

Modeling Services with SoaML

The specification for the new modeling language for services, SoaML (Service-Oriented Architecture Modeling Language), is soon to be published. Which service comprehension forms the basis of SoaML? Which services is SoaML suited towards and which elements are available for modeling contracts, services, providers and consumers?

Modeling with UML has since become an often-used part of the software developing furniture. The question of how services, service providers and service users should be modeled is raised when implementing a service-oriented architecture (SOA) as UML does not have any special elements for modeling an SOA's elements. The OMG (Object Management Group) want to answer this question using SoaML as an extension to UML.

We will use a simple example to show what the SoaML answer looks like. A cinema wishes to offer three services using one internet page: movie information, seat reservation and a shop that sells DVDs; delivery will be provided by a parcel service.

It is easy to see that, unlike the movie information, seat reservation and the DVD online shop are not services which are basic and stateless. They are services which can be interactive, stateful and bi-directional or made by collaborating with other service providers. If you were to only describe basic services (which should be used to put together a service for the customer), the general context would often be lost. This is why SoaML is focused on describing services from both the business and architect view. Service-oriented architecture can be described for business partners, departments and IT systems. SoaML is no longer focused on how preparation of a service is implemented in detail, whether e.g. an interface is realized via file exchange or a web service.

Reusability also needs to be accommodated for in modeling, i.e. the fact that a service is provided by various providers

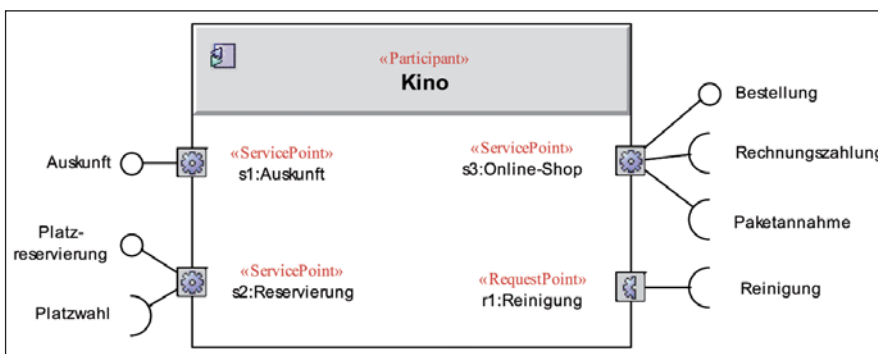


Fig. 1: Cinema with Three Services Offered and One Service Used (Cleaning)

needs to be expressed.

SoaML uses "participants" as an elementary concept; these prepare services via so-called "service points" or use them via "request points". In the example, the cinema participant provides three services

which are specified in more detail by „service interfaces“: movie information, seat reservation and DVD shop. The service interface defines which interfaces the participant provides or requires (Figure 2: Service Interface Reservation). The interfaces used describe the service provider's

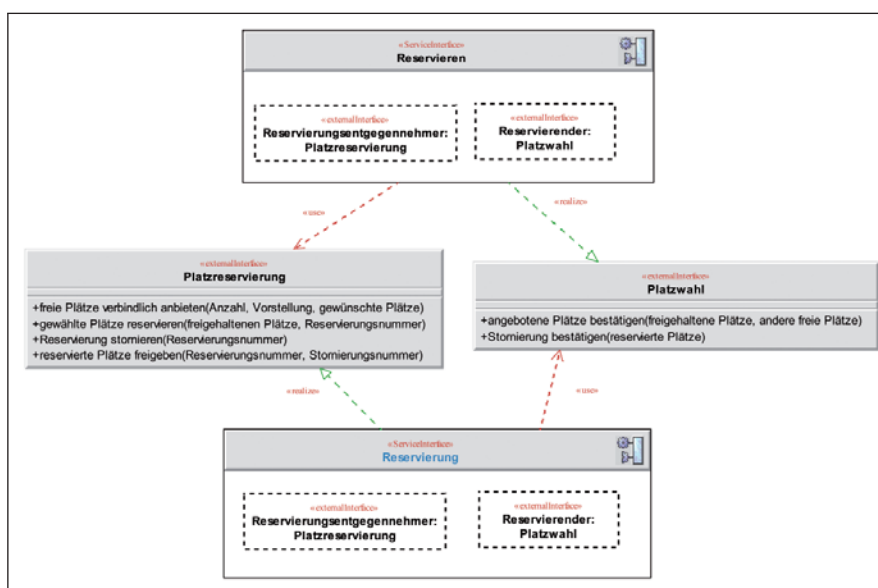


Fig. 2: Service Interface Reservation

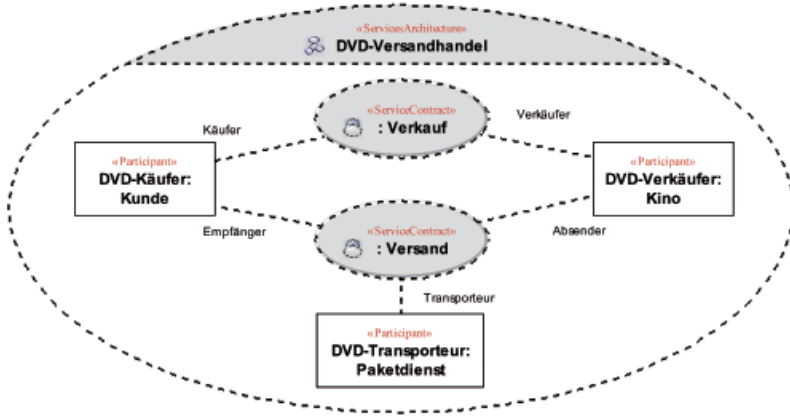


Fig. 3: Services Architecture for DVD Mail Order

responsibilities. This means it is necessary for the person making the reservation to select the seat.

The two other essential modeling constructs are "Service Contracts" and/or "Service Architectures". Both are used to show collaboration between those involved.

A service contract always describes the collaboration of various service interfaces, whereas a services architecture describes the collaboration of various participants.

The example (Figure 3: Services Architecture for DVD Mail Order) shows the services architecture for DVD mail order. The three roles involved, DVD Buyer, DVD Seller and DVD Carrier, are defined in the architecture. For each role, the participant is provided which assumes the role. This means that the cinema is in charge of the DVD shop and the parcel service DVD dispatch. Collaboration between the individual participants is managed using service contracts. Linking the roles shows which roles are adopted in the service contract. This means that the DVD seller is also the DVD sender.

The essential difference between a service contract and a services architecture is that all parties know each other in a service contract and the contract will be explicitly closed. The cinema in the example has a shipping contract with the parcel service. This controls the fact that the parcel service delivers the shipment to the customers. The participants normally only know each other via the service contract for a services architecture. The participants are only loosely linked for the collaboration. The parcel service does not know, for example, that the cinema has sold a DVD to the customer. If the cinema were to sell T-shirts or posters as well as DVDs at a later stage, the shop could be extended without having to change collaboration with the parcel service.

If it is too risky for the customer to be dependent on individual suppliers and service providers if they require a larger service (e.g. when building a house), they may employ the services of a general contractor, who is responsible for coordinating all of the subcontractors. This is then documented using a service contract. The general contractor delegates individual responsibilities to the subcontractors and is responsible for them rather than the customer.

Figure 4 (Service Contract for Mailing Using e.g. a Parcel Service) shows the mailing contract, which is an example of a service contract with more than two roles. The three roles sender, recipient and carrier, were already used in the services architecture for describing the roles of the participants involved. The shipping contract controls which tasks the carrier needs to carry out when shipping. They transport the shipment and deliver it to the recipient. This service is offered by each participant which provides the Transport service interface for a service point; this is the parcel service in this example. The parcel service is then also linked to the "Dispatch" service contract. The same principle applies when using services.

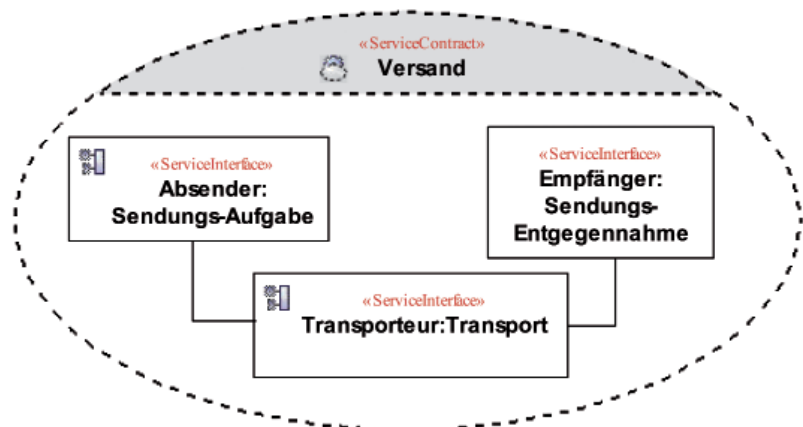


Fig. 4: Service Contract for Mailing Using e.g. a Parcel Service

SA behavior can also be defined as well as structures for service contracts or service interfaces (Figure 5: Protocol for Service Interface Seat Reservation).

A service interface's behavior describes the order in which the provided and required services need to be called, i.e. which order the provider's and user's actions can take place in is defined. Available seats are initially offered as a binding reservation using the reservation acceptance in the example; these are then confirmed by the person making the reservation etc. SoaML builds on the idea of business service by also considering interactive, durable, stateful services. Service providers and users are modeled using participants. Services are typed via service interfaces so that bi-directional communication can be mapped and the services reused.

Rules for preparing and/or using services are defined using service contracts; these can be extended by a behavior description.

All elements correspond to service modeling; this is the reason that many people have voiced the opinion that a more suitable name would be ServiceML. It remains to be seen whether other aspects from SOA will also be added over the next few years. Deployment aspects and service registry are often required in projects. Elements are suitable for displaying entire views or modeling an overview; dependencies can be made clearly visible. It really is worth wading through the 150 page SoaML specification when working in a complex environment.

It is precisely when various systems are interlocked and working together and little has previously been documented on the process that technical dependencies become the victims of generalization. Documentation can be consistently extended by SoaML elements.

SoaML Stereotype	UML Metaclass	Description
ServiceInterface	Class or interface	Defines the interface as a service point or a request point and is the type of role in a ServiceContract Determines how a participant needs to interact to provide or use a service
ServiceContract	Collaboration	Specifies the conditions between providers and users of a service
ServicesArchitectur	Collaboration	High-level view of an SOA which defines how a group of participants work together in a group towards a certain shared purpose by providing and using services
ServicePoint	Port	Offer of a service typed with a ServiceInterface or interface
RequestPoint	Port	Using a service typed with a ServiceInterface or interface
Participant	Component	Offers a service via a ServicePoint or uses a service via a RequestPoint

Table – The Most Important SoaML Stereotypes

Conclusion

SoaML can be used as an aid for understanding and using complex, stateful services.

SoaML is developed by the OMG in a task force that MID GmbH is also part of. Various manufacturers now have profiles for UML modeling tools. MID GmbH is including an SOA profile in the scope of delivery of the newest version (10.1) of the modeling platform, Innovator, which can be used as a basis for the MID Modeling Methodology M³ for SOA.

You can download a free overview poster of the MID Modeling Methodology M³ for SOA under www.mid.de/PosterM3SOA

You can also find a complete SoaML modeling example with a detailed description of UML model elements used under www.mid.de/WhitepaperSoaML

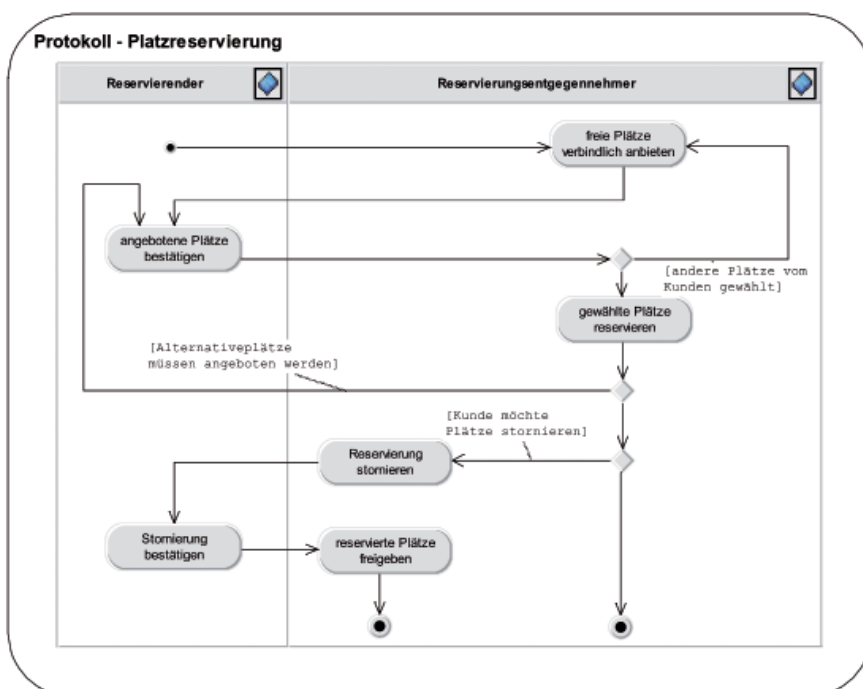


Fig. 5: Protocol for Service Interface Seat Reservation

Free Live Webcast

Fit for SOA -
Modeling with SoaML

Friday, August 7, 2009
11 – 11:30 a.m.

Register here:
www.mid.de/WebcastSoaML



Maria Deeg
Project Manager, MID