Advertising and Matching DAML-S Service Descriptions

Terry R. Payne, Massimo Paolucci, Katia Sycara

Robotics Institute, CMU
5000 Forbes Avenue
Pittsburgh, PA 15213 USA
+1 (412) 268 8780
[*terryp|paolucci|katia*]*@cs.cmu.edu*

ABSTRACT

Software agents and Web-based services extend beyond traditional Web activities (i.e. locating and browsing information) by allowing one to effect some action or change in the world, such as the sale of a product or the control of a physical device. Agents should be able to locate, select, employ, compose, and monitor Web-based services automatically. To do this, a computer-interpretable description of the service is required, as well as a means by which it can be located. This paper presents work in progress on the ATLAS matchmaker; a location service that facilitates the discovery of agents and services described using DAML-S. We outline how services are advertised, and describe how the ATLAS matchmaker identifies matching profiles for a given service request.

INTRODUCTION

As the number and variety of services on the Semantic Web increases, frameworks that allow services and agents to interoperate, seek and cooperate with each other will be required. Infrastructures developed within the MAS community provide such a framework, as they facilitate automatic discovery of agents and web-services on the basis of their capabilities. In this paper we leverage previous experience in developing yellow-pages registry services (e.g. LARKS [6]) to develop a mechanism for automated service discovery within the Semantic Web. As part of the DARPA Agent Markup Language program, an ontology of services[1], called DAML-S [1], has been developed, which provides a set of basic concepts and relations for declaring and describing services, by utilizing the ontology structuring mechanisms provided by DAML [3].

A DAML-S service is characterized by three types of knowledge; a *service-profile*, *process-model*, and *service-grounding*. The *service-profile* describes *what the service does*; it provides the type of information needed by a service requester to determine whether the service has the desired capabilities. The *process-model* describe *how the service works*; i.e. how it is composed, and what happens when the service is executed. A *service-grounding* specifies the details of *how an agent can access a service*. Generally speaking, the *profile* provides the information needed for an agent to discover a service. Taken together, the *process-model* and *grounding* describe how agents and services access, and interoperate with each other.

Service profiles consist of three types of information: a human readable *description* of the service; a specification of the *functionalities* that are provided by the service; and a list of *functional attributes* which provide additional information and requirements about the service that assist when reasoning about several services with similar capabilities. *Service functionalities* are represented as a transformation from the inputs required by the agent to the outputs produced. For example, a news reporting service would advertise itself as a service that, given a date, will return the news reported on that date. *Functional attributes* specify additional information about the service, such as what guarantees of response time or accuracy it provides, or the cost of the service. Table 1 lists the properties defined by the service profile (note that the properties *input* and *output* are defined as sub-properties of *parameter*).

A service provider can register, or advertise its profile with one or more *Middle Agents* [4]. There are several different ways in which middle agents interact with service providers and service requesters [5], depending on factors such as reliability, privacy, efficiency etc. A sub-class of middle agents are generally known as *matchmakers*, *yellow pages* or *directory agent* systems [2][4][5]. These only have knowledge about the capabilities of service providers. Thus, if an agent has some preferences, it can query the middle agent, which then returns a list of agents whose capabilities match the preference query. In contrast, service requesters have *preferences* for particular parameters associated with desired capabilities. Providers advertise their capabilities and service parameters with one or more middle agents, and requesters request agents with particular capabilities and select a provider according to their preferences.

SERVICE MATCHING

A variety of different approaches have been used to match agent advertisements and service requests. The LARKS matchmaker [6] consists of a number of filters, each of which performs partial matching on subsets of the descriptors. Several of the filters perform semantic matches, by determining a semantic distance between co-existent terms within shared ontologies. The ATLAS matchmaker utilizes two separate filters: one compares *Functional Attributes* to determine the applicability of advertisements (i.e. do they

[1] More details about the DAML-S Semantic Markup can be found at http://www.daml.org/services/

support the required type of service or deliver sufficient quality of service); the second compares *Service Functionalities* to determine if the advertised service matches the requested service. A DAML-based subsumption inferencing engine is used to determine if one concept subsumes another.

Description	***Service Functionalities***	***Functional Attributes***
serviceName *intendedPurpose* *textDescription* *role* *requestedBy* *providedBy*	*parameter* *input* *output* *conditionaloutput* *precondition* *accesscondition* *effect* *domainResource*	*geographicRadius* *degreeOfQuality* *serviceType* *serviceCategory* *serviceParameter* *communicationThru* *qualityRating* *qualityGuarantees*

Table 1: Properties of the Service Profile

Matching Functional Attributes

Matching is achieved by performing conjunctive pair-wise comparisons for the properties listed in Column 3 of Table 1. As each of the functional attributes may refer to different ontologies, different types of inference are used to test each pair. For example, *geographicRadius* comparisons may use a different mechanism to determine if the request falls within the geographic scope of the advertisement, to that used to compare *serviceTypes*. As property pairs are assumed to match comparisons cannot be performed (e.g. due to missing property values), the inclusion of property values within the request will provide a way of constraining the set of candidate advertisements that match the request.

Matching Service Functionalities

Different service providers may describe similar services in different ways, by labeling properties as inputs instead of preconditions, effects instead of outputs, etc. For this reason, service functionalities are grouped into two sets; an input set (consisting of the union of *input*, *precondition*, and *accesscondition* properties) and an output set (consisting of the union of *output*, *conditionaloutput*, and *effect* properties). The input and output sets are determined for the request (R_I and R_O) and for each advertisement (A_I and A_O). The input set R_I is compared with an advertisement input set, A_I, and a match is determined if $A_I \subseteq R_I$, i.e.

$$match(R_I, A_I) \Leftarrow (\forall j, \exists i: (i \in R_I) \wedge (j \in A_I) \wedge subs(i, j)) \wedge R_I = \emptyset$$

where $subs(i, j)$ is true when i subsumes j.

Thus, for each concept in the advertisement input set, there is some concept in the request input set that subsumes it. Thus, there is a match when a request can provide all necessary inputs required by an advertisement. However, an agent may not have a model of what inputs may be required, and may need to obtain this information from the advertisement. To support this, input sets also match when the request set is empty.

Output sets match when all the elements in R_O are subsumed by elements in A_O, i.e.

$$match(R_O, A_O) \Leftarrow \forall i, \exists j: (i \in R_O) \wedge (j \in A_O) \wedge subs(j, i)$$

To avoid the possibility of one element matching several others, a greedy approach will be used, whereby each matching pair of elements are removed from the input and output sets. A mapping of corresponding inputs and outputs will also be generated, and returned to the requesting agent if a match is determined.

CONCLUSION

The ATLAS matchmaker provides registration and lookup capabilities of web-based services described using the DAML-S service profiles. Service requests are compared with service advertisements through a subsumption based inferencing mechanism using DAML ontologies. Two separate filters are being developed: one for comparing functional attributes that characterize the service along descriptive dimensions (such as business types, rating mechanisms etc); the second compares the functional capabilities of the service in terms of inputs and outputs.

Several evaluation studies are planned for this system, to determine the effectiveness and speed of each of the filters, to evaluate the utility of Boolean subsumption inferencing for matching DAML concepts; and to evaluate the design of the DAML-S profile descriptions.

ACKNOWLEDGMENTS

This research was supported by a DARPA grant F30601-00-2-0592. Special thanks go to the DAML Services Coalition for their assistance in the development of DAML-S.

REFERENCES

1. Ankolekar, A., Burstein, M., Hobbs, J.R., Lassila, O., Martin, D.L., McIlraith, S.A., Narayanan, S., Paolucci, M., Payne, T.R., Sycara, K., and Zeng, H. DAML-S: Semantic Markup for Web Services. Submitted to International Semantic Web Workshop (SWWS), 2001
2. Genesereth, M., and Katchpel, S. Software Agents. *Communications of the ACM*, 37(7):48-53, 1994.
3. Hendler, J. and McGuinness, D. DARPA Agent Markup Language. *IEEE Intelligent Systems*, 15(6):72-73, 2001
4. Sycara K., Decker, K., Pannu A., Williamson M., and Zeng D. Distributed Intelligent Agents. *IEEE Expert: Intelligent Systems and their Applications*, Vol. 11, No. 6 (December 1996).
5. Sycara K., Decker, K., and Williamson M. Middle-agents for the Internet. In *Proceedings of IJCAI-97*, January 1997.
6. Sycara, K., Klusch, M., Widoff, S. and Lu, J. Dynamic Service Matchmaking Among Agents in Open Information Environments. *SIGMOD Record (ACM SIG on Management of Data)* Vol. 28 No. 1, pp47-53, 1999.