Bridging Semantic eGovernment Applications Using Ontology-to-Ontology Message Translation

Michael Czajkowski, Benjamin Ashpole, Todd Hughes, Tuong Le

Lockheed Martin Advanced Technology Laboratories
3 Executive Campus – 6th Floor
Cherry Hill, NJ 08002, USA
{mczajkow, bashpole, thughes, tle}@atl.lmco.com

Abstract

Developers have built many eGovernment applications using local ontologies to provide a meta-data description of what their service does, how it works, and how to invoke it. Every day more ontologies are written by different developers and posted on web servers around the world. Consequently, effective bridging of semantic web applications for eGovernment is challenging. This paper briefly describes a new technique for bridging disjoint semantic web applications by automatically aligning their ontologies and performing message translation. The challenge is to create a tool to increase efficiency of alignment without reducing accuracy. We demonstrate our solution in an eGovernment scenario of translating driver's license information between two services in two U.S. states.

Introduction

The service interoperability problem must be solved to realize the full potential of eGovernment. Describing an eGovernment application with meta-data tells what the service does, how it works, and how to invoke it (Martin 2005). Semantic applications have been written using homegrown ontologies created by their developers with hopes that that others will find their meta-data and relate to it. However, bridging heterogeneous ontological information is difficult and time consuming. The same problems of XML schemas are occurring for the semantic web: others have to conform to one's semantic descriptions by building their own knowledge translators. Progress has been made for tools to reduce the human cost of semantic linking. Search engines have been proposed (Nejdl 2004) and built (UMBC 2004), and graphical ontology editors have been developed (Kalyanpur et al. 2004). All of these are helpful to humans who must tediously encode ontological knowledge and their relationships to that of others. However, automatic ontology alignment is where the greatest potential lies. The next challenge is to build an automatic translator that converts RDF messages and enables heterogeneous semantic applications to communicate.

Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

To address this challenge, Lockheed Martin Advanced Technology Laboratories (ATL) developed an approach called message-to-message (M2M) translation. Based on automatic ontology alignment, M2M translates meta-data from one semantic application to another. This creates a bridge between applications so services can communicate despite each application using mutually incompatible languages.

Reducing the amount of time to integrate semantic eGovernment services is the chief benefit to the M2M approach. As such, a more tightly integrated set of services can be exploited to generate new capabilities for serving citizens.

This paper begins by discussing background work done at ATL on ontology alignment and composition of semantic web applications. Then, an approach to purpose-driven bridging of disjoint semantic web applications is described. Finally, a simple case example is presented where M2M is applied to an eGovernment scenario of translating driver's license information between two U.S. states.

Component Research

The basis of the M2M methodology relies upon two methologies: ontology alignment and the composition of semantic web application composition.

Ontology Alignment

ATL's Ontology Translation Protocol (Ontrapro) prototype tool automatically discovers semantic correspondences between elements in heterogeneous data models. Ontrapro uses dynamically composable alignment algorithms that compare a comprehensive range of features (e.g., syntactical, lexical, phonetic, and structural) between data models to identify semantic similarities. It uses a filtering mechanism that maximizes alignment precision and recall. In 2004, Ontrapro was showcased at the Information Interpretation and Integration Conference (I3CON) as part of Performance Metrics for Intelligent SystemsWorkshop (PerMIS). For a summary of alignment techniques, see (Shvaiko and Euzenat 2005).

Composing Semantic Web Applications

Developed by ATL, the Meta-Planning for Agent Composition (MPAC) (Czajkowski, Buczak, and Hofmann 2004) framework automatically composes and executes semantic web services written in OWL-S. MPAC dynamically selects semantic web services based on availability through match-making algorithms similar to those described in (Sycara et al. 2003). Once services are chosen, MPAC applies the Java Simple Hierarchical Planner (JSHOP) (Nau 2005) to find an execution sequence that solves given goals with provided input meta-data. After composition, MPAC executes each service with an agent using ATL's Extensible Mobile Agent Architecture (EMAA) (Hofmann et al. 2001). ATL's experience with agents has shown they are ideal for adaptive cognition, highly distributed systems, and execution monitoring.

Approach

Our approach to bridging semantic web applications employs both Ontrapro and MPAC technologies. MPAC can select, compose, and execute semantic eGovernment services of interest. However, MPAC presumes that the ontologies of each service are already linked. Invoking Ontrapro between the executions of two semantic web services in the MPAC agent removes this constraint. Ontrapro translates the inputs, outputs, preconditions, and effects (IOPEs) of the two services. This translation step bridges the two semantic applications that previously could not understand one another.

The approach is outlined as follows (see Figure 1):

1. Select two semantic eGovernment services to bridge together: *A* and *B*.
2. Obtain the set of ontologies used to describe *A* and *B*: ONT_A and ONT_B respectively.
3. Invoke the web service *A*, obtaining output set A_O, and effect set A_E.
4. Perform ontology alignment between all ONT_A and ONT_B , yielding an alignment file M_{AB}.
5. Using M_{AB}, translate the A_O and A_E described in ONT_A into terminology found in ONT_B, particularly the inputs and preconditions of *B:* B_I and B_P.
6. Invoke *B* with B_I and B_P.
7. Repeat the process until all goals are achieved.

First, MPAC selects two services to bridge. The selection of semantic web services *A* and *B* is usually goal driven, based on a client-given desire to achieve some output or effect (e.g., gain access to information reports, share security data). Given a repository of services, MPAC selects semantic applications *A* and *B* using match-making algorithms to meet these goals. Once *A* and *B* are chosen, MPAC uses the semantic web to gather the ontology sets ONT_A and ONT_B that describe *A* and *B* respectively. At this point an agent invokes the first semantic web service *A* given necessary inputs A_I and satisfied preconditions A_P. The result of invoking *A* yields output set A_O and effect set A_E, described in ONT_A.

At this point, Ontrapro builds an alignment file M_{AB} that maps ontologies ONT_A into terms found in ONT_B. Once M_{AB} has been created, Ontrapro translates output messages A_O and effects A_E into terms found in ONT_B. Message translation requires that subjects, predicates, and objects are aligned thoroughly at multiple levels. For M2M ontology alignment, subjects and objects are treated the same. Ontrapro looks at all subjects and objects in ONT_A and uses its heuristics to align them into ONT_B in various ways (e.g., equivalence, subclass, and superclass matching). Predicates are treated differently in the M2M approach because relationships may differ between ONT_A and ONT_B. Predicates are aligned by comparing their subject and object counterparts following symmetric and transitive relationship graphing. The output of translation is a pairing of all subjects, predicates, and objects between A_O , A_E to B_I, B_P. Given preconditions B_P, the service *B* can now be invoked.

Translation Metrics

Automated bridging through M2M translation is successful if *operational* goals can be met. However, semantic web

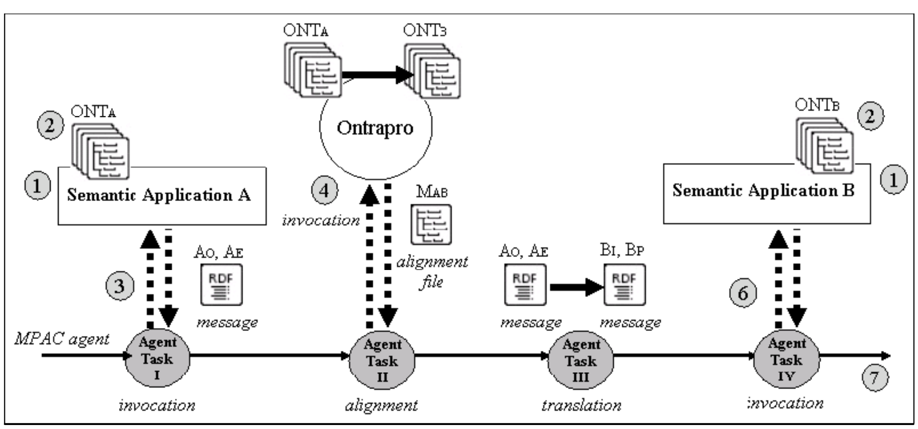

Figure 1. Message-to-Message Translation

research has primarily been focused on preservation of meaning through automatic alignment. For eGovernment, precise preservation is secondary to handling the message properly. M2M translation succeeds if service *B* handles the message with desirable outputs and effects. If *B* fails to understand the message or improperly processes it, M2M has failed.

From examining proper resulting behavior, we derive these metric guidelines:

- *Success versus Failure:* The ratio of service *B*'s successes versus its failures for an Ontrapro alignment heuristic H when translating messages from *A* to *B*.
- *Examining Resulting Behavior:* Compare multiple M2M translations of ONT_A to ONT_B using *H*. Examine the variety of service *B*'s behavior found in B_O, and B_E. For successes, determine if there is a varying level of success from *B*. Certain applications perform better with more accurate translations. For failures, determine which heuristics cause errors versus which cause improper processing.

Notional Use Case

We have developed a use case scenario to show how two web services can be combined to form one eGovernment application despite being described by different ontologies. The scenario contains several semantic applications that produce and consume information found on driver's licenses between U.S. states. The outcome is a constructed composite service, using OWL-S's control constructs (e.g., sequence, if-then-else, iterate) (Martin 2005).

In our scenario, the key constituents are:

- *Producer*: California DMV records service produces information on registered drivers. Input: The name of a citizen. Output: Driver's license information.
- *Consumer*: Nevada Gaming Commission checks the ages of minors attempting to gamble or buy alcohol in Las Vegas. Input: Driver's license information. Output: Boolean, true if legal.

The user might be a law enforcement officer working in Nevada who wishes to check the validity of the age on a Californian license. In this case, the user would need to invoke the California DMV's web service to obtain a message containing driver's license information. Because the ontologies of the producer and consumer are different, the message is fed into Ontrapro for alignment. Ontrapro aligns the message using the aforementioned techniques. Using the translation metrics, success of alignment can be measured when the California DMV's service is invoked, and its outputs and effects are examined. If successful, the outcome is driver's license information in terms that the Nevada Gaming Commission service can understand. Afterwards invocation of the Nevada Gaming Commission service follows, and the result is a boolean message.

This use case shows that two independently developed semantic web services written in distinct ontologies can communicate through automatic M2M translation. The M2M approach can be used to build new capabilities from existing eGovernment services by building bridges between semantic applications.

References

Ashpole, B., Ehrig, M., and Euzenat, J. 2004: Evaluation of Ontology-Based Tools (EON2004), http://km.aifb.uni-karlsruhe.de/ws/eon2004/#oc.

Czajkowski, M., Buczak, A., and Hofmann, M. 2004. Dynamic Agent Composition from Semantic Web Services. In Proceedings of Semantic Web and Databases, 27–40. Toronto, Canada: Springer.

Hofmann, M., Chacon, D., Mayer, G., Whitebread, K., and Hendler, J. 2001.: Cast agents: Network-Centric Fires Unleashed. In Proceedings of National Fire Control Symposium, 12–30. Lihue, Kauai, Hawaii.

Kalyanpur, A., Sirin, E., Parsia, B., and Hendler, J. 2004. Hypermedia Inspired Ontology Engineering Environment: Swoop, http://mindlab.umd.edu.

Martin, D., et al. 2005: Web Ontology Language Services (OWL-S), http://www.daml.org/services/owl-s/1.1.

Nau, D. 2005: JSHOP: Java Simple Hierarchical Ordered Planner, http://www.cs.umd.edu/projects/shop.

Nejdl, W. 2004. How to Build Google2Google an (incomplete) Recipe. In Proceedings of the International Semantic Web Conference, 1-5. Hiroshima, Japan: Springer.

Shvaiko, P., and Euzenat, J. 2005: Tutorial on Schema and Ontology Matching (ESWC 2005), http://www.eswc2005.org.

Sycara, K., Paolucci, M., Ankolekar, A., and Srinivasan, N. 2003. Automated Discovery, Interaction, and Composition of Semantic Web Services. Journal of Web Semantics 1:27-46.

UMBC 2004: Swoogle, http://swoogle.umbc.edu.