Web Service Interface Lesson 1 MD18

All "public" Web services, which clients may access, must have an interface, in other words: a port type definition. A port type accumulates operations that a client can call on a Web service. In the UML model, port types will be designed by using a class stereotyped as port type. It is a special kind of class called **port type**, which has no attributes but only operations. Each operation must be assigned to its implementation (a UML activity diagram). Operations of a port type represent the interfaces of a Web service. Activities implement the behavior of these operations.
More details about port types and operations are described in the xUML Services Reference Guide.

In the next development step, you will define the SOAP interface of the Web service. The E2E Bridge supports SOAP 1.1. as well as Added in Bridge 6.0 SOAP 1.2, whereas SOAP 1.2 is only supported for document-literal encoded services. In this tutorial, you will create an RPC encoded SOAP 1.1 service (which is the default).

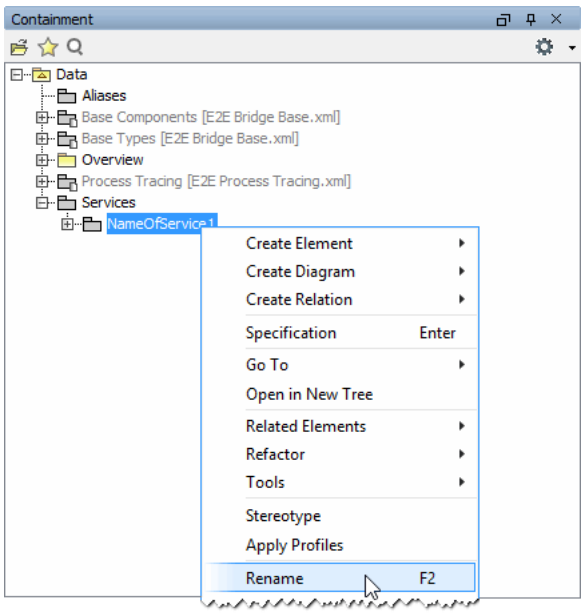
Within a Web service, one or more services can be included. Each service can have one or more port types. Within a port type, one or more operations having input and/or output parameters can be defined. They are the interface to the outside world.

Activities

On this Page:

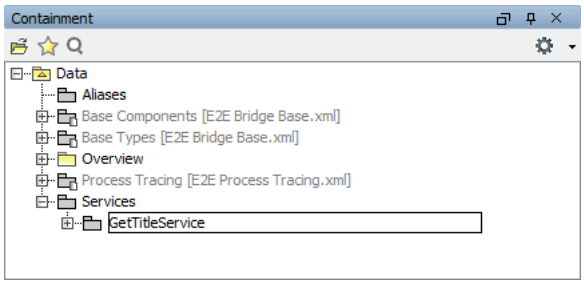
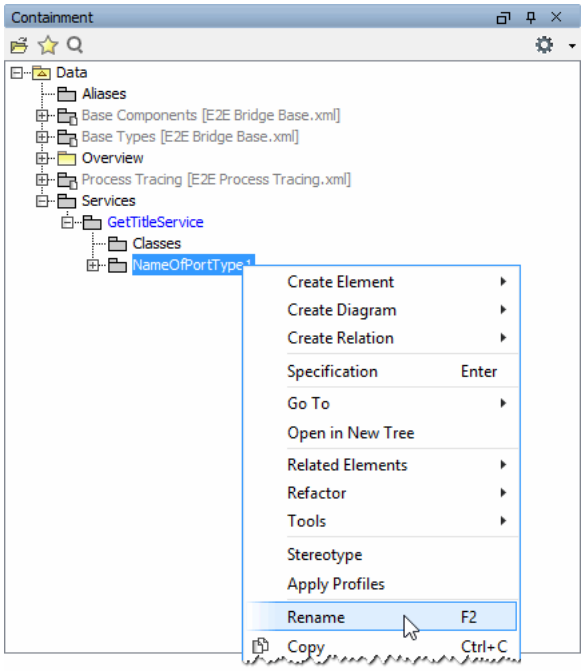
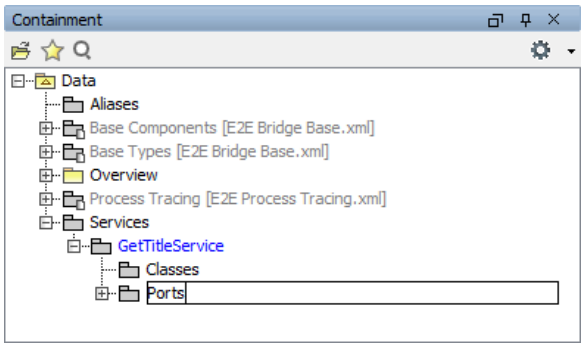
Renaming the Packages

The model lesson1 was created on basis of the E2E model template. Therefore, a default service package is already part of the model.



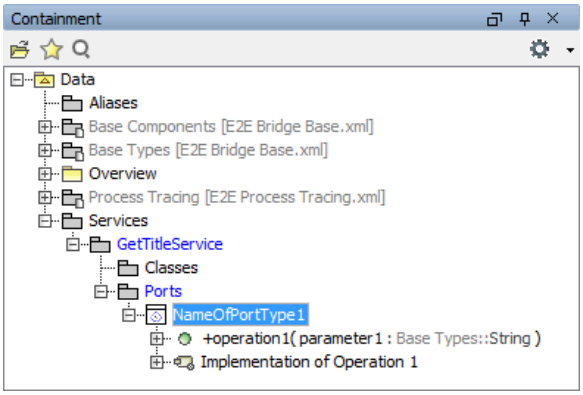
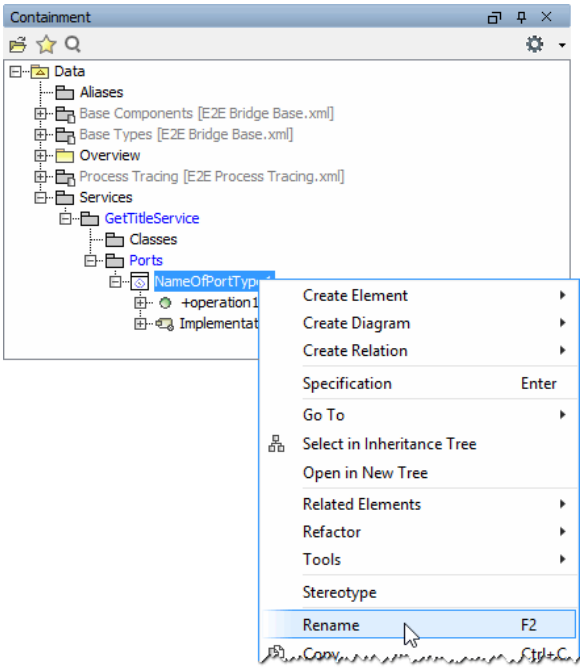
The package **Data / Services** in the containment tree contains services with relevant UML diagrams that will be part of a deployable unit - an xUML service. Adding a SOAP interface to it, it becomes a Web service.

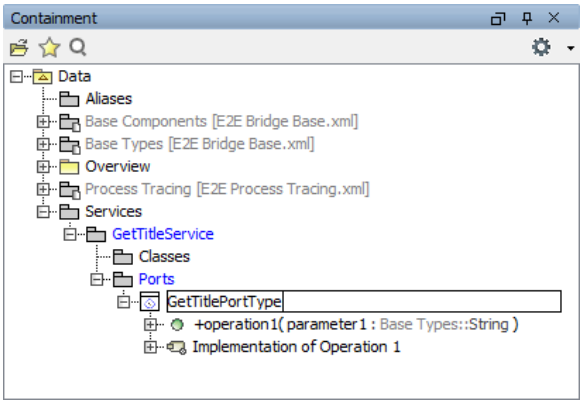
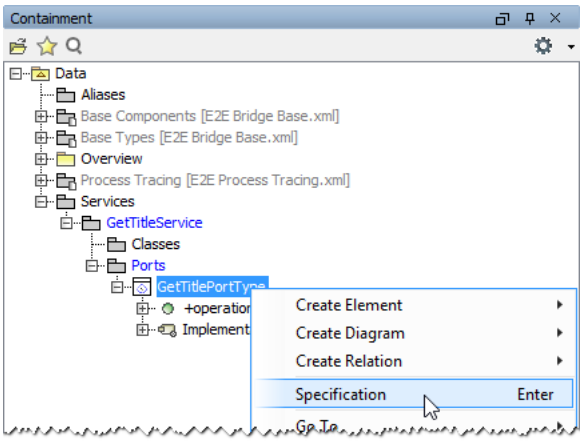
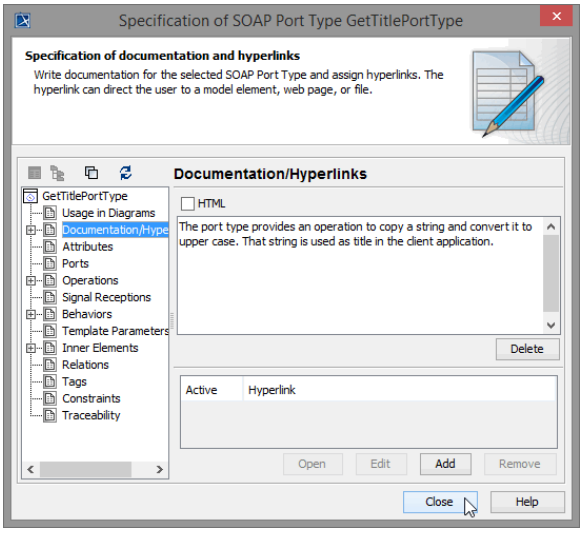
Expand the package **Data / Services**, and click **NameOfService1** with the right mouse button. Select **Rename** from the context menu.

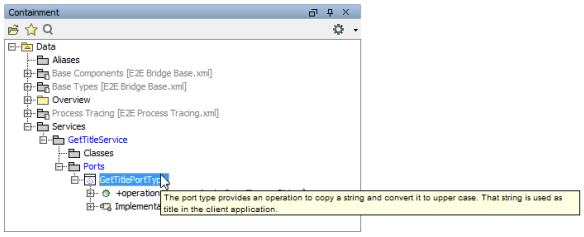
	Rename the package to **GetTitleService** and press the **Enter** key in order to confirm the change. Packages on this level in the containment tree will contain class diagrams, activity diagrams, and port type definitions.
	Expand the package **GetTitleService** and rename the package **NameOfPortType1**.
	Type in the name **Ports**.

Defining the Port Type

A port type, an operation with one parameter, and a default activity implementing the port operation are already defined in the UML model, as they were part of the E2E model template, from which this model has been created.
You will rename the port type and add mandatory documentation for it.

	Open the package **Data / Services / GetTitleService / Ports** and expand the port type class **NameOfPortType1.** Below the port type class, you find the operation **operation1** and its implementation, the activity **Implementation of Operation 1**. We recommend storing the implementation of an operation always below the corresponding class node in the containment tree.
	Expand the package **Ports** and rename the port type **NameOfPortType1**.

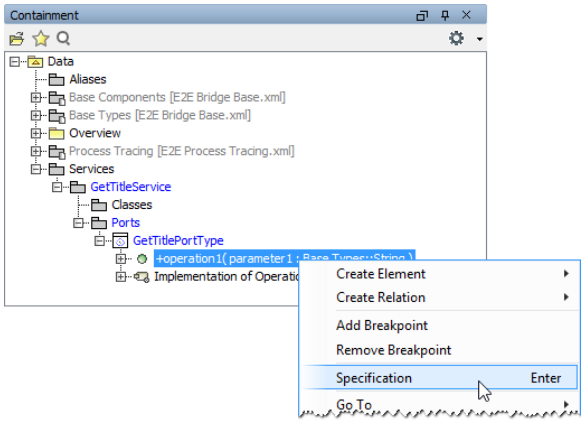
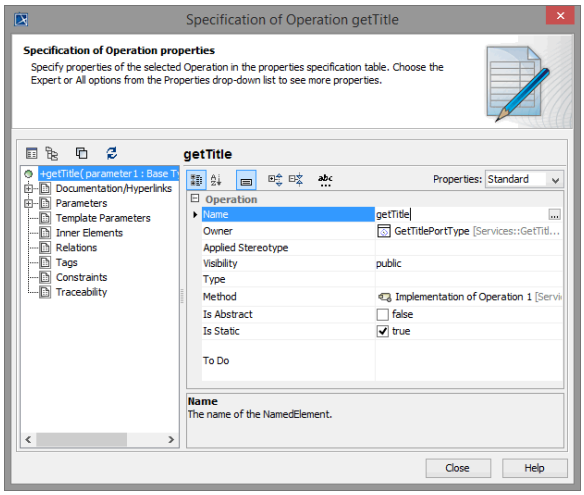
	Type in the name **GetTitle PortType**.
	Documentation for port types and its operations is required by the WSDL standard, therefore the Model Compiler will check for it. In order to enter the documentation of the port type, open its specification dialog. Click the pre-defined port type **GetTitlePortType** with the right mouse button and select **Specification** in the context menu.
	Click the entry **Documentation / Hyperlinks** in the left panel end replace the text **[Add documentation here]** in the **Documentation** panel on the right with the following description: **The port type provides an operation to copy a string and convert it to upper case. That string is used as title in the client application.** Click **Close**.

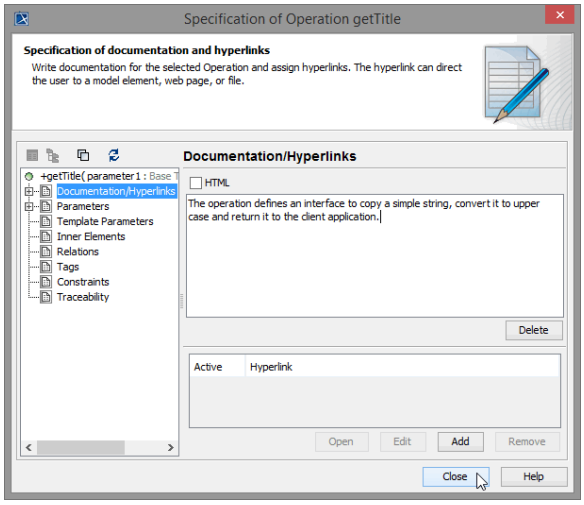
	The documentation will be displayed upon mouse-over on the containment tree element.

Defining the Operation

You have introduced the name of the interface (port type) in the model. The next step will be to define the capabilities of the Web service. The Web service will be capable of taking a string (Title) from the actor, converting it to upper case, and passing this string (Title) back to the actor. This service behavior is exposed to the outside world with an operation.

In the next step, you will define this operation in the model.

	Open the specification dialog of the pre-defined operation **operation1** with the right mouse button, and select the menu item **Specification** from the context menu.
	Rename the operation to **getTitle** in the **Name** field.



Operations need to be documented – this will be verified by the Model Compiler. The documentation text will be inserted into the WSDL file of the xUML service repository, from where possible users of the service can get this information.

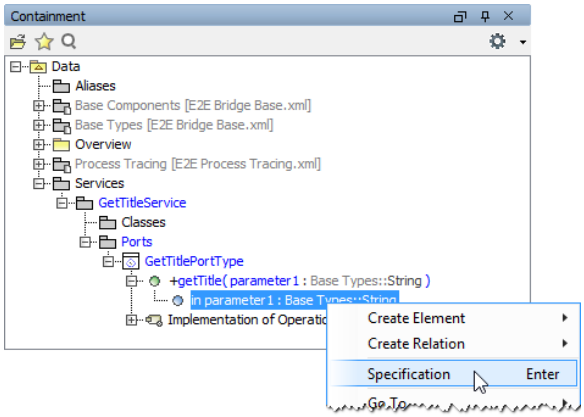
Click the entry **Documentation/Hyperlinks** in the left panel end replace the text **[Add documentation here]** in the **Documentation** panel on the right with the following description:

The operation defines an interface to copy a simple string, convert it to upper case and return it to the client application.

Click **Close**.

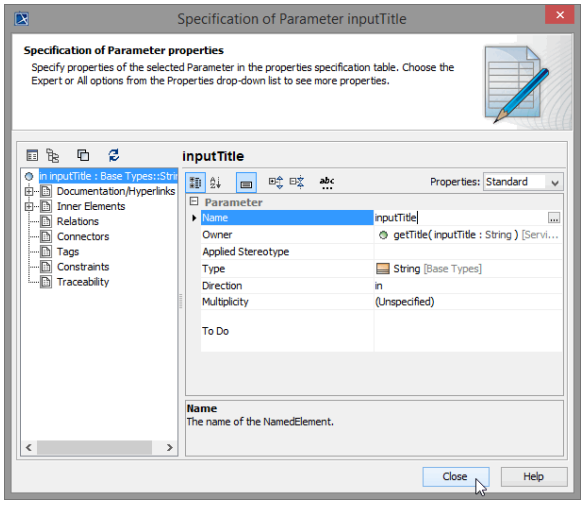
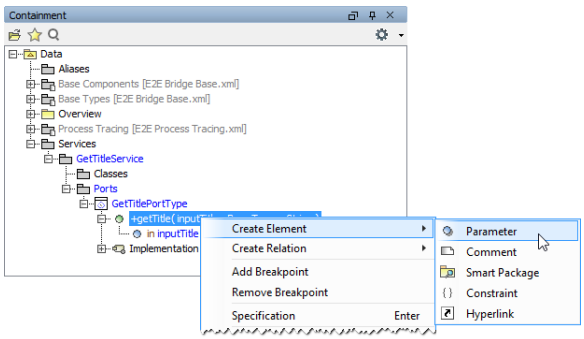
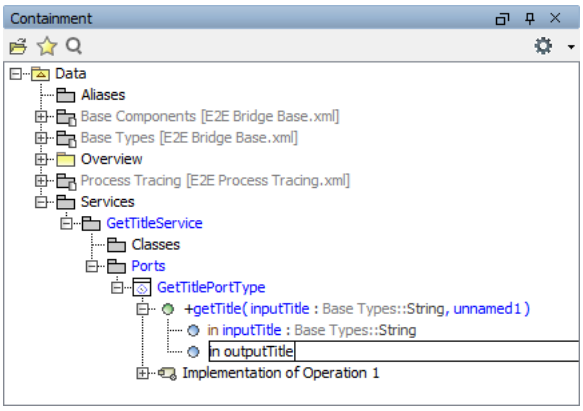
Defining the Operation Parameters

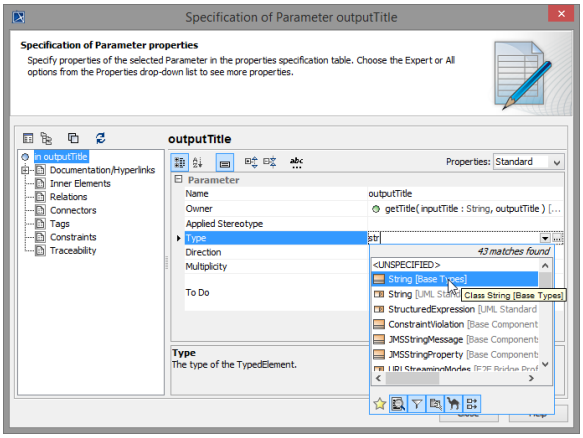
The Web service will be capable of taking a string (Title) from the actor and passing this string (Title) back to the actor. In the next step, you will specify the operation parameters: one input parameter (**inputTitle**) and one output parameter (**outputTitle**).



First, you will define the input parameter.

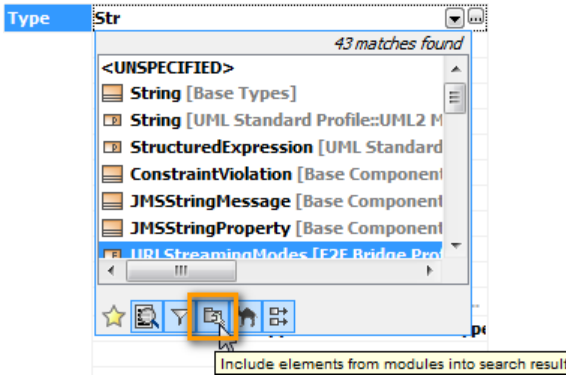
Expand the operation **getTitle**. Right-click the pre-defined opera parameter **parameter1** and select **Specification** from the conte **Parameter** specification dialog opens.

	Rename the parameter to **inputTitle**. As this parameter is only u to the operation , leave the **Direction** on the default **in**. The **typ** already been assigned as well. So just click **Close**.
	The operation now has an input parameter but is still missing th parameter. Right-click the **getTitle** operation and select **Create Element >**
	The new parameter node in the containment tree is in edit mode parameter to **outputTitle** and press the **Enter** key.



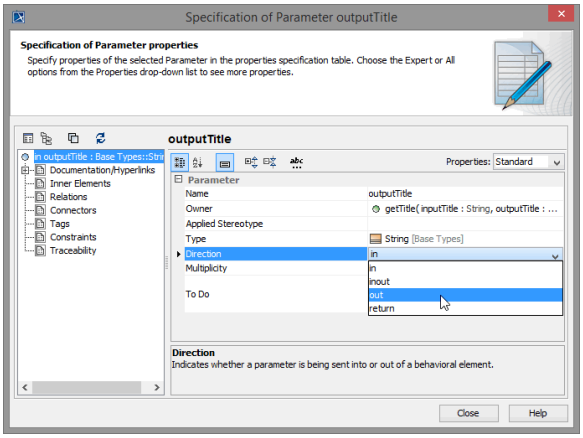
Double-click the new operation parameter **outputTitle** to open i
specification dialog.
Click into the **Type** field and start typing **Str** on the keyboard to
Select **String [Base Types]** with the arrow keys and press **Ente**

If type **String [Base Types]** is not visible in the list of types, you
adjust the filter settings of MagicDraw to include elements from
Select this filter option as shown below:



MagicDraw will remember this setting.

Always make sure to select the E2E base types and **not** the types that are part of the UML standard profile (see **String [UML Standard Profile...]** in the **Type** field.



As this parameter is the output parameter of the operation, select **out** from the pull-down menu **Di rection**.

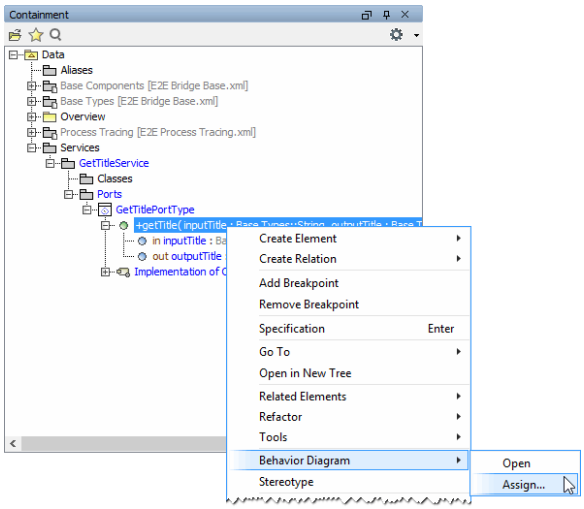
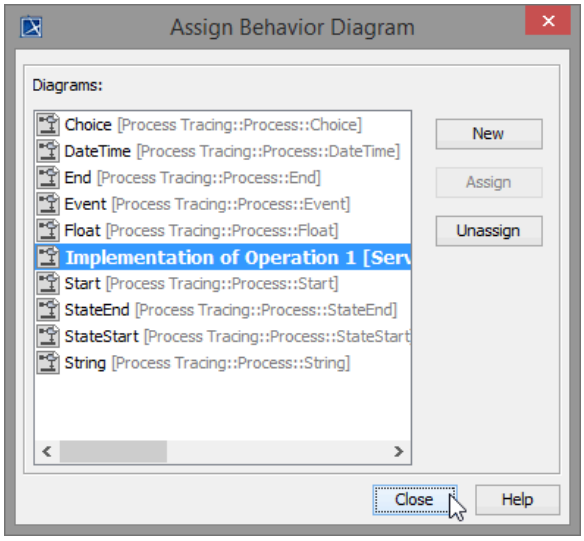
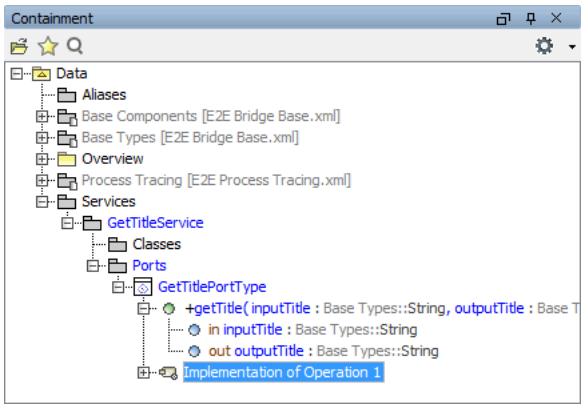
If a parameter is used as input and output at the same time, choose **inout**.

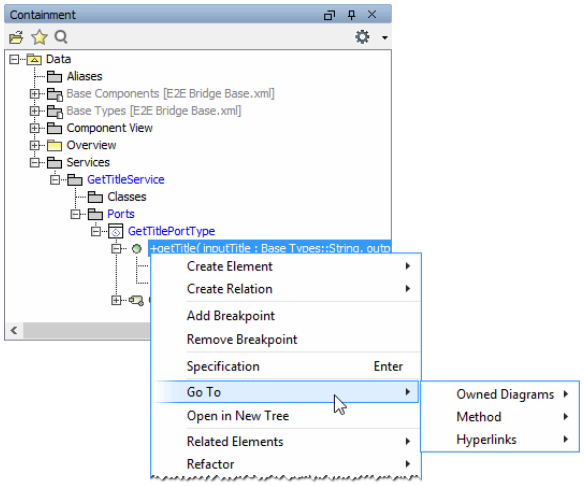
Click **Close**.

Assigning the Activity Diagram to the Operation

At this point, the interface of the Web service is nearly complete.

Each operation must be assigned to an activity diagram of the UML model. Operations of a port type represent the interfaces of a Web service. Activity diagrams implement the behavior of these operations. Each port type operation has to be assigned to the implementing activity diagram.
In the E2E model template you have used to create this UML model, the assignment has already been done for the default operation.

	To check the assignment, select the operation **getTitle** in the containment tree with the right mouse button and choose **Behavior Diagram > Assign...**.
	All diagrams that can be assigned will be listed in the dialog. The button **Assign** is not enabled, because the activity diagram **Implementation of Operation 1** is already assigned to the operation. Close the dialog by clicking the **Close** button.
	Whenever you double-click the operation **getTitle**, the assigned activity diagram will be opened in the diagram pane.



For an overview of all linked items, you can also open the context menu of the operation **getTitle**. Choose **Go To** to see the diagrams this operation is used in (**Owned Diagrams**). The corresponding activity is displayed under **Methods**.

The Web service interface has been finished now. If the operation **getTitle** is called remotely, the actions will be executed as defined in the assigned activity diagram.

Save the UML model.