ITTALKS: An Application of Agents in the Semantic Web [⋆]

Filip Perich, Lalana Kagal, Harry Chen, Sovrin Tolia, Youyong Zou, Tim Finin,
Anupam Joshi, Yun Peng, R. Scott Cost, and Charles Nicholas

Laboratory for Advanced Information Technology
University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250
{fperic1, lkagal1, hchen4, stolia1, yzou1, finin, joshi, peng, cost, nicholas} @ csee.umbc.edu

Abstract. Effective use of the vast quantity of information now available on the web will require the use of "Semantic Web" markup languages such as the DARPA Agent Markup Language (DAML). Such languages will enable the automated gathering and processing of much information that is currently available but insufficiently utilized. Effectively, such languages will facilitate the integration of multi-agent systems with the existing information infrastructure. As part of our exploration of Semantic Web technology, and DAML in particular, we have constructed ITTALKS, a web-based system for automatic and intelligent notification of information technology talks. In this paper, we describe the ITTALKS system, and discuss the numerous ways in which the use of Semantic Web concepts and DAML extend its ability to provide an intelligent online service to both the human community and the agents assisting them.

1 Introduction

With the vast quantity of information now available on the Internet, there is a need to manage this information by marking it up with a semantic language, such as DARPA Agent Markup Language (DAML) [26], and using intelligent search engines, in conjunction with ontology-based matching, to provide more efficient and accurate information search results. The aim of the Semantic Web is to make the present web more machine-readable, in order to allow intelligent agents to retrieve and manipulate pertinent information. The key goal of the DAML program is to develop a Semantic Web markup language that provides sufficient rules for ontology development [20] and that is sufficiently rich to support intelligent agents and other applications [22, 32]. Today's agents are not tightly integrated into the web infrastructure. If our goal is to have agents acting upon and conversing about web objects, they will have to be seamlessly integrated with the web, and take advantage of existing infrastructure whenever possible (e.g., message sending, security, authentication, directory services, and application service frameworks). We believe that DAML will be central to the realization of this goal.

In support of this claim, we have constructed a real, fielded application, ITTALKS, which supports user and agent interaction in the domain of talk discovery. It also provides a simple web-driven infrastructure for agent interaction. In addition, ITTALKS

[⋆] This work was supported in part by the Defense Advanced Research Projects Agency under contract F30602-00-2-0 591 AO K528 as part of the DAML program (http://daml.org/).

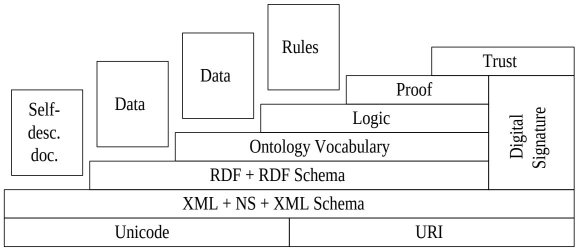


Fig. 1. Tim Berners-Lee's vision of the Semantic Web is founded on a base that includes URIs, XML and RDF.

serves as a platform for designing and prototyping the software components required to enable developers to create intelligent software agents capable of understanding and processing information and knowledge encoded in DAML and other semantically rich markup languages. To date, we have focused on developing the support and infrastructure required for intelligent agents to integrate into an environment of web browsers, servers, application server platforms, and associated supporting languages (e.g., WEB/SQL, WEBL), protocols (e.g., SSL, S/MIME, WAP, eSpeak), services (e.g., LDAP) and underlying technologies (e.g., Java, Jini, PKI).

On the surface, ITTALKS is a web portal offering access to information about talks, seminars and colloquia related to information technology (IT). It is organized around domains, which typically represent event hosting organizations such as universities, research laboratories or professional groups, and which in turn are represented by independent web sites. ITTALKS utilizes DAML for its knowledge base representation, reasoning, and agent communication. DAML is used to markup all the information, which is stored in a knowledge base, to provide additional reasoning capabilities otherwise unavailable. With information denoted in a semantically machine-understandable format, the computer can deduce additional information, a task which is difficult in a traditional database system. For example, if both ITTALKS and the user agree on a common semantics, the ITTALKS web portal can provide not only the talks that correspond to the user's profile in terms of interest, time, and location constraints, but can further filter the IT events based on information about the user's personal schedule, inferred location at the time of the talk, distance and current traffic patterns, etc. ITTALKS can also dynamically update the user's profile with incremental learning of the user's usage patterns.

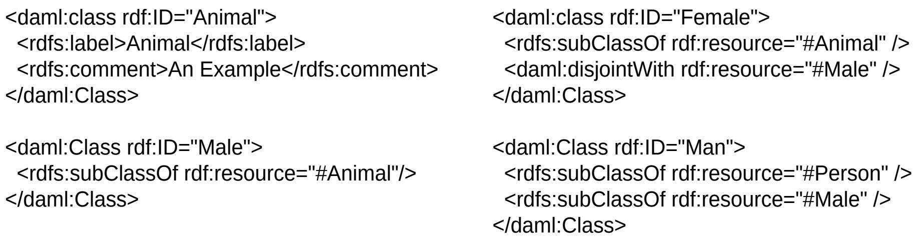
ITTALKS demonstrates the power of markup languages such as DAML for the Semantic Web, drawing on its ability to represent ontologies, agent content languages and its ability to improve the functionality of agents on the web. We have developed DAML-encoded ontologies for describing event, temporal, spatial, personal, and conversational information, which enable us to represent all required knowledge in a DAML-encoded format. Moreover, these ontologies enable us to execute a computer understandable conversation. In addition, we have created several DAML-encoded classification ontologies, which provide us with additional reasoning capabilities in order to find the best matching IT talks for a particular user. Furthermore, in the ITTALKS application, any web page presented on the ITTALKS web sites contains the necessary information for an agent to retrieve the DAML-encoded description of this page as well as the contact information of a responsible agent in order to provide more effective conversation. ITTALKS thus provides each agent with the capability to retrieve and manipulate any ITTALKS-related information via a web site interface or through a direct agent-to-agent conversation. Hence, by combining the features of currently existing web applications with the DAML-based knowledge and reasoning capabilities, ITTALKS presents a true Semantic Web application.

2 Background

DAML is a semantic language being developed by a consortium of U.S.-based academic and business researchers [1], which officially began in August 2000, to address the current web's limitations in providing machine-readable, and more importantly machine-interpretable, information over the Internet. The goal of DAML is to enable the transformation of the currently human-oriented web, which is largely used as a text and multimedia repository only, into a Semantic Web as envisioned by Berners-Lee [4, 3]. This process involves the augmentation of web pages with additional information and data that are expressed in a way that facilitates machine understanding [21, 23].

DAML is built upon the capabilities of an already existing syntactic language, the Extendable Markup Language (XML) [19], and of the Resource Description Framework and Resource Description Framework Schema (RDF/S) [16, 18, 17, 36]. These are XML applications that provide a number of preliminary semantic facilities required in the realization of the Semantic Web vision.

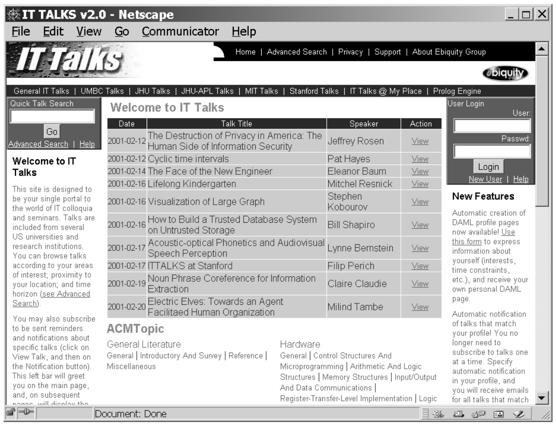
XML was developed by the World Wide Web Consortium (W3C) as a standard for alternative data encoding on the Internet that was primarily intended for machine processing. Moreover, XML is an application profile of the Standard Generalized Markup Language (SGML), and is therefore based on a well-defined and well-understood syntactic language. The XML standard provides the necessary means to declare and use



```
<daml:class rdf:ID="Animal">
  <rdfs:label>Animal</rdfs:label>
  <rdfs:comment>An Example</rdfs:comment>
</daml:Class>

<daml:Class rdf:ID="Male">
  <rdfs:subClassOf rdf:resource="#Animal"/>
</daml:Class>

<daml:class rdf:ID="Female">
  <rdfs:subClassOf rdf:resource="#Animal" />
  <daml:disjointWith rdf:resource="#Male" />
</daml:Class>

<daml:Class rdf:ID="Man">
  <rdfs:subClassOf rdf:resource="#Person" />
  <rdfs:subClassOf rdf:resource="#Male" />
</daml:Class>
```

Fig. 2. An example of DAML-encoded knowledge.

[1] Visit the DAML Program official web site at `http://www.daml.org/`.


IT TALKS v2.0 - Netscape
File Edit View Go Communicator Help
IT Talks
Home | Advanced Search | Privacy | Support | About Ebiquity Group
ebiquity
General IT Talks | UMBC Talks | JHU Talks | JHU-APL Talks | MIT Talks | Stanford Talks | IT Talks @ My Place | Prolog Engine
Quick Talk Search
Go
Advanced Search | Help
User Login
User:
Passwd:
Login
New User | Help
Document: Done

Welcome to IT Talks

Date	Talk Title	Speaker	Action
2001-02-12	The Destruction of Privacy in America: The Human Side of Information Security	Jeffrey Rosen	View
2001-02-12	Cyclic time intervals	Pat Hayes	View
2001-02-14	The Face of the New Engineer	Eleanor Baum	View
2001-02-16	Lifelong Kindergarten	Mitchel Resnick	View
2001-02-16	Visualization of Large Graph	Stephen Kobourov	View
2001-02-16	How to Build a Trusted Database System on Untrusted Storage	Bill Shapiro	View
2001-02-17	Acoustic-optical Phonetics and Audiovisual Speech Perception	Lynne Bernstein	View
2001-02-17	ITTALKS at Stanford	Filip Perich	View
2001-02-19	Noun Phrase Coreference for Information Extraction	Claire Claudie	View
2001-02-20	Electric Elves: Towards an Agent Facilitaed Human Organization	Milind Tambe	View

ACMTopic

General Literature
General | Introductory And Survey | Reference | Miscellaneous

Hardware
General | Control Structures And Microprogramming | Arithmetic And Logic Structures | Memory Structures | Input/Output And Data Communications | Register-Transfer-Level Implementation | Logic

Welcome to IT Talks

This site is designed to be your single portal to the world of IT colloquia and seminars. Talks are included from several US universities and research institutions. You can browse talks according to your areas of interest; proximity to your location; and time horizon (see Advanced Search).

You may also subscribe to be sent reminders and notifications about specific talks (click on View Talk, and then on the Notification button). This left bar will greet you on the main page, and, on subsequent pages, will display the

New Features

Automatic creation of DAML profile pages now available! Use this form to express information about yourself (interests, time constraints, etc.), and receive your own personal DAML page.

Automatic notification of talks that match your profile! You no longer need to subscribe to talks one at a time. Specify automatic notification in your profile, and you will receive emails for all talks that match

Fig. 3. A screenshot depicting the main page of the ITTALKS system.

simple data structures, which are stored in XML documents and which are machine-readable. Subsequently, the information made available in these documents can be processed or translated into additional XML documents to provide the appropriate form for human understanding, such as text-to-voice, graphics or HTML conversion. However, since XML is defined only at the syntactic level, machines cannot be relied upon to unambiguously determine the correct meaning of the XML tags used in a given XML document. Consequently, XML is not suitable as a desired language for representing complex knowledge.

As a result, the W3C Consortium has developed RDF/S with the goal of addressing the XML deficiencies by adding formal semantics on the top of XML. These two standards provide the representation frameworks for describing relationships among resources in terms of named properties and values, which are similar to representation frameworks of semantic networks and rudimentary frame languages such as RDF Schema. Yet, both standards are still very restricted as a knowledge representation language due to the lack of support for variables, general quantification, rules, etc.

DAML is an attempt to build upon 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 similar rules and definitions to RDF/S. In addition, DAML also provides rules for describing further constraints and relationships among resources, including cardinality, domain and range restrictions, as well as union, disjunction, inverse and transitive rules. DAML is, therefore, an effort to develop a universal Semantic Web markup language that is sufficiently rich to provide machines not only with the capability to read data but also with the capability to interpret and infer over the data. DAML will enable the development of intelligent agents and applications that can autonomously retrieve and manipulate information on the Internet and from the Semantic Web of tomorrow.

3 ITTALKS

As part of UMBC's role in the DAML Program, we have developed ITTALKS; a web portal that offers access to information about talks, seminars, colloquia, and other information technology (IT) related events. ITTALKS provides users with numerous details describing the IT events, including location, speaker, hosting organization, and talk topic. More importantly, ITTALKS also provides agents with the ability to retrieve and manipulate information stored in the ITTALKS knowledge base. Below, we discuss various aspects of the system in more detail.

Unlike other web services, ITTALKS employs DAML for knowledge base representation, reasoning, and agent communication. The use of DAML to represent information in its knowledge base, in conjunction with its use for interchangeable type ontologies as described in Section 5.6, enables more sophisticated reasoning than would otherwise be available. For example, a simpler representation scheme might be able to provide the user with talks based on interest, time and location. When both ITTALKS and the user agree on a common semantics, the ITTALKS web portal will be able to perform further filtering, based on more sophisticated inference. In addition to enhancing knowledge representation and reasoning, DAML is used for all communication, including simple messages and queries, using the ITTALKS defined ontology. Moreover, ITTALKS offers the capability for each user to use his/her personal agent to communicate with ITTALKS on his/her behalf and provide a higher level of service.

3.1 Users

ITTALKS can be used anonymously, or, more effectively, with personalized user accounts. Users have the option to register with ITTALKS either by entering information manually via web forms, or by providing the location (URL) of a universally accessible DAMLized personal profile, which includes information such as the users location, his/her interests and contact details, as well as a schedule. This schedule might be as rudimentary as a list of available time periods for given days, or could even include a detailed schedule for each day. Subsequently, this information is used to provide each user with a personalized view of the site, displaying only talks that match the user's interests and/or schedule.

Since DAML is not yet in widespread use, ITTALKS provides a tool for creating a DAML personal profile. Currently, the tool constructs a profile containing only items used by the ITTALKS system. However, we believe that the profile, in one form or another, will ultimately provide a unique and universal point for obtaining personal information about the user, not just for ITTALKS, but for all information needs, and will include any sort of information the user would like to share. In the future, all services that require personal information about the user should access the same user profile, eliminating the need for the user to repeatedly enter the same information for a multitude of services. We believe that the new standard for XML Signature and Encryption under development may provide a mechanism by which users can have some control over access to parts of their profile.

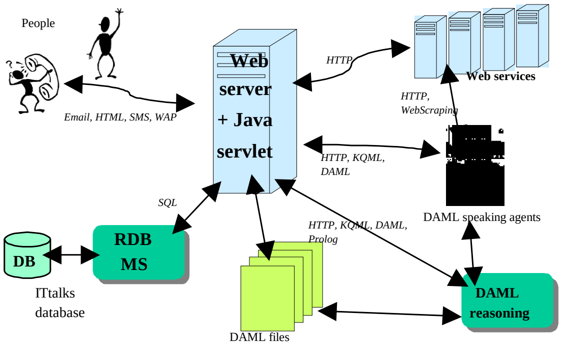


Fig. 4. The architecture for ITTALKS is built around a web server backed by a relational database. Interfaces are provided to human users, software agents and web services.

3.2 Domains

To support our vision of a universal resource for the international IT research community, ITTALKS is organized around domains, which typically represent event hosting organization such as universities, research laboratories or professional groups. Each domain is represented by a separate web site and is independently maintained by a moderator who can, among other things, define the scope of the domain and delegate to other registered users the ability to edit talk entries. For example, the `stanford.ittalks.org` domain might be configured to include only talks hosted at Stanford University. On the other hand, another domain, `sri.ittalks.org`, might be configured to include not only talks about Semantic Web topics that are held at SRI, but also those at Stanford, as well as any talks within 15 mile range of the SRI facility in Palo Alto.

3.3 Access

The ITTALKS system is accessible either to users directly via the web, or to agents acting on their behalf. The web portal provides numerous features, including registration, search, entry and domain administration. An agent-based interface allows interaction with user agents or other services.

Human Interface. The web portal allows a user to browse desired information in a variety of formats, to provide the highest degree of interoperability. It permits a user to retrieve information in DAML, standard HTML format, which includes a short DAML annotation for DAML-enabled web crawlers, or WML [11] format, which supports WAP enabled phones. The ITTALKS web portal also has the ability to generate RDF Site Summary (RSS) [34] files for certain queries. These RSS files can then be used for various external purposes, such as displaying upcoming talks on a departmental web site for some particular university or domain.

Agent Interface. To provide access for agent based services, ITTALKS makes use of Jackal [12], a communication infrastructure for Java-based agents developed by our research group at UMBC. Jackal is a Java package, which provides a comprehensive communications infrastructure while maintaining maximum flexibility and ease of integration. The heart of Jackal is a simple conversation system, serving to maintain context for concurrent threads of conversation while providing a guide for judging behavioral correctness and modeling the actions of other agents. Jackal provides facilities for creating and manipulating user-defined conversation structures of arbitrary extent. Jackal has a very modular, loosely coupled architecture, designed to support maximal concurrency among components, accomplished with the use of multiple threads and buffered interfaces between subsystems. Its concise API allows for comprehensive specification of message requests, and for blocking or non-blocking message retrieval. Currently, it facilitates the use of KQML agent communication language [14] and employs a sophisticated protocol for agent naming, addressing and identity (KNS). Additionaly, it is in the process of adapting to the FIPA standards [15, 2]. In addition, our research group, in cooperation with other universities, is developing a DAML ontology for the necessary conversation protocols.

3.4 Agents

In order to extend the capabilities of the system, we have defined a number of agents that support the operation of ITTALKS. Some can be seen as supporting services (such as external information services), while others we assume will exist in the general environment in the future.

ITTALKS Agent. The ITTALKS agent is a front-end for the ITTALKS system. It interacts with ITTALKS through the same web-based interface as human users, but communicates via an ACL with other agents on the web, extending the system's accessibility. At present, the agent does not support any advanced functionality, but acts primarily as a gateway for agent access.

User Agents. One longtime goal of agent research is that users will be represented online by agents that can service queries and filter information for them. While ITTALKS does not require that such agents exist, we recognize the added power that could be gained by the use of such agents. Therefore, ITTALKS supports interaction with User Agents as well as their human counterparts. The User Agent that we have developed understands DAML, supports sophisticated reasoning, and communicates via a standard agent communication language. Reasoning is accomplished with the use of the XSB, a logic programming and deductive database system for Unix and Windows developed at SUNY Stony Brook.

Calendar Agent. Although a user agent may 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 agent - the calendar agent. This enables the user

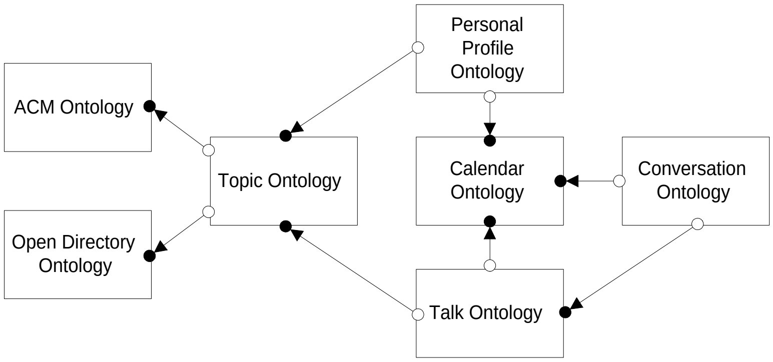


Fig. 5. The relationships among the various ontologies used by the ITTALKS system.

agent not only to consult the user calendar, but also to use the same protocol to consult other calendar agents that may represent other users or groups the user belongs to. In addition, the calendar agent may only represent abstraction to already existing infrastructure, such as Microsoft Outlook [33] or other desktop/server applications. Finally, the calendar agent may also be used to represent a room, and thus allow for re-use of the same principles of participation scheduling as well as event scheduling.

Classifier Agent. ITTALKS uses a Classifier (or recommender) Agent that is invoked when a user is entering a new talk. Based on the talk's abstract, the Classifier returns ACM Classification Hierarchy Classification numbers along with a rank, in descending order. Using a local table of classification numbers and names, ITTALKS suggests to the user ten possible topics.

MapQuest Agent. The MapQuest Agent is a wrapper agent that allows ITTALKS to make use of external services. It interacts directly with agents (e.g. the ITTALKS agent, 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 [29], parses the results, and generates an appropriate response. Note that this agent could be generically named a Distance Agent, and make use of any external service (or combination of several, as needed).

3.5 Ontologies

The ITTALKS system is based on a set of ontologies[2] that are used to describe talks and the things associated with them, e.g., people, places, topics and interests, schedules, etc. Figure 5 shows some of the dependencies that exist among these ontologies. The ontologies are used in the representation and processing of DAML descriptions and

[2] http://daml.umbc.edu/ontologies

also as “conceptual schemata” against which the database and various software APIs are built.

We have developed a general ontology for describing the topics of arbitrary talks and papers. Using this, we have implemented an ontology to describe IT related talks based on the ACM’s Computer Classification System. In addition, we currently are developing a DAML ontology for IT talks based on a portion of the Open Directory, and are considering additional classification ontologies. Figure 7 sketches some of the major classes and properties in these ontologies. These topic ontologies are used to describe talks as well as the users’ interests throughout the system. This includes an automated talk classification, for which we have obtained a training collection for the ACM CCS and are also generating an Open Directory training collection to develop the necessary components. In addition, the DAML ontologies will give a user the ability to add additional assertions in DAML to further characterize their interests. Lastly, we are also in the process of developing a semi-automated component that can map topics in one ontology into topics in another, by utilizing user-specific mapping believes and by taking advantage of the fact that nodes in each ontology have an associated collection of text. This component is further described in Section 5.4.

3.6 Data Entry

Currently ITTALKS requires that information about talks be manually entered via a web form interface, or be available in a DAML description available at a given URL. Although we have made this process as simple as possible (e.g., by supporting automatic form completion using information from the knowledge base and the user’s DAML profile) it is still a time consuming process. Therefore, we are developing a focused web spider to collect talk announcements from open sources on the web. This spider will identify key information items using a text extraction system, and will automatically add information to the ITTALKS knowledge base. We are working with the Lockheed-Martin research group on the above task, and will use their AeroText information extraction system [1].

3.7 Architecture

The current implementation of ITTALKS uses a relational database, in combination with a web server, to provide user access to the system. To enable agents to access the system, the ITTALKS provides an interface for agent-based communication.

Database. The main software packages that are used in the ITTALKS system are the MySQL relational database software and a combination of Apache and Tomcat as the web portal servers. The contents of the ITTALKS knowledge base are stored in a database whose schema is closely mapped to our ontologies describing events, people, topics and locations. We have chosen MySQL because of its known reliability, and because we required software with a license that allows us to make the ITTALKS package available to additional academic and commercial institutions.

Web Server. As stated above, for our web, we have chosen a combination of Apache and Tomcat. This enables us to present the IT talk descriptions to the user using Java servlets and JSP files, which dynamically generate requested information in DAML, XML, HTML, RSS, and WML formats. The current ITTALKS implementation can provide information suitable for viewing on either a standard, computer-based or a WAP-enabled cellular phone.

Extensions. In addition, we are currently employing the Jackal agent communication infrastructure developed at UMBC and the Lockheed-Martin's AeroText information extraction system in order to facilitate ITTALKS-user agent interaction and the automated text extraction, respectively. We are in the process of modifying Jackal to provide support for FIPA ACL interoperability. Also, we are considering the possible replacement of MySQL with native XML database software such as dbXML.

4 Scenarios

We describe here a couple of typical interactions that illustrate some of the features of ITTALKS. The first involves direct use by a human user, and the second, advanced features provided through the use of agents.

4.1 Human Interaction

In this first scenario, a user, Jim, learns from his colleagues about the existence of 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 `stanford.ittalks.org` (a Stanford ITTALKS domain), he selects it, and is presented with a new page listing upcoming talks that are scheduled at Stanford, SRI and other locations within a 15-mile radius (the default distance for the Stanford domain).

Jim browses the web site, viewing announcements for various talks matching his interests and preferred locations (as provided in his explicit search queries). He is impressed that he can see the talk information not only in HTML, but also in DAML, RSS and WML formats. Finding a talk of potential interest to a colleague, Jim takes advantage of the invitation feature, which allows him to send an invitational e-mail to any of his friends for any of the listed talks. Finally, using the personalize link on the bottom of the page, Jim creates his own ittalks.org main page, by providing the URL of his DAML-encoded profile. This customized page, listing talks based on his preferences, will be Jim's entrance to the ITTALKS site whenever her returns.

4.2 Agent Interaction

This scenario assumes that user Jim has already registered with ITTALKS, and has left instructions with the system to be notified of the occurrence of certain types of talks.

In the course of operation, ITTALKS discovers that there is an upcoming talk that may interest Jim, and of which Jim has not been notified. Based on information in Jim's

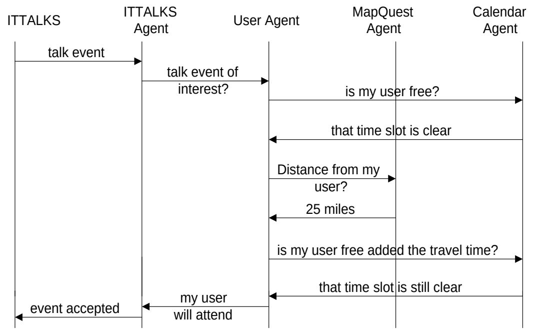


Fig. 6. Interactions between the various agents described in the ITTALKS/Agent scenario.

preferences, which have been obtained from his online, DAML-encoded profile and from information entered directly, ITTALKS opts to notify Jim's User Agent directly. This is done via ITTALKS own agent, which forwards the message using an ACL.

Upon receiving this information, Jim's User Agent needs to know more; it consults with Jim's Calendar agent to determine his availability, and with the MapQuest agent to find the distance from Jim's predicted location at the time of the talk. Some more sophisticated interactions might take place at this time; for example, the Calendar and User agents may decide to alter Jim's schedule, and proceed to contact the User agent of some other individual. In addition, the User agent may request more information about the speaker and the event by contacting other agents or web sites, such as CiteSeer-based agent [5, 30, 6], to obtain more information necessary to make a decision. Finally, after making this decision, the User Agent will send a notification back to the ITTALKS agent indicating that Jim will/will not plan to attend. The ITTALKS agent will make the appropriate adjustments at the ITTALKS site.

In a more complex interaction scheme, Jim may be employed by a research group, which possesses a limited funding and is therefore enforcing a policy that allows only one researcher at a time to attend a particular IT event. As a result, the User agent cannot decide on Jim's participation until it successfully interacts with other agents representing Jim's employer and colleagues. Therefore, the decision whether anyone from the research group could attend the IT event and the further election of the group representative requires an interaction of agent virtual community.

From a slightly different perspective, the User agent will also benefit from participating in virtual communities thanks to recommendations it obtains from other User agents. One User agent may recommend an IT event given its owner's experiences from attending a past talk of the same speaker. Another User agent may decide to share comparisons of two competing times and locations for an identical IT event. Yet another User agent may simply share its owner's intensions on attending a particular IT event. Thus, each member of the virtual community can profit from these and many other positive and negative recommendations, and reflect these social filtering methods in its own decisions.

Finally, in a 'Smart Office' scenario [25, 7], the ITTALKS agent may also be directly contacting an agent representing the location where a given IT event will be held. This 'room' agent may then use varying service discovery techniques [8, 35] to locate a projector presented in the room and inform it to pre-download the powerpoint presentation before the speaker arrival. Moreover, the 'room' agent may also try to contact additional agents in the IT event vicinity to decrease possible noise level from other rooms and to verify that a 'hallway' agent has requested enough refreshments during the IT event.

5 Benefits of DAML

We believe that ITTALKS benefits significantly from its use of a semantic markup language such as DAML. DAML is used to specify ontologies that we use extensively in our system. It is also used for personal profiles, and as an agent content language. Without DAML, specifying schedules, interests and assertions about topics would be very difficult. In ITTALKS, a user can specify that according to the user a couple of topics are equivalent or related or dissimilar, etc. This will allow ITTALKS to tailor the searching of talks to the users needs. As an agent content language, DAML provides more flexible semantics than KIF or other content languages that currently provide syntax only. The ultimate benefit of using DAML then lies in the ability of ITTALKS to independently interact with any DAML-capable agent without the need of a human supervision. Consequently, all these benefits, which are described in further details below, enable more efficient interaction between the system and its users, let them be humans or software agents.

5.1 Interoperability Standard

As an interoperability layer, DAML allows the content of ITTALKS to be easily shared with other applications and agents. For example, a Centaurus room manager agent [25] could watch ITTALKS for events happening in a room for which it is responsible in order to enable better scheduling. DAML also acts as an interoperability standard allowing other sites to make their talks available for inclusion in ITTALKS by publishing announcements marked up in our ontology.

5.2 Agent Communication Language

DAML and ACLs can be successfully integrated. DAML documents will be the objects of discourse for agents that will create, access, modify, enrich and manage DAML documents as a way to disseminate and share knowledge. Agents will need to communicate with one another not only to exchange DAML documents but also to exchange *informational attitudes* about DAML documents. Using an Agent Communication Languages (ACL) agents can "talk" about DAML documents. Integrating ACL work and concepts with a universe of DAML content is our first goal. Using DAML as an agent content language will add more meaning to the message.

5.3 Distributed Trust and Belief

Agents face a difficult problem of knowing what information sources (e.g. documents, web pages, agents) to believe and trust in an open, distributed and dynamic world, and how to integrate and fuse potentially contradictory information. DAML can be used to support *distributed trust and reputation management* [24, 27, 28]. This will form the basis of a logic for *distributed belief transfer* that will enable more sophisticated, semantically-driven rule-based techniques for information integration and fusion.

We are making use of DAML's expressiveness and employing it to describe security policies, credentials and trust relationships, which form the basis of trust management. These policies contain more semantic meaning, allowing different policies to be integrated and conflicts to be resolved relatively easily. Also, it will be possible for other applications to interpret the agent's credentials, e.g. authorization certificates, correctly, making these credentials universal.

Similarly, describing beliefs and associating levels of trust with these beliefs is more straightforward and the deduction of belief is uniform by different applications and services.

5.4 Data Entry Support

ITTALKS supports intelligent form filling, making it easier for users to enter and edit information in their profiles, and also to enter and edit talk announcements and other basic information. In addition, we provide automatic form filling when an editor tries to enter information about an entity (e.g. a talk, person, room) that already present in the knowledge base.

Entering Talks. In order to make ITTALKS successful, as we need to make it as easy as possible new talk descriptions to be entered into the system. We are addressing this problem using three complimentary approaches: an enhanced web interface, accepting marked up announcements, and automated text extraction. DAML plays a key role in the first two and is the target representation for the third.

Enhancing the Web Interface. We have used several techniques to enhance the web form interface for entering talk announcements. One of the simplest and most effective is to recognize then some of the information being entered about an object such as a person, a room or an organization has already been entered into the ITTALKS system and to "pre-fill" the remaining parts of the form from our stored information. For example, most talks at an organization are given in a small number of rooms. Once the complete information about a particular room (e.g., room number, building, address, seating capacity, longitude and latitude, A/V equipment, networking connection, etc.) has been entered for one talk, it need not be entered again.

Although the current implementation of this does not directly use DAML, its use can support a more generalized version of a web form-filling assistant. The approach depends on two ideas: (i) tagging web form widgets with DAML descriptions of what they represent and (ii) capturing dependencies among data items in DAML and (iii)

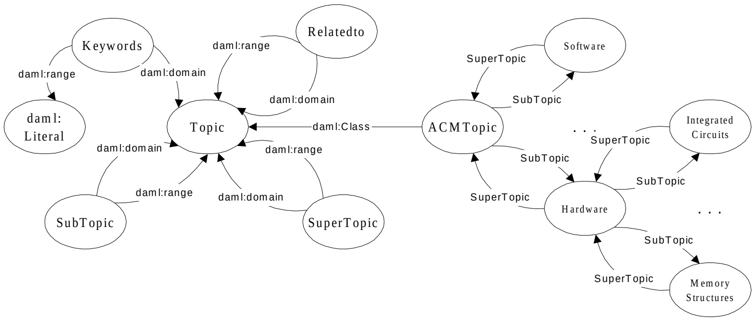


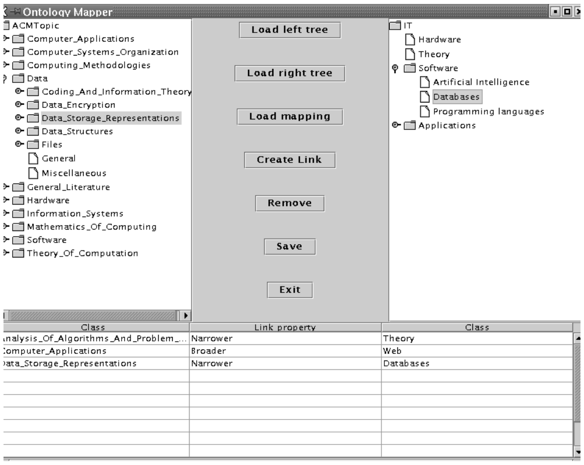
Fig. 7. The Ontologies used by IITALKS are relatively simple, such as the topics ontology used to describe talk topics and user interests.

compiling these dependencies into an appropriate execution form (e.g., JavaScript procedures) that can drive the web form interface.

In addition, we plan to investigate the possibility of a multi-modal support, where user can enter new information via standard keyboard input as well as through voice recognition means. Here, we understand that when presenting the user with a new form, the user will be allowed to use her own voice to enter data in each field. Then upon submittion of voice-filled form, ITTALKS will try to infer the meaning of the recorded sound, obtain additional information based on the knowledge and rules stored in ITTALKS system, and present back the user with a text-prefilled form for verification purposes. This enhancement will then allow ITTALKS to provide talk 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 need to know the appropriate topics for each talk. Initially, we required that users manually select appropriate topic categories from a web interface to the ACM CCS hierarchy. This turns out to be a daunting task requiring the user to navigate in a hierarchy of nearly 300 topics, many of which about whose meaning he will not be sure. Some users will face a similar problem in trying to select topics to characterize their own interests. Ultimately we would like to use more that one topic hierarchy to classify both talk topics and user interests (e.g., ACM CCS and Open Directory nodes), which makes the problem even more difficult for our users.

To address this problem, we have built an automatic text classifier that can suggest terms in a hierarchy that are appropriate for classifying a talk based on its title and abstract. The classifier package used was from the Bag Of Words (BOW) toolkit [31] by Andrew McCallum at CMU. 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, fairly effective, and 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 the text found by searching their web pages.



Class	Link property	Class
ınalysis_Of_Algorithms_And_Problem_...	Narrower	Theory
Computer_Applications	Broader	Web
Data_Storage_Representations	Narrower	Databases

Fig. 8. Snapshot of the Ontology Semi-Automatic Mapper Prototype.

In addition to classifying text into a particular ontology at a time, we are also developing a tool for mapping among multiple ontologies. Such tool will, for example, allow each user to select her preferred topic ontology on the fly, and the ITTALKS system will then immediately adapt and present the personalized filtering results according to this ontology. As a prototype of the mapping tool, we have chosen a semi-automatic approach, wherein user can manually select relations among specific topics accross the ontologies ("landmarks"), e.g. broader, narrower, and similar. Subsequently, the remaining mappings are automatically computed via a combination of the user-specified relations and our automatic text classifier using training sets of documents for the ontologies. The automated mapper then operates in the following manner:

Consider two topic ontologies A and B, each of which is a strict hierarchy. Each node in each hierarchy $(A_1, A_2, ..., A_n)$, $(B_1, B_2, ..., B_m)$ has a set of exemplary documents that have already been classified as being associated with that node. Let *parent*$_i$ be the parent of node i and *text*$_i$ be the string of all the text associated with node i. We have used a text classifier to build a classifier for ontology A and another one for ontology B. The classifier for ontology A is built taking an arbitrary string S to be classified and computing the function $S_a(S, A_i)$ for each node A_i in ontology A. The classifier for ontology B computes $S_b(S, B_i)$. These similarity scores could further be normalized as real numbers between 0 and 1. These two classifiers are used to compute two raw topic similarity matrices $SM_{ab}(A_i, B_j)$ and $SM_{ba}(A_i, b_j)$, for each pair of nodes, one from ontology A and one from ontology B. Therefore,

$$SM_{ab}(A_i, B_i) = S_b(text(A_i), B_j)$$

$$SM_{ba}(A_i, B_j) = S_a(text(B_j), A_i)$$

One simple way to compute the similarity of topic i in ontology A to topic j in ontology B would be to obtain a weighted sum of these two measures:

$$S_{ab}(A_i, B_j) = W * SM_{ab}(A_i, B_j) + (1 - W) * SM_{ba}(A_i, B_j)$$

In this way, the similarity between two nodes can be computed by taking a weighted sum of their raw similarity obtained from the classifier. To further improve mapping results, the the raw similarity scores are combined with the similarity scores of the nodes' parents and children, which enables the system to compute a more precise relations between each pair of nodes. Finally, the computed relations are converted into a DAML file, which is then made publicly available to be accessible by ITTALKS and other Semantic Web applications.

Accepting Marked Up Announcements. One of the simplest ways to enter new talk announcements is to provide them as a document that is already marked up. The current ITTALKS interface allows one to enter a URL for a talk announcement that is assumed to be marked up in ontologies that ITTALKS understands. Currently, these are just the "native" ontologies that we have built for this application. In general, if some talk announcements were available with semantic markup using other ontologies, it might be possible to provide rules and transformation that could map or partially map the information into the ITTALKS ontologies. We expect that, as the Semantic Web develops, it will be more and more likely that talk announcements with some meaningful mark up will be found on the web.

Automated Information Extraction from Text. We would like to be able to process talk announcements in plain text or HTML and automatically identify and extract the key information required by ITTALKS. This would allow us to fill the ITTALKS database with information obtained from announcements delivered via email lists or found on the web. The problem of recognizing and extracting information from talk announcements has been studied before [13, 10] mostly in the context of using it as a machine learning application. We are developing a information extraction use the Aerotext [1, 9] system that can identify and extract the information found in a typical talk announcement and use this to automatically produce a version marked up in DAML which can then be entered in the ITTALKS database.

5.5 User Profiles

We use personal profiles to help ITTALKS meet the requirements of individual users. A profile is a widely accessible source of information about the user, marked DAML, to which other services and individuals can refer. In the future, such a profile may be used by all web-based services that the user wants to access. The profile will ultimately provide a unique and universal point for obtaining personal information about the user for all services, preventing the need for duplication and potential inconsistencies. This profile can be easily shared, and with the use of DAML, 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 that might be encoded in a user agent's knowledge base. Conversely, an agent would likely be able to answer queries about information contained in a profile.

5.6 Modularity

With the use of DAML, we can define several ontologies for topics and switch between them with ease. Furthermore, to restrict the retrieval results, a user can perform the search with respect to a certain set of ontologies, such as the ACM or Open Directory Classification.

5.7 Application Scalability Support

As ITTALKS becomes the central repository of IT related information for various research institutes the ITTALKS knowledge base will be distributed among numerous, and possibly apriori-unknown, locations in order to provide a higher scalability and reliability support. Yet, it will be imperative that users and agents not be required to interact with all locations in order to find or manipulate the desired information. Instead, we envision that each user agent will interact with only one ITTALKS agent, which in turn will be able to efficiently locate and manage the distributed ITTALKS information. For this, we believe that a system of DAML-enabled agents can act as an intermediate between the distributed databases.

6 Future Directions

Since most users do not currently have personal agents, we have been developing one that can be used with this system. It is our goal, however, that ITTALKS be able to interact with external agents of any type. The agent we are developing reasons about the user's interests, schedules, assertions and uses the MapQuest agent to figure out if a user will be able to attend an interesting talk on a certain date.

We are developing a framework to use DAML in distributed trust and belief. DAML expressions on a web page that encodes a statement or other speech act by an agent are signed to provide authentication and integrity. We are working on an ontology to describe permissions, obligations and policies in DAML and allow agents to make statements about and delegate them.

In order to make the process of data entry more efficient, we are developing a focused web spider, which will collect talk announcements from source on the web and to identify the key information in these announcements using a text extraction system. The spider will add all found and relevant information to the ITTALKS knowledge base.

7 Conclusion

Effective use of the vast quantity of information now available on the web necessitates semantic markup such as DAML. With the use of such a tool, we can enable the automated or machine-facilitated gathering and processing of much information that is currently lost to us. ITTALKS, our system for automatic and intelligent notification of Information Technology talks, demonstrates the value of DAML in a variety of ways. DAML is used throughout the ITTALKS system, from basic knowledge representation, to inter-agent communication.

References

1. AeroText. site: http://mds.external.lmco.com/products_services/aero/.
2. Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. Developing multi agent systems with a fipa-compliant agent framework. *Software - Practice and Experience*, 3, 2001.
3. Tim Berners-Lee and Mark Fischetti. Weaving the web: The original design and ultimate destiny of the world wide web by its inventor. *Harper, San Francisco*, 2001.
4. Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.
5. Kurt D. Bollacker, Steve Lawrence, and C. Lee Giles. *Citeseer: An autonomous web agent for automatic retrieval and identification of interesting publications*. Proceedings of the Second International Conference on Autonomous Agents (Agents '98). ACM Press, Minneapolis, 1998.
6. Sergey Brin and Lawrence Page. *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. Proceedings of the 7th International World Wide Web Conference. April 1998.
7. Andrej Cedilnik, Lalana Kagal, Filip Perich, Jeffrey Undercoffer, and Anupam Joshi. A secure infrastructure for service discovery and access in pervasive computing. *Technical report, TR-CS-01-12, CSEE, University of Maryland Baltimore County*, 2001.
8. Dipanjan Chakraborty, Filip Perich, Sasikanth Avancha, and Anupam Joshi. Dreggie: Semantic service discovery for m-commerce applications. *Workshop on Reliable and Secure Applications in Mobile Environment, 20th Symposiom on Reliable Distributed Systems*, October 2001.
9. Lois C. Childs. Aerotext - a customizable information extraction system. unpublished technical report, Lockheed Martin, 2001.
10. Fabio Ciravegna. Learning to tag for information extraction from text. *ECAI Workshop on Machine Learning for Information Extraction*, August 2000. workshop held in conjunction with ECAI2000, Berline.
11. WAPForum Consorscium. Wireless markup language, November 1999. site: http://www1.wapforum.org/tech/documents/SPEC-WML-19991104.pdf.
12. R. Scott Cost, Tim Finin, Yannis Labrou, Xiaocheng Luan, Yun Peng, Ian Soboroff, James Mayfield, and Akram Boughannam. Jackal: A java-based tool for agent development. *Working Notes of the Workshop on Tools for Developing Agents, AAAI '98*, July 1998.
13. T. Elliassi-Rad and J.Shavlik. Instructable and adaptive web-agents that learn to retrieve and extract information. *Department of Computer Sciences, University of Wisconsin, Machine Learning Research Group Working Pap*, 2000.
14. Tim Finin, Yannis Labrou, and James Mayfield. *Software Agents*, chapter KQML as an agent communication. MIT Press, Cambridge, 1997.
15. FIPA. Fipa 97 specification part 2: Agent communication language. *Technical report, FIPA - Foundation for Intelligent Physical Agents*, October 1997.
16. W3C Working Group. W3c resource description framework (rdf), October 1998. site: http://www.w3c.org/RDF.
17. W3C Working Group. W3c resource description framework model and syntax specification, February 1999.

18. W3C Working Group. W3c resource description framework schema (rdfs), March 1999. site: http://www.w3.org/TR/rdf-schema/.
19. W3C Working Group. extensible markup language (xml), October 2000. site: http://www.w3.org/XML.
20. N. Guarino. *Formal Ontology in Information Systems*, chapter Formal ontology and information systems. IOS Press, 1998.
21. Jeff Heflin, James Hendler, and Sean Luke. Shoe: A prototype language for the semantic web. *Linkping Electronic Articles in Computer and Information Science, ISSN 1401-9841*, 6, 2001.
22. James Hendler. Agents and the semantic web. *IEEE Intelligent Systems*, 16(2):30–37, March/April 2001.
23. James Hendler and Deborah McGuinness. The darpa agent markup language. *IEEE Intelligent Systems*, 15(6):72–73, November/December 2000.
24. Lalana Kagal, Harry Chen, Scott Cost, Timothy Finin, and Yun Peng. An infrastructure for distributed trust management. *Autonomous Agents Workshop on Norms and Institutions in Multiagent Systems, AA'01, Montreal, Canada*, May 2001.
25. Lalana Kagal, Vlad Korolev, Harry Chen, Anupam Joshi, and Timothy Finin. A framework for intelligent services in a mobile environment. *Proceedings of the International Workshop on Smart Appliances and Wearable Computing (IWSAWC)*, April 2001.
26. DARPA Agent Markup Language. site: http://www.daml.org/.
27. Ninghui Li, Joan Feigenbaum, and Benjamin Grosof. A logic-based knowledge representation for authorization with delegation (extended abstract). *Proc. 12th IEEE Computer Security Foundations Workshop, Mordano, Italy*, June 1999. IBM Research Report RC 21492.
28. Ninghui Li and BBenjamin Grosof. A practically implementable and tractable delegation logic. *IEEE Symposium on Security and Privacy*, May 2000.
29. MapQuest. site: http://www.mapquest.com/.
30. James Mayfield, Paul McNamee, and Christine Piatko. The jhu/apl haircut system at trec-8. *The Eighth Text Retrieval Conference (TREC-8)*, pages 445–452, November 1999.
31. McCallum and Andrew Kachites. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering, 1996. site: http://www.cs.cmu.edu/ mccallum/bow.
32. Sheila A. McIlraith, Tran Cao Son, and Honglei Zeng. Semantic web services. *IEEE Intelligent Systems*, 16(2), March/April 2001.
33. Microsoft. Outlook. site: http://www.microsoft.com/office/outlook/.
34. Netscape. Rdf site summary (rss). http://my.netscape.com/publish/formats/rss-spec-0.91.
35. Olga Ratsimor, Vladimir Korolev, Anupam Joshi, and Timothy Finin. Agents2go: An infrastructure for location-dependent service discovery in the mobile electronic commerce environment. *ACM Mobile Commerce Workshop*, July 2001.
36. S. Staab, M. Erdmann, and A. Maedche. Ontologies in rdf(s). *Linkping Electronic Articles in Computer and Information Science*, 6, 2001. ISSN 1401-9841.