the elements of the system and how they are related.
- The functional dimension that details the specific functionality of the system, based on the services, tasks, and goals.
- The normative dimension that contains a set of mechanisms for ensuring social order and preventing self-interested behaviors.
- The environmental dimension, that focuses on describing the elements of the environment.

Furthermore, we also see ideas related to organizational communities for knowledge distribution in the shape of a Virtual knowledge Community (VKC). VKC is a platform for knowledge dissemination within groups of agents. Virtual Knowledge Communities are a well suited paradigm for decentralized knowledge exchanges and they have been applied in several domains [SM11]. An innovative approach of implementing the combination of knowledge and behavior through which agents can learn and adapt to new community related behavior without loss of expressivity is proposed in [SM11]. This approach uses Semantic Agent Model (SAM) architecture where knowledge is expressed using OWL and agent behavior defined using rules, SWRL in particular to maintain the knowledgebase as well as inferring new knowledge using a standard reasoner e.g. pellet.

Software personal assistants (SPA) are an active research area that one day might change the face of our human organizations. In the development of SPAs a lot of emphasis is being laid on the abilities of SPAs to overcome cognitive limitations of humans that are thought to limit the performance of organizations. In order to analyze performance of SPAs in organizations, a study was conducted on three basic types of organizational structures: regular hierarchies, rings and random structures [OSS09]. Regular hierarchies are hierarchical structures while rings and random structures are inherently horizontal structures. In the study carried out in [OSS09], the fraction of simulated runs in which the organization successfully completed the plan was used as an approximation of the success rate. For hierarchical organizations, SPA capabilities to manage cognitive and communication “overload” were most important, while for decentralized, flat organizations, SPA technologies that increased the “speed” of communication proved to be very helpful. The conclusion drawn from various experiments was that decentralized organizations perform best overall in the presence of SPAs [OSS09].

As an object oriented programming language, JAVA is the language of choice to establish communication among intelligent agents in the agents based soft-
Software engineering. [Bis08] reviews the Foundation of Intelligent Physical Agents (FIPA)-compliant infrastructure for building agent-based systems and suggests a multi-agent systems framework that merges this infrastructure with the J2EE technologies. Agent-based computing merges two technologies, namely Artificial Intelligence (AI) and object-oriented distributed computing. It is observed that most real-world applications are significantly more complex than before as they contain many dynamically interacting components, each with its own thread of control. Agent-oriented techniques provide a natural way for modeling complex systems, by decomposing its problem space into autonomous agents and their interactions. They enhance the reliability as well as end up reducing the cost and time-to-market of software applications by allowing their development through the assembly of a set of reusable software agents. The development of automated tools to automate the process of agent-oriented analysis and design remains a huge challenge. The merger of the FIPA framework with the J2EE technologies may open up new frontiers for agent architecture that could then be commercially exploited to build new platforms and applications.

Although multi-agent systems seem to provide a good basis for building complex software systems, yet there are certain drawbacks of classical “agent-centered” multi-agent systems (MAS). Recently a lot of interest has been shown in the use of organizational concepts within MAS where the concepts of ‘organizations’, ‘groups’, ‘communities’, ‘roles’, ‘functions’, etc play an important role [FGM03]. It is possible to design an MAS using only organizational concepts. The general principles of an organization centered multi-agent system (OCMAS) are that agents do not need to be coded rather specified with emphasis on expected behaviors. The organizational model also provides a way for partitioning the system, where each partition constitutes a context of interaction for agents [FGM03].

The interaction between the Semantic Web and multi-agent systems technology features groups of intercommunicating agents traversing it and performing complex tasks. A new generic agent model for KC-Agents (agents equipped with a Jess rule engine and a knowledge base (KB)) can be proposed with changes to Knowledge base which can result in altering agent knowledge and behaviour [KKB09]. These techniques form the basis for EMERALD framework, where agents can communicate without having a uniform rulebase. The agents are customizable and their behavior pattern can be altered by altering pattern parameters. Hence, the KC-Agents can be flexibly used in different environments. Despite laying down the basis for a robust MAS i.e. EMERALD, validating the proof whether any answer to a query is trustworthy is an important aspect as well. A defeasible proof validator can provide support in validation of explanations. EMERALD can be extended by adding deductive and defeasible reasoning services. EMERALD implements a number of reasoners offering reasoning services in two major reasoning formalisms:

- Deductive Reasoning: where conclusions are valid when premises of argument are true.
- Defeasible Reasoning: where efficient reasoning is carried out even with incomplete and inconsistent information [KPAB11].

3 Discussion and Analysis

Having laid down the foundation of agents and communities where these agents interact, we can discuss the approaches that were presented earlier in detail.

3.1 Organization Design

In order to map real world organizations into the virtual world of computers, we have to visualize virtual organizations. Although we have discussed a virtual organization model (VOM) [SM11] that can make the task of defining an organization much easier, any such idea needs to have some implementation details in context as well. When we use the term implementation, we are not just talking about frameworks but rules as well which are an important part of the multi-agent systems like JADE and Rule Responder.

In a distributed software architecture paradigm, different components of a system can share information across multiple components and communicate with one another to solve a problem but sharing information or interaction has to be predefined which makes it a rigid distributed architecture environment. Open Agent Architecture (OAA), is a framework for building flexible, dynamic communities of distributed software agents. Human users and software agents, in an OAA, express their requests in terms of what are the things that need to be done. “How it will be done?” is a question that need not be answered, which is not that simple an assumption as we are limited by the tools that we use to implement systems and these tools play an important part in our decision making process regarding the system architecture.

3.2 Components of Semantic Web

A crucial component of the Semantic Web, is an ontology. Web services are a practical way of using the Web ontologies and will be a key enabler for Web agents, creating machine-readable ontologies used by the capable agents to find Web services and automate their use. A well known issue with the Web is that finding the many available Web services is a difficult task. By creating a controlled vocabulary for service advertisements, search engines can find these Web services quite easily. The integration of agent technology and ontologies might significantly affect the user of Web services and the ability to extend programs to perform tasks for users more efficiently and with less human intervention.

3.3 Agent Oriented Analysis and Design

Intelligent agents have various characteristics and capabilities. An Agent can think in a structured way, where agents start with an overall plan to solve the
problem, or in an unstructured way, where actors can start from anywhere and build up a solution eventually. In a scenario with actors and activities, actors have their roles and responsibilities. Actors interact with activities through the message content, speech acts, knowledge, and interaction protocols. The actor will play its role (collection of responsibilities to finish a task) based on its perception (mental states). Either actor or activities can be conceptualized first. This Actor oriented thinking is a precursor to the agent oriented modeling and designing. As an object oriented programming, JAVA is the language of choice to establish communication among intelligent agents in the agents based software engineering. This conceptualization and implementation approach could be a useful in designing a Multi Agent System (MAS) especially when it comes to building large agent based system.

3.4 Multi-Agent Frameworks

A careful analysis of websites reveals that though agent technology has found lot of use in academic world, the business world is still more concerned with presentation of data. Contrary to this thinking, agent technology holds a lot of promise for the business world by making products or services more findable by search engines. In order to avoid burdening agents with their own reasoning engines, there is an approach of utilizing third-party reasoning services. We will be discussing two Multi-Agent frameworks in particular, Emerald and Rule Responder.

EMERALD can effectively provide a platform for agent communications and solve problems without requiring a uniform rule paradigm. A variety of reasoners can be incorporated and reasoning does not depend on translation between rule formalism rather on exchanging the results of the reasoning process of the rule base over the input data. EMERALD offers deductive and defeasible reasoning: in deductive case conclusions are proved to be valid when rule conditions are true e.g., R-Reasoner AND Prova-Reasoner. Defeasible reasoners can be used efficiently in case of incomplete and inconsistent information e.g., DR-Reasoner and SPINdle reasoner. DR-Reasoner modification on R-Reasoner [KOBB11], deals with OO RuleML like syntax, taking address of defeasible logic rule base as input. The facts are stored in an RDF document and the address of these facts are mentioned in the rulebase. This allows both negation-as-failure and classical negation.

Rule Responder is a framework for specifying virtual organizations as semantic multi-agent systems that support collaborative teams. To achieve this rule responder has three classes of agents - an Organizational Agent (OA), Personal Agents (PAs) and an External Agent (EA). The virtual organization as a whole is represented by an OA which uses ontologies and rules to assign and delegate incoming tasks to responsible PAs. For this purpose the OA can use a responsibility assignment matrix represented as an ontology to find an appropriate PA in its organization. The core of a Rule Responder agent is a rule engine such as Prova, OOjDrew, DR Device etcetera which implements the decision and behavioral reaction logic of the agent rules. For communication in agent networks,
Rule Responder uses the Mule based enterprise service bus middle-ware which supports a multitude of synchronous and asynchronous transport protocols. External agents and the other agents can communicate by sending messages that transport queries, answers, or complete rule sets through the public interface of the OA. Reaction RuleML, the de facto standard for XML-serialized reaction rules is used as a platform independent rule interchange format for agent conversations.

Intelligence of agents is limited if there is no communication to share knowledge and experience. Hence, agent communication is pivotal to increasing the efficiency of the agent systems. The core setback in agent interoperation is the variety in representation and reasoning as there is no globally agreed knowledge representation and reasoning formalism. Each agent in a Multi Agent System has its policy, requirements, duties and restrictions thereby making it difficult to communicate and transfer knowledge from another agent in a different MAS. We can make two different MAS, EMERALD [KKB10] and Rule Responder (RR)³, interoperable. The EMERALD Rule Responder (EMERALD-RR) Gateway uses a new java based EMERALD proxy agent called Rule Responder Proxy (RRP) that is directly connected to RR Organizational Agent. This RRP can receive any RuleML query, connect to RR, forward the query to the RR agent and receive the results. The RR-EMERALD Gateway, is based on a new Computational Agent (CA), a low-level RR agent capable of performing automated tasks.

³http://ruleml.org/RuleResponde
This interoperability between different frameworks can lay down the foundation for incorporating various services.

In view of above discussion, we felt the need to implement a reference architecture as shown in Figure 1 for symposium planning based on Rule Responder to support RuleML symposia Q&A websites, the SymposiumPlanner is an instantiation of the Rule Responder framework. 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 multiple conferences i.e IJCAI and BRF.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>RR</th>
<th>EM</th>
<th>Reference RR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separate rule and ontology languages, or combine them in a hybrid or homogeneous manner.</td>
<td>✓</td>
<td>✓</td>
<td>✓ Ontologies + Rules</td>
</tr>
<tr>
<td>Distinguish a performative (pragmatic) layer from a content (semantic) layer.</td>
<td>✓</td>
<td>✓</td>
<td>✓ RIPA Reaction/RuleML</td>
</tr>
<tr>
<td>Organise agent communication in a hierarchically (client-server) or networked (peer-to-peer) or reconfigurable (dynamic) fashion.</td>
<td>✓</td>
<td>✓</td>
<td>✓ Hierarchical architecture</td>
</tr>
<tr>
<td>Can be built on a multi-agent platform such as Jade or be based on an Enterprise Service Bus such as Mule.</td>
<td>✓</td>
<td>✓</td>
<td>✓ MULE ESB</td>
</tr>
<tr>
<td>Can be used to define autonomous agents or semi-autonomous agents.</td>
<td>✓</td>
<td>✓</td>
<td>✓ Semi-autonomous Agents</td>
</tr>
<tr>
<td>Can operate on a private intranet, a cross-organisational extranet or the open Internet.</td>
<td>✓</td>
<td>✓</td>
<td>✓ Intranet + Open Internet</td>
</tr>
<tr>
<td>Can realise virtual organisations in which agents solve various kinds of tasks, including document retrieval, question answering, and information integration.</td>
<td>✓</td>
<td>✓</td>
<td>✓ Agents (OA,PA,TA,CA)</td>
</tr>
</tbody>
</table>

Fig. 2. Dimensions of a Semantic Agent System

Unlike the 2011 version of SymposiumPlanner that decomposed the Organizational Agent into Super and Sub OA configuration, we intend to unify the business logic of multiple OAs into a single OA while maintaining the ‘corporate’ know-how of previous symposiums. This architectural change will not only lessen the flow points of information in the framework which can result in better performance but also make the maintainence easier. The new architecture aims to satisfy the seven dimensions of Semantic Agent Systems like Rule Responder (RR) and EMERALD (EM) as shown in Figure 2. The SymposiumPlanner and many other applications like it, can provide the basis for gradual transformation of our workplace into an efficient and productive environment.

References


[BP07] Harold Boley and Adrian Paschke. Expert querying and redirection with rule responder. In Anna V. Zhdanova, Lyndon J. B. Nixon, Malgorzata Mo-


