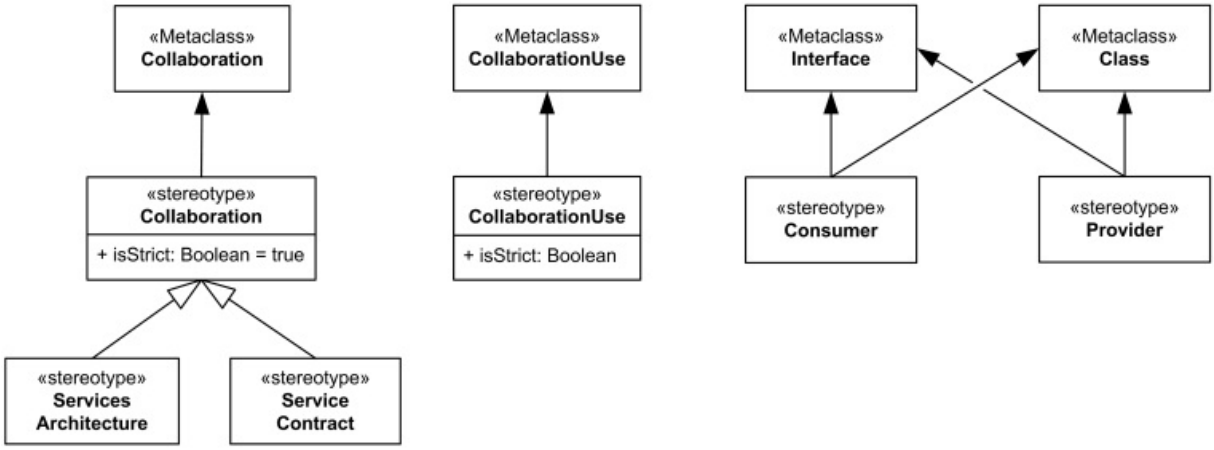
Service Oriented Architecture Modeling Language (SoaML)

UML Profile Diagram Example

UML Profile for Service Oriented Architecture is described in **Service Oriented Architecture Modeling Language (SoaML)** specification. "The goals of SoaML are to support the activities of service modeling and design and to fit into an overall model-driven development approach" ***[SoaML 1.0 Beta 2]***. The SoaML specification contains both a SoaML metamodel and a SoaML UML profile.

The SoaML UML profile supports modeling of service-oriented architectures, including specification of systems of services, specification of individual service interfaces, and specification of service implementations.

A **service** is defined there as a value delivered to another through a well-defined interface and available to a community (which may be the general public). A service results in work provided to one by another.



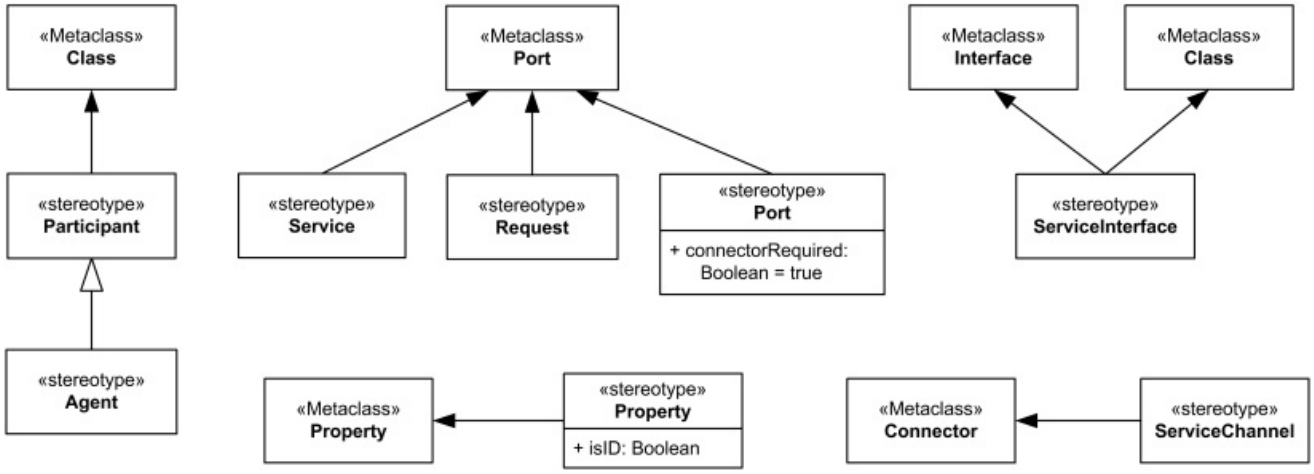
SoaML UML Profile - Contracts

Collaboration, Service Contract or Services Architecture represents a pattern of interaction between roles. In SoaML this interaction may be informal and loosely defined as in a requirements sketch. It may also represent formal agreements or requirements that must be fulfilled exactly.

Service description specifies how participants interact to provide or use a service. In SoaML there are three ways to specify a service interaction – a UML Interface, a Service Interface and a Service Contract.

Consumer models the interface provided by the consumer of a service. The consumer of the service receives the results of the service interaction. The consumer will normally be the one that initiates the service interaction.

It looks weird that Consumer and Provider both extend Interface and Class metaclasses. The SoaML explains that Interface is used in the case of a non composite service contract while Class - in the case of a composite service contract.



SoaML UML Profile - Services

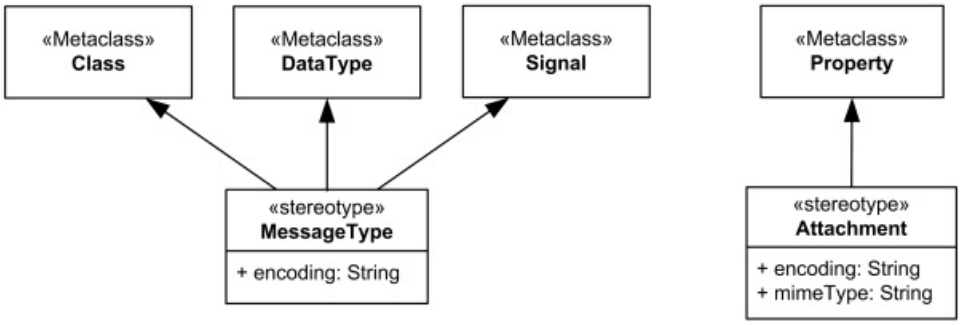
Participants are either specific entities or kinds of entities that provide or use services. Participants can represent people, organizations or information system components. Participants may provide any number of services and may consume any number of services. Participants provide or consume services via ports.

Agent is autonomous entity that can adapt to and interact with its environment. Agent can be software agent, hardware agent, firmware agent, robotic agent, human agent, and so on.

Port is the part or feature of a participant that is the interaction point for a service – where it is provided or consumed. A port where a service is offered may be designated as a «Service» port and the port where a service is consumed may be designated as a «Request» port.

Request extends Port to specify a feature of a Participant that represents a service the Participant needs and consumes from other participants. The request is defined by a Service Interface. A request port is a "conjugate" port - the provided and required interfaces of the port type are inverted, creating a port that uses the port type rather than implementing the port type.

Service represents a feature of a Participant that is the offer of a service by one participant to others using well defined terms, conditions and interfaces. A Service designates a Port that defines the connection point through which a Participant offers its capabilities and provides a service to clients.



SoaML UML Profile - Service Data

Message Type defines information exchanged between service consumers and providers. Message Type should generally only be applied to Data Type since it is intended to have no identity. However, SoaML recognizes that many existing models do not clearly distinguish identity, either mixing Class and DataType, or only using Class. Because of this, another odd thing is that SoaML allows Message Type to extend Class as well as Data Type.

Attachment is a part of a Message that is attached to rather than contained in the message.

Noticed a spelling error? Select the text using the mouse and press Ctrl + Enter.

Follow @uml_diagrams THIS SITE IS ECO-FRIENDLY by Kirill Fakhroutdinov

This document describes UML versions up to ***UML 2.5*** and is based on the corresponding **OMG™ Unified Modeling Language™ (OMG UML®)** specifications. UML diagrams were created in **Microsoft® Visio®** 2007-2016 using ***UML 2.x Visio Stencils***. ***Lucidchart*** is a nice, free UML tool that I recommend for students.
You can send your comments and suggestions to webmaster at **webmaster@uml-diagrams.org**.

Copyright © 2009-2018 uml-diagrams.org. All rights reserved.