

ITTALKS: A Case Study in the Semantic Web and DAML

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

Laboratory for Advanced Information Technology

University of Maryland Baltimore County

{cost,finin,ajoshi,ypeng,nicholas,ian,hchen4,lkagal1,fperic1,yzou1,stolia1}@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) [14, 23], 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, 36]. 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 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

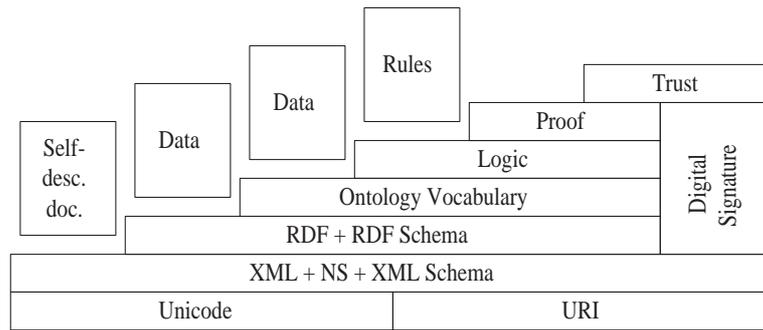


Figure 1: Tim Berners-Lee's vision of the Semantic Web is founded on a base that includes URIs, XML and RDF.

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 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 in the 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.

```

<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>

```

Figure 2: An example of DAML-encoded knowledge.

2 Background

The Semantic Web [4, 3] is a vision in which web pages are augmented with information and data that is expressed in a way that facilitates its understanding by machines. The current human-centered web is still largely encoded in HTML, which focuses largely on how text and images would be rendered for human viewing. Over the past few years we have seen a rapid increase in the use of XML as an alternative encoding, one that is intended primarily for machine processing. The machine which processes XML documents can be the end consumers of the information or they can be used to transform the information into a form appropriate for human understanding (e.g., as HTML, graphics, synthesized speech, etc.) As a representation language, XML provides essentially a mechanism to declare and use simple data structures and thus leave much to be desired as a language in which to express complex knowledge. Recent enhancements to basic XML, such as XML Schema, address some of the shortcomings, but still do not result in an adequate language for representing and reasoning about the kind of knowledge essential to realizing the Semantic Web vision.

RDF (Resource Description Framework) [40] and RDFS (RDF Schema) [41] attempt to address these deficiencies by building on top of XML. They provide representation frameworks that are roughly the equivalent to semantic networks in the case of RDF and very simple frame languages in the case of RDFS. However, RDFS is still quite limited as a knowledge representation language, lacking support for variables, general quantification, rules, etc. DAML is one attempt to build on XML, RDF and RDFS and produce a language that is well suited for building the Semantic Web.

The goal of the DAML program (<http://www.daml.org/>), which officially began in August 2000, is to develop a universal Semantic Web markup language that is sufficiently rich to support intelligent agents and other applications. DAML can dramatically improve traditional ad hoc information retrieval because its semantics will improve the quality of retrieval results. Also, it will allow the intelligent agents of tomorrow to retrieve and manipulate the information on the semantic web.

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 5.5, 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.

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.

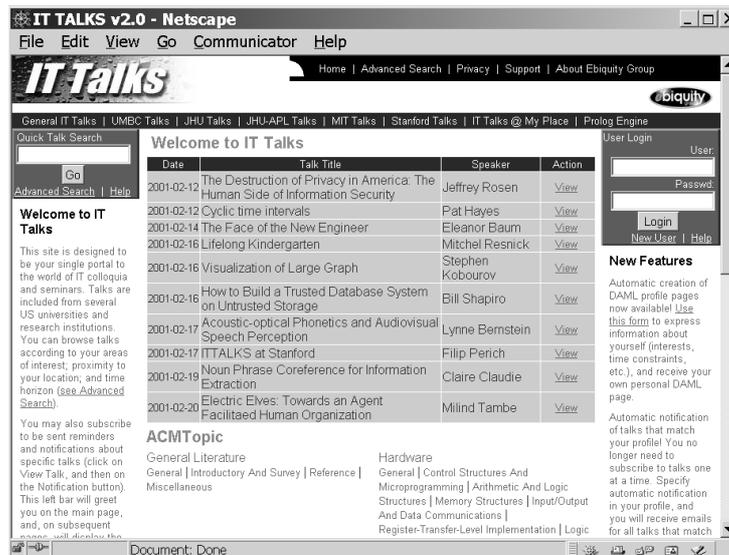


Figure 3: A screenshot depicting the main page of the ITTALKS system.

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.

3.3.1 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 [42] format, which supports WAP enabled phones. The ITTALKS web portal also has the ability to generate RDF Site Summary (RSS) [37] 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.

3.3.2 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 specifica-

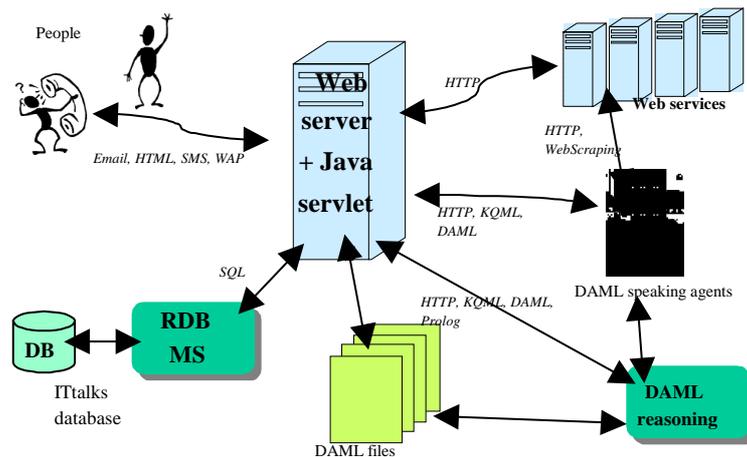


Figure 4: The architecture for ITTALKS is built around a web server backed by a relational database. Interfaces are provided to human users, software agents and web services.

tion of message requests, and for blocking or non-blocking message retrieval. Currently, it facilitates the use of KQML agent communication language [18] and employs a sophisticated protocol for agent naming, addressing and identity (KNS). Additionally, it is in the process of adapting to the FIPA standards [19, 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.

3.4.1 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.

3.4.2 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.

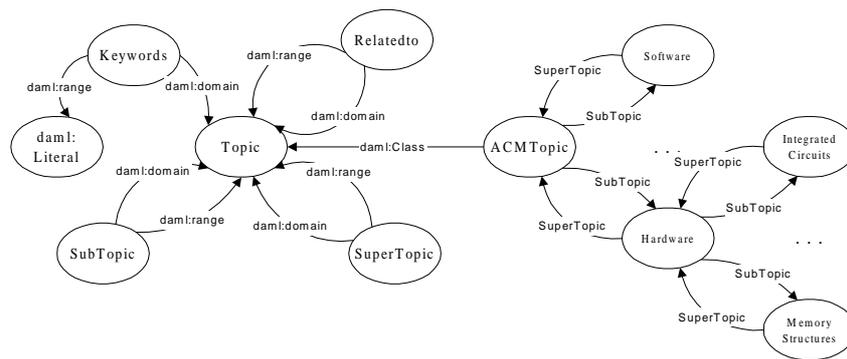


Figure 5: The Ontologies used by ITTALKS are relatively simple, such as the topics ontology used to describe talk topics and user interests.

3.4.3 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.

3.4.4 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 [33], 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 ¹ that are used to describe talks and the things associated with them, e.g., people, places, topics and interests, schedules, etc. Figure 6 shows some of the dependencies that exist among these ontologies. The ontologies are used in the representation and processing of DAML descriptions and 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 5 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

¹See <http://daml.umbc.edu/ontologies>.

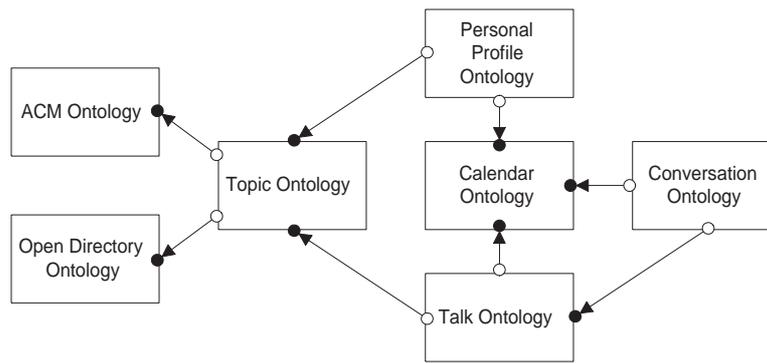


Figure 6: The relationships among the various ontologies used by the ITTALKS system.

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 component that can map topics in one ontology into topics in another, by taking advantage of the fact that nodes in each ontology have an associated collection of text as well as DAML information.

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.

3.7.1 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.

3.7.2 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.

3.7.3 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 a Stanford ITTALKS domain (`stanford.ittalks.org`), 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 preferences, which have been obtained from his online, DAML-encoded profile and from information

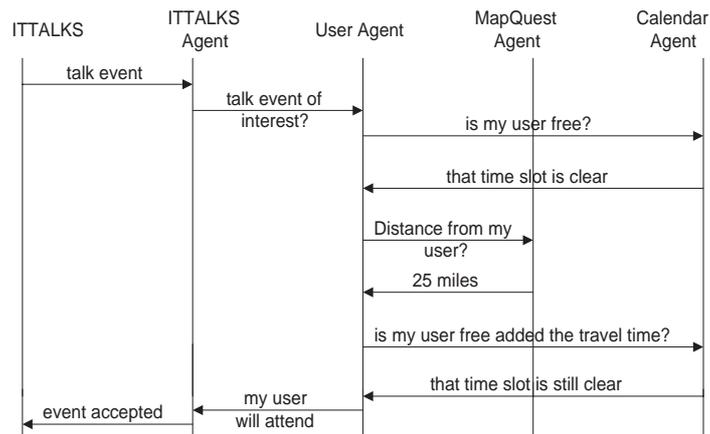


Figure 7: Interactions between the various agents described in the ITTALKS/Agent scenario.

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 [8, 34, 9], 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.

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 from his/her perspective, two or more topics are equivalent, related, 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, be they 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 [26] 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 *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* [25, 31, 32]. 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.

Authorization in a distributed system is quite different from that in a centralized system. Various schemes for decentralized security have been suggested, like Access Control Lists, Role based Access Control [38, 24], PolicyMaker [6, 5], etc. Although the above mentioned mechanisms are powerful, individually they are unable to meet all the requirements of trust management. Generally, security systems should not only authenticate users, but also allow them to delegate their rights and beliefs to other users securely, and have a flexible mechanism for this delegation. Most schemes either support only authentication, ignoring delegation altogether, or they support delegation to some extent without providing the required flexibility, or they provide insufficient restrictions on delegation of rights.

We have tried to solve this problem through the application of a chain of trust, using rights and delegations. In this system, we model permissions as the rights of an agent and associate rights with actions, so that possession of a right permits the corresponding agent to perform a certain action. These permissions can be extended by delegation from an authorized agent. We are also working with obligations, entitlements, and prohibitions and the delegation of these propositions.

Our work on distributed trust represented actions, privileges, delegations and security policy as horn clauses encoded in Prolog. In order to develop an approach that is better suited to sharing information in an open environment, we are recasting this work in DAML. We have defined an initial ontology² that covers the basic concepts including actions, agents, roles, privileges, prohibitions, obligations, security policies and other key classes and their properties. In applying our framework, one must extend the initial ontology by defining domain specific class of actions, permissions, etc. and creating appropriate individuals. A simple example of a delegation is given in Figure 8.

We hope to use this approach to distributed trust in ITTALKS to express the security policies which govern who can create, delete and edit talk announcements and who can further

²<http://daml.umbc.edu/ontologies/trust-ont.daml>

```

<rdf:RDF
  xmlns:rdf      = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs    = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns:daml    = "http://www.daml.org/2001/03/daml+oil#"
  xmlns         = "http://daml.umbc.edu/ontologies/trust-ont#" >

  <delegation rdf:ID="Delegation1">
    <from>susan-agent</from>
    <to>marty-agent</to>
    <permission>
      <from>susan-agent</from>
      <to>marty-agent</to>
      <starttime>2001:8:1:10:00</starttime>
      <endtime>2001:8:5:24:00</endtime>
      <readfileaccess>
        <name>ReadFileAccess</name>
        <description >
          Accessing a file in read mode
        </description>
        <actor>umbc-agent</actor>
        <objects>susan-files</objects>
        <redelegatable>
          <permission>
            <readfileaccess>
              <actor>umbc-agent</actor>
              <objects>file123.txt</objects>
            </readfileaccess>
          </permission>
        </redelegatable>
        <precondition>
          <request>
            <readfileaccess>
              <objects>susan-files</objects>
            </readfileaccess>
          </request>
        </precondition>
      </readfileaccess>
    </permission>
  </delegation>

```

Figure 8: This DAML expression represents a delegation from susan to marty, allowing marty to access in read mode all susan's files marty is also given the ability to redelegate readfileaccess to file123.txt to all agents from UMBC.

redelegate these privileges to others. The host ITTALKS system might have a base policy that specified, for example, the initial privileges and obligations that the *root* user of each ITTALKS domain would have and any constraints on their delegation (e.g., that privileges could only be delegated to registered ITTALKS users). One of the privileges that a root user could have would be to extend the security policy for his domain through delegations. As root of the *umbc.ittalks.org* domain, I might delegate to all department faculty the right to create new talk announcements and the right to re-delegate this privileges to individual graduate students. I might further delegate the right to edit or delete a talk announcement to the agent who initially created it.

5.3 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 is already present in the knowledge base.

5.3.1 Entering Talks

In order to make ITTALKS successful, we need to make it as easy as possible for new talk descriptions to be entered into the system. We are addressing this problem using three complementary 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.

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

5.3.3 Text Classification

In order for ITTALKS to filter talk announcements on topic matches, it needs 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 than 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 [35] 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.

5.3.4 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.

5.3.5 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 [16, 11] mostly in the context of using it as a machine learning application. We are developing an information extraction tool using the AeroText [1, 10] 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.

5.4 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.

5.5 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.

5.6 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 a priori-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.

5.7 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.

6 Current Status/Observations

We have currently implemented the web site and display normal HTML with embedded DAML. There is an option for viewing only DAML content for a certain page, talk, or user. All requests are made via HTTP. We also provide a form-based interface to add/modify the database, including talks and users. We have a two level moderation, with the root being the highest. The root can delegate rights to certain users, making them editors and allowing them to edit a particular domain. We also offer tools to generate personal profiles. We allow users to filter talks by interest and location. We also have a MapQuest agent that calculates the distance between a user’s location and a talk.

7 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 the 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.

Currently we use only HTTP, but plan to move to Jackal for agent communication. Jackal currently supports KQML and we are in the process of adapting it to the FIPA standards. In addition, our research group, in cooperation with other universities, is developing a DAML ontology for the necessary conversation protocols.

In order to make the process of data entry more efficient, we are developing a focused web spider, which will collect talk announcements from open sources 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.

8 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.

DAML is intended to significantly enhance the usability of web content in a number of ways. It is expected that such an advance will have some cost; with DAML, this is largely in complexity. DAML-marked text is difficult for human users to read or construct. Unlike languages like HTML, which, with some small learning curve, can be used by hand, DAML requires the use of mechanized assistance. We encountered this in a number of places in the construction of ITTALKS; in the need for a classifier and other tools for using DAML-marked ontologies, in the automatic construction of a DAML user profile, in the hiding of DAML markup in general from the user in the presentation of data.

This first point leads to a second problem, which is the duplicate representation of data in multiple modes. Using a simpler markup-language, it is possible to use a single, common representation which may be used by both humans and machines. With a more complex markup such as DAML, there are good reasons to prefer the separation of data into different, paired documents. While this has some advantages, it leads to greater problems of synchronization. Note that presently, ITTALKS uses an internal relation database representation, and markup is applied to output data, so synchronization in this case is not a problem. Were multiple modes used, however, it would still be necessary for a user referencing one document to be able to identify and acquire the corresponding alternate documents.

These aspects of DAML are a necessary result of its increased expressive power. It is not necessarily the case that simpler is better - in fact, the reason DAML has been advanced is that existing markup frameworks are not sufficient to support the kind of sophisticated use of information on the web that we would like. But they are difficulties that must be addressed in order to encourage DAML's widespread acceptance.

Another point to bear in mind is that DAML's success is largely dependent on the success of efforts at constructing, representing, and merging/reconciling ontologies, something which is not always immediately obvious. Although many systems can easily make use of a language such as DAML internally in a number of ways (e.g. for representation, communication), their integration with systems in the community at large still depends on advances in our use of ontologies.

9 Acknowledgments

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/>).

References

- [1] AeroText. AeroText web 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] M. Blaze, J. Feigenbaum, A. Keromytis, and J. Ioannidis. The keynote trust-management system, 1998.
- [6] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. *IEEE Proceedings of the 17th Symposium*, 1996.
- [7] Matt Blaze, Joan Feigenbaum, and Angelos D. Keromytis. The role of trust management in distributed systems security. In *Secure Internet Programming*, pages 185–210, 1999.
- [8] Kurt D. Bollacker, Steve Lawrence, and C. Lee Giles. Citeseer: An autonomous web agent for automatic retrieval and identification of interesting publications. In *Proceedings of the Second International Conference on Autonomous Agents (Agents '98)*, Minneapolis, 1998. ACM Press.
- [9] 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.
- [10] Lois C. Childs. AeroText - a customizable information extraction system. Technical report, Lockheed Martin, 2001. Unpublished.
- [11] Fabio Ciravegna. Learning to tag for information extraction from text. In Fabio Ciravegna and Roberto Basili, editors, *Proceedings of the ECAI Workshop on Machine Learning for Information Extraction, in conjunction with ECAI 2000*, Berlin, August 2000.
- [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. In Jeremy Baxter and Chairs Brian Logan, editors, *Working Notes of the Workshop on Tools for Developing Agents, AAI '98*, number WS-98-10 in AAI Technical Reports, pages 73–82, Minneapolis, Minnesota, July 1998. AAI, AAI Press.
- [13] Neal Coulter. Computing classification system 1998: Current status and future maintenance, report of the ccs update committee. *Computing Reviews*, January 1998.
- [14] DARPA. DARPA agent markup language website. <http://www.daml.org/>.
- [15] Stefan Decker, Frank van Harmelen, Jeen Broekstra, Michael Erdmann, Dieter Fensel, Ian Horrocks, Michel Klein, and Sergey Melnik. The semantic web - on the roles of XML and RDF. *IEEE Internet Computing*, September/October 2000.
- [16] T. Eliassi-Rad and J. Shavlik. Instructable and adaptive web-agents that learn to retrieve and extract information. Technical Report 2000-1, Machine Learning Research Group, Department of Computer Sciences, University of Wisconsin, 2000.
- [17] D. Fensel et al. OIL in a nutshell. In R. Dieng et al., editor, *Knowledge Acquisition, Modeling and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*, Lecture Notes in Artificial Intelligence. Springer-Verlag, October 2000.
- [18] Tim Finin, Yannis Labrou, and James Mayfield. KQML as an agent communication language. In Jeff Bradshaw, editor, *Software Agents*. MIT Press, 1997.

- [19] FIPA. FIPA 97 specification part 2: Agent communication language. Technical report, FIPA - Foundation for Intelligent Physical Agents, October 1997.
- [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 webs. *Linköping Electronic Articles in Computer and Information Science*, 6, 2001. <http://www.ep.liu.se/ea/cis/1997/013/>.
- [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] Herzberg, Mass, Mihaeli, Naor, and Ravid. Access control meets public key infrastructure, or: Assigning roles to strangers. In *RSP: 21th IEEE Computer Society Symposium on Research in Security and Privacy*, 2000.
- [25] Lalana Kagal, Harry Chen, Scott Cost, Timothy Finin, and Yun Peng. An infrastructure for distributed trust management. In *Working Notes of the Second Workshop on Norms and Institutions in Multi-Agent Systems, Fifth International Conference on Autonomous Agents (Agents '01)*, Montreal, Quebec, Canada, May 29 2001.
- [26] Lalana Kagal, Vlad Korolev, Harry Chen, Anupam Joshi, and Timothy Finin. Centaurus: A framework for intelligent services in a mobile environment. In *Proceedings of the International Workshop on Smart Appliances and Wearable Computing (IWSAWC)*, April 2001.
- [27] Robert Kass and Tim Finin. *Architectures for intelligent interfaces: Elements and prototypes*. 1996.
- [28] Craig A. Knoblock, Kristina Lerman, Steven Minton, and Ion Muslea. Accurately and reliably extracting data from the web: A machine learning approach. *Data Engineering Bulletin*.
- [29] Ora Lassila. Web metadata: A matter of semantics. *IEEE Internet Computing*, 2(4):30–37, 1998.
- [30] Ora Lassila and Deborah McGuinness. The role of frame-based representation on the semantic web. *Linköping Electronic Articles in Computer and Information Science*, 6, 2001. <http://www.ep.liu.se/ea/cis/1997/013/>.
- [31] Ninghui Li, Joan Feigenbaum, and Benjamin Grosf. 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.
- [32] Ninghui Li and Benjamin Grosf. A practically implementable and tractable delegation logic. *IEEE Symposium on Security and Privacy*, May 2000.
- [33] MapQuest. MapQuest website. <http://www.mapquest.com/>.
- [34] 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.
- [35] Andrew Kachites McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering, 1996. <http://www.cs.cmu.edu/mccallum/bow/>.
- [36] Sheila A. McIlraith, Tran Cao Son, and Honglei Zeng. Semantic web services. *IEEE Intelligent Systems*, 16(2), March/April 2001.
- [37] Netscape. RSS website. <http://my.netscape.com/publish/formats/rss-spec-0.91>.
- [38] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 20(2):38–47, 1996.
- [39] S. Staab, M. Erdmann, and A. Maedche. Ontologies in RDF(S). *Linköping Electronic Articles in Computer and Information Science*, 6, 2001. <http://www.ep.liu.se/ea/cis/1997/013/>.
- [40] W3C. RDF website. <http://www.w3c.org/RDF>.
- [41] W3C. RDFS website. <http://www.w3c.org/TR/rdf-schema>.
- [42] WAP Forum. WML spec. <http://www1.wapforum.org/tech/documents/SPEC-WML-19991104.pdf>.