DISCOVERY AND CONTRACTING OF SEMANTIC WEB SERVICES

Position Paper for the Workshop on Frameworks for Semantics in Web Services, Innsbruck, Austria, June 2005

Authors:

Rubé n Lara*

Daniel Olmedilla+

*Tecnología, Información y Finanzas (TIF), Spain
Email: rlara 'at' afi.es

+L3S Research Center and Hanover University, Germany
Email: olmedilla 'at' l3s.de

ABSTRACT

The automatic location of services that fulfill a given need is recognized as a key step towards dynamic and scalable integration. In order to achieve such automation, a model that considers the dynamic aspects of service provision and how they affect service descriptions, and that enables an efficient and accurate discovery and contracting of relevant services, is required. In this paper we propose such a model, paying special attention to the distinction between services and Web services and how these two concepts relate to each other. The border between Web service discovery and service contracting is discussed, and what kind of descriptions can be expected at each step is analyzed. We outline requirements on frameworks for semantic description of Web Services in the light of our model, and propose techniques for realizing the dicovery and contracting steps.

INTRODUCTION

Semantic descriptions of Web services can be exploited to improve the process of locating services that achieve a given requester goal, enabling its (total or partial) automation. While a number of proposals for automating the discovery of Web services are available (e.g. [Paolucci et al., 2002], [Li and Horrocks, 2003]), these hardly consider the dynamic aspects of service provision and how

they affect service descriptions. As a consequence, these approaches do not explicitly address important issues such as the limited accuracy of service descriptions or the interaction between parties needed to perform service contracting. In addition, they often regard the terms 'service' and 'Web service' as synonymus. We believe these two terms are not equivalent and, furthermore, it is relevant to distinguish them and to explore their relation, as done in [Preist, 2004] and [Baida et al., 2004].

Based on our previous work in [Keller et al., 2005] and [Kifer et al., 2004], this paper presents our understanding of the terms 'service' and 'Web service', why they should be distinguished, and how they relate. Given this distinction, we explore what kind of Web service descriptions can be expected and how dynamic aspects of service provision influence them. Finally, we propose a model for Web service discovery and service contracting and identify suitable formalisms and techniques for achieving accurate and efficient discovery and contracting. We limit ourselves to discovery and contracting based on funtional constraints; the consideration of non-functional constraints and capabilities [Arroyo et al., 2004] is out of the scope of the paper.

SERVICE AND WEB SERVICE

The notion of service is semantically overloaded [Preist, 2004]. Different communities have a different understanding of the term "service". For example, in the business comunity a service is seen as a business activity that often results in intangible outcomes or benefits [Baida et al., 2004], while in computer science the terms service and Web service are often regarded as interchangeable to name a software entity accessible over the Internet.

In order to reach a common understanding of the problem we want to address, it is crucial to precisely define the terms service and Web service and, therefore, what kind of entities we aim at semantically describing and locating.

Service

A service is defined in [Preist, 2004] as a provision of value in some domain. As an example, let us consider a user who wants to book a flight ticket from Madrid to Innsbruck on a given date. The service he is looking for is the provision of a flight ticket with the specified constraints. Such provision is independent on how the supplier and the provider interact, i.e., it does not matter whether the requester goes to an airline tickets office or uses the airline Web site to book his flight. We understand the term service in this sense, that is, as a provision of value.

Web service

A Web service is defined in [Preist, 2004] as a computational entity accessible over the Internet (using Web service standards and protocols) . Coming back to our previous example, an airline might provide a software component accessible via Web service standards, i.e., a Web service to request the booking of a flight; the Web service is an electronic means to request a service, but not the service itself. We understand the term Web service as a means to request a service over standard protocols, described using widely accepted standards.

Web service descriptions

As introduced before, a Web service is used to request the actual provision of a service fulfilling some requester need. In order to locate an appropriate Web service and, in turn, to request the provision of the desired service, the Web service must describe what services it is able to provide. However, while the requester needs will be often concrete e.g. flying from Madrid to Innsbruck on a concrete date, a Web service will often be used to provide a set of related services e.g. booking of flights from a given airline. Even though we would expect a Web service to describe the complete set of services that can be requested, two observations make this expectation rather unrealistic:

Information volume: describing all the services that can be requested using a given Web service will, in many cases, imply the duplication of the provider database e.g. a Web service to book flights from a given airline will have to describe all the flights, from all origins to all destinations, and on all dates that the airline can provide. The information volumes involved in such scenario are rather unmanageable.

Dynamics: the set of services accessible via a Web service is dynamic. For example, the set of flights that can be booked using a Web service offered by an airline will depend on seat availability, which will change over time. Maintaining an accurate list of the services provided by the Web service would imply updating the Web service description every time some dynamic condition changes.

Given the observations above, we do not expect Web services to accurately describe the complete set of services they can provide. Instead, we expect Web services to describe the abstract set of services that can be requested i.e. a static characterization of the kind of services that can be accessed via the Web service. For example, the airline Web service from our example will declare that it can be used to request the booking of flights from a given airline and e.g. from

any destination in Europe to any other destination in Europe, not detailing all the possible concrete itineraries and dates, and independently of dynamic conditions such as seat availability or weather conditions. We call such static characterization of the services provided by a Web service, following the terminology in [Preist, 2004], an abstract service description. The concrete service that will be finally provided will be represented by a concrete service description. As pointed out in [Preist, 2004], an abstract service description will be complete i.e. every concrete service that can be accessed via the Web service will be contained in the description, but not always correct i.e. some services that are part of the abstract service description will not be actually provided.

CONCEPTUAL MODEL FOR WEB SERVICE DISCOVERY AND SERVICE CONTRACTING

Web service discovery is the process of locating Web services that can be used to request a service that fulfills some user needs. Service contracting is the process of actually contracting, via a discovered Web service, a concrete service fulfilling such needs.

Goal discovery

In order to achieve the automatic discovery of Web services, requester needs have to be explicitly stated. We expect such needs to be expressed as goals, corresponding to the description of what concrete service is sought. Such description will include: 1) the information the requester wants to receive as the result of the service provision (post-conditions) e.g. train schedules, and 2) the state of the world resulting from it (effects) e.g. booking of a flight. Goals need to be formalized in order to automate the Web service discovery process. However, it is unrealistic to expect requesters to be able to formalize their desires from scratch. For that reason, we introduce the goal discovery step in our model. Given a requester goal expressed in natural language, the goal discovery process will locate, using keyword-matching, one or more pre-defined goals that represent a generic formalization of his desire e.g. booking of flights. One of the located pre-defined goals will be selected and refined, either automatically from the textual desire or manually using appropriate tools, in order to express the concrete requester goal e.g. booking a flight from Madrid to Innsbruck on June 1st, 2005.

DL-based Web service discovery

Given a formalized requester goal, a Web service offering a service that fulfills this goal has to be selected from the set of all available Web services. As has been discussed in the previous section, this selection will be based on the abstract service descriptions of Web services and, therefore, we will not have a guarantee that the selected Web services can actually be used to request a service fulfilling the goal.

However, as the set of available Web services is potentially huge, an efficient filtering of relevant Web services is essential. Our assessment is that Description Logics [Baader et al., 2003] are a good candidate for this purpose. As shown in [Li and Horrocks, 2003], once a set of DL descriptions has been classified, querying the relation of a new DL description to the set of classified descriptions is highly efficient, independently of the size of the classified descriptions. Given this fact, our suggested approach is to classify the set of available services at publication time, based on their abstract service descriptions in DL. Such descriptions will only characterize what post-conditions and effects will result from the abstract service provision, not dealing with how they relate to the input given to the Web service. Determining whether a Web service is relevant to the goal at hand is reduced to computing the subsumption relation between the goal description in DL and the abstract services accessible through the correspondent Web services. Details on what subsumption (or set-theoretic) relations are relevant and how they are to be interpreted can be found in [Keller et al., 2005].

Transaction Logic-based Web service discovery

So far, we have only considered the post-conditions and effects in the abstract service descriptions associated to Web services, independently of the input information that has to be provided to the Web service in order to request a concrete service. However, in most cases this input information will determine what service will be actually provided e.g. if we give the origin and destination cities for a flight, the flight booked as the result of the service provision will be between these two cities.

While DL is a good formalism for classification tasks, it still has some limitations in our context. One of these limitations is its little ability to express the relation between the input given to a Web service and the post-conditions and effects resulting from the service determined by that input. This relation is important as it captures the Web service functionality, and as it defines what information is required to provide the desired service. If the requester does not have such information available, the Web service cannot be used to request the desired service. For that reason, we have studied other formalisms to complement the DL-based Web service discovery and to further refine the set of selected services. As presented in [Kifer et al., 2004], transaction logic is a good candidate for that purpose. Using F-Logic [Kifer et al., 1995] for describing

the abstract service that can be requested through a given Web service, we can capture the relation between the Web service inputs and the concrete service provided. Given such description and the requester input information, we can use transaction logic to simulate the results of the provision of the service determined by the input, and check fulfillment of the requester goal.

The requester input information is needed to further refine the set of relevant Web services and to determine whether the Web service can be actually used to request the service sought. However, as different Web services might require different input information, we do not expect the requester to specify all the input information he is able to provide as part of its goal at request time. Instead, we suggest that the set of information the requester has available will be part of an information repository kept locally by the requester and accessed during the Transaction Logic-based discovery to retrieve the input information required by the Web service. As the descriptions of relevant Web services will be accessible to the user from the DL-based Web service discovery step, as well as his own available information, transaction logic-based discovery does not need any kind of interaction with the provider.

As a result, we will have a set of Web services that can be potentially used to request the desired service, and whose input information requirements can be fulfilled by the requester. Furthermore, we know that the input information that can be provided will lead to the appropriate concrete service provision. However, as the abstract service description is not necessarily correct due to the abstraction done when formalizing it and to dynamic factors, we still do not have a complete guarantee that the Web service can actually be used to request the desired service.

Service Contracting

The result of Web service discovery is a set of Web services that can potentially be used to request a service fulfilling the requester goal. However, for a complete guarantee that an appropriate concrete service will be provided, and as a result of the possible incorrectness of abstract service descriptions and the presence of dynamic conditions affecting the service provision, communication between the requester and the provider is necessary. In addition, and even though the input information required by the Web service is available from the requester, the disclosure of such information might be conditional e.g. a requester might only disclose his credit card information to trusted providers.

In order to actually contract a service, the requester and the provider will establish a dialogue in which relevant information will be exchanged. If such dialogue succeeds, it will imply that the provider guarantees the provision of the requested service. For example, the requester will provide his itinerary information, the provider will check the availability of a flight for such itinerary,

and will communicate it to the user. If the flight is available, the airline guarantees the provision of the booking for a given period of time.

The exchange of information between parties will follow the choreography interfaces i.e. the description of the public behaviour exposed by both the requester and the provider. However, the rules guiding the conversation will, in many cases, be kept private. Rule-based policies [Bonatti et al., 2004] might condition the evolution of the conversation e.g. the requester might require a certificate to be sent by the provider in order to disclose his credit card details.

For expressing the choreographies of the interacting parties, there are several suitable approaches, such as WSMO choreography descriptions, OWL-S process models or SWSL processes. Regarding the policies affecting the disclosure of information, Peertrust [Gavriloaie et al., 2004] is regarded as a good candidate, as it integrates well with F-Logic, has an implemented negotiation engine, and preliminary research on its integration with WSMO is already available [Olmedilla et al., 2004].

SUMMARY OF POSITION

The main aspects of our position can be summarized as follows:

Distinction between services and Web services
> Services and Web services are closely related to each other but still different. While a service is a provision of value in a given domain, independently of how the actual provision happens, a Web service is a software component accessible through standard protocols that can be used to request the provision of a set of services.

Realistic expectations on Web service descriptions
> Due to the potentially big volumes of information required to accurately describe the set of all services that can be requested through a Web service, and to dynamic conditions that affect the service provision, we cannot expect Web service descriptions to be correct. They will be a static characterization of the set of such services, so that the descriptions are kept manageable.

Distinction between Web service discovery and service contracting
> In order to request a given service, Web services that can potentially be used to issue the request have to be efficiently located. Only when a reduced set of relevant Web services has been identified, interaction with these Web services will take place. Such interaction will accurately determine whether the concrete service sought can be requested through a given Web service.

Accurate but still efficient discovery
> Directly interacting with all available Web services to contract the service

sought does not scale, as it is time-consuming and the number of published Web services might be huge. For that reason, we propose a two-steps Web service discovery process to efficiently and as accurately as possible restrict the set of Web services before any interaction happens.

Requirements on frameworks for the semantic description of Web services

We regard Description Logics, and F-Logic plus transaction logic, as suitable formalisms for Web service discovery and contracting. More generally, Description Logics and a rule language that can work together with state-based reasoning is required. Such formalisms are supported by WSML and SWSL. OWL-S does not provides support for F-Logic, but support for rule languages with similar expressivity to F-Logic could be sufficient. For service contracting the public behaviour (choreography) of both requesters and providers must be described. WSMO, OWL-S and SWSL propose different formalisms for that purpose. Extensions with policy languages that enable the description of information disclosure policies is desirable but not explicitly supported by any of the proposals mentioned. Still, a rule language can be used for that purpose, including F-Logic. Summarizing, the frameworks proposed partially fulfill the requirements imposed by our model. A more detailed evaluation of what is missing from each of these frameworks will be part of our future work.

ACKNOWLEDGEMENTS

REFERENCES

[Arroyo et al., 2004] S. Arroyo, C. Bussler, J. Kopecky, R. Lara, A. Polleres, M. Zaremba: Web Service Capabilities and Constraints in WSMO. W3C Workshop on Constraints and Capabilities for Web Services, Redwood Shores, CA, USA, October 2004.

[Baader et al., 2003] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, P. F. Patel-Schneider (eds.): The Description Logic Handbook, Cambridge University Press, 2003.

[Baida et al., 2004] Z. Baida, J. Gordijn, B. Omelayenko, H. Akkermans: A Shared Service Terminology for Online Service Provisioning, Proceedings of the Sixth International Conference on Electronic Commerce (ICEC04), Delft, The Netherlands, 2004.

[Bonatti et al., 2004] P. A. Bonatti, N. Shahmehri, C. Duma, D. Olmedilla, W. Nejdl, M. Baldoni, C. Baroglio, A. Martelli, V. Patti, P. Coraggio, G. Antoniou, J. Peer, N. E. Fuchs: Rule-based Policy Specification: State of the Art and Future Work. Project deliverable D1, Working Group I2, EU NoE REWERSE, Sep. 2004

[Gavriloaie et al., 2004] Rita Gavriloaie, Wolfgang Nejdl, Daniel Olmedilla, Kent Seamons, Marianne Winslett: No Registration Needed: How to Use Declarative Policies and Negotiation to Access Sensitive Resources on the Semantic Web. 1st European Semantic Web Symposium, May. 2004, Heraklion, Greece

[Keller et al., 2004] U. Keller, R. Lara, H. Lausen. A. Polleres, D. Fensel: Automatic Location of Services. Proceedings of the 2nd European Semantic Web Conference, Heraklion, Greece, May 2005.

[Kifer et al., 1995] M. Kifer, G. Lausen, and J. Wu: Logical foundations of object-oriented and frame-based languages. Journal of the ACM, 42:741-843, July 1995.

[Kifer et al., 2004] M. Kifer, R. Lara, A. Polleres, C. Zhao, U. Keller, H. Lausen, D. Fensel: A Logical Framework for Web Service Discovery. In Semantic Web Services Worshop at ISWC, Hiroshima, Japan, November 2004.

[Li and Horrocks, 2003] Lei Li and I. Horrocks. A Software Framework for Matchmaking Based on Semantic Web Technology. In WWW' 03, Budapest, Hungary, May 2003.

[Olmedilla et al., 2004] Daniel Olmedilla, Rubén Lara, Axel Polleres, Holger Lausen: Trust Negotiation for Semantic Web Services. 1st International Workshop on Semantic Web Services and Web Process Composition in conjunction with the 2004 IEEE International Conference on Web Services, Jul. 2004, San Diego, California, USA

[Paolucci et al., 2002] M. Paolucci, T. Kawamura, T. Payne, and K. Sycara. Semantic Matching of Web Service Capabilities. In ISWC, pages 333– 347. Springer Verlag, 2002.

[Preist, 2004]Chris Preist. A Conceptual Architecture for Semantic Web Services. In Proceedings of the International Semantic Web Conference 2004 (ISWC 2004), November 2004.