Modeling Properties of Services

Jens Hündling
Hasso-Plattner-Institute for IT-Systems Engineering
at the University of Potsdam, Germany
jens.huendling@hpi.uni-potsdam.de

1. MOTIVATION

Nowadays, companies are coalescing, building virtual enterprises and are achieving common business goals together [1]. Simultaneously, companies try to stem dependencies and want to stay flexible and agile. Information technology is the core enabler for tightening integration of companies, and Service Oriented Computing (SOC) is the arising paradigm to handle flexibility [5]. Even more, SOC increases automation based on publishing standardized descriptions of offered services. Although SOC undoubtedly offers a promising approach to achieve this goal, it is still in its infancies and important features are missing. One key aspect is handled in this paper by introducing a model for properties of services, which enables enriched service description that exceed technical interface specifications. These properties are often subsumed under the terms Quality of Service (QoS) [2]or non-functional properties [6]. The basic principles of SOC is manifested by the Service Oriented Architecture (SOA) [5], defining three roles and their interactions. *Service Providers* publish descriptions of their services in a *Service Repository*. *Service Consumers* can find the published descriptions at the repository, where several descriptions are stored, e.g. in a categorized catalogue. Such a service description allows the consumer to use the service using automatically generated messaging.Hence, an essential requirement are standardized description languages and standard communication protocols as well as a common understanding of the querying, the categories and other objectives of the service repository. Currently, de-facto standards for today's most popular implementation of SOC are existing: Web services [5]. Additionally, for specific industrial sectors, standardized names for categories and services are evolving or are already existing, and the semantic web [4] is an emerging technology to handle this with approaches like WSMO [3].

A model for service description is given in Figure 1, using the terminology of the Web Services Description Language (WSDL) 2.0 [8]. This model enables separating the description of the abstract functionality offered by a service from concrete details of a service description such as 'how' and 'where' that functionality is offered. Thus, an abstract level describes a service in terms of the communication, i.e. the *messages* it sends and receives. One or more messages, their exchange pattern and the logical roles of the participants are aggregated in an *operation*. Finally, an *interface* groups logically related operations. At the concrete level, a *binding* specifies transport and wire format details for interfaces. Adding a network address with a binding results in an *endpoint*, and finally, a *service* groups together endpoints implementing a common interface [8]. The different levels of abstraction will also be relevant for the service properties model introduced in this paper.

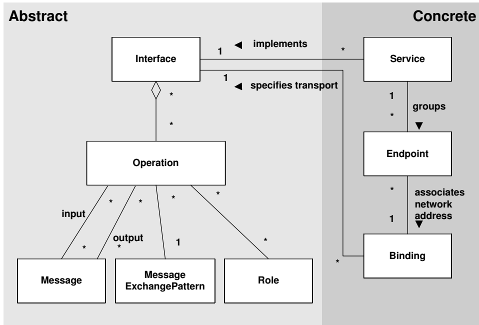

Figure 1: The core elements of WSDL 2.0.

To illustrate the requirements, a short example of a transportation service will be introduced. The service offers transportation and delivery of goods. It is invoked by a message containing procurement information, start and destination address. Additional information like possible climatic conditions of the transportation, e.g. temperature and humidity in climatic containers, are also available. Cost and delivery time can also be provided and security options for messaging like encrypted payment issues are available. Furthermore, some providers offer delivery state monitoring, e.g. by a web portal (web), per automatic mails at special events like delay or delivery (mail) or by a 24h phone-hotline (phone). Table 1 shows service implementations S_{11} to S_{31} with additional properties. These services are implementing a common transportation interface I_T (not described in more detail). The i in S_{ij} depicts a provider; j is a counter for the services of one provider. As could easily be recognized, the diversity of the properties show the complexity of the requirements for a service property model.

2. SERVICE PROPERTIES

As mentioned above, before consumers use services, they have to find them, and the consumer has to decide, whether using a specific service implementation or another implementation from a competitor. The model presented so far concentrates on technicalities like messages, message exchange patterns and URI's of the endpoints. Generally speaking, a whole set of different properties about different aspects is related to a service and relevant for several usage scenarios like discovery, negotiation, composition, substitution and management. These properties are often categorized in functional and non-functional properties [6] or Quality of Services (QoS) [2]. *Functional properties* describe what the service does and *non-functional properties* are used to describe how the service does it. This separation seems feasible and useful, because functional equivalence can be defined on two services with the same

Service	Climatic Temperature	Humidity	Cost	Time	Security Option	State Monitor
S_{11}	–	–	100-150 €	12-48 hours	–	web portal
S_{21}	–	–	150 US-$	1 day	–	mail, phone
S_{22}	max. 20°F	–	320 US-$	8 hours	128-bit key	mail, phone
S_{31}	$\leq$+4°C	$20-80\%$	400 €	24-36 hours	512-bit key	phone, web portal

Table 1: Sample transportation services and their properties

functional properties. Thus, a service can easily be replaced by a functional equivalent service with more suitable non-functional properties, e.g. lower cost, faster execution, and higher security. This goes along with the basic SOC principle of loose coupling. Nevertheless, the decision whether a property is functional or not has to be made individually for each request and is not dependent on the property itself. A model for service properties is necessary and the properties have to be classified using certain aspects.

2.1 Scales of Measurement

In this paper, a first aspect to categorize the properties will be introduced: Scales of measurement. Generally speaking, scales of measurement as know from mathematics refer to ways in which variables or numbers are defined and categorized. The scales are *nominal* (e.g. state monitor), ordinal (e.g. security), interval (e.g. temperature) and *ratio* (e.g. cost). It has to be mentioned that each scale has its allowed arithmetics functionality. This has consequences for service discovery, which is basically done by specifying the needs, e.g. services implementing interface I_T. Additional restrictions on the service's property, like budget restrictions, temporal deadlines and necessary security protocols can be added to the discovery query. On the other side, a provider can offer his service not with precise properties, but with ranges or at different levels. All of these can be found in the example in Table 1. If a consumer searches a service, the restrictions and other aspects like optimization criteria should be described in a declarative way. Thereby, methods well known from Description Logic could be applied for matchmaking. Standardization efforts like the Web Ontology Language (OWL) [7] are suspected to build a semantic foundation for describing such restrictions.

2.2 Service Properties Model

The model sketched in this paper can be mapped to the service entities as shown in figure 2. The **Quality of Services** is a set of properties. For each **Property**, the **Domain**, a **Scale** and a **Metric** has to be defined. This abstract set of QoS properties is bound to an **Interface**, specifying each interface's set of applicable properties. The concrete counterpart is a **ServiceLevel** which in turn is related to a service. When a **Provider** stores its service and the service level in a repository, ranges etc. can be defined. A **Provider** and a **Requestor** can negotiate on more precise properties and create a **ServiceLevelAgreement** as a contract.

3. CONCLUSION AND OUTLOOK

In this extended abstract a service property model is sketched. It is based on the principles of domains and scales of possible properties. These properties are grouped as a set of properties that is related to an interface. For a concrete service implementation, the specific service level can be defined and enhanced requests for services can be generated. Nevertheless, many things that are already in the example are not mentioned in detail in this paper. Considering the sample service S_{11}, it is likely that the cost depends on

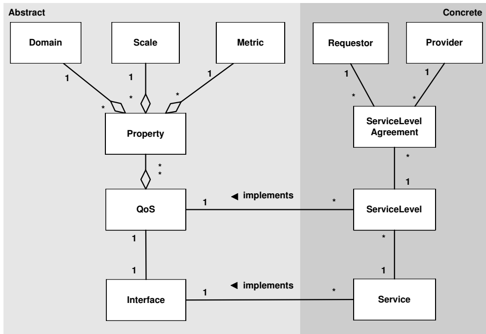

Figure 2: Mapping Service PSroperties to the Service Entities.

the delivery time negotiated as well as the distance between starting point and delivery point. Additionally, the metric has been left out of the discussion as well as the difference of the measurement units like US-$ and € or °C and °F. These aspects are future work, and defining properties in OWL [7] seams to be a fruitful approach which we currently investigate. Additionally, using QoS for processes and process (re-)planning which is still under research.

4. REFERENCES

[1] C. Bussler. *B2B Integration - Concepts and Architecture*. Springer, June 2003.

[2] J. Cardoso, A. Sheth, J. Miller, J. Arnold, and K. Kochut. Quality of service for workflows and web service processes. *Elsevier Journal of Web Semantics*, 2004.

[3] J. de Bruijn, C. Bussler, J. Domingue, D. Fensel, M. Kifer, J. Kopecky, R. Lara, E. Oren, A. Polleres, and M. Stollberg. Web Service Modeling Ontology (WSMO). `http://www.wsmo.org/TR/d2/v1.1/`, Feb. 2005.

[4] D. Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, Berlin, 2 edition, 2003.

[5] J. Hündling and M. Weske. Web services: Foundation and composition. *EM - Electronic Markets Journal*, 13(2):108–119, June 2003.

[6] J. O'Sullivan, D. Edmond, and A. H. M. ter Hofstede. Formal description of non-functional service properties. Technical Report FIT-TR-2005-01, Queensland University of Technology, Australia, 2005.

[7] M. K. Smith, C. Welty, and D. L. McGuinness. OWL Web Ontology Language Guide. `http://www.w3.org/2004/OWL/`, 2004.

[8] W3C. Web Services Description Language (WSDL) version 2.0. `http://www.w3.org/TR/wsdl20/`, August 2004.