

# Towards a Normative SOA Model

Gregory Kohring  
C&C Research Laboratories,  
NEC Europe Ltd.

## 1 Defining the Modeling Language

As noted in the SOA-RM's charter, there are many informal models of SOAs of varying quality already available on the Internet [1][2][3]; hence, we might ask what is to be gained by this TC introducing yet another informal model? One answer is that we can hope to achieve a tightening of our concepts used in defining an SOA as well as a better understanding of what does and does not constitute an SOA. However, we will still be left with the problem of comparing different models of an SOA and determining whether or not those models are compatible. For example, are the models discussed in [1][2][3] compatible with the submission from MacKenzie and Nickull[4]? At a certain level, one would agree that all four models share a common understanding as to what properties an SOA should possess; but are the models truly equivalent, i.e., does the W3C model[2] comply with Microsoft's model[3]?

By developing a standard model in terms of a standard modeling language, we stand a much better chance of determining when two competing models, written within the same modeling language, are equivalent. Of course, formal models do not remove all ambiguity, but, depending on the power of the underlying modeling language, they can at least isolate the areas containing inconsistencies.

In this paper we would propose the use of UML (Unified Modeling Language) [5] for the definition of an SOA reference model. UML is a widely known modeling language with which most developers are familiar; furthermore, the basics of UML diagrams, if not their nuances, are relatively easy for non-experts to learn. One advantage of using UML is that it has an associated normative form, namely XMI. [5] With XMI, it is also possible to create a normative model which other users can work with.

## 2 A Proposed Starting Point

This section discusses one proposed starting point for of an SOA model written in UML. Figure 1 shows a high level view of the major conceptual parts in our SOA model. On the surface, this model is similar to the other four models [1][2][3][4] already mentioned, and if these other models had also been written in UML, it would be possible to transform one into the other and establish their equivalence (or dissonance).

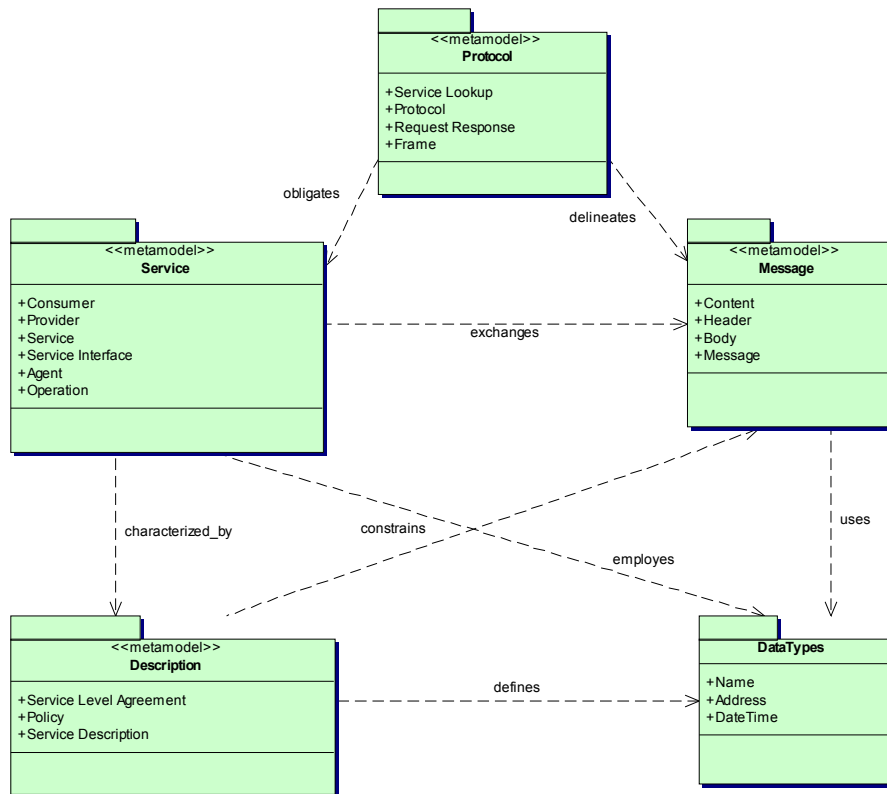


Figure 1 A high level model of an SOA.

As in other models this model contains the usual concepts of “Service”, “Message”, “Protocol”, “Description” and “DataTypes”. (In this short position paper I am not going to precisely define all the terms, as that is the task of the SOA-RM TC over the course of the next 12 months, rather I will assume the terminology is familiar to everyone and leave the task of precise definition until later.) The dotted lines denote dependencies between concepts and the labels convey non-normative semantic descriptions of those dependencies. What is more important is the information inside the package symbol, as this information is what constitutes the normative part of the model.

Drilling downwards into the “Service” package for example, we can create Figure 2 representing a model for a “Service”. Here we see that a service has a name, an address, a provider (or owner) and a set of “interfaces” through which clients wishing to use the service must interact. (One can take the view that the “Service Interface” is actually more fundamental than the service itself, since multiple services can support the same interface.) A service also contains a collection of policies governing how it operates and set of SLAs (Service Level Agreements) defining the quality of service this particular service is willing to offer.

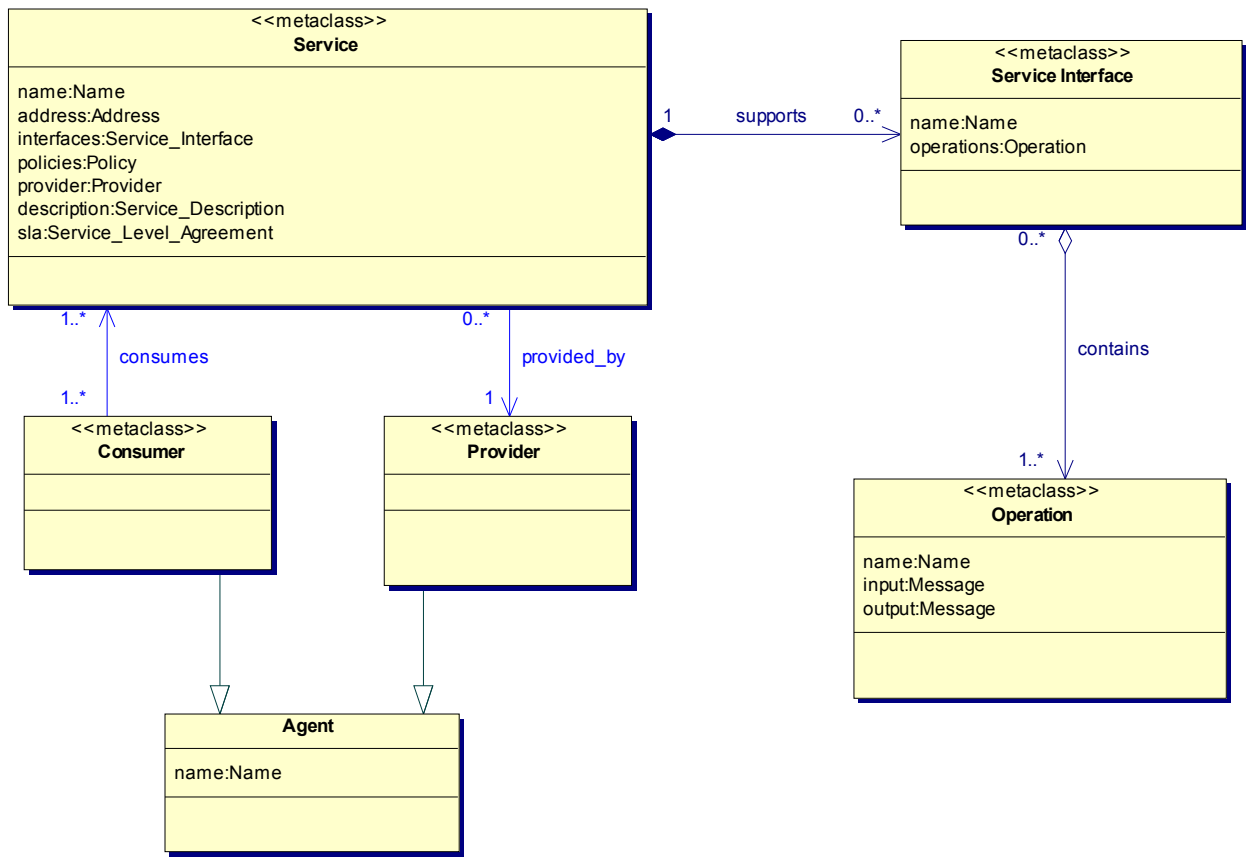


Figure 2 A model of a service

At first glance, one might complain that this model is too much software oriented, given the coffee shop example mentioned during the inaugural teleconference; however, if you look more closely you will notice that a coffee shop does indeed possess a name, address and owner, along with some service interfaces (over the counter sales), policies (when in LA only accept US dollars) and SLAs (satisfaction guarantees).

Obviously, we can continue this exercise for the other high level concepts depicted in Figure 1, but at this time such an exercise is better left until the TC has agreed upon those concepts as its starting point.

### **3 Summary**

In this note we have stated the case for a normative SOA model based upon UML and XMI. This is not to say that there is not a place for non-normative documents using the informal notation presented by MacKenzie and Nickull[4], as such documents might be easier for non-experts to understand. By the same token, the case for a normative model as required by the expert practitioner is also clearly given.

In conclusion, it is probably best to create a non-normative document describing the model using either UML or an informal notation along with an XMI document containing the normative model.

## References

- [1] *Service-oriented architecture (SOA) definition*, <[http://www.service-architecture.com/web-services/articles/service-oriented\\_architecture\\_soa\\_definition.html](http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html)>.
- [2] *Web Services Architecture*, <<http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>>.
- [3] *Understanding Service-Oriented Architecture*, <<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnmaj/html/aj1soa.asp>>.
- [4] C.M. MacKenzie and D.A. Nickull, *Position Paper on the Definition of an SOA-RM*, <[http://www.oasis-open.org/apps/org/workgroup/soa-rm/download.php/11977/soa\\_rm\\_paper\\_matt\\_duane\\_05-03.pdf](http://www.oasis-open.org/apps/org/workgroup/soa-rm/download.php/11977/soa_rm_paper_matt_duane_05-03.pdf)>.
- [5] *UML Resource Page*, <<http://www.uml.org/>>.