ITtalks: A Case Study in the Semantic Web and DAML+OIL

R. Scott Cost, Tim Finin, Anupam Joshi, Yun Peng, Charles Nicholas, Ian Soboroff, Harry Chen, Lalana Kagal, Filip Perich, Youyong Zou, and Sovrin Tolia,
University of Maryland, Baltimore County

Semantic Web markup languages will improve the automated gathering and processing of information and help integrate multiagent systems with the existing information infrastructure. In this article, the authors describe their ITtalks system and discuss how Semantic Web concepts and DAML+OIL extend its ability to provide an intelligent online service.

Semantic Web developers seek to make the Web more machine-readable, to let intelligent agents retrieve and manipulate pertinent information. Achieving this will require seamless agent integration with the Web and taking full advantage of the existing infrastructure (such as message sending, security, authentication, directory services, and application service frameworks). We believe that the Semantic Web markup language DAML+OIL (DARPA Agent Markup Language plus Ontology Inference Layer) will be central to this goal's realization. The DAML Program aims to develop a Semantic Web markup language that provides sufficient rules for ontology development and that supports intelligent agents and other applications.[1,2] (For more information, see the "DAML+OIL" sidebar.)

As part of the University of Maryland, Baltimore County's role in the DAML Program, we constructed a fielded application, ITtalks, which facilitates user and agent interaction for locating talks on information technology. ITtalks also lets developers design and create intelligent software agents that can understand and process information encoded in DAML+OIL or other semantically rich markup languages. Primarily, we focused on developing the support and infrastructure for integrating intelligent agents into the Web.

ITtalks features

ITtalks is a Web portal that offers access to information about talks, seminars, and colloquia related to information technology. ITtalks users can view such information as location, speaker, hosting organization, and talk topic. ITtalks also lets agents retrieve and manipulate information stored in the ITtalks knowledge base.

ITtalks uses DAML+OIL for knowledge base representation, reasoning, and agent communication (see Figure 1). We used DAML+OIL to mark up stored information to provide additional reasoning capabilities. Additionally, ITtalks uses DAML+OIL for all communication, including simple messages and queries, using the ITtalks-defined ontology.

Personalized accounts

You can access ITtalks anonymously or through a personalized account. You register by entering information manually through Web forms or by providing the location (URL) of a DAML+OIL personal profile, which could include your location, interests, contact details, and schedule. Your schedule could be rudimentary (a list of available time periods for given days) or significantly more detailed. ITtalks uses this information to personalize the site, displaying only talks that match your interests, schedule, or both.

Because DAML+OIL is not yet in widespread use, ITtalks provides a tool for creating a DAML+OIL personal profile. The tool constructs a profile containing only items that the ITtalks system uses. However, we believe that the profile will ultimately provide a unique and universal point for obtaining personal user information for all information needs, not just for ITtalks, and could then include any information you wish to share.

Customized domains

To help create a universal resource for the international IT research community, we organized ITtalks around domains, which typically represent event-hosting organizations such as universities, research laboratories, or professional groups. A separate Web site exists per domain, each independently maintained by a moderator who can define the domain's scope

1094-7167/02/$17.00 © 2002 IEEE

DAML+OIL

DAML+OIL (DARPA Agent Markup Language plus Ontology Inference Layer) is a semantic language being developed by a consortium of US-based academic and business researchers. The DAML program (www.daml.org), launched in August 2000, seeks to address the Web's limitations in providing machine-readable and, more importantly, machine-interpretable information. DAML+OIL aims to transform the currently human-oriented Web, which is largely used as a text and multimedia repository, into a Semantic Web as envisioned by Tim Berners-Lee (see Figure A).[1] This process augments Web pages with additional information to facilitate machine understanding.[2]

DAML+OIL is built on the capabilities of XML (Extensible Markup Language) and RDF(S) (Resource Description Framework and Resource Description Framework Schema). These are XML applications that help provide preliminary facilities for the Semantic Web.

The World Wide Web Consortium (W3C) developed XML as a standard for alternative data encoding on the Internet, primarily for machine processing. XML is an application profile of the well-defined and well-understood syntactic language SGML (Standard Generalized Markup Language). The XML standard lets you declare and use simple data structures, which are stored in machine-readable XML documents. You could then process or translate the information in these documents into additional XML documents that permit human understanding, such as text-to-voice, graphics, or HTML conversion. However, XML is defined only at the syntactic level, so you cannot rely on machines to unambiguously determine the correct meaning of the tags. Consequently, XML is not suitable for representing complex knowledge.

The W3C Consortium developed RDF(S) to address XML deficiencies by adding formal semantics on top of XML. These two standards provide representation frameworks for describing relationships (named properties and values) among resources that resemble representation frameworks of semantic networks and rudimentary frame languages (such as RDF Schema). Yet, both standards are still restrictive as knowledge representation languages because they lack support for variables, general quantification, rules, and so on.

DAML+OIL tries to build on XML and RDF(S) to produce a language that is well suited for building the Semantic Web. It follows the same path for representing data and information in a document as XML and provides rules and definitions similar to RDF(S). Additionally, DAML+OIL also provides rules for describing further constraints and relationships among resources, including cardinality; domain and range restrictions; and union, disjunction, inverse, and transitive rules. Thus, DAML+OIL endeavors to develop a universal Semantic Web markup language that can enable machines to read data and interpret and draw inferences from it.

References

1. T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, vol. 284, no. 5, May 2001, pp. 34–43.
2. J. Hendler and D. McGuinness, "The DARPA Agent Markup Language," *IEEE Intelligent Systems*, vol. 15, no. 6, Nov./Dec. 2000, pp. 72–73.

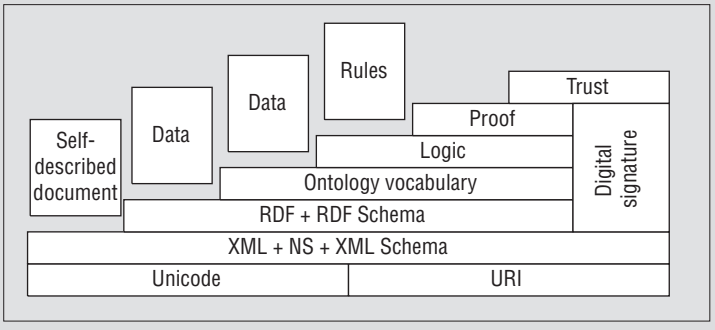

Figure A. Tim Berners-Lee's vision of the Semantic Web is founded on a base that includes URIs (uniform resource identifiers), XML (Extensible Markup Language), and RDF (Resource Description Framework).

and let other registered users edit talk entries. For example, the http://umbc.ittalks.org moderator might configure the domain to include only talks hosted at the University of Maryland, Baltimore County (UMBC).

Multiple access modes

You can access the ITtalks system through the Web or through agents acting on your behalf. The Web portal offers features including registration, search, entry, and domain administration.

To increase interoperability, you can browse ITtalks information (see Figure 2) in DAML+OIL, standard HTML with DAML+OIL annotation, or Wireless Markup Language (WML). The ITtalks Web portal can also generate RDF Site Summary (RSS) files in response to certain queries. You can use these RSS files for various external purposes, such as displaying upcoming talks on a departmental Web site.

To provide access for agent-based services, ITtalks uses Jackal, a communication infrastructure for Java-based agents developed by our research group at UMBC.[3] Jackal facilitates the use of the Knowledge Query and Manipulation Language (KQML) for agent communication.[4] We're adapting Jackal to the Foundation for Intelligent Physical Agents (FIPA) standards.[5,6] Our research group, along with other universities, is also developing a DAML+OIL ontology for the necessary conversation protocols.

Interchangeable ontologies

We based ITtalks on a set of ontologies (http://daml.umbc.edu/ontologies), which describes talks and the things associated with them such as people, places, topics and interests, and schedules. Figure 3 shows some of the dependencies among these ontologies. We use the ontologies to represent and process DAML+OIL descriptions and also as a conceptual schemata against which we built the database and various software APIs.

We developed a general ontology for describing talk and paper topics. This ontology contains information required for describing talks and the relationships among them, such as "topic," "supertopic," and "related-to." Using this, we implemented an ontology to describe IT-related talks based on the ACM computer classification system (CCS). We are also developing a DAML+OIL ontology for IT talks based on a portion of the Open Directory and are considering additional classification ontologies. Figure 4 sketches some of the major classes and properties.

We use these topic ontologies to describe

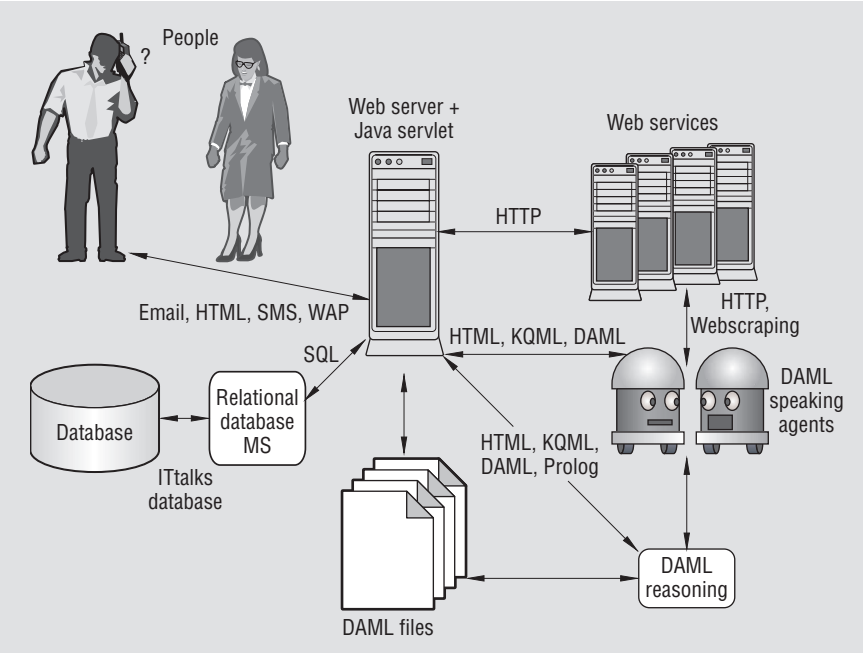

Figure 1. The architecture for ITtalks is built around a Web server backed by a relational database. ITtalks has interfaces for human users, software agents, and Web services.

```
<rdf:RDF>
 <rdf:Description about="&itlink;742">
  <Talk rdf:parseType="Resource"
   <Title>Invisible Web</Title>
   <Abstract>
    Accessing information anytime, anywhere has...
   </Abstract>
   <BeginTime>
    <time:Year>2002</time:Year>
    <time:Month>01</time:Month>
    <time:Day>17</time:Day>
    <time:Hour>12</time:Hour>
    <time:Minute>00</time:Minute>
    <time:Second>00</time:Second>
   </BeginTime>
   <EndTime>
    <time:Year>2002</time:Year>
    <time:Month>01</time:Month>
    <time:Day>17</time:Day>
    <time:Hour>13</time:Hour>
    <time:Minute>00</time:Minute>
    <time:Second>00</time:Second>
   </EndTime>
   <Location rdf:parseType="Resource">
    <Institution>UMBC</Institution>
    <Building>Math-Physics</Building>
    <Room>401</Room>
    <Street1>1000 Hilltop Circle</Street1>
    <City>Baltimore</City>
    <State>MD</State>
    <Zip>21250</Zip>
    <Country>USA</Country>
   </Location>
  …
   </Talk>
 </rdf:Description>
</rdf:RDF>
```

Figure 2. An example of DAML+OIL-encoded knowledge.

talks and the users' interests throughout the system. The DAML+OIL ontologies let users add assertions in DAML+OIL to further characterize their interests. The topic ontologies support an automated talk classification, for which we obtained an ACM CCS training collection and are generating an Open Directory training collection to develop the necessary components. We are also developing a semiautomated component that can map topics in one ontology to topics in another, by using user-specific mapping *believes* and by exploiting the fact that each ontology's nodes have an associated text collection.

Data entry support

Although we have simplified data entry—by supporting automatic form completion using information from the knowledge base and the user's DAML+OIL profile—it still needs improvement. Therefore, we, along with the Lockheed Martin research group, are developing a focused Web spider to collect talk announcements from open sources on the Web. This spider will identify and automatically add key information to the ITtalks knowledge base using Lockheed Martin's AeroText information extraction system.

Architecture

ITtalks uses a relational database combined with a Web server to provide user access to the system. ITtalks also has an interface for agent-based communication.

Database

The ITtalks system uses the MySQL relational database software. We store the contents of the ITtalks knowledge base in a database whose schema is closely mapped to our ontologies, describing events, people, topics, and locations. We chose MySQL because of its known reliability and because we needed software with a license that lets us offer the ITtalks package to academic and commercial institutions. We are considering replacing MySQL with a native XML database software such as dbXML.

Web server

We used a combination of Apache and Tomcat as the Web portal servers. This lets us present IT talk descriptions to the user through Java servlets and JavaServer Pages files, which dynamically generate requested information in DAML+OIL, XML, HTML, RSS, and WML. ITtalks can also deliver information viewable on either a standard, computer-based phone or a WAP-enabled cellular phone.

A typical scenario

To better portray user interaction with ITtalks, let's look at a simple, typical interaction. In this scenario, Jim learns from his colleagues about the ITtalks Web portal as a source of IT-related events in his area; Jim is affiliated with Stanford University.

Jim directs his browser to the www.ittalks.org main page. Seeing a link to http://umbc.ittalks.org, he selects it and is presented with a new page listing upcoming talks scheduled at UMBC, Johns Hopkins University, and other locations within a 15-mile radius (the default distance for the UMBC domain).

Jim browses the Web site, viewing announcements matching his interests and preferred locations (as provided in his explicit search queries). He requests to view the talk information in WML. Finding a talk of potential interest to a colleague, Jim takes advantage of the invitation feature, which lets him send an invitational email to any of his friends for any

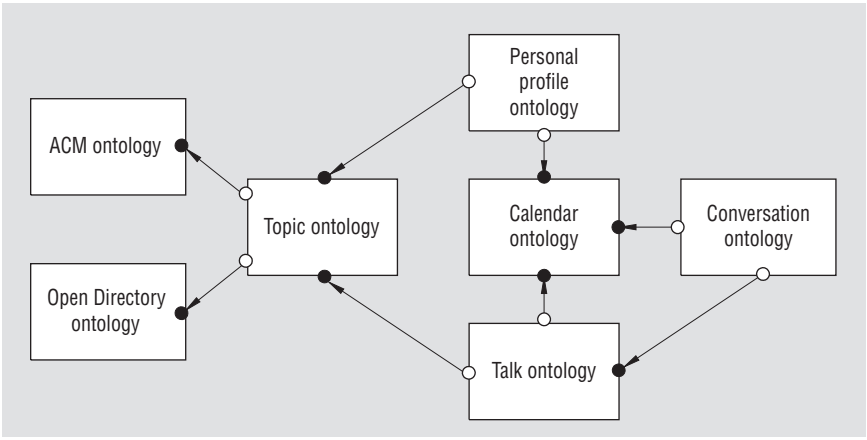

Figure 3. The relationships among the various ontologies that ITtalks uses.

of the listed talks. Finally, using the Personalize link on the bottom of the page, Jim creates his own http://ittalks.org main page, by providing the URL of his DAML+OIL-encoded profile. This customized page, listing talks based on his preferences, will be Jim's entrance to the ITtalks site whenever he returns.

Agents

We'd like ITtalks to demonstrate, among other things, how DAML+OIL facilitates integration of service agents with online information resources. To this end, and to extend ITtalks' capabilities, we defined several agents that support ITtalks' operation. You can view some as supporting services (such as external information services); we assume that others will exist in the general environment in the future.

ITtalks

The ITtalks agent is a front end for the ITtalks system. The agent interacts with ITtalks through the Web-based interface used by humans but communicates via an ACL (Agent Communication Language) with other agents on the Web. The agent acts primarily as a gateway for agent access and does not support any advanced functionality.

User

Agent research has long aimed to represent human users online through agents that can service queries and filter information for them. Although ITtalks does not require such agents to exist, we recognize the power that such agents add. Therefore, ITtalks supports interaction with user agents as well as their human counterparts. The user agent that we developed understands DAML+OIL, supports sophisticated reasoning, and communicates via a standard ACL. It reasons by means of XSB, a logic programming and deductive database system for Unix and Windows developed at the State University of New York, Stony Brook.

Calendar

Although a user agent might contain the necessary knowledge about its user's schedule, we believe that it will benefit from assigning the calendar-based facts and preferences to a separate calendar agent. This lets the user agent use the same protocol to consult the user calendar and other groups' or users' calendar agents. The calendar agent can only represent abstraction to an already existing infrastructure such as Microsoft Outlook or other desktop–server applications. The calendar agent can also represent a room and thus permit reuse of the same principles of participation and event scheduling.

Classifier

ITtalks uses a classifier, or recommender, agent that is invoked when a user enters a new talk. On the basis of the talk's abstract, the classifier returns ACM CCS numbers along with a rank, in descending order. Using a local table of classification numbers and names, ITtalks will suggest to the user 10 possible topics.

MapQuest

The MapQuest agent is a wrapper agent that lets ITtalks use external services. It interacts directly with agents (for example, the ITtalks and user agents) and accepts requests for information such as the distance between two known locations. It then phrases an appropriate request to the MapQuest system, parses the results, and generates a response. We could also generically name this agent a distance agent and use any external service (or combination of several, as needed).

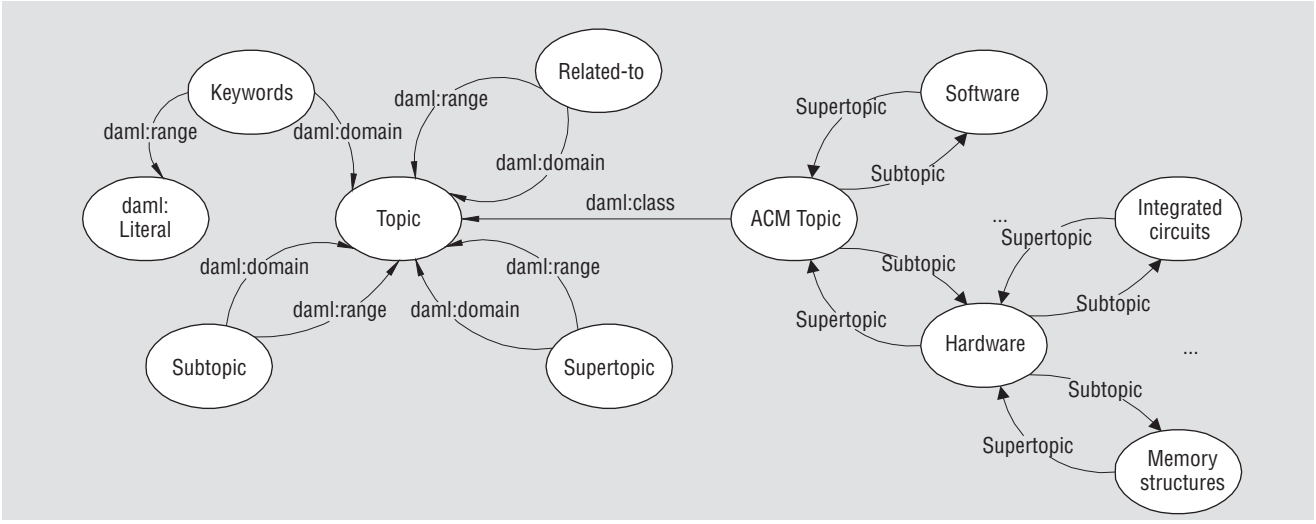

Figure 4. The ontologies that ITtalks uses are relatively simple, such as the topics ontology it uses to describe talk topics and user interests.

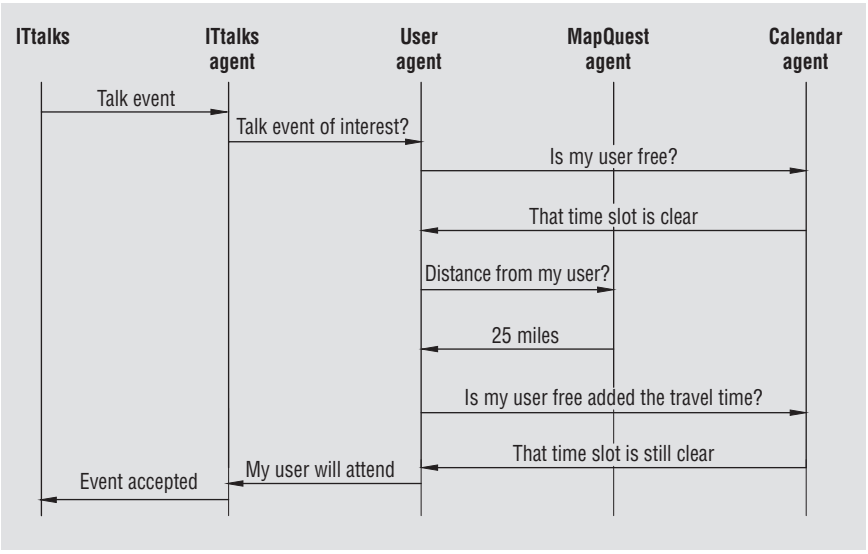

Figure 5. Agent interactions in the ITtalks agent scenario.

Scenario: Advanced features

We continue our scenario with user Jim to demonstrate more advanced interactions that ITtalks offers. Jim has registered with ITtalks and left instructions for the system to notify him when certain types of talks are scheduled.

ITtalks discovers an upcoming talk that might interest Jim. Based on his preferences, ITtalks opts to notify Jim's user agent directly via an ITtalks agent, which forwards the message using an ACL. Jim's user agent then consults Jim's calendar agent to determine his availability and the MapQuest agent for the distance from Jim's predicted location at the time of the talk (see Figure 5).

Some more sophisticated interactions, which we plan to implement, could potentially take place at this time. For example, the calendar and user agents might decide to alter Jim's schedule and contact a colleague's user agent. Or, the user agent could request more information about the speaker and event by contacting other agents or Web sites, such as a CiteSeer-based agent.[7] Finally, after making a decision, the user agent could notify the ITtalks agent, indicating that Jim will or will not attend. The ITtalks agent would then make the appropriate adjustments to the ITtalks database.

Future implementations might also allow for even more complex interactions. Say that a research group employs Jim but has limited funding. The group lets only one researcher at a time attend a particular IT event. Consequently, the user agent cannot decide on Jim's participation until it interacts with other agents representing Jim's employer and colleagues. In this case, a decision and election of a group representative requires an interaction involving an agent virtual community.

The user agent could also benefit from participating in virtual communities thanks to recommendations it obtains from other user agents. One user agent might recommend an IT event given its owner's experiences from attending a past talk of the same speaker. Another user agent might decide to share comparisons of two competing times and locations for an identical IT event. Yet another user agent might simply share its owner's intension to attend a particular IT event. Thus, each virtual community member could profit from these and other recommendations and reflect these social filtering methods in its own decisions.

Finally, future implementations might factor in smart offices.[8,9] For example, the ITtalks agent could directly contact an agent representing an IT event location. This room agent could use varying service discovery techniques[10,11] to locate a projector in the room and instruct it to download a PowerPoint presentation before the speaker's arrival. Moreover, the room agent might also try to contact additional agents in the IT event's vicinity to decrease the possible noise level from other rooms and to verify that a hallway agent has requested enough refreshments during the event.

Benefits of DAML+OIL

ITtalks benefits significantly from its use of a semantic markup language. DAML+OIL specifies the ontologies that we use extensively in our system for personal profiles and as a content language for agent communication. Without DAML+OIL, specifying topic schedules, interests, and assertions would be very difficult. As an ACL, DAML+OIL offers more flexible semantics than KIF (Knowledge Interchange Format) or other content languages that only provide syntax. The greatest benefit DAML+OIL gives ITtalks is the ability to interact with any DAML+OIL-capable agent without human supervision. Consequently, all these benefits enable more efficient interaction between the system and its users—humans or software agents. (See the "Related Work" sidebar.)

Interoperability standard

Using DAML+OIL lets us easily share ITtalks content with other DAML+OIL-aware applications and agents. In the future, we could also offer a simple extension to let arbitrary agents register and interact with ITtalks for various purposes. For example, a Centaurus room manager agent[8] could watch ITtalks for events happening in a particular room to enable better scheduling. DAML+OIL also acts as an interoperability standard, by letting other sites publish announcements marked up in our ontology, making their talks available for inclusion in ITtalks.

Agent communication language

In the future Semantic Web, agents will create, access, modify, enrich, and manage DAML+OIL documents as a way to disseminate and share knowledge. Because DAML+OIL documents will be the objects of discourse, DAML+OIL and ACLs must be successfully integrated. Agents will need to exchange DAML+OIL documents and exchange *informational attitudes* about DAML+OIL documents. Using an ACL, agents can talk about DAML+OIL documents. Thus, we hope to integrate ACL work and concepts with a universe of DAML+OIL content.

Distributed trust and belief

Agents have difficulty knowing what information sources (for example, documents, Web pages, and agents) to believe or trust in an open, distributed, and dynamic world and how to integrate potentially contradictory information. We can use DAML+OIL to support *distributed trust and reputation management*,[12,13] by forming the basis of a logic for *distributed belief transfer* that will enable more sophisticated, semantically driven rule-based techniques for information integration and fusion. Distributed trust involves authentification and access control based on the conformance of a user's credentials to a predefined security policy. We are considering *distributed belief* as a way for agents to garner required information. For example, an agent would believe that the

Related Work

ITtalks must address two general problems: generating the Semantic Web markup for talk announcements and integrating and fusing information from different systems.

ITtalks uses text classification to automatically generate topic descriptors for markup, using the Carnegie Mellon University Rainbow program (www-2.cs.cmu.edu/~mccallum/bow). Kamal Nigam and his colleagues have also written a good reference for text classification.[1]

The Lockheed Martin AeroDAML project (http://ubot.lockheedmartin.com/ubot) offers a more powerful approach to automatically generating markup for natural language documents. The project tackles the markup generation problem by combining its proprietary natural language processor, AeroText, with the DAML+OIL language for knowledge representation. Consequently, software agents can use AeroDAML to automatically annotate existing Web pages to overcome the syntactic complexity and semantic ambiguity of otherwise inaccessible information.

ITtalks uses a multiagent systems approach (http://agents.umbc.edu) to integrate information from different sources. Specifically, it wraps each information source or destination with code to enable it to act as an agent. The resulting system of agents communicates with each other using the Foundation for Intelligent Physical Agents (http://fipa.org) agent communication language and associated standards.

The WebScripter (www.isi.edu/Webscripter) system, developed at the Univ. of Southern California Information Sciences Institute, offers another approach to information fusion on the Semantic Web. WebScripter collects DAML-encoded information from multiple, heterogeneous Web sources and combines them by extracting and fusing the encoded information into reports. WebScripter comprises two software components: the report-authoring environment for defining reports, and the report instantiation component that generates new reports from DAML instances and makes them available on the Web.

Reference

1. K. Nigam et al., "Text Classification from Labeled and Unlabeled Documents Using EM," *Machine Learning*, vol. 39, nos. 2–3, 2000, pp. 103–134.

http://w3c.org/xml page would always have current information about XML.

We use DAML+OIL's expressiveness by employing it to describe the security policies, credentials, and trust relationships that form the basis of trust management. Policies specified in this way contain semantic meaning, which facilitates their integration and improved conflict resolution. Also, other applications will be able to interpret the agent's credentials (such as authorization certificates) correctly, universalizing these credentials. Similarly, describing beliefs and associating levels of trust with them is straightforward, and deducing belief is uniform for different applications and services.

Data entry support

ITtalks supports intelligent form filling, making it easier for users to enter and edit information in their profiles and talk announcements. Additionally, it provides automatic form filling when an editor tries to enter information about an entity that already exists in the knowledge base.

Entering talks

To make ITtalks successful, we must simplify the entry of new talk descriptions into the system. We address this problem using three complementary approaches: an enhanced Web interface, acceptance of marked-up announcements, and automated text extraction. DAML+OIL plays a key role in the first two and is the target representation for the third.

Enhancing the Web interface

Although our implementation of automatic form filling for entities already entered in the knowledge base does not directly use DAML+OIL, it can support a more generalized version of a form-filling assistant. This depends on

- Tagging Web form widgets with DAML+OIL descriptions of what they represent
- Capturing dependencies among data items in DAML+OIL
- Compiling these dependencies into an appropriate execution form (for example, JavaScript procedures) that can drive the Web form interface

Additionally, we plan to investigate multimodal support, where users can enter new information through keyboard or vocal input. When filling in a form, the user would speak to enter data in each field. On receipt of the voice-filled form, ITtalks would try to infer the recorded sound's meaning, obtain additional information based on the knowledge and rules stored in the ITtalks system, and present the user with a completed form for verification. This enhancement would let ITtalks provide voice-entry support for devices with limited keyboard functionality, such as PDAs or cellular phones.

Text classification

For ITtalks to filter talk announcements on topic matches, we must know the appropriate topics for each talk. Initially, we required users to manually select appropriate topic categories from the ACM CCS hierarchy. However, the user faces a daunting task of navigating a hierarchy of nearly 300 topics, many with unfamiliar meanings. Trying to select their own topics to characterize their interests can be equally daunting. Ultimately, we would like to use more than one topic hierarchy to classify both talk topics and user interests (for example, ACM CCS and Open Directory nodes), which further complicates the problem.

To address this, we built an automatic text classifier to suggest appropriate terms in a hierarchy for classifying a talk based on its title and abstract. We used the classifier package from the Bag Of Words toolkit by Andrew McCallum at Carnegie Mellon University. This library provides support for a wide variety of text classification and retrieval algorithms. We used the Naive Bayes algorithm, which is widely used in the classification literature and fairly effective, and was quick to learn the 285 classes in our test collection. We plan to use the same classification agent to suggest interest terms for users based on text found by searching their Web pages.

Additionally, we are also developing a tool for mapping multiple ontologies. Such a tool will, for example, let each user select a preferred topic ontology on the fly, and the ITtalks system will immediately adapt and present the personalized filtering results. As a prototype of the mapping tool, we chose a semiautomatic approach that lets users manually select relations (*landmarks*) among specific topics across the ontologies—for example, broader, narrower, and similar. Subsequently, the tool automatically computes the remaining mappings via the user-specified relations and our automatic text classifier, using training sets of documents for the ontologies.

The automated mapper then performs two operations. First, it accepts marked-up announcements. An easy way to enter new talk announcements is to supply an already marked-up document. The ITtalks interface lets you enter a URL for a talk announcement marked up in ontologies that ITtalks understands. Currently, these only include the native ontologies we built for this application. If talk announcements were available with semantic markup using other ontologies, we might be able to provide rules and transformation that could map or partially map the information into the ITtalks ontologies. We expect to encounter such marked-up announcements as the Semantic Web develops.

In the next step, the automated mapper automatically extracts the information. We would like to process talk announcements in plain text or HTML and automatically identify and extract the key information. This would let us fill the ITtalks database with information obtained from email or Web announcements. Others have studied the problem of recognizing and extracting information from talk announcements, mostly for use in a machine learning application.[14,15]

User profiles

We use personal profiles to help ITtalks meet individual user requirements. You can easily share this profile, and the use of DAML+OIL will allow more expressive content for schedules, preferences, and interests. The notion of a personal profile and a user agent are closely linked; a user might have one or the other, or both. The profile would likely express much of the information encoded in a user agent's knowledge base. Conversely, an agent would likely be able to answer queries about information in a profile.

Future directions

Because most users don't have personal agents, we have been developing one that you can use with this system. However, we'd like ITtalks to be able to interact with external agents of any type. The agent we are developing reasons about the user's interests, schedules, and assertions and uses the MapQuest agent to determine if a user can attend a particular talk.

We are developing a framework to use DAML+OIL in distributed trust and belief. DAML+OIL expressions on a Web page that encodes an agent's statement or other speech act are signed to provide authentication and integrity. We are working on an ontology for the description of permissions, obligations, and policies in DAML+OIL, which will support their use, exchange, and delegation by agents.

To make the data entry process more efficient, we are developing a focused Web spider, which will collect talk announcements from the source and identify key information using a text extraction system. The spider will add all found and relevant information to the ITtalks knowledge base.

In the future, all services that require personal user information should access the same user profile, eliminating the need to repeatedly enter the same information for a multitude of services. We believe that the new standard for XML Signature and Encryption under development might provide a mechanism by which users could control access to different parts of their profile.

Our system demonstrates the value of markup languages to the Semantic Web through its ability to improve Web agent functionality and to represent ontologies and ACLs. Each ITtalks Web page contains the necessary information for an agent to retrieve the page's DAML+OIL-encoded description and a responsible agent's contact information to provide more effective conversation. ITtalks thus gives each agent the ability to retrieve and manipulate all ITtalks-related information through a Web site interface or a direct agent-to-agent conversation. Hence, by combining the features of existing Web applications with the DAML+OIL-based knowledge and reasoning capabilities, we believe ITtalks presents a true Semantic Web application. ■

Acknowledgments

The Defense Advanced Research Projects Agency under contract F30602-00-2-0 591 AO K528 as part of the DAML program (www.daml.org) partly supported this research.

References

1. J. Hendler, "Agents and the Semantic Web," *IEEE Intelligent Systems*, vol. 16, no. 2, Mar./Apr. 2001, pp. 30–37.
2. S.A. McIlraith, T.C. Son, and H. Zeng, "Semantic Web Services," *IEEE Intelligent Systems*, vol. 16, no. 2, Mar./Apr. 2001, pp. 46–53.
3. R.S. Cost et al., "Jackal: A Java-Based Tool for Agent Development," *Proc. Workshop Tools for Developing Agents* (AAAI 98), AAAI Press, Menlo Park, Calif., 1998, pp. 73–82.
4. T. Finin, Y. Labrou, and J. Mayfield, "KQML as an Agent Communication," *Software Agents*, MIT Press, Cambridge, Mass., 1997, pp. 291–316.
5. *FIPA 97 Specification Part 2: Agent Communication Language*, tech. report, Foundation for Intelligent Physical Agents, 1997; www.fipa.org (curent Jan. 2002).
6. F. Bellifemine, A. Poggi, and G. Rimassa, "Developing Multi Agent Systems with a FIPA-Compliant Agent Framework," *Software: Practice and Experience*, vol. 31, no. 2, Feb. 2001, pp. 103–128.
7. K.D. Bollacker, S. Lawrence, and C.L. Giles, "Citeseer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications," *Proc. 2nd Int'l Conf. Autonomous Agents* (Agents 98), ACM Press, New York, 1998, pp. 116–123.
8. L. Kagal et al., "A Framework for Intelligent Services in a Mobile Environment," *Proc. Int'l Workshop Smart Appliances and Wearable Computing* (IWSAWC), 2001, pp. 195–201.
9. A. Cedilnik et al., *A Secure Infrastructure for Service Discovery and Access in Pervasive Computing*, tech. report TR-CS-01-12, Computer Science and Electrical Eng. Dept., Univ. of Maryland Baltimore County, Baltimore, Md., 2001.
10. D. Chakraborty et al., "Dreggie: Semantic Service Discovery for m-Commerce Applications," *Proc. Workshop Reliable and Secure Applications in Mobile Environment*, 2001, pp. 28–31.
11. O. Ratsimor et al., "Agents2go: An Infrastructure for Location-Dependent Service Discovery in the Mobile Electronic Commerce Environment," *Proc. ACM Mobile Commerce Workshop*, ACM Press, New York, 2001.
12. L. Kagal et al., "An Infrastructure for Distributed Trust Management," *Proc. Autonomous Agents Workshop Norms and Institutions in Multiagent Systems* (Agents 01), 2001.
13. N. Li and B. Grosof, "A Practically Implementable and Tractable Delegation Logic," *IEEE Symp. Security and Privacy*, IEEE Computer Soc. Press, Los Alamitos, Calif., 2000, pp. 27–42.
14. T. Eliassi-Rad and J. Shavlik, *Instructable and Adaptive Web-Agents That Learn to Retrieve and Extract Information,* tech. report 2000-1, Machine Learning Research Group, Dept. of Computer Sciences, Univ. of Wisconsin, Madison, 2000.
15. F. Ciravegna, "Learning to Tag for Information Extraction from Text," *Proc. 14th European Conf. Artificial Intelligence Workshop on Machine Learning for Information Extraction*, 2000.

The Authors

R. Scott Cost is a research assistant professor of computer science and electrical engineering at the UMBC. His research interests include software agents and agent communication, information retrieval, electronic commerce, and the Semantic Web. He received an AB from Colgate University, an MSE from Johns Hopkins University, and a PhD from UMBC—all in computer science. He is a member of the ACM, IEEE, and AAAI. Contact him at the Laboratory for Advanced Information Technology, Univ. of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; cost@csee.umbc.edu.

Tim Finin is a professor in the Department of Computer Science and Electrical Engineering and director of the Institute for Global Electronic Commerce at UMBC. His research interests include the applications of AI to problems in information systems, intelligent interfaces, and language processing. He received an SB in electrical engineering from MIT and a PhD in computer science from the University of Illinois. He is a former AAAI councilor and currently serves as the AAAI's representative on the Computing Research Association's board of directors. Contact him at the Laboratory for Advanced Information Technology, Univ. of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; finin@csee.umbc.edu.

Anupam Joshi is an associate professor of computer science and electrical engineering at UMBC. His research interests include networked computing, intelligent systems, data mining, the Semantic Web, content-based retrieval of video data from networked repositories, and networked high-performance computing and communications. He has a BTech in electrical engineering from the Indian Institute of Technology, Delhi, and an MS and a PhD in computer science from Purdue University. He is an associate editor of *IEEE Transactions on Fuzzy Systems*. He is a member of the IEEE, IEEE Computer Society, and ACM. Contact him at the Laboratory for Advanced Information Technology, Univ. of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; joshi@csee.umbc.edu.

Yun Peng is assistant director of the graduate program for the Department of Computer Science and Electrical Engineering at UMBC. His research interests include a variety of subjects in artificial intelligence, neural networks, and artificial life. He has a BS in electrical engineering from the Harbin Engineering Institute, China, an MS in computer science from Wayne State University, and a PhD in computer science from the University of Maryland, College Park. He is an associate editor of *Pattern Recognition* and *Computers in Biology and Medicine*. He is a member of the AAAI and the international Neural Network society. Contact him at the Laboratory for Advanced Information Technology, Univ. of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; ypeng@csee.umbc.edu.

Charles Nicholas is a professor of computer science at UMBC. He is the director of the UMBC's Center for Architectures for Data-Driven Information Processing, sponsored by the US Department of Defense. His research interests include document engineering, information retrieval, and programming languages. He received a BS from the University of Michigan, Flint, and an MS and a PhD from Ohio State University—all in computer science. He is general chair of the 2002 ACM International Conference on Information and Knowledge Management (CIKM 02). He is a member of the ACM and IEEE. Contact him at the Laboratory for Advanced Information Technology, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; nicholas@csee.umbc.edu.

Ian Soboroff is a researcher with the Retrieval Group at the National Institute of Standards and Technology. His research interests include information filtering, distributed retrieval systems, and methods for scaling current information retrieval, laboratory-style evaluations to Tbyte-sized document collections and beyond. He received his BS, MS, and PhD in computer science from UMBC. He is a member of the ACM and USENIX. Contact him at NIST, 100 Bureau Dr., Stop 8940, Gaithersburg, MD, 20899-8940; ian.soboroff@nist.gov.

Harry Chen is a PhD student and a graduate research assistant in the Department of Computer Science and Electrical Engineering at UMBC. His research interests include mobile computing and artificial intelligence. He received a 2001 PhD Research Fellowship from HP Labs. He received his BS and MS in computer science from UMBC. He is a student member of the ACM, IEEE, and AAAI. Contact him at the Laboratory for Advanced Information Technology, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; hchen4@csee.umbc.edu.

Lalana Kagal is a PhD student and graduate research assistant in the Department of Computer Science and Electrical Engineering at UMBC. Her research interests include artificial intelligence, security, and mobile computing. She has a BS and an MS in computer science from Pune University, India. Contact her at the Laboratory for Advanced Information Technology, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; lkagal1@csee.umbc.edu.

Filip Perich is a PhD student and graduate research assistant in the Department of Computer Science and Electrical Engineering at UMBC. His research interests include artificial intelligence, mobile computing, and mobile data management. He has a BA in mathematics from Washington College, Maryland. He is a member of the ACM. Contact him at the Laboratory for Advanced Information Technology, Univ. of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; fperic1@csee.umbc.edu.

Youyong Zou is a PhD student at UMBC. His research interests include Semantic Web language (DAML) reasoning, using Semantic Web languages in the Foundation for Intelligent Physical Agents, and ontology mapping. He has a BS and an MS in computer science from NanKai University, China. Contact him at the Laboratory for Advanced Information Technology, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; yzou1@csee.umbc.edu.

Sovrin Tolia is a graduate student at UMBC, pursuing his MS in computer science. His research interests include mobile data management, sensor networks, agents, and the Semantic Web. He has a BS in information systems from the Birla Institute of Technology and Science, India. Contact him at the Laboratory for Advanced Information Technology, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; stolia1@csee.umbc.edu.