

Usage of Choices

Example File (Builder project Advanced Modeling/UI):

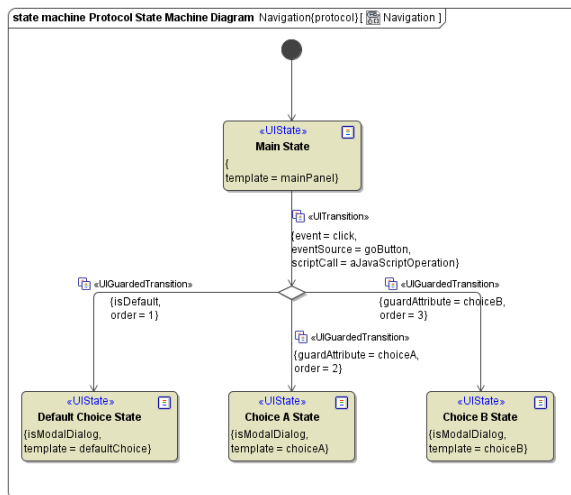


<your example path>\Advanced Modeling\UI\uml\uiChoice.xml

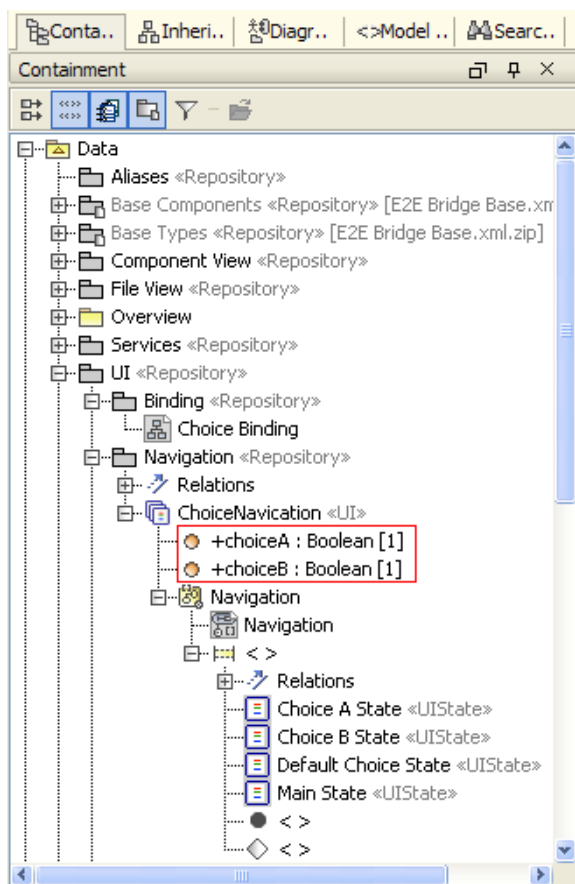
Related Pages:

- [Authentication and Authorization](#)
- [File Upload](#)
- [HTTPS](#)
- [History State](#)
- [Form and Form Validation](#)
- [Calling a UI from external Applications](#)
- [Usage of Choices](#)
- [Service Calls](#)
- [HTTP Proxy](#)
- [Controller States](#)
- [Back Button and Browser History](#)
- [Mock-Ups](#)

Choices are common to steer user interface flows. A very common usage is the implementation of wizards or the role dependent control of UI elements. Depending on the value a user sets on page A, the next page will be page C and not page B. Scenarios like this are done by implementing choices within the navigation's **State Machine Diagram**.



Choices are defined by using the Choice diamond element splitting to the different choices. Each choice is a **<<UIState>>** and the transition leading from the choice to the destination state is of type **<<UIGuardedTransition>>**. These specialized transitions hold a **guardedAttribute** which is linked to a Boolean variable defined on the **<<UI>>** controller (**choiceA: Boolean**, **choiceB: Boolean**). With each choice there is always a default case which will be followed in case no choice was made or none of the options which follow to a specific page were chosen. The default case is specified by setting the **<<UIGuardedTransition>>** parameter **isDefault** to **true**.



The `<<UI>>` choice variables are bound to the user interface elements (in the examples attached to this topic radio buttons), so depending on the choice the user does, these Boolean values are set to true or to false.