Personal Agents on the Semantic Web *

Anugeetha Kunjithapatham, Mithun Sheshagiri, Tim Finin, Anupam Joshi, Yun Peng
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore MD 21250 USA
{anu1,mits1,finin,joshi,ypeng}@cs.umbc.edu

1 Introduction

The Semantic Web is a vision to simplify and improve knowledge reuse and dissemination on the world wide web. Efforts are underway to define the format and meaning of the language of such a Semantic Web that could serve both humans and computers. The EU-NSF strategic workshop report on the semantic web identifies 'the applications for the masses such as intelligent personal assistants' as one of the key applications enabled by the semantic web. Personal assistants gather and filter relevant information and compose it into a coherent picture with regard to the user's preferences. An intrinsic and important pre-requisite for a personal assistant or rather any agent is to manipulate information available on the Semantic Web in the form of ontologies, axioms, and rules written in various semantic markup languages. ;the means of information gathering being centralized (event notification services) or de-centralized (peer agents).In this paper, a model architecture for such a personal assistant, that deals with real-world semantic markup is described.

2 Personal Agents (PA) and the Semantic Web

As the amount of information grows on the web, the average user is overwhelmed by the cognitive load involved in making decisions and making choices. Our endeavour involves delegating some of the tasks to the PA thereby helping the user make better use of his time. We demonstrate this concept using a talk notification service with a human and an agent interface;an illustration of this concept is provided in the figure below. The tasks performed by our PA involve information filtering and filtering through peer collaboration. The PA has a model of the user's preferences expressed in DAML+OIL[daml.org, 2001]. The use of DAML helps the PA leverage semantic inferencing. The PA interacts with the user using MS Outlook Calendar and schedules talks based on criteria like user's interest in the talk, user's availability, and recommendation from peers. The PA makes use of a host of third party services that aid the agent in its decision making. These third parties are wrappers that convert unstructured information on web pages (like mapquest) to structured facts in the agent's KB. We feel that in the near future more and more services will be offered that cater to machines rather than humans. We use DAML+OIL as the inter-lingua for communications among PAs and between agents and service providers. We have used the JADE agent framework to build our agents. Please refer to the extended version of this paper[1] for a more detailed description.

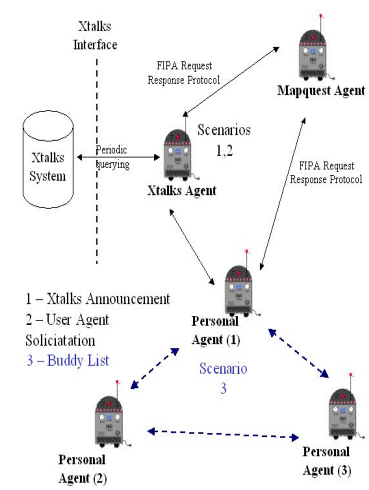

Figure 1: **Multi-Agent Scenario and Interactions**

3 Reasoning in PAs

We use the Java Expert System Shell (JESS) for storing knowledge and inferencing. The PAs beliefs of the world are stored as VSO triple-based facts in JESS. DAMLJessKB[Kopena and Regli, 2003] provides a set of axioms that are used for reasoning over RDF[Lassila and Swick, 1999] and DAML+OIL[w3.org, 2001]. As new facts are entered into the KB, rules corresponding to these axioms fire and new facts are asserted into the KB. This is the mechanism for inferencing in our PAs. Apart from the DAML axioms, user's preferences are also expressed as rules in JESS. Certain rules also fire off events that are captured and an appropriate indication is made to the user. For example, when all conditions for scheduling a talk are met and the corresponding rule

*This research was supported in part by DARPA contract F30602-97-1-0215.

[1] http://users.ebiquity.org/docrepos/2003/paper/PersonalAgents-ISWC03.pdf

fires, this event is captured and the Bridge2Java API is used to schedule the talk in the user's Outlook Calendar.

4 Interaction with Peer PAs

The PA also has an implicit module that enables it to interact and collaborate with peer agents. The PA, on receiving the talk notification, consults a list of PAs that are regarded as buddies. A FIPA-ACL[FIPA, 2000] message is sent to peer-PAs and as per the query-ref FIPA interaction protocol, the PA receiving the message is obliged to send back a reply. The content of these messages are DAML+OL assertions. The peer PAs are determined through a buddy-list that the user maintains. We also have developed a discovery mechanism for discovering buddies. Our mechanism makes use of a popular search engine to locate the homepage of the owner of the peer agent. Inspite of the vast size of the web, it is easy for a search engine engineered to index billions of pages to locate the homepage of a person with reasonable web presence. For the sake of simplicity, we refer to the person initiating the discovery as the user and the person being located as the owner. A HTML META tag in the homepage points to the owner's profile in DAML. The profile among other things includes the location (ip:port) of owner's agent. A FIPA *subscription request* is sent to the owner's agent. The owner's agent on receiving the request sends its owner an email. This mail is in the form of a HTML with embedded scripts. The e-mail contains user's details and hyperlinks to capture the owner's decision. In response to the owner's response a corresponding FIPA *inform* message is sent back to the user's agent which might/might not update the buddy list based on the owner's decision.

An agent can also pose queries to peer agents. We have developed a querying mechanism that uses a combination of DAML Query Language(DQL)[DQL, 2002], JESS *defqueries* and FIPA agent communication protocols. DQL enables us to describe queries in DAML, and FIPA protocols provide the transport mechanism for the queries by defining the interaction between the agents involved. The query is framed as a set of PSO triples with unbound variables and sent to one of the buddy agents as a FIPA *query-ref* message. The receiving agent on receiving the query, converts the triples into a JESS *defquery* and fires it in its KB. The triples acquired by firing the query are packed into multiple FIPA *inform-result* and sent back to the querying agent. For more information on DQL and our extensions to it, please refer[Sheshagiri and Kunjithapatham, 2003].

5 Trust and Privacy in PAs

The PA possesses a wide range of knowledge including some persistent data and dynamic data acquired through notification services, interaction with peers and reasoning performed at various stages. Such information could be of great use to the peer PAs and hence it would be worthwhile to share it with interested parties. While interacting with a peer for sharing information, the PA will have to determine if the requested information exchange can be shared. It may be impossible for the PA to come up with a decision emulating it's user's choice of action in such situations; but, we believe that a simple and straight-forward mechanism to determine the credibility of the requesting party and the nature of information requested would help the PA to take a user desirable decision. We describe below a mechanism that we propose:

In our model architecture, the data in the PA's Knowledge base is categorized as *sharable, non-sharable* or *sharable with the user's consent*. Personal information, past appointments and class schedule information of the user are classified as *sharable* facts. *Non-sharable facts* consists of confidential information. Facts such as the current location, future appointments etc. are categorized as facts *sharable with user's consent*. The PA on receiving a query responds based on the type of information requested. If the information is categorized as *sharable with user's consent*, the PA sends a mail to its user about the request and replies according to the user's response.

Additional rules based on the user's relationship with the requestor and the requester's role are also defined. To enable the PA to identify the appropriate rules to execute, we have come up with a set of rules to identify the order of thier execution. Some of the implemented rules based on the relationship with the requestor are as follows: (1)*If the requestor is a friend and not a family member - share only information marked as sharable*. (2)*If not a friend/family member but Advisor- share SSN, schedule information*. (3)*Peer agents belonging to family members have access to all information* A *cache* component has been designed to keep track of rejected queries, and to allow the PA to determine the urgency of the query, possibly based on the number of times the PA got the same query.

6 Conclusion

We have demonstrated a Personal Agent application that leverages the capabilities of semantic web languages and agent technology to perform some of the user's tasks. Automation achieved through applications like this can help the user manage his/her time more efficiently.

References

[daml.org, 2001] DAML+OIL Specification daml.org. http://www.daml.org/2001/03/daml+oil, 2001.

[DQL, 2002] DQL. Daml query language http://www.daml.org/dql, 2002.

[FIPA, 2000] FIPA. http://www.fipa.org/, 2000.

[Kopena and Regli, 2003] Joe Kopena and William Regli. DAMLJessKB: A tool for reasoning with the semantic web. *IEEE Intelligent Systems*, 18(3):74–77, 2003.

[Lassila and Swick, 1999] Ora Lassila and Ralph Swick. Resource description framework model and syntax specification http://www.w3.org/rdf, 1999.

[Sheshagiri and Kunjithapatham, 2003] Mithun Sheshagiri and Anugeetha Kunjithapatham. A fipa compliant query mechanism using daml query language (dql) http://www.cs.umbc.edu/ finin//papers/dqlfipa.html, 2003.

[w3.org, 2001] DAML +OIL Reference Description w3.org. http://www.w3.org/tr/daml+oil+reference, 2001.