Automatic Matchmaking of Web Services

Sudhir Agarwal
Institute of Applied Informatics and Formal Description Methods (AIFB),
University of Karlsruhe (TH), Germany.
agarwal@aifb.uni-karlsruhe.de

Anupriya Ankolekar
Institute of Applied Informatics and Formal Description Methods (AIFB),
University of Karlsruhe (TH), Germany.
ankolekar@aifb.uni-karlsruhe.de

Categories and Subject Descriptors: H.3.5

General Terms: Algorithms, Design, Theory, Verification

Keywords: Semantic Web Services, Matchmaking

1. INTRODUCTION

Cross-organizational business processes are often offered as web services mainly due to open standards like XML and SOAP. To achieve automation of various tasks like finding and composing relevant web services, machines should be able to reason about various properties of web services and of the involved data. Web Ontology Language (OWL) provides a formalism for describing data and thus also static semantic constraints of web services.

However, since invoking a web service can actually trigger a complex business process involving many business partners it becomes necessary to have a formalism that allows the users to reason over the temporal aspects of a web service automatically. For example, if a user is only interested in web services that perform certain tasks in desired order, he must be able to specify this requirement and find the web services that fulfill the requirement automatically. On the other hand, a web service provider may wish to publish the structure of his web service partially or completely if his web service is not acting as a mediator and thus he is not willing to guarantee the functioning of the component web services.

In this paper, we propose a novel combination of π-calculus and DLs for describing not only static semantic constraints but also the access control policies and temporal aspects of a web service semantically [3, 6]. Our formalism does not force a provider to specify the behavior of his web service, but allows the users to reason over the behavior of a web service, in case the information is present. We also show how requests for restricting the semantics, security and temporal properties of web services can be specified and present the main features of our matchmaking algorithm.

2. SPECIFICATION OF WEB SERVICES

From now on, $\mathsf{o_1.o_2}$ represents the sequential execution of the operations $\mathsf{o_1}$ and $\mathsf{o_2}$. Further, (x) represents bound occurrence of x (variable), $\overline{\mathsf{c}}\mathsf{x}$ represents the activity for sending x along the channel c and $\mathsf{c(x)}$ represents the activity for receiving a value along the channel c and saving it in the variable x. The expressions inside [] are conditions.

2.1 Temporal Constraints

Consider a web service that upon receiving an order and credit card charging information from a user, first places the order and then charges the credit card. The web service can be specified with our formalism as follows:

$$\mathsf{user(Order).user(ChargingInfo).}$$
$$\overline{\mathsf{placeOrder}}\ \mathsf{Order.}\overline{\mathsf{chargeCC}}\ \mathsf{ChargingInfo}$$

2.2 Mediation

Analogous to subsumption relationship between concepts of an ontology, we propose to use simulation relationship for web services or components of a web service in order to resolve the problem of heterogeneity of web services. For example, $\overline{\mathsf{service_1}}\mathsf{B.P}$ simulates $\overline{\mathsf{service_2}}\mathsf{A.P}$ if $\mathsf{A} \sqsubseteq \mathsf{B}$, in case the services $\mathsf{service_1}$ and $\mathsf{service_2}$ are query answering services.

2.3 Semantic Constraints

Consider a web service that returns the set of all professors working in a certain university. The user has to provide the value of the university, for which he wishes to have the list of professors working in the university.

$$\mathsf{user(u).\{[valueOf(u)\ instanceOf\ University]}$$
$$\overline{\mathsf{select}}\ \text{“}\mathsf{Prof} \sqcap \exists \mathsf{worksIn.\{valueOf(u)\}}''\mathsf{.select(x).}\overline{\mathsf{user}}\,\mathsf{x\}}$$

2.4 Security Constraints

We foresee an input parameter for the set of certificates a user has to show in order to prove his eligibility to access a web service. Further, we use a special predicate $\mathsf{CCD(C,P)}$, that is true iff the set C of certificates fulfills the access control policy P according to the certificate chain discovery algorithm [1, 4].

Consider a web service from earlier example with the slight modification that only the employees of a university have access to it.

$$\mathsf{user(C).[CCD(valueOf(C), UniEmployee)]}\overline{\mathsf{select}}\ \text{“}\mathsf{Professor}\sqcap$$
$$\exists\mathsf{worksIn.(University}\sqcap\exists\mathsf{locatedIn.\{Germany\})}''\mathsf{.select(x).}\overline{\mathsf{user}}\,\mathsf{x}$$

3. MATCHMAKING

3.1 Request Specification

With a request a user specifies conditions a web service must fulfill in order to be considered as a match.

Conditions on Output: A requester specifies with $\overline{\mathsf{select}}\ \mathsf{query.select(x).}\overline{\mathsf{user}}\mathsf{x}$ conditions on the output x, where in this example query is a DL concept expression also determining the type of the output.

Conditions on Input: Depending on what a user exactly wants, inputs of a web service can sometimes mean restriction, sometimes flexibility. If a requester wishes to provide an input, the request can be defined as $\mathsf{user}(\mathsf{u}).\overline{\mathsf{select}}\ \mathsf{query}.\mathsf{select}(\mathsf{x}).\overline{\mathsf{user}}\mathsf{x}$, where query may be dependent on the value of u.

Conditions on Access Control Policies: A user can specify with the predicate $\mathsf{CCD}(\mathsf{C},\mathsf{P})$ conditions on the access control policy, e.g. to restrict the set of matches to only those web services that are accessible to scientific staff of a university by using $\mathsf{CCD}(\mathsf{valueOf}(\mathsf{C}),\mathsf{SciStaff})$. If $\mathsf{SciStaff} \sqsubseteq \mathsf{UniEmployee}$, our matchmaking algorithm will find the web service from section 2.4.

Conditions on Sequence of Operations: Sometimes, a requester wishes to define constraints on the sequence of certain actions. For example, credit card should not be charged before the order has been placed. However, a web service may or may not perform some tasks after the order placement and before the action for charging the credit card. To model such constraints, we use a symbol $\square$ to denote a set of *don't-care activities*.

In case, a user wants that the credit card is charged some time after but not necessarily immediately after the order has been placed, the request can be defined as follows[1]:

$$\mathsf{user}(\mathsf{Order}).\mathsf{user}(\mathsf{ChargingInfo}).$$
$$\overline{\mathsf{placeOrder}}\ \mathsf{Order}.\square.\overline{\mathsf{chargeCC}}\ \mathsf{ChargingInfo}$$

3.2 Matchmaking Algorithm

The main feature of our matchmaking algorithm is that it does not produce mere match/no-match answers, but in case a web service is match only if certain conditions are fulfilled by the user, also the conditions for such matches. Further, our matchmaking algorithm supports not only the types of the input and output parameters of a web service but more expressive requests and simulation relationships among web services or components of a web service. Due to space limitations, we do not present the algorithm. However, we like to point out that we have a running prototypical implementation of our approach, that consists of a client for modeling web services and a server that performs the matchmaking.

4. RELATED WORK

WSDL-S proposes to annotate WSDL operations with pre- and postconditions. However, WSDL-S does not mandate any particular logic for specifying the pre- and postconditions [2]. Note, that the decidability and complexity of algorithms for reasoning about WSDL-S descriptions depend on the logic chosen for specifying the conditions.

OWL-S is an OWL ontology for modeling various properties of a web service [8]. As a result, OWL-S enables reasoning about web service properties with the help of a DL reasoner e.g. KAON2[2]. OWL-S has constructs for modeling pre- and postconditions of a web service as well as the process model of a composite web service. Though OWL-S also does not mandate any logic for specifying conditions, it suggests SWRL or DRL as possible candidates. However, SWRL in its pure form is undecidable.

OWL-S proposes to model pre- and postconditions of a web service as properties of the web service. As a result, even if a decidable subset of SWRL, e.g. DL-safe rules as proposed in [7], is used, one cannot reason over the pre- and postconditions of a web service with a DL reasoner directly. The reason is, that DLs cannot capture the semantics of the rules that are reachable via some property only. As a consequence, though OWL-S provides primitives for modeling pre- and postconditions, the OWL-S based matchmaking algorithms consider only the types of input and output parameters of a web service [8].

WSMO initiative addresses the issue of goal definition and web service discovery in much more detail than the above mentioned approaches [5]. WSMO also addresses the issue of heterogeneity of descriptions of requesters and providers. Similar to OWL-S matchmaker, WSMO differentiates between different types of matches. For example, exact-match, subsumes-match, plugin-match and intersection match. However, to the best of our knowledge, there is currently no implementation of a WSMO based matchmaking approach.

5. CONCLUSION

We have proposed a novel technique for specifying semantic, security and temporal aspects of a web service. We introduced a request specification language that is more expressive than existing approaches. We have developed and implemented a matchmaking algorithm that returns not only boolean answers but also the set of conditions under which a web service is a match.

Acknowledgments This work was supported by the Federal Ministry of Education and Research (BMBF) under the Internetökonomie-SESAM project and the European Commission under contract IST-2003-506826 SEKT.

6. REFERENCES

[1] S. Agarwal and B. Sprick. Specification of Access Control and Certification Policies for Semantic Web Services. In *6th International Conference on Electronic Commerce and Web Technologies*, August 2005.

[2] R. Akkiraju, J. Farrell, J.Miller, M. Nagarajan, M. Schmidt, A. Sheth, and K. Verma. Web Service Semantics - WSDL-S, " A joint UGA-IBM Technical Note, version 1.0,. Technical report, April 2005.

[3] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory Implemenation and Applications.* Cambridge University Press, 2003.

[4] D. E. Clarke, J.-E. Elien, C. M. Ellison, M. Fredette, A. Morcos, and R. L. Rivest. Certificate chain discovery in SPKI/SDSI. *Journal of Computer Security*, 9:285–322, 2001.

[5] U. Keller, R. Lara, A. Pollers, I. Toma, M. Kifer, and D. Fensel. WSMO Web Service Discovery, November 2004. http://www.wsmo.org/TR/d5/d5.1/v0.1.

[6] R. Milner, J. Parrow, and D. Walker. A Calculus of Mobile Processes, Part I+II. *Journal of Information and Computation*, pages 1–87, September 1992.

[7] B. Motik, U. Sattler, and R. Studer. Query Answering for OWL-DL with Rules. volume 3298 of *LNCS*, Hiroshima, Japan, November 2004. Springer.

[8] K. Sycara, M. Paolucci, A. Ankolekar, and N. Srinivasan. Automated Discovery, Interaction and Composition of Semantic Web Services. *Journal of Web Semantics*, 1(1):27–46, December 2003.

[1]Note, that the agents $\mathsf{placeOrder}$ and $\mathsf{chargeCC}$ must not be necessarily the same agents as described as components of some web service, but may have simulation relationships with the components of a potential match.

[2]http://kaon2.semanticweb.org