What is SoaML? How to Draw SoaML Diagrams?

Written Date : September 3, 2013

SoaML in Visual Paradigm

Visual Paradigm supports the modeling of SOA with SoaML. In Visual Paradigm, the SoaML profile is organized into five SoaML diagram types, namely, the Service Interface Diagram, Service Participant Diagram, Service Contract Diagram, Services Architecture Diagram, and Service Categorization Diagram. Each of these provides a unique view to describe and help understand services and the services architecture. Combined with the use of UML diagrams like sequence diagram, activity diagram, BPMN business process diagram, and OMG business motivation model (BMM), you can describe SOA as well as indicate its technical and business relevance.

What is This Tutorial About?

This tutorial is written to explain what SoaML is, how to use SoaML for SOA, and how to draw the various SoaML diagrams in Visual Paradigm. There are mainly 5 parts in this tutorial. Each part explains one of the SoaML diagram types in detail, with SoaML tool description, diagram definition, and the steps to create the diagram.

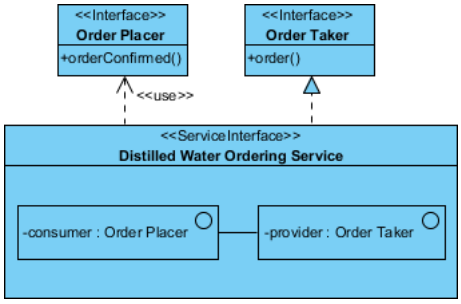
The example that will be used in this tutorial is a basic version of a distilled water supplier. You will draw different SoaML diagrams to explain the different aspects of the distilled water ordering and delivery service.

Preparation

To perform the steps in this tutorial, make sure you have Visual Paradigm downloaded and installed. You may click here to download Visual Paradigm if you do not have it installed.

To avoid disrupting your production environment during the tutorial, please create a new project to perform the steps in this tutorial.

Part I - Drawing a Service Interface Diagram



Service Interface Diagram Tools

- Service Interface
- Interface
- Role
- Connector
- Capability
- Expose

- Dependency
- Realization
- Usage
- Message Type
- Milestone

What is a Service Interface Diagram?

A Service Interface Diagram is one of the most important SoaML diagram types. To understand what a service interface diagram is, you must first know about a key concept of SoaML - service.

As specified in the SoaML specification, a service is "value delivered to another through a well-defined interface". In SoaML, a service can be specified using three approaches: simple interface, service interface, and service contract. A simple interface defines a one-way service that does not require a protocol. Such a service can be used with anonymous callers, and the participants know nothing about the caller. A service interface allows for bi-directional services. Such a service involves communication between the provider and consumer of services in completing services. A service contract defines how participants work together to exchange value, and we will talk about it when introducing the service contract diagram.

A Service Interface Diagram allows for the modeling of service specifications. You can model simple interfaces and service interfaces in a service interface diagram. Now, follow the steps below to draw a service interface diagram.

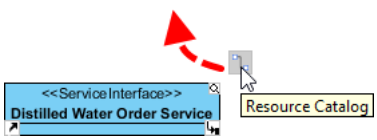
How to Draw a Service Interface Diagram?

1. In a new project, create a service interface diagram by selecting **Diagram > New** from the toolbar. In the **New Diagram** window, enter "service interface diagram" in the search field, click **Next**. Then, fill in the **Diagram Name** and **Description** (if any). Click **OK** to confirm diagram creation.

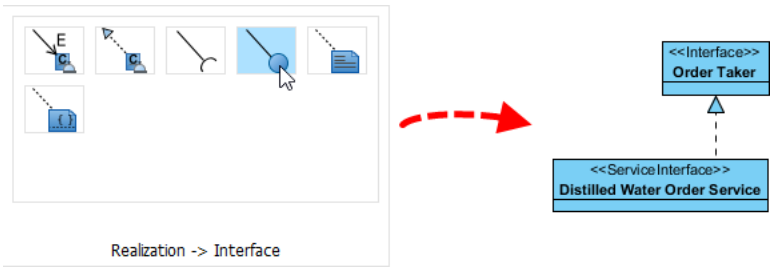
2. We are going to create a service interface for the distilled water ordering service. Select **Service Interface** from the diagram toolbar.

3. Click on the diagram to create a service interface. Name it *Distilled Water Ordering Service*.

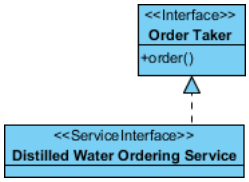
4. We are going to define the interface for the provider of the distilled water ordering service. Move your mouse pointer over the service interface *Distilled Water Ordering Service*. Drag out the **Resource Catalog** icon at the top right.



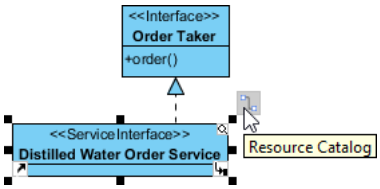
5. Release the mouse button above the service interface. Select **Realization -> Interface** from the Resource Catalog to create the interface, and name it *Order Taker*.



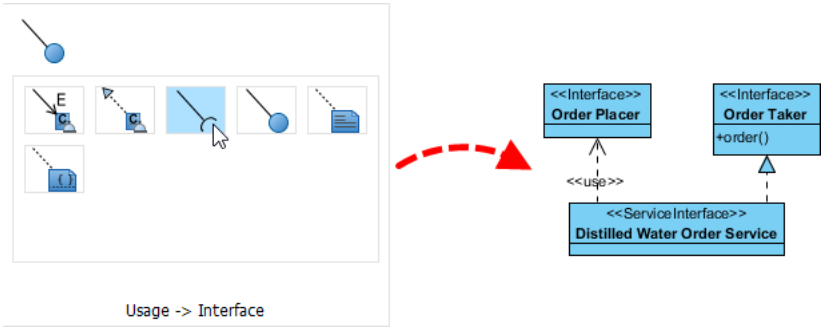
6. The provider interface contains operations that may be invoked during the course of the service. It is a must for the service provider to support the operation defined. The Order Taker is responsible for processing customers' orders. Add an operation *order()* in *Order Taker* by right-clicking on *Order Taker* and selecting **Add > Operation** from the popup menu.



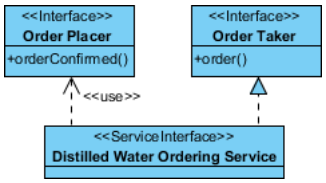
7. We are going to define the interface for the consumer of the distilled water ordering service. Move your mouse pointer over the service interface *Distilled Water Ordering Service*. Drag out the Resource Catalog icon at the top right.



8. Release the mouse button above the service interface. Select **Usage -> Interface** from the Resource Catalog to create the interface, and name it *Order Placer*.

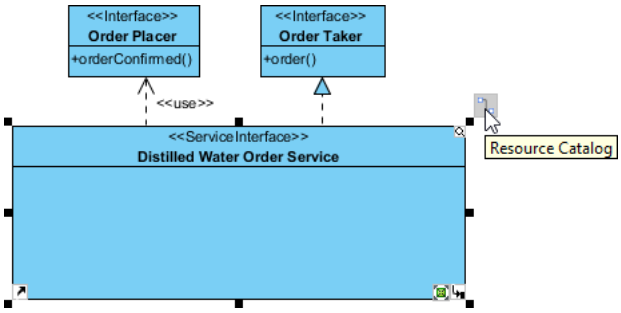


9. Add an operation *orderConfirmed()* in it. This indicates that the consumer of the service must support the *orderConfirmed()* operation.

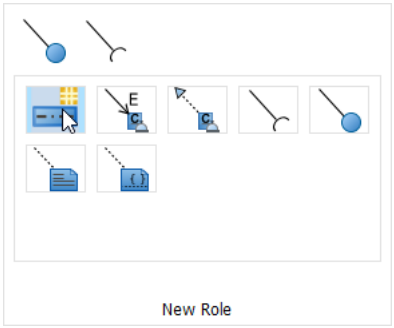


10. Let's define the roles of the provider and consumer in the distilled water ordering service. The roles have to be defined inside the body of the service interface. So let's resize the service interface *Distilled Water Delivery Service* to make it bigger.

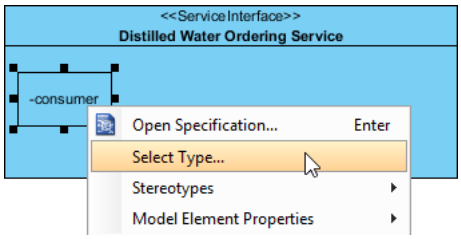
11. Click on the **Resource Catalog** icon of the service interface *Distilled Water Ordering Service*.



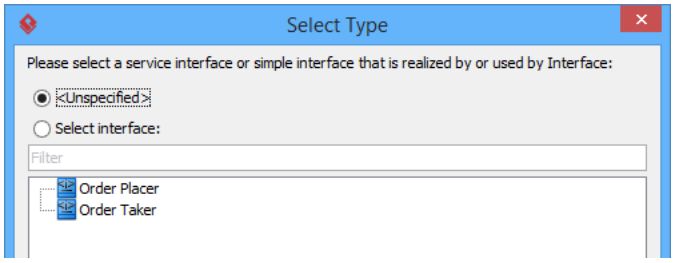
12. Select **New Role** from the Resource Catalog.



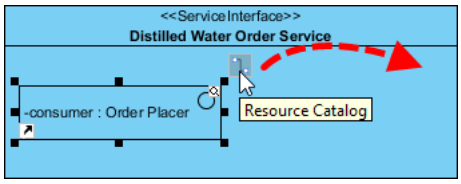
13. The type of consumer role is *Order Placer*. Right-click on the role and select **Select Type...** from the popup menu.



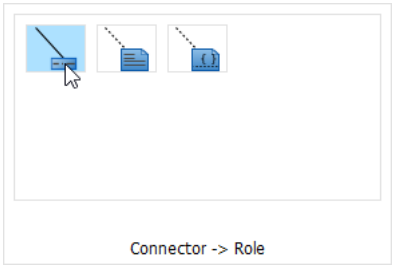
14. Select *Order Placer* in the popup window. Click **OK**.



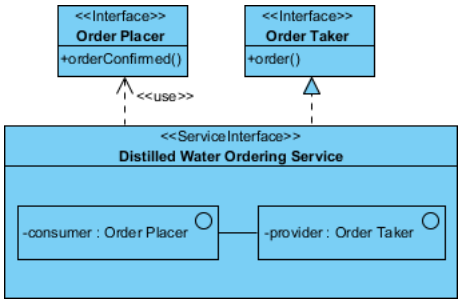
15. Create the *provider* role from the *consumer* role. Drag out the **Resource Catalog** icon from the *consumer* role.



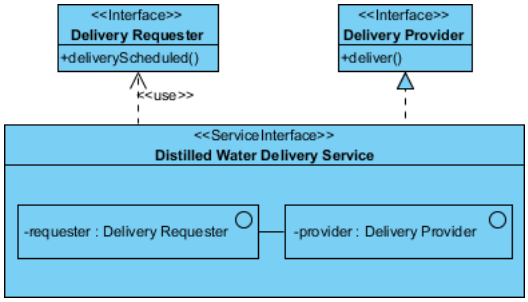
16. Select **Connector Role** from the Resource Catalog.



17. Similar to how you set the type for the consumer role, set *Order Taker* to be the type of the provider role. Up to now, your service interface diagram should look like this:



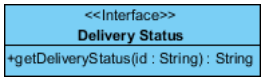
18. You've completed defining the distilled water ordering service. Now, apply the skills you've learned in the previous steps to define the distilled water delivery service in a new service interface diagram. Your diagram should look like this:



19. Before we end this section, let's also try to create a simple interface. As mentioned before, a simple interface defines a one-way service that requires no protocol. Create a new service interface diagram first.

20. Let's create a simple interface for querying the order delivery status. Click on the diagram to create a simple interface and name it *Delivery Status*.

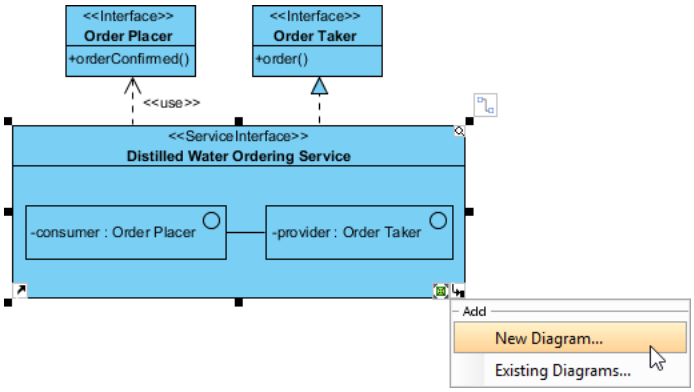
21. Add an operation *getDeliveryStatus(id:String):String* to it.



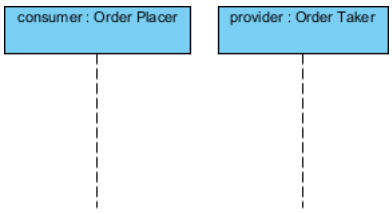
Part II - Specifying Choreography Using a UML Sequence Diagram

Service choreography defines the interaction between the provider and consumer in completing a service. A UML sequence diagram can be used for specifying service choreography. In Visual Paradigm, you can add a sub-sequence-diagram to a service interface for such a purpose. Now, try to specify the service choreography for the service interface Distilled Water Ordering Service defined in the previous section.

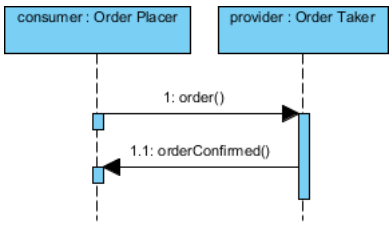
1. Open the first service interface diagram where the service interface *Distilled Water Ordering Service* was defined.
2. Click on the service interface *Distilled Water Ordering Service*.
3. Click on the tiny resource icon at the bottom right of the shape and select **New Diagram...** from the popup menu.



4. The **New Diagram** window is opened. In the **New Diagram** window, enter "sequence diagram" in the search field, click **Next**. Then, fill in the **Diagram Name** and **Description** (if any).
5. Click **OK** to confirm diagram creation.
6. A sequence diagram is created, with lifelines *consumer* and *provider* in it. The lifelines are created by following the roles you defined in the service interface.

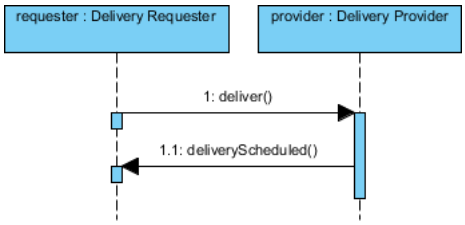


7. Specify how the consumer interacts with the provider of the service by drawing sequence messages between the two lifelines. The *consumer* begins by invoking the *provider's order()* method. The provider will then react by calling the *consumer's orderConfirmed()* method.

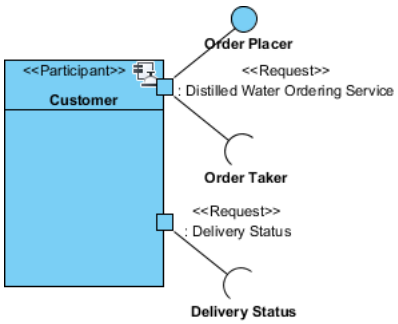


In practice, you can specify an optional call with the use of an opt fragment.

8. Now, try to specify the choreography of the distilled water delivery service, using a UML sequence diagram. Your diagram should look like this:



Part III - Drawing a Service Participant Diagram



Service Participant Diagram Tools

- Participant
- Agent
- Part
- Property
- Service Port
- Request Port
- Port
- Service Channel
- Connector
- Capability
- Usage

What is a Service Participant Diagram?

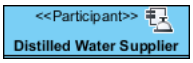
In SoaML, a participant represents a certain party or component that provides and/or consumes service(s). Participants can be software components, organizations, systems, or individuals.

A Service Participant Diagram allows for modeling primarily the participants that play role(s) in services architectures. It also presents the services provided and used by these participants. Now, try to draw a service participant diagram for the distilled water supplier.

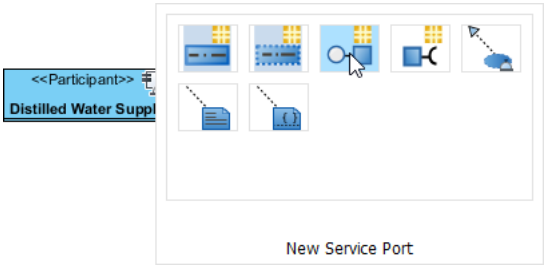
How to Draw a Service Participant Diagram?

1. Create a service participant diagram by selecting **Diagram > New** from the toolbar. In the **New Diagram** window, enter "participant diagram" in the search field, click **Next**. Then, fill in the Diagram Name and Description (if any), click **OK** to confirm diagram creation.

2. The distilled water supplier participant offers the distilled water ordering service. Let's create a participant for it. Select **Participant** from the diagram toolbar and click on the diagram to create a participant. Name it *Distilled Water Supplier*.

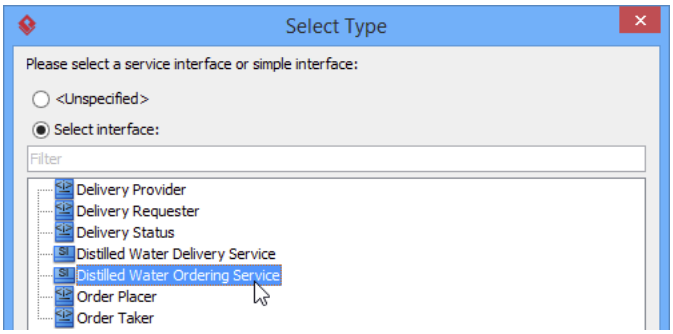


3. In SoaML, a port is used to represent the point of interaction on a participant where a service is provided or consumed. A **<<Service>>** port typed by a service interface is known as a service port. It represents the point where the service is offered by the participant, based on the typed service interface. Let's create such a port on the *Distilled Water Supplier* participant. Click on the **Resource Catalog** icon at the top right of the *Distilled Water Supplier* participant, and select **New Service Port** from the Resource Catalog.

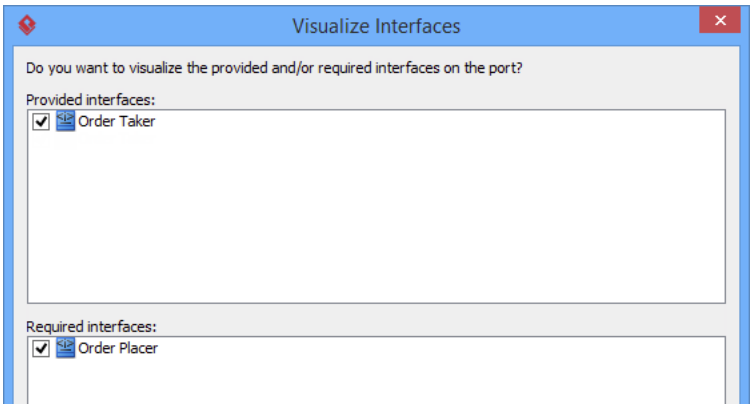


4. A service port is created on the left-hand side of the participant, following the convention. Let's type the port. Right-click on the port and select **Select Type...** from the popup menu.

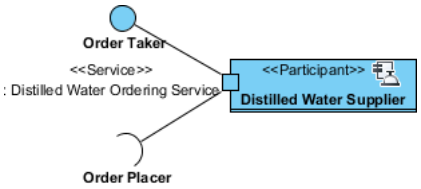
5. In the **Select Type** window, select *Distilled Water Ordering Service* and click **OK**.



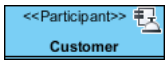
6. You are prompted to visualize the provider and consumer interface associated with the service interface *Distilled Water Ordering Service*. Keep the default selection and click **OK**.



The port, provider, and consumer interfaces are created automatically. Note that the port is said to provide the *Order Taker* interface and requires the *Order Placer* interface.



7. Just as we want to define the service provided by the distilled water supplier, we want to define what services a participant consumes. The Customer participant is a consumer of the distilled water ordering service. Create the *Customer* participant on the diagram.



8. A participant requests services from other participants (who provide the services). The request for service can be defined using a **<<Request>>** port. Make use of the resource-centric interface to create a **<<Request>>** port on the Customer participant.

9. Type the **<<Request>>** port. Right-click on the **<<Request>>** port and select **Select Type...** from the popup menu.

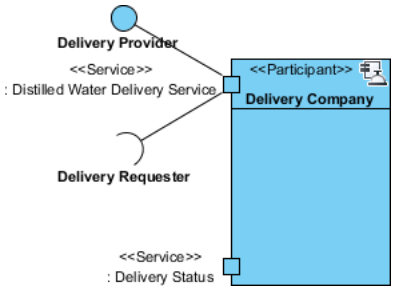
10. In the **Select Type** window, select *Distilled Water Delivery Service* and click **OK**.

11. You are prompted to visualize the provider and consumer interface associated with the service interface *Distilled Water Ordering Service*. Keep the default selection and click **OK**. Up to now, your diagram should look like this:

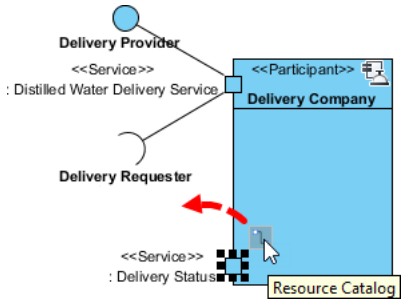
Note that both the *Customer* and *Distilled Water Supplier* have a port typed with the *Distilled Water Ordering Service*. The distilled water supplier is the provider of the service, while the customer is the consumer of the service. The distilled water supplier provides the *Order Taker* interface and requires the *Order Placer* interface. The customer provides the *Order Placer* interface and requires the *Order Taker* interface.

12. A participant can be the provider and consumer of many services. Remember the distilled water delivery service? The delivery company is the provider of the service, and the distilled water company is the consumer. Apply the skills you have just learned to add the participant, **<<Service>>** and **<<Request>>** port to the diagram. The result should look like this:

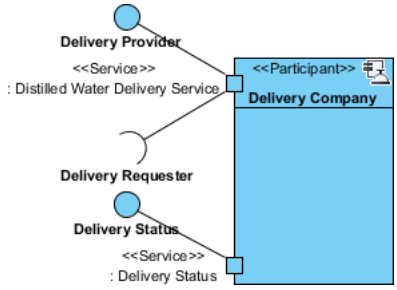
13. The delivery company also provides the delivery status service. Add a **<<Service>>** port to the *Delivery Company* participant and type it with the *Delivery Status* service interface.



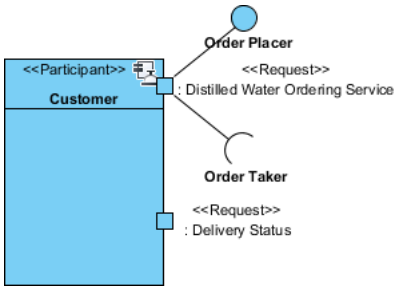
14. Visualize the provided interface of the port. Drag out the **Resource Catalog** icon from the port and select **Realization -> Interface** from the Resource Catalog.



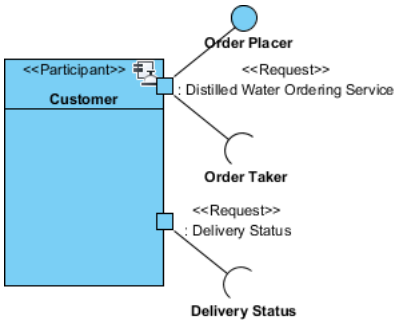
15. When you release the mouse button, the provided interface is visualized - you don't need to name it manually.



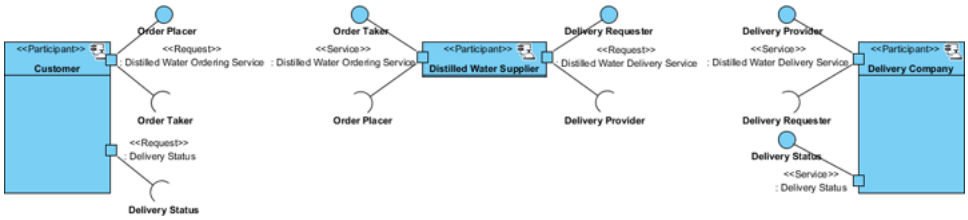
16. The customer can check the delivery status. Add a **<<Request>>** port to the *Delivery Company* participant and type it with the *Delivery Status* service interface.



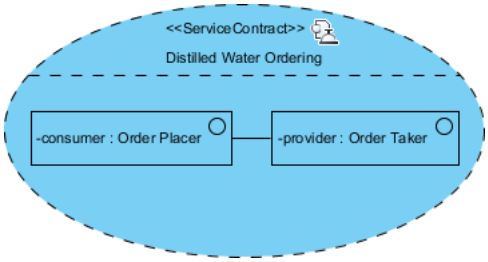
17. Make use of the **Usage -> Interface** resource to visualize the required interface of the port.



Up to now, your service participant diagram should look like this:



Part IV - Drawing a Service Contract Diagram



Service Interface Diagram Tools

- Service Contract
- Role
- Provider
- Consumer
- Connector
- Dependency

What is a SoaML Service Contract Diagram?

As mentioned before, there are three approaches to specify a service. You've learned how to specify a service with the two interface-based approaches - simple interface and service interface. In this section, we will talk about a service contract.

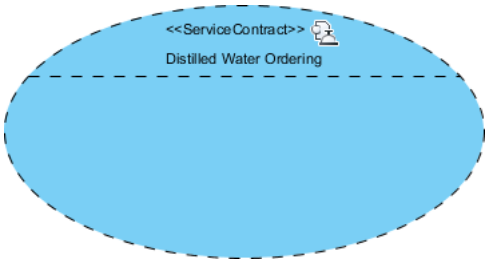
A Service Contract defines the agreement between parties about how a service is to be provided and consumed. "Agreement" here refers to interfaces, choreography, and any terms and conditions. Interacting participants MUST agree to the agreement in order for the service to be enacted.

How to Draw a SoaML Service Contract Diagram?

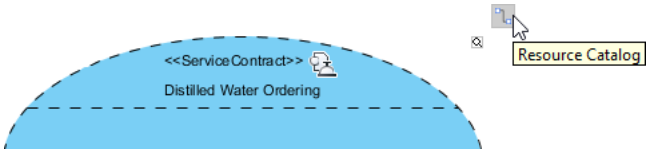
Let's draw service contract diagrams to show the agreements of the three services.

1. To create a Service Contract Diagram, select **Diagram > New** from the toolbar. In the **New Diagram** window, enter "service contract diagram" in the search field, click **Next**. Then, fill in the **Diagram Name** and **Description** (if any), click **OK** to confirm diagram creation.

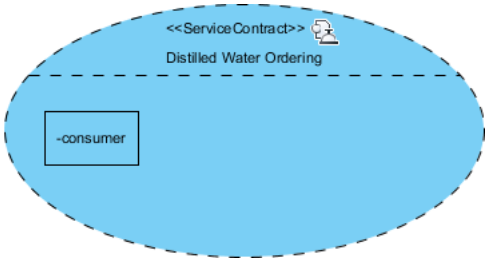
2. Select **Service Contract** from the diagram toolbar and click on the diagram to create a service contract. Name it *Distilled Water Ordering*.



3. Visualize the roles of the consumer and provider in the distilled water ordering service. Click on the **Resource Catalog** icon of the *Distilled Water Ordering* contract, and then select **New Role** from the Resource Catalog.

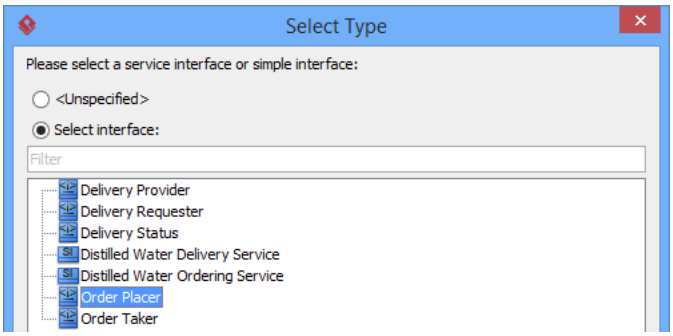


4. Name the role *consumer*. A *consumer* is the participant that has some needs and requests for a service from a provider.



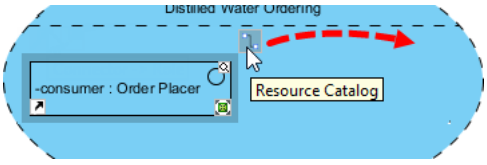
5. Let's type the role. Right-click on the role and select **Select Type...** from the popup menu.

6. In the **Select Type...** window, select *Order Placer* and click **OK**.



The type of *consumer* role is *Order Placer*. This is the interface that a consumer will and must implement to use the distilled water ordering service.

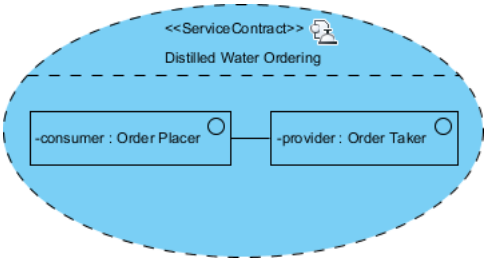
7. Visualize the role of the provider in the distilled order ordering service. Create a new role from the consumer role with the help of the Resource Catalog.



8. Name the role *provider*. It is the role that provides something of value to the consumer.

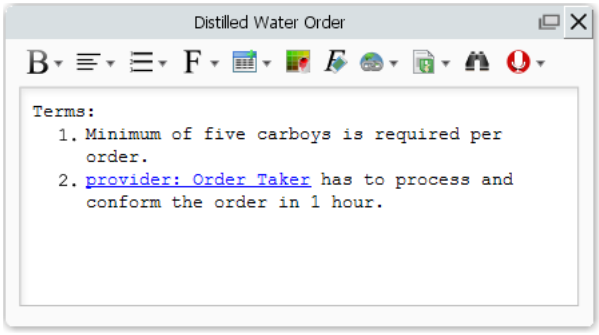
9. Let's type the role. Right-click on the role and select **Select Type...** from the popup menu.

10. In the **Select Type...** window, select *Order Taker* and click **OK**. The type of *consumer* role is *Order Taker*. This is the interface that a provider will and must implement to provide the distilled water ordering service.



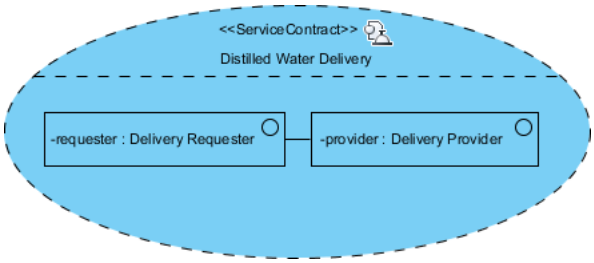
11. We can specify the terms and conditions of the contract in the description of the service contract. Select the **Distilled Water Order** contract on the diagram.

12. In the **Description Pane**, enter the following terms that the consumer and provider have to follow. There is no standard or rule to govern the way you describe the terms. Just make sure everything is clearly described.

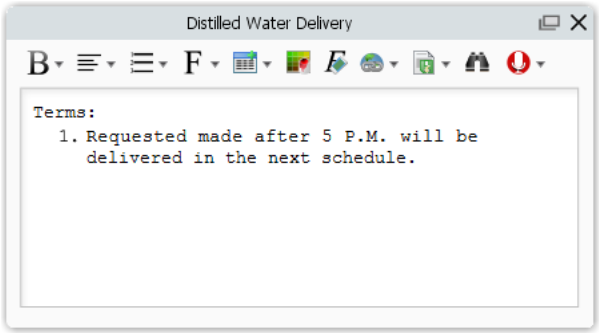


As a side note, you can add a model element link, which is the hyperlink shown in the image above, by clicking the **Add Model Element...** button in the toolbar of the **Description Pane**.

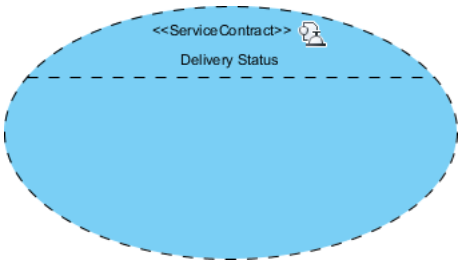
13. Apply the skills you have just learned to draw a new service contract for the distilled water delivery service. The contract should involve the consumer role *requester* and provider role *provider*, and are typed with *Delivery Requester* and *Delivery Provider* respectively. The result should look like this:



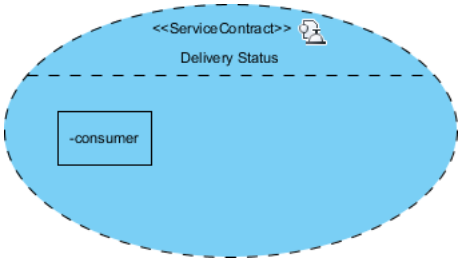
And these are the terms of the *Distilled Water Delivery* contract:



14. Let's create a contract for the *Delivery Status* service. The case is a bit tricky, so let's walk through the steps together. Create a new service contract diagram and a *Delivery Status* contract first.

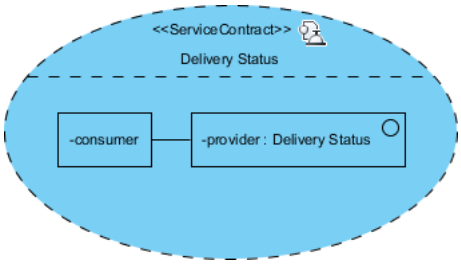


15. Create a role *consumer* in the contract. A *consumer* is the consumer of the delivery status service.

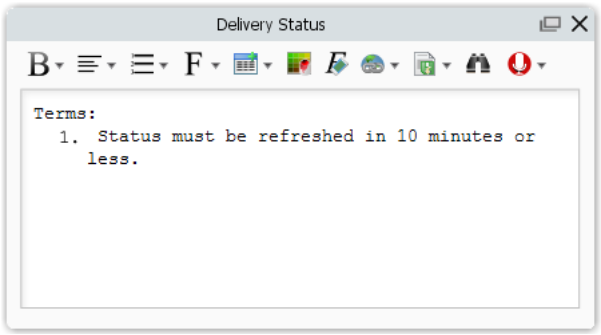


Here we have a problem. Remember, the delivery status service was defined as a simple service (in our service interface diagram)? A simple service is a one-way service that does not require a protocol. The provider need not know about the consumer. Thus, there is no required interface defined for such a service. So how do we type the role *consumer* here? The answer is - you don't need to type it. Just leave the interface part unspecified when you need to indicate the use of a simple service.

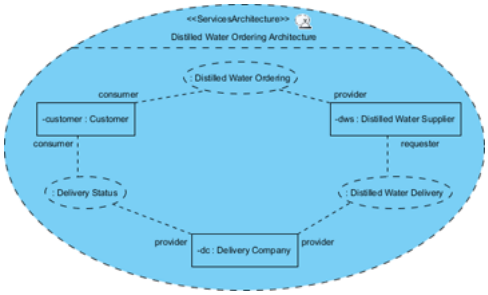
16. Complete the rest of the service contract diagram to make it look like:



And here are the terms:



Part V - Drawing a Services Architecture Diagram



Service Architectures Diagram Tools

- Services Architecture
- Internal Participant
- External Participant
- Service Contract Use
- Role Binding

What is a SoaML Services Architecture Diagram?

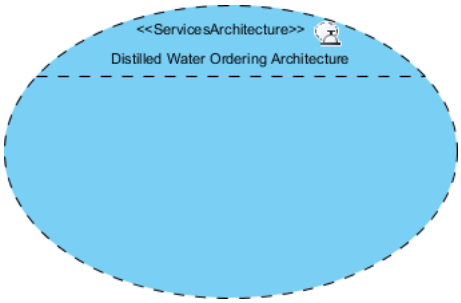
Until now, we have defined and specified the services to use in distilled water ordering. We also have defined the participants. However, nothing shows us how the customer, distilled water supplier, and delivery company work together, providing and using the services in the services architecture. To represent this information, we can draw a services architecture diagram.

A services architecture, abbreviated as SOA, shows the participant roles that provide and consume services to fulfill a certain purpose.

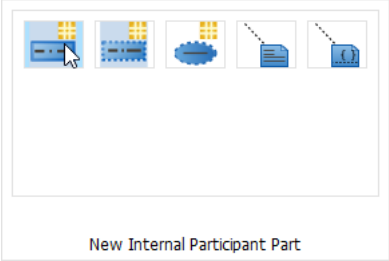
How to Draw a SoaML Services Architecture Diagram?

1. To create a Service Contract Diagram, select **Diagram > New** from the toolbar. In the **New Diagram** window, enter "services architecture diagram" in the search field, click **Next**. Then, fill in the **Diagram Name** and **Description** (if any), click **OK** to confirm diagram creation.

2. Select **Services Architecture** from the diagram toolbar and click on the diagram to create a services architecture. Name it *Distilled Water Ordering Architecture*.

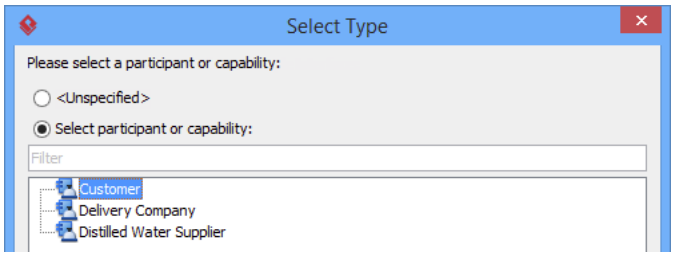


3. Let's start by adding the participant role customer to the services architecture. Click on the Resource Catalog icon of *Distilled Water Ordering Architecture* and select **New Internal Participant Part** from the Resource Catalog.

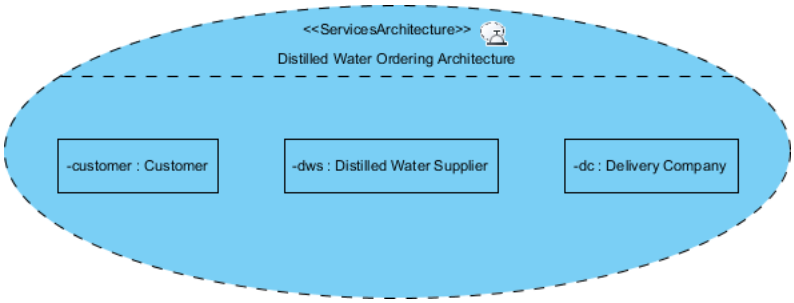


4. Let's type the participant role. Right-click on the part and select **Select Type...** from the popup menu.

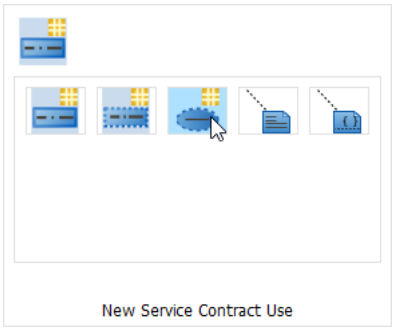
5. In the **Select Type...** window, select *Customer* and click **OK**. *Customer*, *Delivery Company*, and *Distilled Water Supplier* are the participants we defined in the service participant diagram.



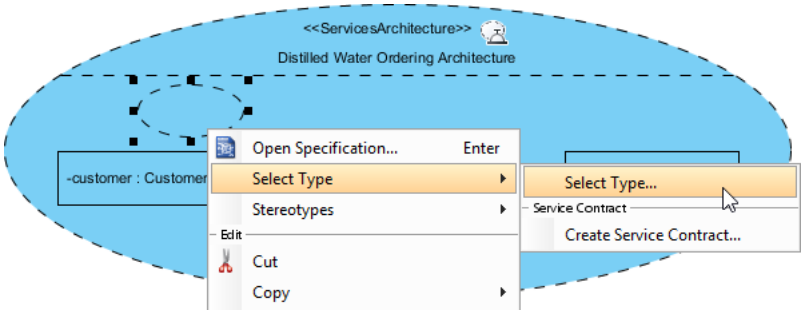
6. Add two more parts: *dws* and *dc*. Type them with *Distilled Water Supplier* and *Delivery Company*. Your services architecture diagram should look like this:



7. Let's visualize the service contracts that the participants provide or use within the SOA. Click on the **Resource Catalog** icon of *Distilled Water Ordering Architecture* and select **New Service Contract Use** from the Resource Catalog.

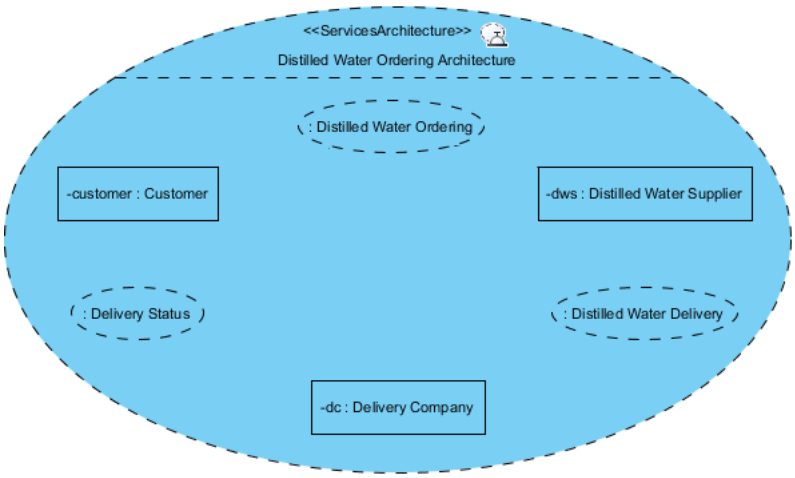


8. Give an empty name to the service contract use. Type it by right-clicking on it and selecting **Select Type > Select Type...** from the popup menu.



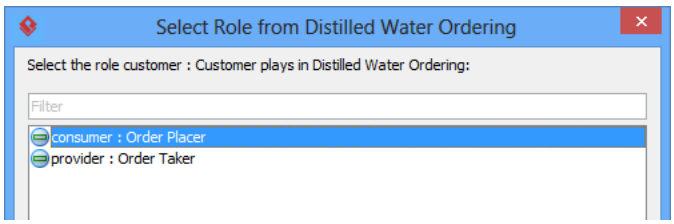
9. In the **Select Type...** window, select *Distilled Water Order* and click **OK**. *Distilled Water Ordering*, *Distilled Water Delivery*, and *Delivery Status* are the service contracts we defined in the service contract diagram.

10. Add two more service contract uses. Type them with *Distilled Water Delivery* and *Delivery Status*. Your services architecture diagram should look like this:

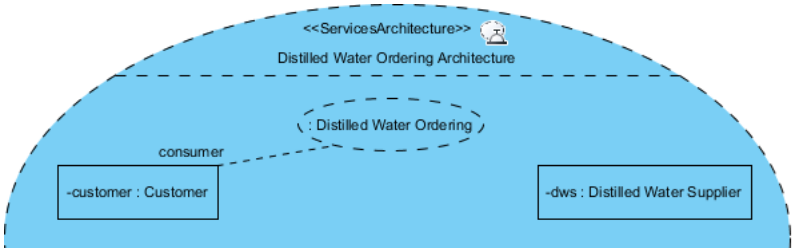


11. Participants provide and use the services within the SOA. This is modeled as role bindings to the related service contracts. Drag the **Resource Catalog** icon from the participant role *Customer* to service contract use: *Distilled Water Ordering* to bind them.

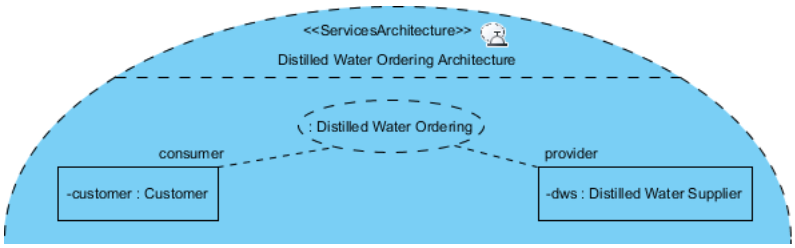
12. You are prompted to select the role the customer plays in the distilled water ordering service. Select consumer: *Order Placer* and click **OK**.



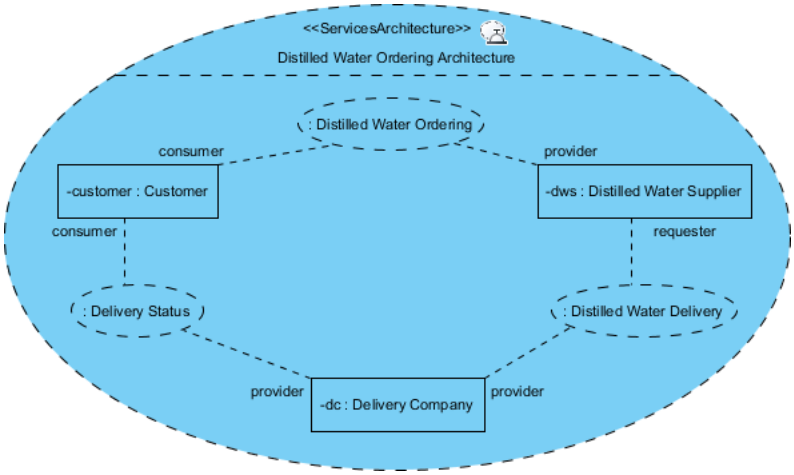
The participant role and service contract use are connected.



13. The distilled water supplier provides the distilled water ordering service. Let's bind the participant role *Distilled Water Supplier* with service contract use *Distilled Water Ordering*. Select *provider* to be the role.



14. Apply the skills you have just learned to bind the other participant roles and service contract uses. The result should look like this.



Summary of SoaML Notations

The SoaML notations you can apply in your SoaML diagrams are listed below. Notation descriptions were extracted from OMG SoaML specification and OMG UML specification.

SoaML Notations	Description
Service Interface	A service interface represents the service itself.
Interface	An interface is a simple interface that defines a one-way service that does not require a protocol.
Role	A role defines the role of a provider or consumer in a service.
Connector	A connector connects roles and interfaces.
Capability	The ability to produce an outcome that achieves a result.
Expose	Expose is a kind of dependency used to indicate what capabilities that are required by or are provided by a participant should be exposed through a service interface.
Dependency	Indicates that an element depends on another.
Realization	A service interface specifies the receptions and operations it receives through interface realization.
Usage	A service interface specifies its required needs through usage dependencies to interfaces.

Message Type	A message type is a specification of information that is exchanged between the consumers and providers of a service.
Milestone	A milestone defines a signal that is sent to an abstract observer.
Participant	A participant represents some party or component that provides and/or consumes services. Participants can be people, organizations, systems, etc.
Agent	Autonomous entities that can adapt to and interact with their environment. Agents can be software agents, hardware agents, firmware agents, robotic agents, human agents, etc.
Part	The part has a type that specializes/realizes the type of role.
Property	An identifying property for uniquely identifying the containing classifier. This is also known as a primary key.
Service Port	A service port is a kind of port that represents the point of interaction on a participant where a service is actually provided.
Request Port	A request port is a kind of port that represents the point of interaction on a participant where a service is actually consumed.
Port	A service uses a port to indicate the feature or interaction point through which a classifier interacts with other classifiers.
Service Channel	A service channel is a communication path between services and requests within an architecture.
Service Contract	A service contract is the specification of the agreement between providers and consumers of a service as to what information, products, assets, value, and obligations will flow between the providers and consumers of that service.
Provider	For modeling the interface provided by the provider of a service.
Consumer	For modeling the interface provided by the consumer of a service.
Services Architecture	Services Architecture describes how participants work together for a purpose by providing and using services expressed as service contracts.
Internal Participant	An internal role that is involved in a services architecture.
External Participant	An external role that is involved in a services architecture.
Service Contract Use	Indicates the ability of an owning classifier to fulfill a service contract or adhere to a services architecture.
Role Binding	A role binding binds each of the roles to a part of the containing architecture to indicate what role each part in the owning architecture play.

Resources

1. Complete SoaML Example - Distilled-Water-Ordering_SoaML.vpp

Related Links

- Visual Paradigm Feature - SoaML Modeling

Attributions

- Object Management Group - Service oriented architecture Modeling Language (SoaML) Specification
- Object Management Group - Unified Modeling Language (UML) Specification

Visual Paradigm home page
(https://www.visual-paradigm.com/)

Visual Paradigm tutorials
(https://www.visual-paradigm.com/tutorials/)