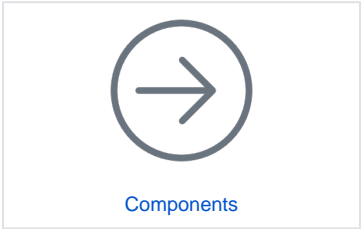


# Sequences Lesson 1 MD18



In the next modeling step, you will draw a sequence diagram documenting the Web service. A sequence diagram is a time-oriented view of the interaction between objects. An interaction may be modeled at any level of abstraction within the system design, from subsystem interactions to instance-level interactions for a single operation or activity.

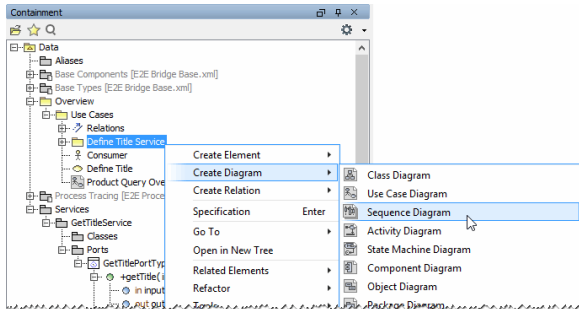
The sequence diagram has two dimensions: the vertical axis represents time, while the horizontal axis represents participating objects. Horizontal ordering of the objects is not vital to the operation, so you may rearrange them as necessary.



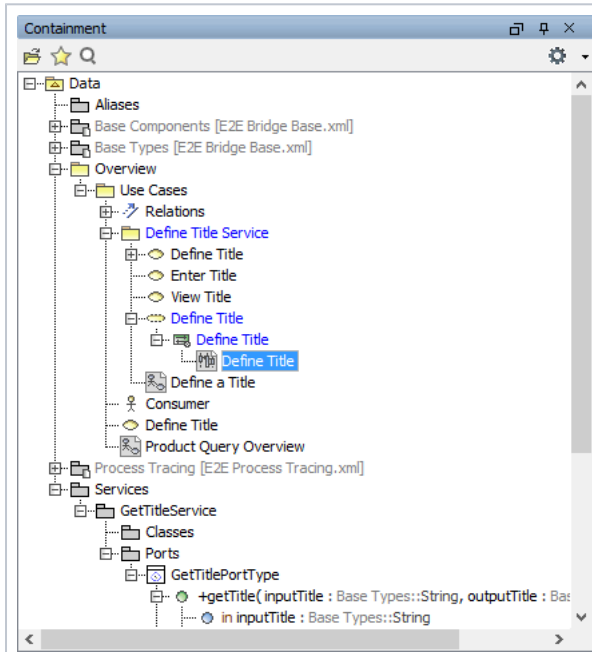
Components

The sequence diagram is optional and will not be considered by the E2E Model Compiler as it is used for documentation purposes only.

In the containment tree, expand the package **Data / Overview / Use Cases**, select the package **Define Title Service**, and open its context menu. Select **Create Diagram > Sequence Diagram**.

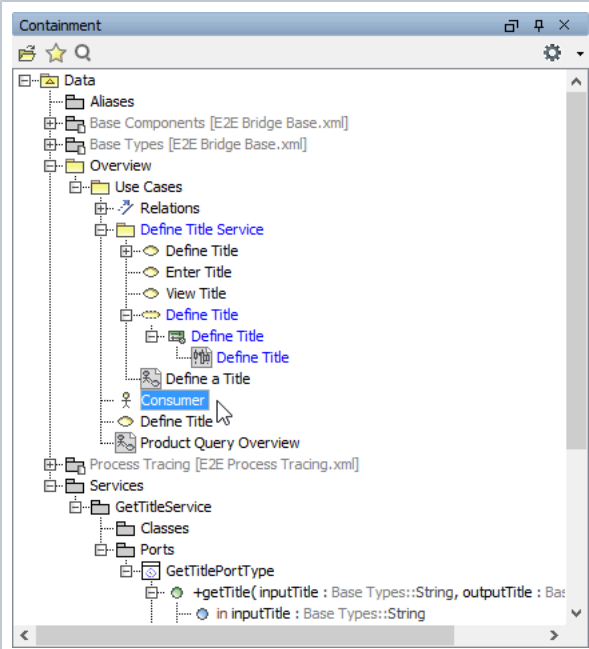
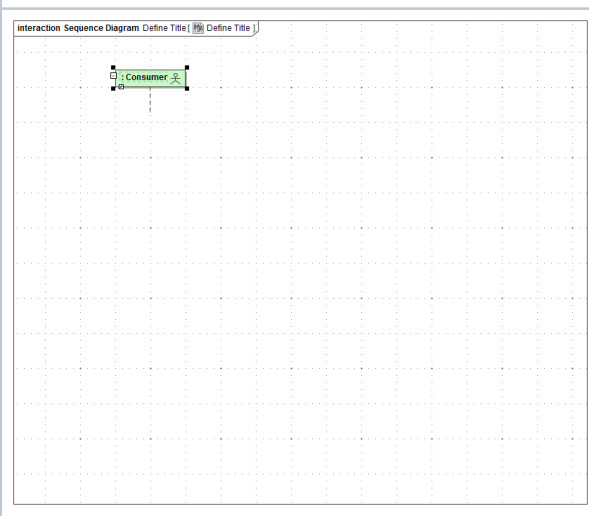


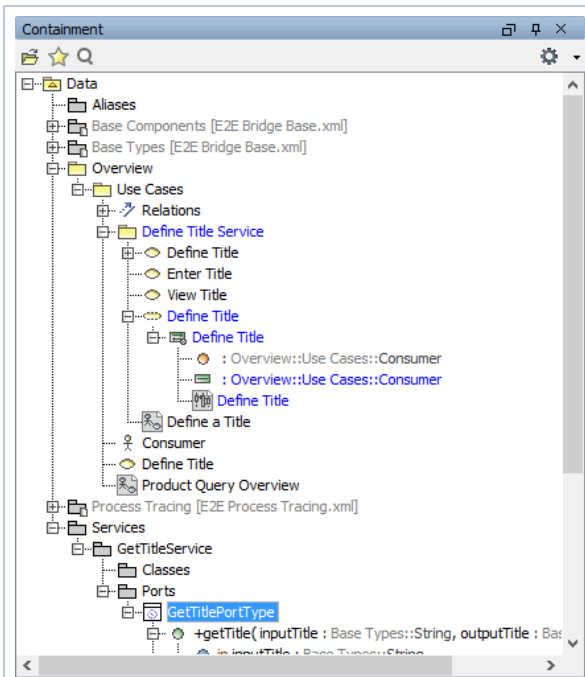
The sequence diagram will be created. In the containment tree, it is wrapped in an **Interaction**, a unit of behavior. The interaction is part of a **Collaboration**, which represents how elements of the UML model cooperate to perform an essential behavior.



The name of the sequence diagram is in editing mode. Start typing **Define Title** and press **Enter**. Note that the owner nodes (interaction and collaboration) have been renamed as well.

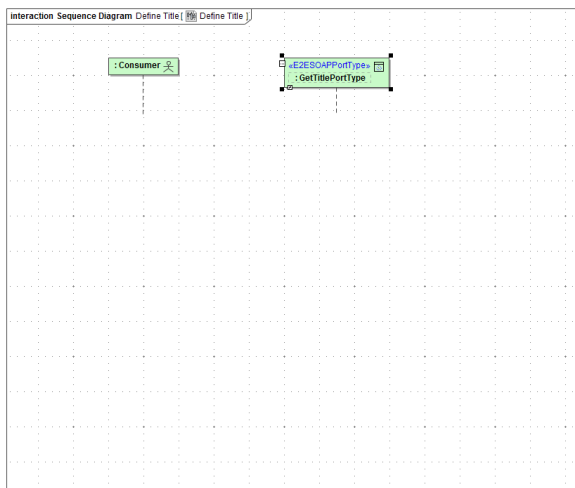
The sequence diagram **Define Title** is now included in the package **Use Cases**.

	<p>The actor <b>Consumer</b> in package <b>Data / Overview / Use Cases</b> is part of the process and can be directly used in the sequence diagram.</p> <p>Select the actor <b>Consumer</b>.</p>
	<p>Drag the actor from the containment tree onto the diagram pane.</p> <p>The first <b> Lifeline</b> is created.</p>

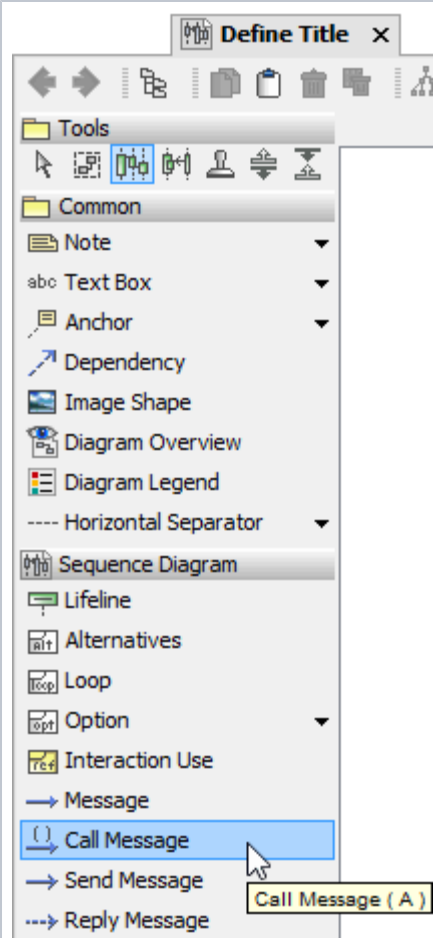
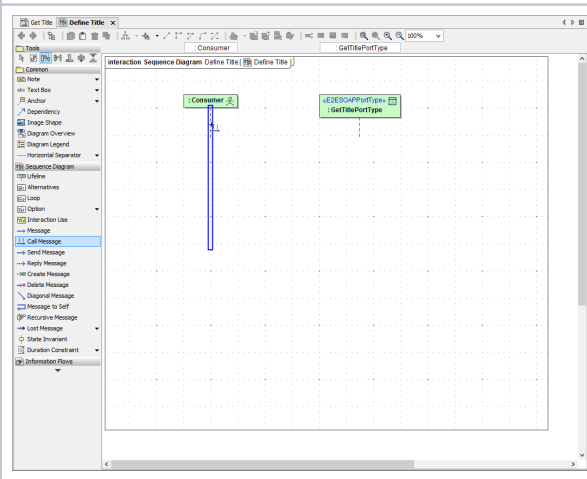


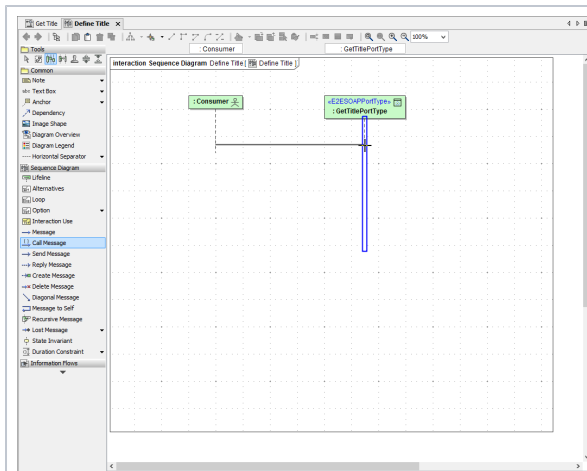
The actor **Consumer** uses the Web service, which is accessible via the SOAP interface. You can use the port type **GetTitlePortType** in the sequence diagram as well.

Select the port type class **GetTitlePortType** in package **Data / Services / GetTitleService / Ports**.

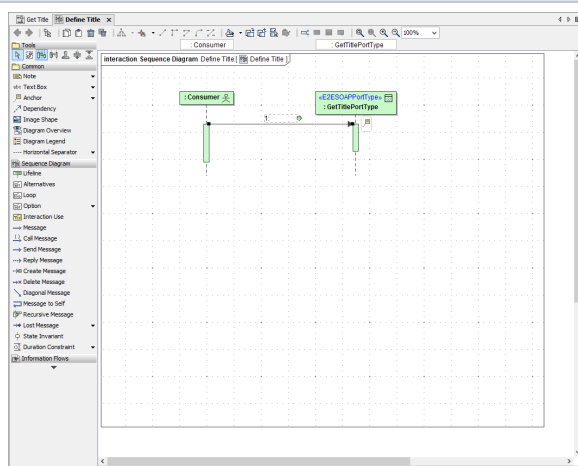


Drag **GetTitlePortType** onto the diagram pane and drop it next to the lifeline **Consumer**.

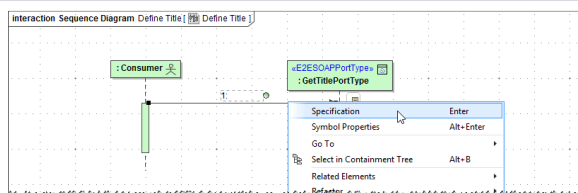
	<p>The interaction is designed by using messages that are exchanged between the lifelines. In the next step, draw the call message, which is sent by the actor <b>Consumer</b>.</p> <p>Select the <b>Call Message</b> icon from the diagram toolbar.</p>
	<p>Move the mouse cursor over the consumer's lifeline and click when the blue frame appears.</p>



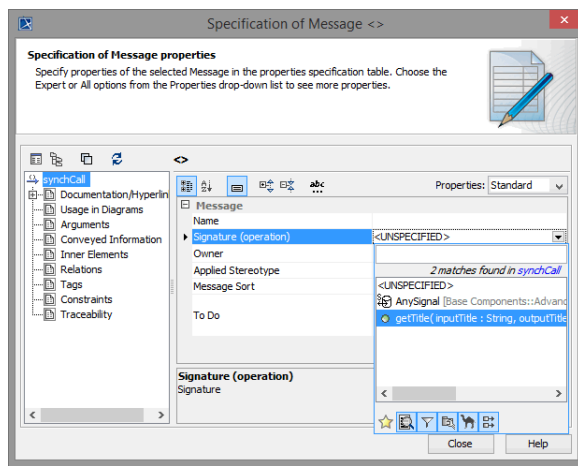
Move the mouse cursor over to the lifeline of the port type. When the blue frame appears, click again to draw the message.



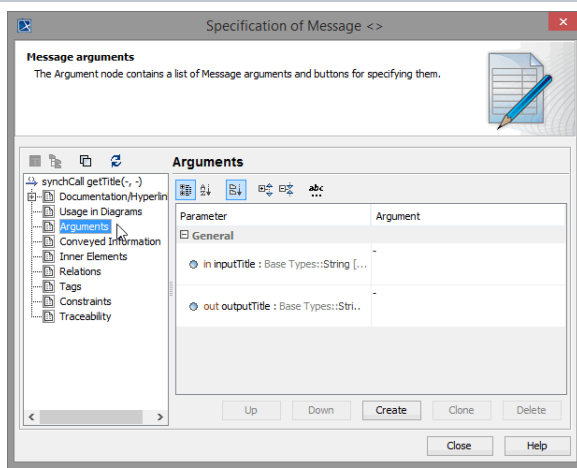
You have now created a call message from actor **Consumer** to port type **«E2SOAPPortType» :GetTitlePortType**. Now, you need to specify, which operation is called on the port type.



Select the call message and open the **Message** specification dialog via the context menu.

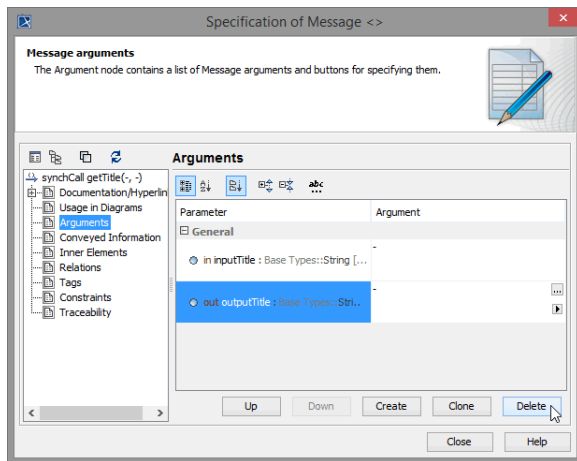


Define the **Operation** by selecting the operation **getTitle** from the drop down list.



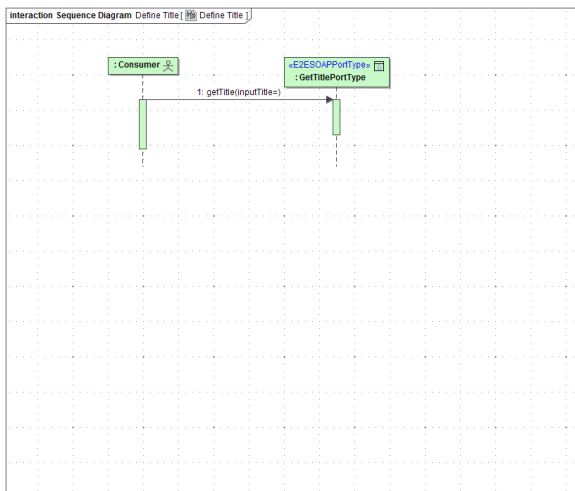
Next, click **Arguments** in the left panel of the dialog.

The argument list shows two arguments, which correspond to the input and output parameters of the port type operation. Both are having default values (-), which will not be changed as no specific values need to be entered for this message. Since you only pass **inputTitle** as parameter to the SOAP operation, you can delete the argument **outputTitle**.



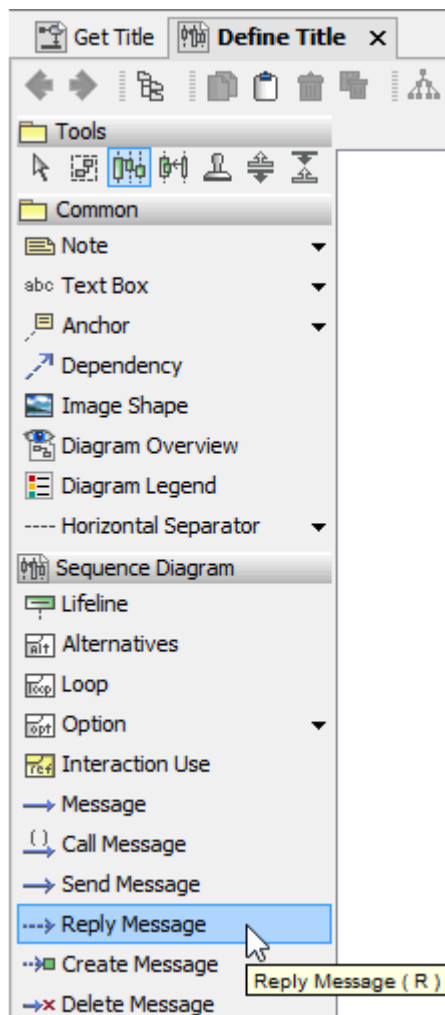
Click the second argument (parameter **outputTitle**) and delete it by clicking the **Delete** button on the lower right corner of the dialog.

Close the dialog window.

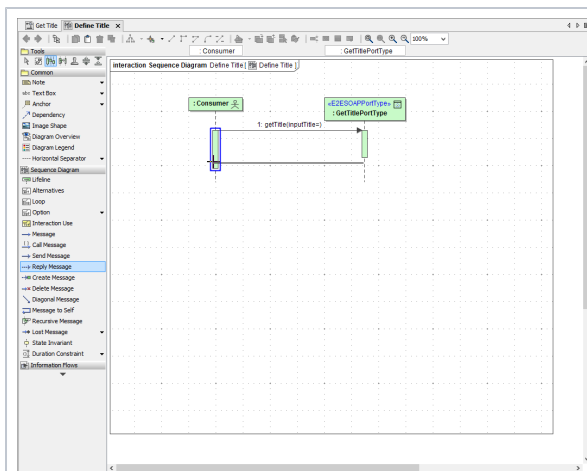


The call message has been defined. The actor is calling the port type operation **getTitle** and is passing the input parameter **inputTitle**.

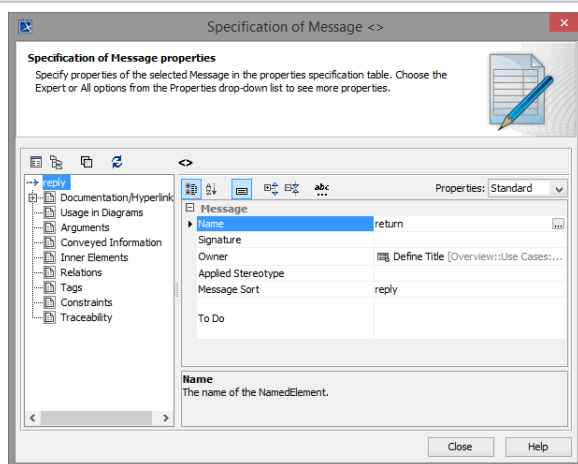
In the next step, you will draw the reply message from port type **GetTitlePortType** back to the service consumer **Consumer**.



Select the **Reply Message** icon from the diagram toolbar.

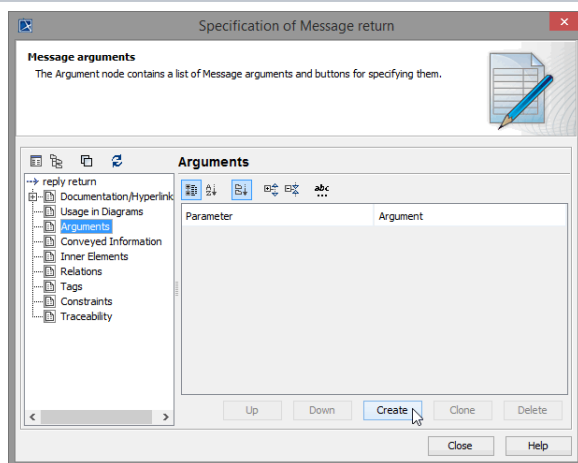


Draw the reply message from the lifeline of the port type to the lifeline of the actor.



Double-click the return message to open the **Message** specification dialog.

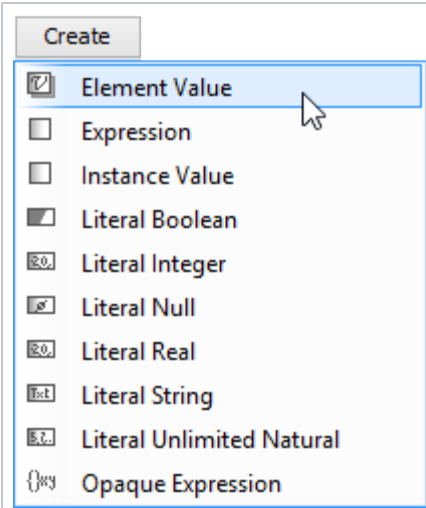
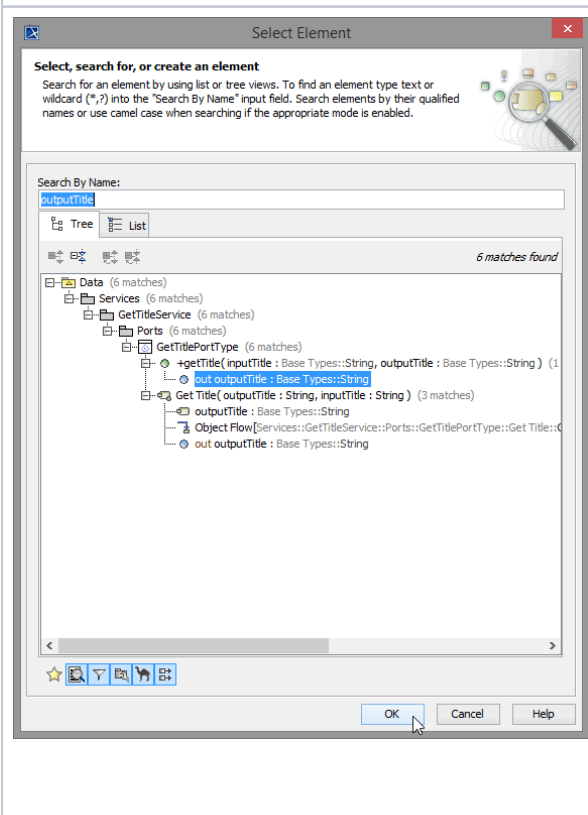
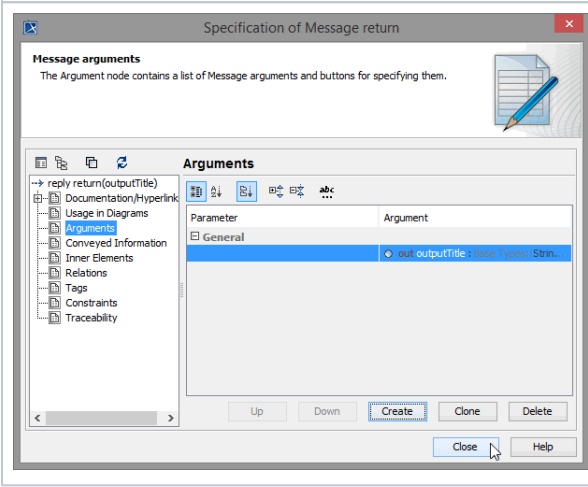
Enter **return** in the **Name** field.



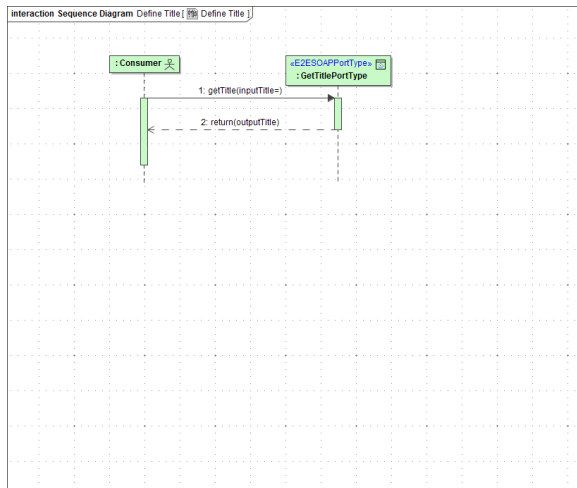
Now, you will assign the output parameter **outputTitle**.


Click **Arguments** in the left panel of the dialog. Then, click the **Create** button.



 <p>A list showing all possible argument types appears.</p> <p>Select <b>Element Value</b>.</p>	
 <p>The <b>Select Element</b> dialog window opens showing the content of the container tree.</p> <p>Enter <b>outputTitle</b> in the search field. Select the operation parameter <b>outputTitle</b> from <b>Data / Services / GetTitleService / Ports / GetTitlePortType / getTitle / outputTitle</b> and click <b>Ok</b>.</p>	
 <p>The parameter is displayed in the argument list now.</p> <p>Close the <b>Message</b> specification dialog.</p>	

The sequence diagram should look like in the picture below.



Save  the UML model.