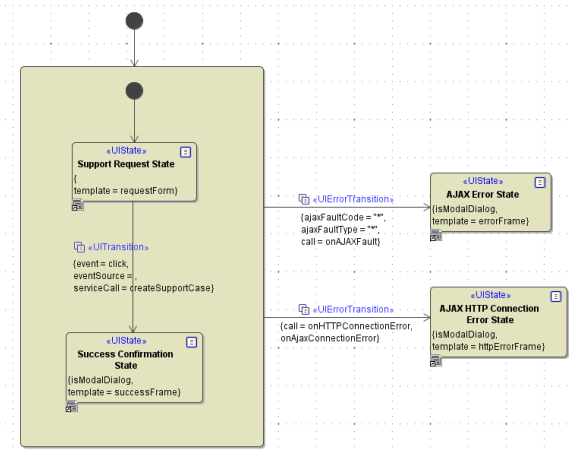


Catching AJAX Errors

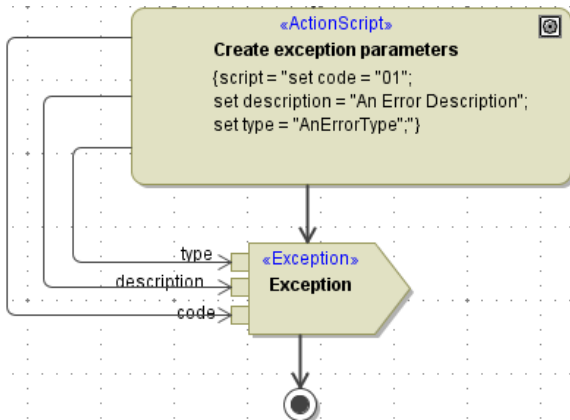
The AJAX Errors are caught within the State Machine Diagram. The **<<UIStates>>** are wrapped by a surrounding Composite State which will catch all exceptions and errors which might occur and then delegate these via **<<UIErrorTransitions>>** to the error handling **<<UIStates>>**.



The definition of which type of AJAX error, a fault or a connection error, should be caught, depends on how the **<<UIErrorTransitions>>** are specified. Any fault or error happening within the **<<UIStates>>** and are held by the Composite State will be caught.

AJAX Fault Handling

Due to almost all UI requests going to an xUML service, the service exceptions need to be caught and delegated to the user interface. The exception signal/event holds more information on the error that occurred and all this information can be displayed to the user.



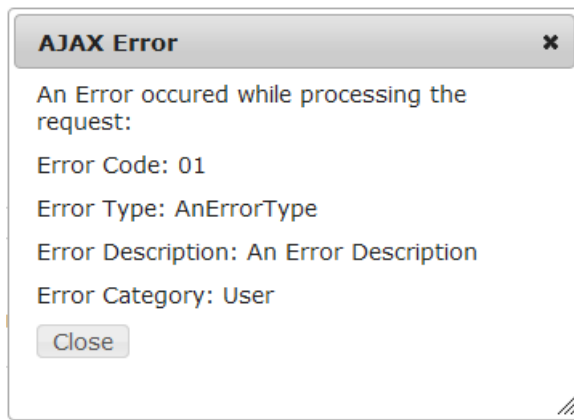
As seen in the above figure, a service handling an AJAX request, sends an **<<Exception>>**. This exception will be caught by the xUML UI and can be e.g. display that error to a user as below figure shows.

On this Page:

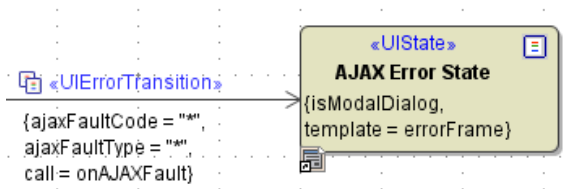
- [AJAX Fault Handling](#)
- [AJAX Connection Error Handling](#)

Related Pages:

- [Catching Errors in JavaScript Operations](#)

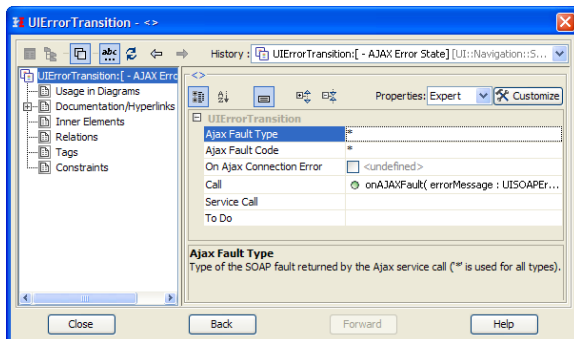


The Exception signal is caught within the `<<UIErrorTransition>>` where it is sent to a JavaScript operation having this fault event as an input object. The JavaScript operation is defined on the `<<UI>>` controller and is registered in the call properties of the `<<UIErrorTransition>>` as seen in below figure.



The `<<UIErrorTransition>>` further defines which type of error should be caught by defining the **ajaxFault Code** as well as the **ajaxFaultType**. In case of the above figure all exceptions are caught by using the ******* in both cases. This allows having different user interfaces for different types of errors.

Figure: Configuration to Catch AJAX Faults

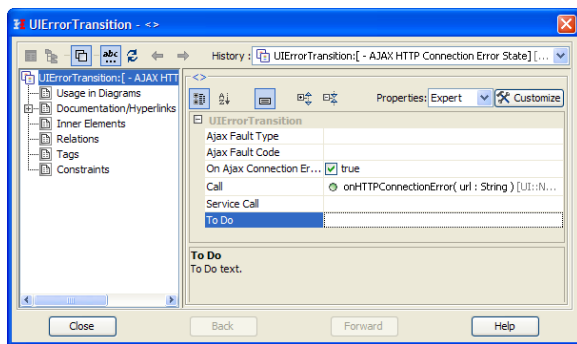


The JavaScript operations input object has to be of type **UISOAPErrorMessage**. The object holds the following information.

Attribute	Description
code	An error code, defined and set in the service where the <code><<Exception>></code> signal is sent.
type	The type of error, defined and set in the service where the <code><<Exception>></code> signal is sent.
description	A description of the error, defined and set in the service where the <code><<Exception>></code> signal is sent.
category	The category of the error e.g. User or System

AJAX Connection Error Handling

Connection errors occur when a request to a service times out or is not possible at all due to the service is e.g. not running. The error which is caught returns back the URL it tried to connect to. The implementation within the model follows the same principle as with AJAX faults. The error is caught by the Composite State and is redirected to the `<<UIErrorTransition>>` which handles the connection error. The difference is in the settings of the transition.



To catch connection errors, the parameter **onAjaxConnectionError** needs to be set to true. The JavaScript operation called has an input parameter called url and is of Base Type String.

