Towards a Meta-Model for Service Properties

by Jens Hündling
Dagstuhl, Germany 2005
Dagstuhl Seminar (05462) on Service Oriented Computing (SOC)

Service Oriented Computing offers a promising approach for global businesses and integrated, virtual enterprises that achieve common business goals. For realizing an exhaustive Service Oriented Architecture, some basic research is still necessary and a consensus has to be reached about key aspects. We argue that one key aspect is a common (meta-) model of enriched service descriptions, especially for all use cases related to service discovery and service selection. A first step towards enriched service description is modelling the service's properties, also known as Quality of Services (QoS).

The envisioned service properties are often subsumed either under Quality of Services (QoS) or non-functional properties. Since these terms are used with controversial meanings, they will be discussed here shortly. Functional properties describe *what* the service does and non-functional properties describe *how* the service does it. The first step towards using a service is requesting the service broker for available and suitable services. This request is currently based on the functional properties of the requested services. There is a lot of research already underway to handle this; e.g. some current approaches suggest semantic annotations for service descriptions. Thus, functional equivalence can be defined on two concrete services, i.e. 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 functional equivalence goes along with the basic SOC principle of loose coupling.

Nevertheless, if there are several services available and functionally adequate for the request, how does the requestor select one specific service? The provider of the service might be a first selection criterion, but what if a provider offers several service, e.g. a "gold" and a "silver service" or a "premium, deluxe and a standard edition" of his service? Again, the obvious answer is selecting the *cheapest* and the *fastest*; but what about *reliability*, *availability* and *security*? Even more, what about properties like *purity*, *insurance*, *colour*, *taste* and *temperature*? All of these (non-functional) properties might be selection criteria in a specific domain. Generally speaking, some kind of ranking or, more generally, comparison must be possible. For automated discovery and selection, service descriptions need to be specified in a formal, computer-readable way. Additionally, properties of services that are beyond technical interface specifications should possibly be modelled, e.g. by including QoS properties.

Additionally, in a service-oriented environment, it is often hard to decide whether a certain property is functional or non-functional. Consider the property duration (typically considered to be non-functional) in a request like: "Transportation of goods within in 24 hours". A consumer (i.e. service requestor) uses some (other) technology to find services with this capability. Thus, the consumer gets a set of adequate services. The request is specified more precisely by stating "within 24 hours", which might also be considered as functional requirement, because a functionally equivalent service has to match this mandatory property. Similar examples can be found for other properties like costs or security, which are often regarded as non-functional properties. To sum up, whether a property is functional or non-functional, is not depending on the property itself, but on the request. Quality of Services is also a widely used term. QoS constraints are sometimes defined as a synonym for non-functional constraints. Other approaches see QoS as a subset of properties of Web services. Furthermore, some approaches separate properties in functional, non-functional and quality aspects. There are also authors claiming that QoS implies aspects, which can be both, functional and non-functional.

Considering all the problems mentioned above, we suggest a meta-model for service properties. To mention some important aspects of the meta-model, it should be based on the domain (i.e. the permissible values like "all positive numbers" or "Red, Green, White"), the mathematical scale (e.g. nominal, ordinal, interval or ratio scale), and the metric (i.e. how and where to measure the values as well as the measuring units). These aspects have some implications. For example, for a nominal scale it is not possible to give a common ranking, because no ordering is defined per se. Thus, the desirability has to be specified somehow, e.g. by stating maximal/minimal values or explicitly stating a single best and worst value of the domain. Using the meta-model, it should be possible to specify the properties of interest for a specific domain and thereby building a model for the service properties of the domain. In the next step, the providers of that domain can specify their concrete service properties and requestors can define constraints (e.g. budget restrictions and deadlines) and their desirability ("Red is better than Green and White is the worst").

Service properties, or more precisely the meta-model, have to be embedded in a general service model for Service Oriented Architectures and should be related to Service Level Agreements. In a next step, the meta-model should help to aggregate the properties of a composite service. A composite service is build from several services that are orchestrated, i.e. control- and data-flow is defined. Thus, a composite service builds a process with services as process steps. Obviously, the properties of the composite service are somehow related to or dependent of the properties of the composed services.

References

- Jens Hündling: *Modelling Properties of Services.* Proceedings of the First European Young Researchers Workshop on Service Oriented Computing April 21-22 - 2005, Leicester , U.K.

Contact

Jens Hündling
Hasso Plattner Institute for IT-Systems Engineering at the University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3, D-14482 Potsdam, Germany
Jens.Huendling@hpi.uni-potsdam.de
http://bpt.hpi.uni-potsdam.de/