Ontology based Specification of Web Service Policies

Stephan Grimm[2], Steffen Lamparter[1],
Andreas Abecker[2], Sudhir Agarwal[1], Andreas Eberhart[1]
[1] Institute of Applied Informatics and Formal Description Methods (AIFB),
University of Karlsruhe
{lamparter,agarwal,eberhart}@aifb.uni-karlsruhe.de
[2] Forschungszentrum Informatik (FZI), Karlsruhe
{grimm,abecker}@fzi.de

Abstract: An ever-growing number of XML-based languages are used to describe Web Service related issues such as security (WS-Security Policy), access control (XACML), or privacy (P3P-WS). While it is desirable to specify policies in a declarative way, these languages expose great diversity in both syntax and semantics making it hard to realize a unified system. Our contribution to this problem is twofold. First, we present an expressive formal notation for policies. Second, we show how requester-, provider-, and third-party policies can be used for choosing a suitable service while making sure that all relevant policies are obeyed.

1 Introduction

Web Services are loosely coupled, reusable software components, which can be invoked using standard web protocols. They facilitate communication between different applications on heterogeneous computer systems by using standard protocols such as SOAP and service descriptions like WSDL. Higher-level languages like BPEL4WS [A$^+$03] allow orchestrating various services into complex business processes.

Currently, policies are a new focus within the Web Service community and languages for expressing them are emerging. The term policy is generally defined as directive issued by a higher instance, e.g. a company, organization, or government. Policies are decided upon, and subsequently affect the way business and communication is conducted. Besides the general term, we find many definitions for fiscal policy, privacy policy, or foreign policy, which apply the general concept to individual domains. Policies affect many different aspects of an organization and are defined independently from a specific application area or business process. Consequently, on a technical level, policies often talk about issues such as quality of service, security, or privacy, which can also be found across application boundaries. It is important to note, that it is very hard to draw a clear line between general policies and application specific issues. Consider car parts being ordered in a supply chain management scenario. A policy stating that any communication must be encrypted with at least a 128 bit key is definitely independent of supply chain management. However, a policy that only accepts electronic orders from premium partners uses application- and company-specific terms such as order and premium partner.

After having introduced the term policy, we will now outline the role of policies in the Web Service usage scenario. First, the requester needs to discover potential services. Today, this is done by querying a UDDI registry for all endpoints implementing an agreed-upon B2B standard. In the second step, requester and provider needs and requirements have to be matched. Policies play an important role here, since non-functional aspects have to be taken into account as well as end point specific behavior. Consequently, a legal policy might disqualify an endpoint even though it fulfills the functional requirements specified by the B2B standard. The matching might include some form of negotiation that tries to optimize parameter settings for all participants. The Web Service community is working on declarative policy languages and the respective policy interpreters in order to automate this step, which today is carried out by the developer explicitly picking an endpoint from the query result obtained from the UDDI registry. After this matching phase, the endpoint can finally be invoked, for instance from within the workflow engine executing a business process.

In this paper we apply techniques from the semantic web and formal knowledge representation to the domain of Web Service policies. We sketch a generic policy notation, which is expressive enough in order for existing proposals such as WS-Security Policy, WS-P3P, or XACML to be translated into it. These policies can be formalized in terms of a common ontology covering the domains relevant to Web Services. This mechanism allows for the declarative specification of policies and for background knowledge - formalized in an ontology - to be included in an elegant way. Furthermore, policies from heterogeneous sources such as business partners or government agencies can be integrated and evaluated seamlessly within a single unified engine. This dramatically reduces the cost of maintaining policies as well as the complexity of dealing with several different policy languages.

2 Formal Specification of Policies

Policies impose restrictions on properties of a service. For an ontological description of policies these restrictions have to be expressed in an underlying knowledge representation formalism. This way requesters and providers of services can describe their policies with respect to a common ontology in terms of meaningful concepts and relations. By referring to common domain ontologies they assure the usage of an agreed terminology. One approach to represent policies by means of ontological descriptions is to specify property restrictions using the Description Logic (DL) formalism[1] [BCM+03]. DLs allow the definition of classes via complex concept constructors giving them meaning through a well defined formal semantics. They especially support restrictions on class properties, which suits our notion of policy. For the description of policies we use several aspects of the DL formalism reflecting different ways of restricting properties [LH03]. A property can be restricted to a fixed value, a set of possible values or a value range. Furthermore, it can be either mandatory or facultative as well as single-valued or multiple-valued. Sev-

[1] Subsequent examples are expressed in the DL formalism, which is motivated by its relation to the Web Ontology language OWL (http://www.w3.org/2001/sw/WebOnt/). In case further expressivity is required, it is possible to augment OWL with rules as in the Semantic Web Rule Language (http://www.daml.org/2003/11/swrl/).

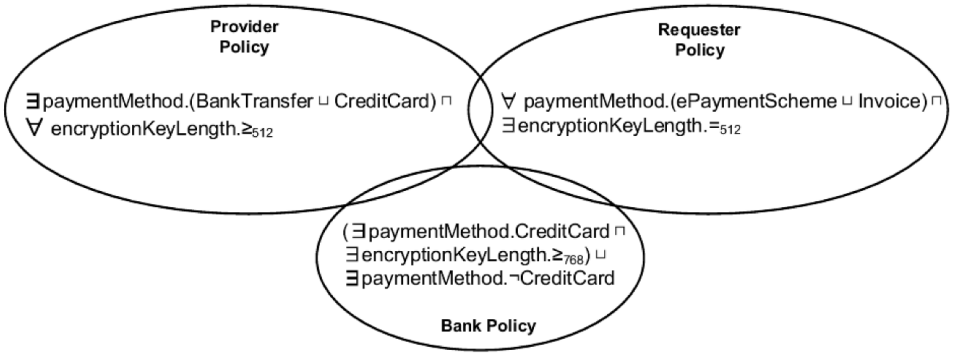

Figure 1: No match is achieved, since policies of requester, provider, and third party do not have a common intersection.

eral restrictions can be combined disjunctively and conjunctively. Further, complemental notions can be used to express restrictions in a negated way.

Formally described policies can be checked for compatibility via matching. In this context, matching of policies means checking whether their restrictions are compatible and do not contradict each other. This can be accomplished by a standard DL inference that verifies the satisfiability of the conjunction of the policy concept expressions involved [T$^+$02]. To prove a match between policy descriptions $P_1 \ldots P_n$ the statement $P_1 \sqcap \ldots \sqcap P_n \sqsubseteq \bot$ must be shown to be contradictory. In this case the intersection of the extensions of all the policy concept expressions is non-empty. Here the extension of a policy concept expression can be seen as all the possible parameter instantiations it allows, which is illustrated by the Venn-style diagram in Figure 1. (The details of the example are explained in the following section.)

3 Example Scenario

As an example, consider an employee of a company whose job is to purchase goods via an electronic service interface. This person plays the role of a requester of a service. The policy of his company requires either credit card or invoice as payment method. Further it requires the transferred account data or credit card number to be encrypted for security reasons, but the software used for encryption only supports a key length of 512 bit. On the other hand, a provider of an appropriate service has a policy that requires the payment method to be either bank transfer or electronic payment and the key length for encryption to be at least 512 bit. These two policies are captured by appropriate DL concept expressions in Figure 1. We assume the commonly used ontology to entail the fact $CreditCard \sqsubseteq ePaymentScheme$, which states that credit card is a special kind of electronic payment scheme[2]. In principle these two policies are not contradictory, which

[2]For a proper matching behavior, further axiomatizations are required such as certain properties being functional and certain classes being disjoint.

is illustrated by their overlapping extensions. As long as they use credit card and a key length of 512 bit the provider and requester can do business with each other.

Now assume that an additional policy is taken into account in which the bank, associated with the credit card to be used, requires that credit card numbers have to be encrypted with keys of length 768 bit or more. This is achieved by a policy stating that services use either anything but a credit card or a credit card together with an encryption key of appropriate length. This additional policy is compatible with the constraints of the provider and with those of the requester, but not with the intersection of both. A credit card is the only payment method the requester and provider can use in common, but since the requester's key length restriction contradicts with the bank's constraint for credit card usage, there is no intersection between all the three policies involved. Hence the requester has to look for another provider meeting his as well as the bank's policies.

The policies in the example combine aspects from the security and the financial sector. In general, a policy description can incorporate restrictions from a variety of domains. It is worth noting that in this approach they can all be handled within a single technical infrastructure for description and compatibility matching.

Technically, the scenario is realized as follows. The policies are marked up in OWL, which allows them to be published and exchanged on the web. The policies relevant for the requester, namely the requester and bank policies, are known in advance. Given the request for locating a suitable B2B partner, a list of potential candidates from e.g. a UDDI repository can be obtained. One by one, the candidates' policies are retrieved and their conjunction with the requester's and third party policies is fed into a DL reasoner. If the resulting intersection is non-empty, the service can be invoked without violating the provider's or any of the locally known policies.

4 Related Work, Conclusion and Outlook

Initiatives such as OWL-S [A+02, S+03] or WSMO [K+04b] try to incorporate Semantic Web techniques for describing Web Service semantics in an unambiguous and computer-interpretable way. These descriptions are to be used for advanced functionality such as dynamic service composition based on formally specified goals. Our work can nicely supplement this line of research by orthogonally adding policy-related aspects using the same techniques for description and reasoning.

In contrast to those few approaches which try to formalize particular policy aspects in a semantics-oriented manner, IBM's WS Policy framework[3] offers a "syntactic", yet comprehensive, framework to manage different aspects of policies. We are currently investigating how our work can be combined with this framework.

Current policy-related approaches mostly address security issues. [D+03] deals with security markup in the OWL-S predecessor DAML-S, developing security-related ontologies and two step matchmaking. [K+04a] proposes ontologies to annotate OWL-S input and

[3] http://www-106.ibm.com/developerworks/library/ws-polfram/

output parameters with respect to their security characteristics, such as encryption and digital signatures. Complementing our work, [A+04] shows how access control for Semantic Web Services can be specified and integrated with Semantic Web Service descriptions.

In this paper, we illustrated how expressive knowledge representation formalisms, as for instance description logics, can be used for specifying service policies. Starting from use cases and examples from existing policy languages, we sketched different kinds of property restrictions for policy formulation. Further, we showed how policies from various sources can be considered in choosing an appropriate service automatically. This framework paves the way for dynamic substitution of endpoints in an application or a business process. We believe that an integrated approach to modelling policies offers great benefits by simplifying implementations of policy engines and by allowing users to specify policies for different domains in a single declarative language. As future work, we plan to further investigate the way in which the chosen formalism can be appropriately applied to policy specification such that the descriptions capture the modeler's intention. To achieve the desired matching behavior, restrictions on the usage of DL expressions have to guide the modeler in describing policies by a set of intuitive modelling primitives.

References

[A+02] Ankolekar, A. et. al.: DAML-S: Web Service Description for the Semantic Web. In: *ISWC2002: Ist Int. Semantic Web Conf., Sardinia, Italy*. LNCS. S. 348–363. Springer. 2002.

[A+03] Andrews, T. et. al. Business process execution language for web services version 1.1. http://www.ibm.com/developerworks/library/ws-bpel/. May 2003.

[A+04] Agarwal, S. et. al.: Credential based access control for semantic web services. In: *AAAI Spring Symposium 2004 - Semantic Web Services*. March 2004.

[BCM+03] Baader, F., Calvanese, D., McGuinnes, D., Nardi, D., and Patel-Schneider, P. (edt.): *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press. January 2003.

[D+03] Denker, G. et. al.: Security for DAML web services: Annotation and matchmaking. In: *2nd Int. Semantic Web Conf.* volume 2870 of *LNCS*. Springer. October 2003.

[K+04a] Kagal, L. et. al.: Authorization and privacy for semantic web services. In: *Proc. of AAAI Spring Symposium on Semantic Web Services*. March 2004.

[K+04b] Keller, U. et. al. Web service modeling ontology, working draft. http://www.wsmo.org/2004/d2/v0.3/. February 2004.

[LH03] Li, L. and Horrocks, I.: A software framework for matchmaking based on semantic web technology. In: *Proc. of the 12th Int. Conf. on WWW*. 2003.

[S+03] Sycara, K. et. al.: Automated discovery, interaction and composition of semantic web services. *J. of Web Semantics*. 1(1):27–46. 2003.

[T+02] Trastour, D. et. al.: Semantic web support for the business-to-business e-commerce lifecycle. In: *Proc. of the 11th Int. Conf. on WWW*. S. 89–98. 2002.