THE APPLICATION OF SOFTWARE AGENT TECHNOLOGY TO PROJECT MANAGEMENT INFRASTRUCTURE

Amin Andjomshoa
Mohammad Hassan Shafazand

Shahid Bahonar University of Kerman, Iran

amin@isolab.at
shafazand@ipsi.fhg.de

Alexander Schatten
A Min Tjoa

Institute for Software Technology and Interactive Systems
Vienna University of Technology
Favoritenstr. 9-11/188
1040 Vienna, Austria
{schatten, tjoa}@ifs.tuwien.ac.at

KEYWORDS
Intelligent Software Agent, Project Management, Distributed Systems, Ressource Management, User Management, Webservices, Metadata

ABSTRACT

Agent technology promises to increase the flexibility and power of distributed management systems and services. This paper describes the suggested work that is aiming to understand the practical implications and benefits of applying agent technology to the management of distributed project and resources using Simple Object Access Protocol (SOAP) as the underlying messaging service provider and RDF (Resource Description Framework) as metadata structure. The OSWP (Open Science Workplace) is a tool planned to manage distributed resources and projects on the web. Agents seem to play a central role in enhancing flexibility, user friendliness and productivity both for the end users and top managers. Agents can support the finding of resources and provide solutions for managing tasks.

1. INTRODUCTION

1.1.Open Science Workplace

The Open Science Workplace was initiated as a joint project between the Iranian Technical Cooperation Office (TCO) of the Presidency and the Vienna University of Technology. The main goal of OSWP is to provide an integrated infrastructure for managing projects and tasks using distributed members and resources (Schatten et.al., 2001). To understand the usage of agents in OSWP we briefly describe the scenario of managing a sample project.

A project is composed of a hierarchical tree of tasks. Each task is a piece of work, which is assigned to be performed by a project member within a specified period of time. When a member logs into the system he obtains a view of his tasks with the reported progress . He can also view the related resources and documents attached to the task or project by other members. The project manager can also monitor the current status of the project and its subtasks. Each task will be evaluated by the task owner to meet the definition goals. There is also an estimated progress that compares the current progress with the time estimation, which is calculated by OSWP using task's dates. Moreover OSWP provides some other utilities such as group mailing, to do lists, news, resource management, chat, notes to provide a computer aided environment to mange and accomplish tasks and projects.

As complex applications like project, ressource management and communication support tools like OSWP store an amount of different types of data as as described above--structured as well as semi- and unstructured data--one could imagine this situation as beeing confronted with an information-biotop rather than an information mono-culture. Worse, multiple OSWP servers could be installed at different locations with different focal points and the need will arise, to solve problems needing ressources from more than one server...

1.2. Software Agent Technolog?

Although there is no common agreement for the term *agent*, a popular definition, from Wooldridge and Jennings (1995), describes an agent as a software entity that has the characteristics of autonomy (acts independently), pro-activity (goal-based), reactivity (responds in a timely fashion to events) and social ability (communicates with other agents to achieve goals collaboratively). Other characteristics frequently quoted include mobility (the ability to move from one host to another) and learning (the ability to improve performance overtime based on previous experiences).

Problems described in this article are difficult to address with only conventional “seach” strategies, as it is desired to perform more complex logic during the interactions e.g. when trying to find fitting persons to open tasks. As agents work autonomously, but are able to communicate, problem solving between agents is more powerful as negotiation can be seen as a back and forth process between multiple agents, not only as a "one-way" search process. Furthermore a broad scope of different tasks will require interaction between servers, hence a more general approach is intended.

Of course it may be possible to achieve cost saving and performance boosting solutions without agents, but agent technology provides a more natural model of the real world (i.e. a community of entities each with their own goals, communicating and often working together to achieve mutual benefit) compared to existing software paradigms, such as object-orientation.

In a nutshell, agent technology can be understood as a "negotiation" platform, that allows, as soon as implemented, integration of different application-and-search logic, eventually also with agent code moving from server to server when needed. This is possible as in some agent implementations the "moving agent" is in fact code, that "travels" from one server to another. So if a special retrieval strategy (logic) is needed, a distinctive agent can be designed at one place, that implements the necessary functionality and once put into the agent environment it will send copies to the other server in the (OSWP) network.

1.3. Simple Object Access Protocol (SOAP)

SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined data types, and a convention for representing remote procedure calls and responses. SOAP can be used by definition in combination with a variety of other protocols like HTTP and SMTP. (W3C, 2000).

SOAP does not itself define any application semantics such as a programming model or implementation specific semantics; rather it defines a simple mechanism for expressing application semantics by providing a modular packaging model and encoding mechanisms for encoding data within modules. This allows SOAP to be used in a large variety of systems ranging from messaging systems to RPC. A major design goal for SOAP is simplicity and extensibility.

1.4. Ressource Description Framework (RDF)

The Resource Description Framework (RDF) is an infrastructure that enables the encoding, exchange and reuse of structured metadata. RDF is an application of XML that imposes needed structural constraints to provide unambiguous methods of expressing semantics. RDF additionally provides a means for publishing both human-readable and machine-process able vocabularies designed to encourage the reuse and extension of metadata semantics among disparate information communities. The structural constraints RDF imposes to support the consistent encoding and exchange of standardized metadata provides for the interchangeability of separate packages of metadata defined by different resource description communities.

RDF supports the use of conventions that will facilitate modular interoperability among separate metadata element sets. These conventions include standard mechanisms for representing semantics that are grounded in a simple, yet powerful, data model.

RDF additionally provides a means for publishing both human-readable and machine-process able vocabularies. Vocabularies are the set of properties, or metadata elements, defined by resource description communities. The ability to standardize the declaration of vocabularies is anticipated to encourage the reuse and extension of semantics among disparate information communities. (RDF Link)

Since XML has a limited capability to describe the relationships with respect to objects, the use of XML to provide metadata is not enough. In our suggested work we will try to use RDF as the metadata language.

2. AGENTS IN PROJECT MANAGEMENT – A CASE STUDY

2.1. Prerequisites

Achieving the goals of a typical project requires that a number of people collaborate with one another in order to share their knowledge, resources and capabilities. Managing this collaborative environment is a complex task, which depends on many parameters. The project management can be viewed from two aspects:

1. Project view including resources, goals and stages
2. Project management view including controlling techniques and responsibility accounting.

In both views we should handle some static elements like resources, stages, documents, products and some dynamic elements like the transition between stages and organizational rules to mange changes and updates.

It goes without saying that the most important role in process of a successful project is the role that a project manager plays. The project manager is the expert who should have enough knowledge in defining stages, assigning tasks and evaluating the artifacts created by developers. Any mistake in these steps may lead to unexpected results and waste of time and resources. Hence support of this function has the main focus in this agent application.

2.2. The Application of Agents

Task Assignment
In theOSWP project management system, each project is broken to some subtasks which are assigned to OSWP members. Assigning a task to a member requires enough information about the members' skills and capabilities. This information is extracted from data source of member skills. Still there are some other parameters which effect the assignment of task to a member. For example:

- The number of current assigned tasks and projects
- The experience of member on similar tasks
- The reputation of member in the previous assigned tasks.

As we discussed in the previous section, the project manager plays a centric role in assigning tasks to appropriate members, and the correct assignments can guarantee the success of the whole project.

Agent supporting systems can help the project manager to find appropriate human resources. Since these resources could be geographically distributed, the agents should negotiate with each other to share their knowledge (beliefs) and create a priority-sorted list of human resources. These communicative agents will use their experiences to candidate the most appropriate members. Hence a kind of Intelligence should be applied to their goal to make them capable of making decisions.

Planning subtasks of a project
Another important issue is sub tasking the project. There are so many discussions on methods of sub tasking, length of a task and number of sub tasks, but there is no standard set of rules that can be applied to all projects. However there are some templates that can be applied to standardize the process of project management. For example RUP (Rational Unified Process) gives a set of artifacts, roles, guidelines and rules which can be applied to a project. The project manager may break the project to stages that RUP suggests.

To satisfy the standards of RUP, the manager should define tasks, attach guidelines and checklists to them and clarify the artifacts. As you see these tasks are somehow time consuming and boring. The situation gets even worse, when the manager does not know about this methodology. At this point an agent may help him to apply a standard template or a template of some successful project to this new project. The agent uses his own knowledge to retrieve the appropriate template or calls the other agents for template of similar projects.

3. IMPLEMENTATION

The core goal of our implementation is to make the distributed resources available to remote users. So first of all we should find a structured web metadata, which uniformly describes our resources on distributed OSWP sites. As we discussed earlier, RDF will be used as metadata language for annotating existing OSWP resources with additional machine-readable information.

In our suggested work we will apply SOAP serialization to RDF data model. In a WWW9 presentation Nielsen demonstrated how RDF can be encoded by SOAP Serialization syntax (Nielsen, 2000). Each resource in OSWP should be mapped into an RDF abstract model. For example, consider the following statement:

VISION.DOC resource with ID 1234 is an "RUP Vision Document" entitled "Airport vision document", is created by John Smith (smith@oswp.org) on July 20, 2002 and it describes the project requirement and user needs for "Airport project".

An example of what a resource might look like in RDF is shown in Figure 1.

```
<rdf:Description
about="http://www.oswp.org/resources/VISION.DOC"
rdf:ID="1234">
   <res:creator>
      <rdf:Description>
         <mem:name>John Smith</mem:name>
         <mem:email>smith@oswp.org</mem:email>
      </rdf:Description>
   </res:creator>
   <res:descriptions>Project  requirement  and  user
needs</res:descriptions>
   <res:title>Airport Vision Document</res:title>
   <res:date>July 20, 2002</res:date>
   <res:type>Vision Document</res:type>
</rdf:Description>
```

Figure 1: RDF Serialisation

This RDF schema can become more detailed by adding related projects or tasks, additional members or resource information. With this graph we have a semantic view of resources which are logically interconnected to members and projects. This might be the basis for a decentralized, distributed resource manager.

The RDF-XML shown in Figure 1, will be embedded in the soap message to move between OSWP servers. The client application then receives not only the required resources but also the net of semantic relations between this resource and other OSWP objects. As we discussed in the SOAP section, SOAP lacks the ability of communication at business and legal requests, instead it does the communications at programming language APIs' level. In the presented solution RDF will complete the SOAP by communicating OSWP at business and semantic level.

To implement the agents we will use the SOAP-Agents which expose their services as web services. A Web service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via internet-based protocols (W3C, 2002).

In each resource center an agent has the responsibility of knowledge mining and filtering the data. This resource center only needs a semi structured resource of skills and projects which in OSWP project is already available (Schatten et.al., 2001). By these assumptions the scenario will be as follows:

The local agent queries his own knowledge to response the local lookups. If the request is to be queried in remote OSWP sites, then the local agent send this request to the registered coworkers (other OSWP sites) to receive the required information. The agents in this platform should be able to analyze and wrap the requests as a SOAP message. Then SOAP web service responsible for this query is called by local agent. On the other side SOAP server receives the request and activate an agent to handle the request and create the result RDF serialization. These results will be embedded in SOAP response and sent to the calling agent.

Since the ontology of the local and remote agents may have been customized for the members, agents should be able to transform the conversations to and from their local ontology. For example a German agent may sent a request containing the word "Projekt", then the receiving English agent should translate this word to "project" which is known in it's ontology. This translation can be easily applied with an XSL transformation of requests or responses.

4. CONCLUSION AND VISIONS

Providing the metadata about the available resources improves the way we work with the resources. Fortunately such a metadata standard exists and RDF is a W3C proposed standard for defining the architecture necessary for supporting web metadata. The SOAP protocol provides the transfer layer over the network and programming language APIs. It makes our framework more abstract an independent of different platforms. The suggested work suggests new framework for agent communications. It uses both RDF and SOAP, which will fit into each other and handle the agent communications.

The broader vision is, to get a grip on the immanent complexity of the described server spanned tasks not by designing complex applications that query multiple servers, but writing rather simple agents that "create" this complex behaviour by interaction among themselves, with the servers and the users.

Yet more important: the rise of webservices (SOAP, XML-RPC, ...) and metadata initiatives like the described RDF but also UDDI/WSDL gives promise to a re-vitalisation of the multi-agent buzzword-scenario in the middle of the 1990th (Nwana and Ndumu, 1999), which seemed to fail on a broader scale--among others by the problem to easily connect to various different information pools. Web-services combined with metadata initiatives and web-ontologies like the semantic web (Berners-Lee et.al., 2001) could leverage this access and OSWP can draw an initial scenario for such an integration effort.

The next step off course dealing with the (more general) problem of interoperation, going beyond the concrete agent problem is to try to embed the OSWP XML interfaces into semantic web structures as soon as they evolve to a steady state. Agents in *this* framework are planned to be used for automated search of information and resources and support for complex manual activities. The future work includes the detail investigation into agent's anatomy and scalability of these collaborative agents.

REFERENCES

Tim Berners-Lee, James Hendler and Ora Lassila, 2001, “The Semantic Web”, *Scientific American*, May

Faramarz Farhoodi and Peter Fingar, 1997, “Distributed Object Computing, ‘DOC’” *Magazine The Endorsed Publication of the Object Management Group* part 1: October

Henric Frystyk Nielsen, 2000, “Soap, RDF and the Semantic Web.”, *WWW9*

H. S. Nwana & D. T. Ndumu, 1999 “A Perspective on Software Agents Research”, *The Knowledge Engineering Review*, Vol 14, No 2, pp 1-18

Alexander Schatten, A Min Tjoa, M. Hassan Shafazand, Amin Andjomshoa, 2001, “Building an Web-Based Open Source Tool to Enhance Project Management, Monitoring and collaboration in Scientific Projects”, *iiWAS Conference*

Wooldridge, M., Jennings, N. R., 1995, "Intelligent Agents: Theory and Practice", *Knowledge Engineering Review*, Vol. 10, No. 2, pp115-152

“Simple Object Access Protocol (SOAP) 1.1”, 2000, *World Wide Web Consortium, Note,* 8 May www.w3.org/TR/SOAP

“Web Services Architecture Requirements”, 2002, *World Wide Web Consortium,* http://www.w3.org/TR/2002/WD-wsa-reqs-20020429