

ITTalks Event Notification Service

An illustrative case for services in the Agentcities Network

Sovrin Tolia
University of Maryland
Baltimore County
stolia1@cs.umbc.edu

Deepali Khushraj
University of Maryland
Baltimore County
deepali1@cs.umbc.edu

Timothy Finin
University of Maryland
Baltimore County
finin@cs.umbc.edu

ABSTRACT

Agentcities is a worldwide network of agent platforms aimed at providing an open, heterogeneous and scalable infrastructure for service provisioning. To realize this vision, a host of research issues need to be tackled. We believe that the application of semantic web concepts, for service description, service composition and service availability, are some of the key ones for which feasible solutions need to be developed. To gain insights into the additional issues posed by such environments, we are building an event information service for ITTalks. ITTalks is a web-based system for automatic and intelligent notification of information technology talks, seminars and colloquia. It illustrates some of the ways in which DAML+OIL extends its ability to provide intelligent services to humans and the agents assisting them. In the context of the Agentcities initiative, our work encompasses building service description, composition and invocation ontologies using DAML-S. We are extending the JADE directory facilitator to provide a more robust rule-based partial search mechanism based on DAML-S service descriptions.

General Terms

Agentcities, Service Discovery, Service Composition, Service Availability

Keywords

Semantics, Ontologies, Applications, Services, DAML-S

1. INTRODUCTION

Agentcities[12] is a network of agent platforms aimed at providing an open, autonomous, distributed and heterogeneous backbone infrastructure for service providers to host their services. The goal is to deploy a large number of services interoperating with one another and facilitating composition of compound services from a host of component agent services. This network of services provided by Agentcities along with the semantic web[10] will serve as a backbone information infrastructure for agents. To realize this vision,

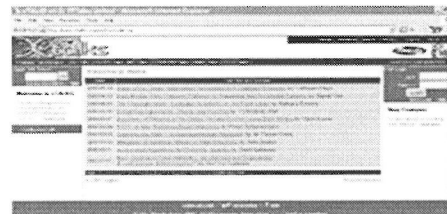


Figure 1: Snapshot of ITTalks

a host of research issues need to be tackled and we believe that the application of semantic web concepts, for service description, service composition and service availability, are some of the key ones for which feasible solutions need to be developed. In this paper, we share some insights on these issues and present our initial work in prototyping ITTalks[7] Event Notification Service for the Agentcities Network.

ITTalks¹ is a vertical portal that offers access to information about talks, seminars and colloquia. It employs DAML+OIL[2] for knowledge-base representation, reasoning and agent communication. The web view gives users event information customized to their personal profiles(expressed in DAML+OIL). The use of DAML+OIL to represent information in it's knowledge-base, enables sophisticated reasoning by software agents. Figure-1 gives a snapshot of the web interface for the ITTalks system.

As a part of our prototype, we have built service description, composition and invocation ontologies using DAML-S. Here we present our initial experiences with DAML-S and it's suitability for Agentcities type of environments. To enable a more robust search for services in the agent platforms, we are enhancing the directory facilitator in JADE to provide a rule-based partial mechanism for searching services.

In the next section we describe the effectiveness of DAML-S for service description and present some issues on service composition. In section three we discuss a few ways to increase service availability. In Section four we outline our initial design and implementation and section five concludes the paper.

¹Access ITTalks at <http://www.ittalks.org/>

2. SERVICE DESCRIPTION AND COMPOSITION

For services to effectively discover each other, expressive service descriptions are required. Moreover, these descriptions should be represented in the form that is understood by other agents in the network. To facilitate a more robust service negotiation between agents, it is important to describe these services using standardized tools and ontologies having constructs for rich expressibility. DAML-S[1] is one such ontology defining concepts to describe services and their invocation models.

DAML-S provides constructs enabling automated service discovery, execution, composition and interoperation. It consists of three parts. First, the Service Profile is used to describe what a service provides. Second, the Service Model describes service parameters (including inputs, outputs, preconditions and side effects) and the execution flow. Third, the Service Grounding is used to describe how it is accessed. We have designed a service profile and service model ontology for ITTalks Event Notification Service, Language Identification Service and Research Paper Recommendation Service.

Based on our design experience, we find that having a standard ontology govern the service description process reduces the problem of interoperation when it comes to service discovery. The process model simplifies the task of specifying different types of inputs and outputs associated with a service. Its representation in DAML+OIL enables sophisticated reasoning for effective matchmaking.

To invoke the services that were discovered, it is vital to know the underlying communication protocol. This can become complex in a heterogeneous environment like Agentcities and to simplify it, the services need to express the communication protocols they support. Although Service Grounding in DAML-S envisions this, no concepts have been defined yet. As the services become more complex (for example, composite service consisting of many simple services), both the service requester and the provider would like to monitor the state of execution. As the network of such services grows, it will be important to automate the monitoring of service execution. DAML-S does introduce the notion of execution state monitoring, but does not define concepts for the same.

The service profile in DAML-S defines various concepts that can be used to describe the attributes of the service. However, it does not allow services to declare those attributes on the basis of which it would like to be searched. There is a need for additional constructs like keywords, classification that enable searching for services on the Agentcities network easier. For example, when an agent wants to search for a weather service, there is no standard concept in DAML-S that enables this agent to specify this. Implementation of such search mechanisms is left to the designer. Some implementations might search this string against service type while others might search this against service description text.

The invocation process in services might not be as simple as a request-reply paradigm. More sophisticated interac-

tions like contract-nets are possible, in which the requester of the service and the service provider will involve in a multi-stage conversation. This might require such interactions for service negotiation to be modelled in DAML-S. This definition will become complex, as there will be multiple rounds of messaging requiring maintenance of state information at the service endpoints. It seems to us that initial draft of DAML-S does introduce interesting concepts about services and their process models, however, a complete specification of concepts is required for service agents to use it during service negotiation.

Service composition refers to the process of forming compound services from component agent services. It facilitates answering queries that ordinarily cannot be answered by individual services. The problem of service composition has been studied well in the literature [5, 9, 11] but in the context of the Agentcities network, there are new challenges that arise. These stem from the fact that in forming compound services, the component agent services would be adhering to different ontologies making the planning phase of the composition complex. These would be built using diverse technologies and hosted on different agent platforms, making the execution phase difficult.

Another key design issue is deciding on the entity that would be doing this service composition. One approach is to have dedicated agents doing service composition. However, it is not clear as to how these agents would describe their services and do composition independent of the application domain. A more scalable approach is the one in which several component service agents do service composition. For example, the ITTalks agent on Agentcities network might receive a query like, "Are there any talks in the language Chinese next month?" It knows that it can provide talks happening next month, but it cannot provide the languages of the talks. In that case, it might initiate a composition process involving a search for an agent that provides languages given some text information.

DAML-S defines concepts to model a composite service comprising of simple or composite services. We have built a simple example of a composite service called "SearchTalkLanguage" that is comprised of simple services called "SearchLanguage" and "SearchTalk". The description of this service is provided in the ontology².

3. SERVICE AVAILABILITY

For the Agentcities network to serve as a backbone services infrastructure, it is important to have a large number of diverse services. Not having a large number of diverse services in the network would really diminish the efficacy of the infrastructure. We envision that the Agentcities network should not be restricted to just agent-based services. To leverage the advantage of a large number of services already present on the web, designs have to be developed to incorporate information about such web services in the network of platforms. For example, UMBC has developed a server-side cache-enabled apache module called Damlator³

²Ontology can be accessed at <http://agentcities.umbc.edu/ontologies/>

³Access Damlator at <http://daml.umbc.edu/downloads/>

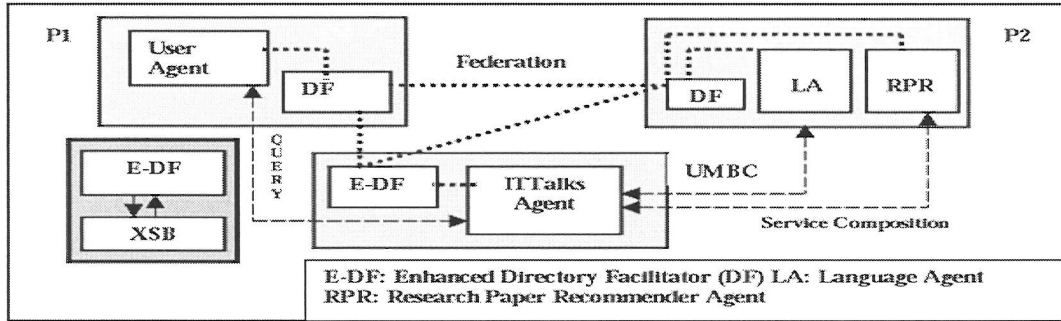


Figure 2: Architectural View of Current Prototype.

that supports conversion of DAML+OIL into different formats. Web services like Damlator fall into the class of Helper Services and are proving to be useful for other domain services. This is just one facet of making the Agentcities network really resourceful for finding relevant services.

Increasing the availability of already hosted services is another facet. By availability we don't mean the "up-time" of services, but ensuring that relevant services do figure in various searches. One way to increase the availability is to have a federation of directory facilitators cooperating with each other to redirect search requests. Realizing such a federation would become complex in a network where directory facilitators are built using different technologies. Another way to increase the availability of services is by improving the search mechanism in each of these directory facilitators. The enhanced search mechanism should provide results based on partial matches.

For example, the directory facilitator in JADE[4] does a complete object-based search for services. This imposes an inherent constraint of not allowing partial matches. Moreover, the registration process requires such services to be described by name-value pairs denoting its properties. If service descriptions are expressed in DAML-S or other similar rich markup, this attribute-value pair mechanism for service registration will not be very effective. We envision more powerful, rule-based partial search enabled directory facilitators in the Agentcities network enabling a more robust search for services. The directory facilitators need to evolve themselves from time to time to support various service description technologies that emerge as a result of standardization. In the implementation section, we outline our preliminary architecture for such a directory facilitator. Similar work on semantic service discovery has been done in the context of the Jini service discovery system[6].

4. DESIGN

We are prototyping an ITTalks Event Notification Service to be hosted on UMBC's Agentcities platform. It provides descriptions about Information Technology talks based on personal profiles, time-period and location. It accepts queries and provides results that are expressed in DAML-S using its process model. The agent implementing this service serves as a gateway to the ITTalks system. It transforms service requests expressed in agent communication language(ACL) into servlet-driven requests understood by ITTalks.

For investigating the issues in service composition, we are building a Language Agent Service and Research Paper Recommender Agent Service that represent atomic services in the composition process. These services are expected to be hosted on "other" platforms in the Agentcities network. Both these services accept queries and return results marked-up in DAML-S. The use of DAML-S in returning results is to enable the requestor agent to identify the outputs. The Language Agent service is an agent wrapper over the language identification system built at Georgetown University[8]. It is able to identify languages based on text descriptions and provide the associated confidence measures. The research paper recommender agent service suggests publications relevant to a given abstract or a title. It uses the NEC ResearchIndex[3] scientific digital library to make those recommendations.

To enable partial matches for service discoveries, we are enhancing the directory facilitator in JADE. The enhanced directory facilitator accepts service descriptions expressed in DAML-S during registration and allows agents to search for services based on partial service information.

4.1 Motivating Scenario

John would like to know all the talks presented in the language Chinese that are happening next month in UMBC. Also, he wants research papers relevant to these talks.

ITTalks as a system does not have all the information that such a query needs. However, if the query is analyzed, there are three atomic services that are actually required. Information Technology talks that will take place next month are provided by the ITTalks agent, a service that is able to provide a language given the text description and a service that is able to provide relevant research papers. This scenario illustrates the need for having expressive service descriptions for automated discovery and execution and the need for forming compound services based on atomic services.

4.2 Initial Prototype

The component agent services required by the scenario are described using DAML-S. Figure-3 gives a snippet of the ontology describing the ITTalks Event Notification Service. In our initial prototype, service composition is carried out by the ITTalks agent once it receives the query from the

```

<service:ServiceProfile rdf:ID="Profile_ITTalks-On-AgentCities">
  <service:isPresentedBy rdf:resource="xtag:ITTalks-AgentCities-Agent"/>
  <profile:serviceName>xTalks Agent on Agentcities</profile:serviceName>
  <profile:textDescription>Search IT Talks.</profile:textDescription>
  <profile:providedBy>
    <profile:ServiceProvider rdf:ID="UNBC">
      <profile:name>DAML Research Group @ UNBC</profile:name>
      <profile:webURL>http://agentcities.umbc.edu</profile:webURL>
    </profile:ServiceProvider>
  </profile:providedBy>
  <profile:geographicRadius rdf:resource="http://daml.umbc.edu/ontologies/place-ont.daml#Place"/>
  <profile:qualityRating rdf:resource="http://www.w3.org/2000/10/XMLSchema#string"/>
  <profile:has_process rdf:resource="http://daml.umbc.edu/ontologies/xtalks-agentcities.daml#SearchTalk"/>
  <process:input>
    <profile:ParameterDescription rdf:ID="Search-Based-On">
      <profile:parameterName>Search Based On</profile:parameterName>
      <profile:restrictTo rdf:resource="http://daml.umbc.edu/ontologies/xtalks-agentcities.daml#Search-Types"/>
      <profile:referTo rdf:resource="http://daml.umbc.edu/ontologies/xtalks-agentcities.daml#Based-On"/>
    </profile:ParameterDescription>
  </process:input>
  <process:input>
    <profile:ParameterDescription rdf:ID="Number-Of-Results">
      <profile:parameterName>Number of Results</profile:parameterName>
      <profile:restrictTo rdf:resource="http://www.w3.org/2000/10/XMLSchema#integer"/>
      <profile:referTo rdf:resource="http://daml.umbc.edu/ontologies/xtalks-agentcities.daml#Num-Results"/>
    </profile:ParameterDescription>
  </process:input>
  <process:input>
    <profile:ParameterDescription rdf:ID="Logical-Connecting-Operators">
      <profile:parameterName>Logical Operator Connectors</profile:parameterName>
      <profile:restrictTo rdf:resource="http://daml.umbc.edu/ontologies/xtalks-agentcities.daml#Operators"/>
      <profile:referTo rdf:resource="http://daml.umbc.edu/ontologies/xtalks-agentcities.daml#Logical-Operators"/>
    </profile:ParameterDescription>
  </process:input>
  <process:output>
    <profile:ParameterDescription rdf:ID="Search-Results">
      <profile:parameterName>SearchResults</profile:parameterName>
      <profile:restrictTo rdf:resource="http://daml.umbc.edu/ontologies/xtalks-agentcities.daml#Talk-Profile"/>
      <profile:referTo rdf:resource="http://daml.umbc.edu/ontologies/xtalks-agentcities.daml#Talk-Description"/>
    </profile:ParameterDescription>
  </process:output>
</service:ServiceProfile>

```

Figure 3: DAML-S Profile of ITTalks Search Service

user agent. Figure-2 provides an architectural view of our system along with an interaction flow.

When a user poses such a query, the useragent queries the directory facilitator of it's agent platform. The federation of directory facilitators enables the useragent to know that ITTalks agent would be able to answer this query(enabled by the way of ITTalks agent publishing the capability to do service composition). The useragent now queries the ITTalks Agent, which in turn does the service discovery and invocation of the language and research paper identification services. After carrying out this composition, it returns the results back to the useragent. In our prototype, the ITTalks agent was aware of the plan for doing service composition, but did not know the component agents who would be providing the required services.

JADE is being used for the implementation of these service agents. We are enhancing the directory facilitator in JADE to include an XSB based reasoning engine for supporting partial searches for services. The agents register their services with the directory facilitator with a DAML-S service profile. Other agents wishing to discover services will provide a partial/complete DAML-S description of the service attributes. A rule-based search is carried out to determine the matching services and the degree of match.

5. CONCLUSION

Currently our prototype makes agent services available only across homogeneous platforms. In future, when agents from different agent platforms need to communicate with one another, the heterogeneity of communication protocols and agent frameworks will pose a big challenge in making them interoperate.

The agents in our prototype shared a common ontology. However, in Agentcities, this will not always be the case. We believe that Ontology Mapping Services would be very useful to partially solve this problem.

On one side "we view Agentcities as a propelling force to bring the agent technology from research laboratories into production environments so as to provide useful information and services to end-users". On the other side "it would serve as a key backbone to the global agent-based service-centric computing infrastructure". However, there are numerous challenges that need to be addressed before this vision becomes a reality. In this paper we have expressed our views and outlined the work in progress on service description, composition and availability in the context of open environments like Agentcities.

6. ACKNOWLEDGMENTS

We acknowledge all the people involved with the Agentcities initiative for expressing their ideas and bringing out new research directions. More information about Agentcities can be had from <http://www.agentcities.org>

7. REFERENCES

- [1] Daml-s: Semantic markup for web services.
- [2] The darpa agent markup language. <http://www.daml.org/>.
- [3] Nec researchindex scientific digital library. <http://citeseer.nj.nec.com/>.
- [4] F. Bellifemine, A. Poggi, and G. Rimassa. Developing multi agent systems with a fipacompliant agent framework. *Software – Practice And Experience*, pages 103–128, 2001.
- [5] F. Casati and S. I. et al. Adaptive and dynamic service composition in eflow. Technical Report, HPL-200039, Software Technology Laboratory, Palo Alto, CA, March 2000.
- [6] D. Chakraborty and F. P. et al. An agent discovery architecture using ronin and dreggie. In *First GSFC/JPL Workshop on Radical Agent Concepts (WRAC)*, 2002.
- [7] R. Cost, T. Finin, A. Joshi, Y. Peng, C. Nicholas, I. Soboroff, H. Chen, L. Kagal, F. Perich, Y. Zou, and S. Tolia. Ittalks: A case study in the semantic web and daml+oil. *IEEE Intelligent Systems Special Issue*, 17(1):40–47, January/February 2002.
- [8] S. Huffman. The genetic classification of languages by n-gram analysis: A computational technique. <http://epsilon3.georgetown.edu/~cball/languageid/>.
- [9] R. Katz, E. A. Brewer, and Z. Mao. Fault-tolerant, scalable, wide-area internet service composition. Technical Report. UCB/CSD-1-1129. CS Division. EECS Department. UC. Berkeley, January 2001.
- [10] T. B. Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.
- [11] D. Mennie and B. Pagurek. An architecture to support dynamic composition of service components. Systems and Computer Engineering. Carleton University, Canada.
- [12] S. N. Willmott, J. Dale, B. Burg, C. Charlton, and P. O'brien. Agentcities: A worldwide open agent network. *Agentlink News*, (8):13–15, November 2001.