

Modeling User Interfaces

What is a BRIDGE UI, and what is it NOT

BRIDGE User Interfaces enables developers working with the BRIDGE to model and integrate simple, web based user interfaces. Typical use cases are administration interfaces for xUML services or user interfaces for interacting with processes – especially if they are modeled in the BRIDGE using BPMN. Although BRIDGE User Interfaces can be used for complex and large web based user interface applications as well, it always is recommended to evaluate the needs against other technologies like AIR or AngularJS.

The strength of BRIDGE UI is the ability to rapidly implement a UI prototype using the flexible and advanced options that Magic Draw offers. The step from a simple Magic Draw UI model, for functional prototype presentations to an actual running framework of a web application is only a few mouse clicks away. This is possible by using the xUML Importer. The Magic Draw UI Prototype Importer creates a ready to run web application the way it was designed in the functional Magic Draw prototype. The only steps that need to be done further, is the implementation of backend services and fine tuning of the user interface design if needed. From a best practice point of view using the Magic Draw Prototype Importer is the fastest and easiest way for creating web applications. Nevertheless, it is important to know what lays behind the generated framework and for this reason there is a walkthrough tutorial which goes step by step through a manual development process.

On this Page:

- [What is a BRIDGE UI, and what is it NOT](#)
- [General Concept](#)
 - [Model](#)
 - [View](#)
 - [Control](#)

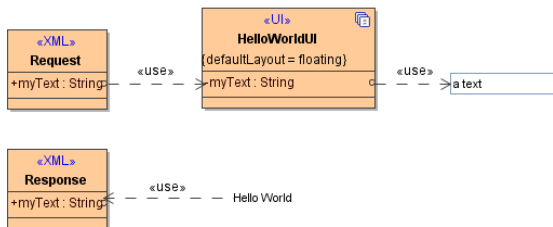
General Concept

Designing user interfaces in UML using MagicDraw is implementing the common Model-View-Controller (MVC) concept. However, due to the fact, that implementing BRIDGE UI prototypes and applications within UML models is a new approach to building UIs, the implementation of the MVC conception needs some explanation.

Model

The data model holding the data being displayed within a view is, in the SOA context, derived from the document encoded SOAP service. This is where an BRIDGE SOAP service is coming into use and the bridge to the BRIDGE UI is built. The data model in fact holds the input and output parameters of a SOAP service.

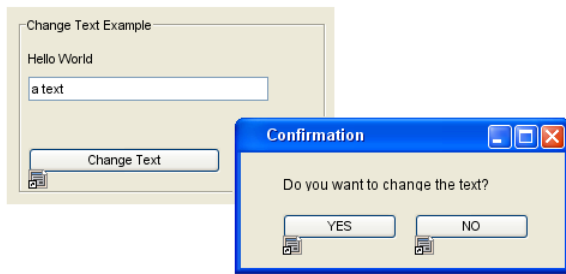
Figure: HelloWorld UI data model



View

The view, the actual user interface, is modeled with the new UI modeling environment MagicDraw offers starting with version 16. It is possible to create almost all types of user interfaces using a wide selection of user interface elements which are needed to create realistic prototypical screen designs. The user interfaces can additionally be linked to other diagrams or even diagram elements to be used as a "clickable", more realistic prototype. When deciding to use the MagicDraw UI Prototype Importer the linking between the screens will result in the state machine flow of the <<UI>> controller and therefore is important.

Figure: HelloWorld UI User Interfaces



Control

The control and flow of the different BRIDGE UI screen elements is represented by a UML State Machine Diagram. Each view, be it a main panel screen, a modal dialog or just a UI fragment, is attached to one **<<UIState>>**. The **<<UIState>>** itself is connected with other states using **<<UITransitions>>** which steer the behavior between different user interface states, e.g. a detail modal dialog window containing all the data from a table row selection. These transitions also define how the user interface implements events which trigger state changes.

Figure: HelloWorld UI State Machine

