

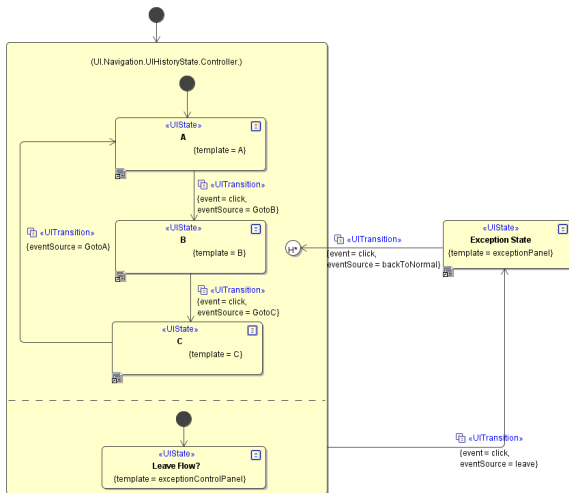
# History State

Example File (Builder project Advanced Modeling/UI):



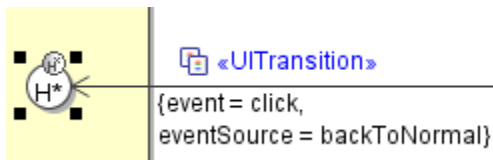
<your example path>\Advanced Modeling\UI\uml\uiHistoryState.xml

In a lot of cases it is important for an end user to be able to return to a previous state of the web application, e.g. if an error occurred and the end user is redirected to an error page. To be able to guide the user back to a previous application screen, which might be a input form with lots of fields, so the user can correct the data or resubmit in case of a temporary system error, the usage of the history state can be of great help.



In the example state machine above, the user can simulate an error by clicking on the leave button while stepping through a simple application flow simulation (states **A** to **C**). This will exit the application state and end in the **Exception State**. This might happen if a back-end system went off-line while working with the application. Now lets assume that this happened at the application state **B** and the user just filled out a large form in step **A**. It would be frustrating to have to re-enter data again, just because of a temporary technical fault. To work against a situation like this, the history state is implemented to lead back to the application's state machine. The history state persisted the data until the state the exception occurred and lets the user continue where he stopped and that would be at step **B**.

To implement the history state within a xUML UI application, the application's state machine needs to be wrapped in a composite state. In the above example this is the **Application Composite State**. The history state, that needs to be modelled within the applications state machine, is symbolized with the white circled **H\*** element. Magic Draw knows two types of histories, the one to use though is the deep history marked with a star.



It is important, that the Exception State or any other state leading to the history state is outside the application flow.

The history state element does not have a specific stereotype but own a tagged value where the **Kind** is set. To make it work, the value needs to be set to **deepHistory**.

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](#)

