

Modellierung von Services mit SoaML

Die Spezifikation für SoaML (Service oriented Architecture Modeling Language), der neuen Modellierungssprache für Services steht kurz vor der Veröffentlichung. Welches Service-Verständnis liegt der SoaML zu Grunde? Für welche Services eignet sich SoaML und welche Elemente stehen für die Modellierung von Contracts, Services, Provider und Consumer zur Verfügung?

Modellierung mit UML ist inzwischen häufig ein fester Bestandteil in der Software-Entwicklung. Bei der Einführung einer serviceorientierten Architektur (SOA) stellt sich die Frage, wie Service, Service-Provider und Service-Nutzer modelliert werden sollen, da in der UML keine speziellen Elemente für die Modellierung der Elemente einer SOA zu finden sind. Mit der SoaML als Erweiterung der UML will die OMG (Object Management Group) jetzt eine Antwort auf diese Frage geben.

Um zu zeigen, wie die Antwort der SoaML aussieht, betrachten wir das Ganze an einem einfachen Beispiel. Ein Kino möchte über eine Internet-Seite drei Dienste anbieten: Eine Programmauskunft, eine Platzreservierung und einen Shop, in dem DVDs verkauft werden, wobei die Lieferung über einen Paketdienst erfolgt. Abbildung 1: Darstellung des Participants Kino mit drei angebotenen Diensten und einem genutzten Dienst (Reinigung).

Es lässt sich leicht erkennen, dass es bei den letzten beiden - anders als bei der Programmauskunft - nicht um Dienste geht, die elementar und zustandslos sind, sondern auch um Dienste wie Platzreservierung und DVD-Verkauf, die interaktiv, zustandsbehaftet und bidirektional sein können oder in Zusammenarbeit mit anderen Dienstleistern erbracht werden. Würde man ausschließlich die elementaren Dienste (mit denen die Leistung für den Kunden zusammengebaut werden soll) beschreiben, ginge häufig der Gesamtzusammenhang verloren. Der Fokus der SoaML liegt daher auf der Beschreibung von Diensten aus der geschäftlichen und architektonischen Sicht. Die serviceorientierte Architektur kann für Geschäftspartner,

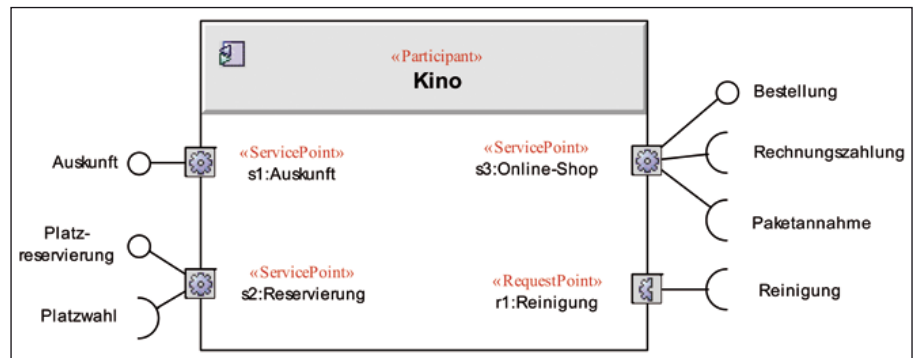


Abb. 1: Darstellung des Participants Kino mit drei angebotenen Diensten und einem genutzten Dienst (Reinigung)

Abteilungen und IT-Systeme beschrieben werden. Wie die Bereitstellung eines Dienstes im Detail implementiert, ob beispielsweise eine Schnittstelle über Dateiaustausch oder über einen Web-Service realisiert wird, ist nicht mehr Schwerpunkt der SoaML.

Außerdem soll der Wiederverwendbarkeit in der Modellierung Rechnung getragen werden. D. h., es muss ausgedrückt werden können, dass ein Dienst von mehreren verschiedenen Anbietern zur Verfügung gestellt wird.

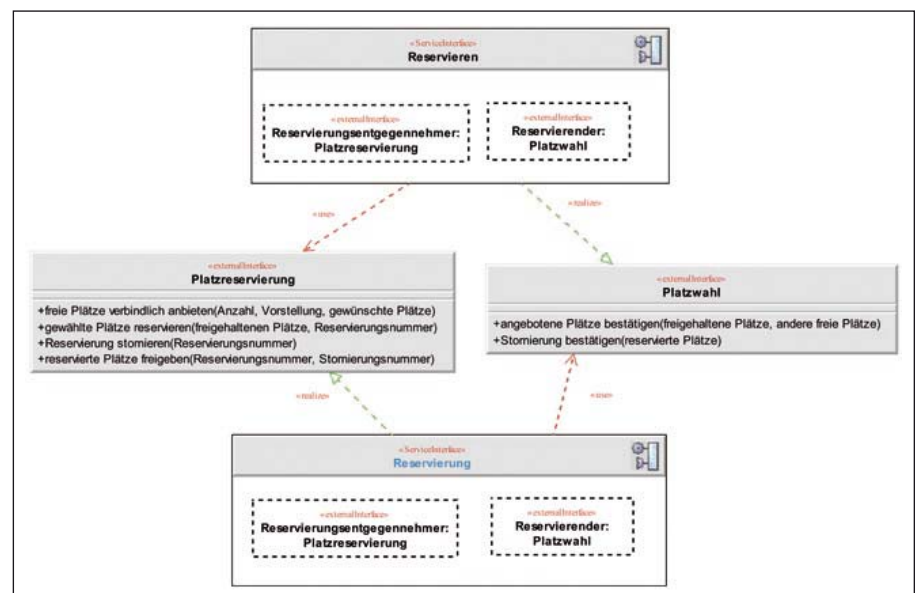


Abb. 2: Service-Interface Reservierung

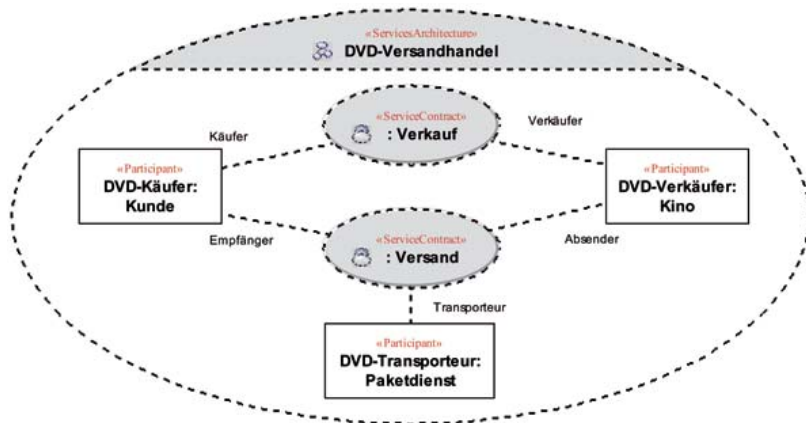


Abb. 3: Services-Architektur für den DVD-Versandhandel

Als elementares Konzept benutzt SoaML „Participants“, die über sogenannte „Service-Points“ Dienste bereitstellen oder diese über „Request-Points“ nutzen. Im Beispiel bietet der Participant Kino drei Dienste an, die durch „Service-Interfaces“ näher spezifiziert werden: Programmkunft, Platzreservierung und DVD-Verkauf. Über das Service-Interface ist definiert, welche Schnittstellen der Participant bereitstellt oder benötigt (Abbildung 2: Service-Interface Reservierung). Die genutzten Schnittstellen beschreiben dabei die Pflichten des Dienst-Nutzers. So ist es nötig, dass bei der Platzreservierung der Reservierende die Plätze auswählt.

Zwei weitere wesentliche Modellierungskonstrukte sind „Service-Verträge“ bzw. „Services-Architekturen“. Beide werden genutzt, um die Zusammenarbeit zwischen verschiedenen Beteiligten darzustellen. Dabei beschreibt ein Service-Vertrag stets die Zusammenarbeit zwischen mehreren Service-Interfaces, wohingegen eine Services-Architektur die Zusammenarbeit mehrerer Participants beschreibt.

Im Beispiel (Abbildung 3: Services-Architektur für den DVD-Versandhandel) sehen wir die Services-Architektur für den DVD-Versandhandel. In der Architektur sind die drei beteiligten Rollen DVD-Käufer, DVD-Verkäufer und DVD-Transporteur definiert. Für jede Rolle ist der Participant hinterlegt, der die Rolle einnimmt. So übernimmt den DVD-Verkauf das Kino, den DVD-Versand der Paketdienst. Zwischen den einzelnen Participants ist die Zusammenarbeit über Service-Verträge geregelt. Die Rollenbindungen zeigen wiederum, welche Rolle innerhalb des Service-Vertrages übernommen wird. So ist der DVD-Verkäufer auch der Absender der DVD.

Der wesentliche Unterschied zwischen einem Service-Vertrag und einer Services-Architektur besteht darin, dass sich bei einem Vertrag alle Parteien kennen und der Vertrag explizit geschlossen wird. Im Beispiel hat das Kino einen Transport-

Vertrag mit dem Paketdienst. Dieser regelt, dass der Paketdienst die Sendung an den Kunden überbringt. Bei einer Services-Architektur kennen sich die Beteiligten meist nur über die Service-Verträge. Die Beteiligten sind für die Zusammenarbeit nur lose gekoppelt. Der Paketdienst weiß beispielsweise nicht, dass das Kino dem Kunden eine DVD verkauft hat. Sollte das Kino später zusätzlich zu DVDs auch T-Shirts oder Poster verkaufen, kann der Shop erweitert werden, ohne dass die Zusammenarbeit mit dem Paketdienst geändert werden muss.

Ist es dem Kunden bei größeren Leistungen (wie beispielsweise dem Bau eines Hauses) zu riskant, auf die einzelnen Lieferanten und Dienstleister angewiesen zu sein, dann bestellt er in der Praxis häufig einen Generalunternehmer, der als Gesamtverantwortlicher die Subunternehmer koordiniert. Dies wird dann über einen Service-Vertrag dokumentiert. Der Generalunternehmer delegiert die einzelnen Pflichten an die Subunternehmer, ist aber stets dem Kunden gegenüber verantwortlich. Als Beispiel für einen Service-Vertrag mit mehr als zwei Rollen, ist in Abbildung 4 (Service-Vertrag für den Versand von Sendungen durch beispielsweise einen Paketdienst) der Versand-Vertrag dargestellt. Die drei Rollen Absender, Empfänger

und Transporteur wurden schon in der Services-Architektur für die Beschreibung der Rollen der verschiedenen beteiligten Participants verwendet. Der Versandvertrag regelt, welche Aufgabe der Transporteur beim Versand übernehmen muss. Er transportiert die Sendung und liefert sie dem Empfänger. Diesen Dienst bietet jeder Participant an, der an einem Service-Point das Service-Interface „Transport“ bereitstellt, in diesem Beispiel der Paketdienst. Damit ist der Paketdienst auch an den Service-Vertrag „Versand“ gebunden. Dasselbe Prinzip gilt für die Nutzung von Services. Zusätzlich zu den Strukturen kann für Service-Verträge oder Service-Interfaces ein Verhalten definiert werden (Abbildung 5: Protokoll für das Service-Interface Platzreservierung). Das Verhalten eines Service-Interfaces beschreibt, in welcher Reihenfolge die bereitgestellten und die benötigten Leistungen aufgerufen werden müssen. D. h. es wird definiert, in welcher Reihenfolge die Aktionen von Anbieter und Nutzer stattfinden können. Im Beispiel werden zuerst freie Plätze durch den Reservierungsentgegennehmenden verbindlich angeboten, die dann durch den Reservierenden bestätigt werden usw.

SoaML baut auf dem geschäftlichen Servicegedanken auf, bei dem auch interaktive, langlebige, zustandsbehaftete Services betrachtet werden sollen. Mit Participant werden Dienst-Anbieter und -Nutzer modelliert. Die Dienste werden über Service-Interfaces typisiert, um eine bidirektionale Kommunikation abbilden und die Dienste wieder verwenden zu können. Über Service-Verträge und Services-Architekturen werden Regeln für die Bereitstellung oder/und Nutzung der Dienste definiert, die durch eine Verhaltensbeschreibung ergänzt werden können. Alle Elemente beziehen sich allerdings auf die Service-Modellierung, weshalb schon Stimmen laut wurden, die meinten, der treffendere Name wäre ServiceML. Hier bleibt abzuwarten, ob in den kommenden Jahren auch andere Aspekte der SOA aufgenommen werden. In den Projekten wird

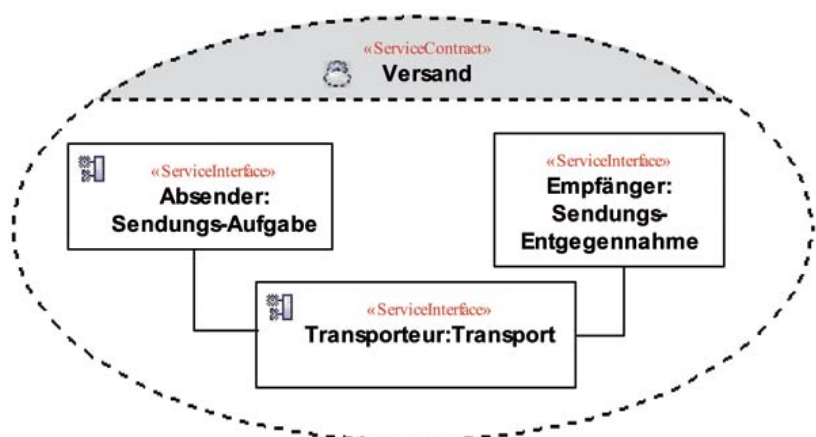


Abb. 4: Service-Vertrag für den Versand von Sendungen durch beispielsweise einen Paketdienst

SoaML Stereotyp	UML Metaklasse	Beschreibung
ServiceInterface	Class oder Interface	Definiert die Schnittstelle zu einem Service Point oder einem RequestPoint und ist der Typ der Rollen in einem ServiceContract Legt fest, wie ein Participant interagieren muss, um einen Service anzubieten oder zu nutzen
ServiceContract	Collaboration	Spezifiziert die Abmachungen zwischen Anbietern und Nutzern eines Service
ServicesArchitectur	Collaboration	High-Level-Sicht einer SOA, die definiert, wie eine Gruppe von Participants in einer Gemeinschaft für einen bestimmten Gesamtzweck zusammenarbeiten, indem sie Services anbieten und nutzen
ServicePoint	Port	Angebot eines Services typisiert mit einem ServiceInterface oder Interface
RequestPoint	Port	Nutzung eines Services typisiert mit einem ServiceInterface oder Interface
Participant	Component	Bietet einen Service über einen ServicePoint an oder nutzt ihn über einen RequestPoint

Tabelle - Die wichtigsten SoaML-Stereotypen

vor allem nach Deployment-Aspekten sowie der Service-Registry gefragt. Die Elemente eignen sich sehr gut, um Gesamtansichten darzustellen oder einen Überblick zu modellieren, Abhängigkeiten können gut sichtbar gemacht werden. Auch wenn es etwas Aufwand macht, die ca. 150 Seiten lange Spezifikation der SoaML zu verstehen, lohnt es sich, wenn man in einem komplexen Umfeld unterwegs ist. Gerade wenn mehrere Systeme verzahnt zusammenarbeiten und bisher wenig prozessorientiert dokumentiert wurde, sind häufig die fachlichen Abhängigkeiten der Verallgemeinerung zum Opfer gefallen. Mit den Elementen der SoaML kann hier die Dokumentation konsequent ergänzt werden.

Fazit

SoaML hilft, komplexe, zustandsbehaftete Services zu verstehen und zu nutzen.

Die SoaML wird bei der OMG durch eine Arbeitsgruppe, an der auch die MID GmbH beteiligt ist, entwickelt. Inzwischen gibt es von einigen Herstellern Profile für UML Modellierungswerkzeuge. So liefert die MID GmbH die neueste Version 10.1 der Modellierungsplattform Innovator mit einem SoaML-Profil aus, das als Grundlage für die MID ModellierungsMethodik M³ für SOA dient.

Ein Überblicksposter zur MID ModellierungsMethodik M³ für SOA gibt es als Gratis-Download unter www.mid.de/PosterM3SOA

Ebenso finden Sie das vollständige SoaML-Modellierungsbeispiel mit einer detaillierten Beschreibung der verwendeten UML-Modellelemente unter www.mid.de/WhitepaperSoaML

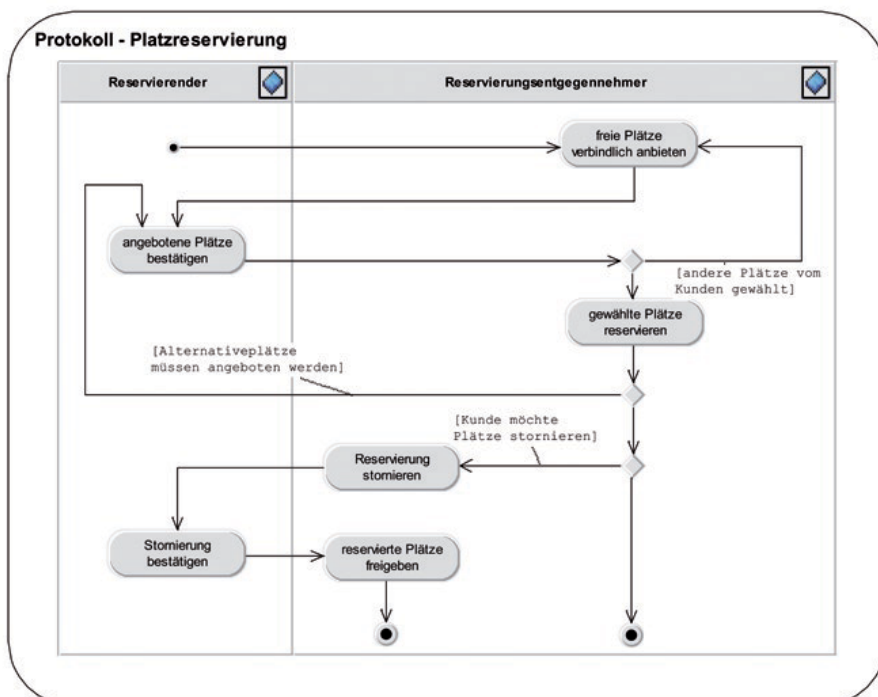


Abb. 5: Protokoll für das Service-Interface Platzreservierung

Gratis-Live Webcast

Fit für SOA - Modellieren mit SoaML

Freitag, 7. August 2009
11:00 - 11:30 Uhr

Anmelden unter:
www.mid.de/WebcastSoaML



Maria Deeg
Project Manager, MID