Ontology Mapper: A Multi-Agent System for Knowledge Sharing

Suryakant Sansare  
University of Maryland Baltimore County  
Department of Computer Science  
ssansa1@cs.umbc.edu

Urvi Shah  
University of Maryland Baltimore County  
Department of Computer Science  
ushah1@cs.umbc.edu

Abstract
Ontologies are built to provide semantics to real world terms. This obviously creates multiple representation of the same term by different societies. Such diversified views necessitate the need for some kind of mediator, which facilitates sharing of knowledge that can be used for effective communication of the meaning. Ontology Mapper is a MAS that provides mapping service to department agents to facilitate cross-domain topic ontology understanding. Using agent communication mechanisms the system allows understanding of ontologies and terms across departments and consequently to the users.

Introduction
Ontology is the buzzword in the Artificial intelligence mainstream. Ontology is the semantic representation of concepts in a machine understandable form. Due to lack of common understanding of even the most well known concepts e.g.; topics in the branch of Science, ontology for these concepts differ. A mapper service provides a method to resolve these differences to a reasonable point. Our mapper is designed to handle the concept of “TOPIC”, the definition of which, if provided here will be debatable.

Multi-agent systems add another dimension to agent-oriented systems. Formalism for a multi-agent system must deal with the multiplicity of agents; group properties of agent systems, such as common knowledge and joint intention; interaction among agents, such as communication and cooperation.

We try to explore the use of agents in a particular domain viz. Universities. Universities have a natural interaction with each other and share lot of information amongst themselves. Since it’s common for universities to have their own definition or belief in topic hierarchies there is a need for a tool that provides meaning of the terms as can be understood by the individual university or any department. We can visualize the mapper as being a service provider agent, limiting our domain to Academic environment. The agents representing each department hold, amongst other information, the ontology for “Topics” for their branch of science. The department agents request relation of a topic between any ontology, which the mapper provides with a certain degree of certainty. We have implemented the system for a single university domain.

Background
Ontology
Ontology[1] is an explicit specification of a conceptualization. The term is borrowed from philosophy where “ontology” is a systematic account of existence. To support the sharing and reuse of formally represented knowledge among AI systems, it is useful to define the common vocabulary in which shared knowledge is represented. A specification of a representational vocabulary for a shared domain of discourse -- definitions of classes, relations, functions, and other objects -- is called ontology. Pragmatically, a common ontology defines the vocabulary with which queries and assertions are exchanged among agents. There are various tools for building an ontology e.g. Protégé, Ontolingua etc.

Mapping
Imagine that you have an inferencing engine that already has information about project managers and so on in one company, but then is given information about a different corporate hierarchy, which could be subtly or dramatically different, in terms of the names of its parts and the way they are defined and relate together. In this case reasoning about data with respect to the new way of looking at the world within the old framework would require a ‘cross-walk’ mapping items and connections in one ontology to
the other. Similarly, if you want to combine ontologies to talk about different aspects of objects, then you need to describe how the ontologies relate to each other. Although this can be a very difficult and time-consuming problem, there is definitely a need for this service.

Agents
Agent based technology has generated lots of excitement in recent years because of its promise as a new paradigm for conceptualizing, designing and implementing software systems. The next generation of agent technology will perform information gathering in context. This requires that agents be able to interoperate and coordinate with each other in peer-to-peer interactions. The main characteristics of MAS are 1) each agent has incomplete information for solving the problem, and hence limited viewpoint. 2) there is no global control 3) data are decentralized. [2] The agent technology has thus been the center of focus for some time due to the similarities between the characteristics of the agent and nature of real world problems.

Architecture
The system consists of a set of department agents in the university environment. Each agent holds an ontology that represents its department. In addition the agents provide the service of ontology mapping. The agents register their mapping service with the agent platform. The users query their departmental agent requesting the mapping of a term, defined with a set of document/documents onto another known ontology or across all departmental ontologies in the agent’s knowledgebase. The user may also request the mapping of its ontology onto another ontology. The agent searches the DF for the agent possessing the appropriate ontology and communicates the mapping request. The service provider agent replies with the result of the request, which is returned to the user in the desired form. The interaction between department agents is done using Peer-to-Peer messages or broadcast messages. The conversation state is maintained by the means of conversation-id slot in ACL messages.

Implementation
JADE[3] was used as the platform for building agents. JADE is a rich object oriented technology implemented in Java. It simplifies implementation of Multi-agent systems through a middleware, which is FIPA complaint. The communication architecture in JADE offers flexible and efficient messaging by creating and managing a queue of incoming ACL messages private to each agent. We use the GUI provided by the platform for monitoring and controlling the status of agents. The agents perform the following functionalities:
- registration of its service with the DF
- maintain the knowledge about its ontology
- search within the DF for agent/agents having the required ontology.
- provides user interface for querying the agent.
- provides mapping service
- receives requests from other agents and sends back the results as a response.

The agents are GuiAgents that send messages when an event occurs and exhibit CyclicBehaviour for receiving messages from other agents. The core of the mapper service is the “Rainbow Classifier” with a perl script interface. Rainbow[4] is a program that performs statistical text classification using the Bow library, written in C, using one of the different methods including naïve bayes, TFIDF, knn, maximum entropy and SVM. We run “Rainbow” on a pre-defined port at each host. The perl script provides data to rainbow for classification and retrieves the results. This script is invoked from within the agent on receipt of a REQUEST message.

The ontology held by each agent is written in DAML[5] (DARPA Agent Markup Language), which is a markup language designed to have a greater capacity that XML for describing objects and relationships between objects. DAML creates a higher level of interoperability among sites and expresses semantics to the objects which facilitates the use of software programs to find or interpret it.

Mapping Scenario
A user in the Computer Science (CS) department can give three kinds of mapping requests.

Case 1: He desires to know the mapping of the term “Genetic Algorithms” to the ontology in Bio-Informatics (BI) department. The user requests its agent to carry out this task. The agent finds the Bio-Informatics agent by querying platform’s Directory Facilitator (DF).
**Case 2:** He desires to map his ontology with the ontology in the BI department. The user requests its agent to carry out this task. The agent finds the Bio-Informatics agent by querying platform’s Directory Facilitator (DF).

**Case 3:** He desires to know the meaning of the term “Genetic Algorithms”. The user gives the request to the agent who searches the DF for all registered agents providing the mapping service.

The CS agent sends a REQUEST message to that agent with the term/ontology’s supporting documents. The receiving agent accepts this message and in turn forwards the message to the mapper program and awaits results. This result is sent back as an INFORM message to the CS agent. The user gets the result in his preferred form.

---

**Related Work**

There has been ongoing work on using personal ontologies to browse the Web. This was accomplished by mapping from reference ontology to a personal ontology. Then, each user was allowed to browse a particular site with their personal ontology for Personalize Information Filtering on the WWW. [6] Microsoft BizTalk Mapper is a translation tool that is used to map records and fields between two different specification formats.

The CAIMAN[7] system facilitates the exchange of relevant documents between geographically dispersed people in Communities of Interest. The ontology mapping in CAIMAN is based on a novel approach, which considers the concepts in an ontology implicitly represented by the documents assigned to each concept. Using machine-learning techniques for text classification, a concept in a personal ontology is mapped to a concept in community ontology.
There is very little research about the application of ontology mappings for community support. In (Takeda, Matsuzuka, & Taniguchi 2000)[10], a system has been presented that performs collaborative document recommendations by finding related folders in users bookmarks. The links from related folders of a peer user are recommended to each user. The quality of the recommended links seemed to be insufficient while the quality of the mapping of folders has been judged satisfactory. This may be due to the fact that the mapping of the folders is based on a naive keyword comparison. Moreover, we consider the direct exchange of documents between users problematic, as perspectives on a knowledge domain may vary too much to find suitable mappings. This is not as much the case for a mapping between the community and user perspective. The relatively stable community perspective makes it more feasible to find a suitable mapping over time. In (Takeda, Matsuzuka, & Taniguchi 2000), the structure of the bookmark hierarchies is not considered at all.

In (Mitra, Wiederhold, & Kersten 2000)[11], a graph-based model for expressing ontology interdependencies is presented. Algebra for set operations with ontologies is constructed. The inter dependencies of two source ontologies are expressed with a third ontology, the articulation ontology. The articulation ontology consists of concepts, which subsume concepts from both source ontologies. The two source ontologies are connected to the articulation ontology by links, which are termed semantic implication. The mapping is performed manually by the user on the explicit representation of the ontology. The ONION project in the Stanford DB group seed matching rules from the user to generate new matching rules. In our implementation the agents presents the results to the user and it is up to the users discretion to accept the matching. None of these systems make use of the agent technology to support the systems. Agent technology aids in automating the service and better user compatibility.

Future Work

This agent framework could be integrated with the ITTalks project at UMBC. Presently the user does not have the option to provide his interest using any other Computer Topics ontology but ACM, which is the default. Using our framework the user can specify his interest and/or the related field, which gives more flexibility to the user.

In the present implementation we assume a single university domain with one ontology per department. The department agent provides only the service of ontology mapping. This framework can be expanded to span across various universities and also business organizations where there is a need for exchanging and understanding various terminologies. The agents can cache results and inferences can be drawn on future queries.

Conclusion

This framework will aid better understanding of terminology of any topic across domains. This gives the user more space to express himself and assures that he is reasonably understood. The Mapping service facilitates sharing of views whilst following ones own belief. This can be achieved since the mapping service aids in expressing our beliefs in a way that can be comprehended by individuals that may not have the same belief.

References

5. DARPA Agent Markup Language site: www.daml.org


12. Mark d’Inverno, Michael Fisher, Alessio Lomuscio, Michael Luck, Maarten de Rijke, Mark Ryan and Michael Wooldridge: Formalisms for Multi-Agent Systems