Function Block (FB) Workbench (FBench) Developer's Guide

William Dai, Arthur Shih, Cheng Pang, Jason Park, David Huang, David Gommans
Supervised by *Valeriy Vyatkin*

Content

How to insert new items to the menu in FB Workbench

1) Declare “private JMenuItem itemname = createMenuItem(“ICON Name”,”Name”,”The function calling”,”Comment”);” in FB Workbench.java of org.oooneida.wb package.
2) Declare “private JMenu MenuName = new JMenu("Name");” in FB Workbench.java of org.oooneida.wb package.
3) Add MenuName.add(Itemname); in menuBarInit() in FB Workbench.java of org.oooneida.wb package.

How to add/design a new panel in the FB Workbench Editor

Please refer to the LibraryElementView.java of org.oooneida.wb package

1) Declare any type of field you want to display in the frame, for example JTextField, JSplitPane or any graphical windows, in FB Workbench the internal graphical view class named GraphView, in “Some”View.java.
2) Define a constructer to load the document object (DOM) to form the window structure and also initial the components in the window.
3) Add eventlistener handler into the class
4) Finally, put into the org.oooneida.wb package. You can create a new panel by create a new instance for this class in FB Workbench.java

How to make a new XSLT Translation Table for supporting new language in the FB Workbench Compiler

Please refer to DOMtoJava.xsl of org.oooneida.wb.dom package as an example.

1) Add an XSLT translation table in org.oooneida.wb.dom like DOMtoC++.xsl
2) Complete the XSL translation table – Please refer to the XSLT tutorial
3) Add an extra view in LibraryElementView.java in org.oooneida.wb package by adding an extra tab:

```
srcAreaName.setSelectionColor(Color.yellow);
srcAreaName.setEditable(false);
srcAreaName.setEditorKit(new HTMLEditorKit());
new DOMTranslationModel(document,srcAreaName,"DOMtoC++.xsl");
JScrollPane srcViewName = new JScrollPane(srcAreaName);
srcViewName.setPreferredSize(new Dimension(400, 150));
tabbedPane.add("C++", srcViewName);
```

How to give graphical representation of an object in FB Workbench

All the graphic components (Basic components) are defined in the package org.oooneida.wb.graph. Users can define your own shape by creating codes of java.awt library to plot the graphics. Users can join several components together to form a model. The internal models are defined in the package org.oooneida.wb.graph.model. The FB Workbench will plot the graphs by deploying the models or components in the library.

How to customize event listeners, table and tree structure in FB Workbench

Users are allowed to create their own events and related listeners in the package org.oooneida.wb.events. Also table and tree structure can be used in the Frame view can be defined in org.oooneida.wb.table and org.oooneida.wb.table respectively.

FB Workbench Useful Internal Classes/Functions

1) SearchFile.java
   Package: org.oooneida.wb
   SearchFile is a class which can return the file content given by the user by scanning all folders on the local hard disc.

2) DOMParser.java
   Package: org.oooneida.wb.debug
   DOMParser class is a parser which can load document object model (DOM) into a series of tree structured nodes. These nodes can be further process by any other class/function.

Dialogs

ECC, FBNetwork and FBType Interface dialogs extend the abstract class DialogModel, dialogs can be found in theorg.oooneida.wb.graph.popup package.
The dialogs modify the document tree according to the user selection.

Drag

Left drag inside ECC, FBNetwork, FBType and its sub-classes will perform a transition of the selected component and right drag won't be allowed, unless the user is performing right-drag in ECC. Right-drag in ECC, from one ECState to another ECState, creates an ECTransition. They were done by checking 'mouseEntered' and 'mouseExited' methods of MouseListener in ECState.

Context Menu

The user selection in the tree is detected for context menu purposes are done by calling the DOMTree instance in LibraryElementView.java(a global field called tree), when the mouse is clicked in the main graphical display, the mouse handler checks the DOMTree instance and shows the appropriate context menu. The menu is hidden when the mouse clicks somewhere else than the menu.

Delete

The Context menu is invoked from the mouse listener in the respective GraphElement children. The delete function is implemented in the main graphical display area, and the document tree is modified since the DOM Element can be accessed directly from GraphElement children(via. getElement() function).

Developers Log

Summer 2006: David and Jason (November 24th 2006 to February 28th 2007)

The main focus of this development period is to focus on the user interface aspect of the FB Workbench. Initially the issue with the FB Workbench was that the previous developers focused on developing functional aspects of the program (using the FB Workbench to run function blocks, the debug feature, and compile to java… etc.). There are a few areas to be addressed from the initial assessment of the FB Workbench:

1. Incapable of adding/modifying/removing FB Interface list
2. Incapable of adding/modifying/removing ECC list
3. Incapable of adding/modifying/removing FBNetwork list
4. Incapable of modifying Algorithms

These issues together renders the FB Workbench practically unusable as a graphical editor (every editing aspect is text based), it has only simulation functions.

To expand on the mentioned issues:

1. adding/modifying FB Interface list:

There are a total of 5 elements in the interface list category, with an invisible element. In our reference goal (FBDK), there are various dialogs that can be accessed through the context menu, each offering the editing of every attribute associated with the particular element. Also there is context assistance for the selection of types and actions.

2. adding/modifying ECC list

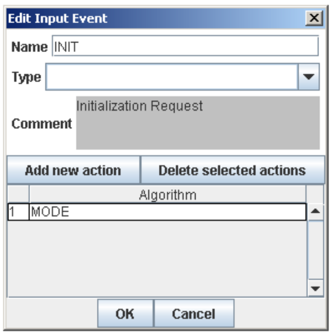
ECC list contains 2 elements, ECState and ECTransition. ECState contains a child named ECAction. The addition of the ECState in the reference goal is done with the context menu, and the position of the created ECState is determined on the mouse location. The ECState location is edited with left drag (drag with the left mouse button). The ECTransition is created with left drag with the alt button pressed, however the condition is defaulted to 1. All the other attributes are edited with respective dialogs.

3. adding/modifying ECC FBNetwork list

FBNetwork contains FB, EventConnections and DataConnections as its children. FB has Parameter as its children. FB is added with the context menu in the reference goal, and connections are added with left drag with the alt button pressed. All the other fields are edited with respective dialogs.

The Graphical User Interface (GUI) we implemented encapsulates the functions we observed in our reference goal, and is further refined to be more intuitive to the user.

FB Interface list:



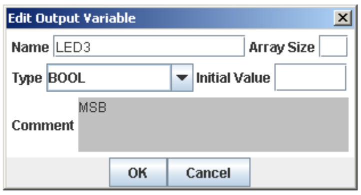
Edit Input Event

Name: INIT

Type:

Comment: Initialization Request

	Algorithm
1	MODE



Edit Output Variable

Name: LED3

Array Size:

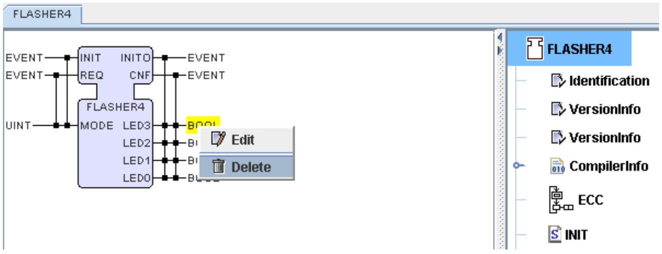
Type: BOOL

Initial Value:

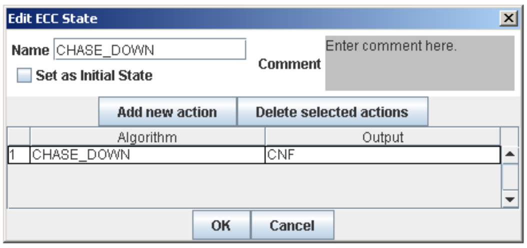
Comment: MSB

The dialog contains the editing function for all the fields associated with the elements in the FB Interface list. (Duplicate entries should not be allowed in the table model for the Event dialog.)

There is also delete function to remove a particular element.



ECC List:

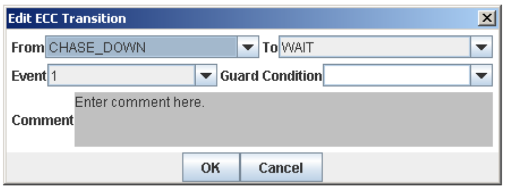

Edit ECC State

Name CHASE_DOWN

□ Set as Initial State

Comment: Enter comment here.

	Algorithm	Output
1	CHASE_DOWN	CNF

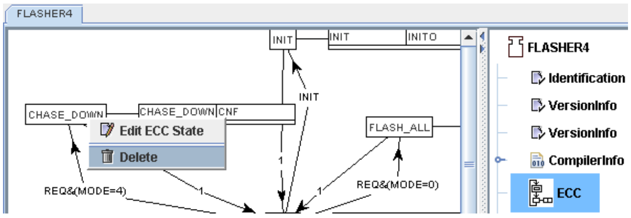

Edit ECC Transition

From CHASE_DOWN To WAIT

Event 1 Guard Condition

Comment: Enter comment here.

The dialog contains the editing function for all the fields associated with the elements in the ECC list. The set as initial state check box would move the selected element to the top of the list in its parent list, which is how the language determines the initial state for ECC. (Multiple selections are allowed in the table model for the ECState to allow easy removal of unwanted actions.)



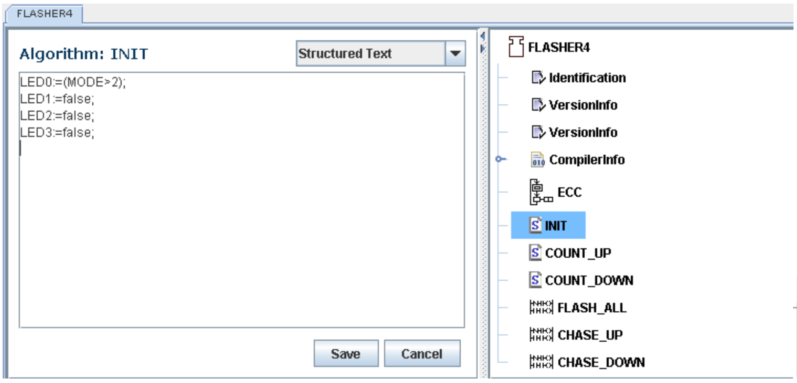
There is also delete function to remove a particular ECState or ECTransition.

Left Drag (drag with the left mouse button pressed) with the mouse on an ECState would modify its location.

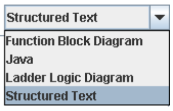
Right Drag (drag with the right mouse button pressed) with the mouse from an ECState to another ECState would pop up the “New ECC Transition” dialog, where the user can edit the condition for the ECTransition.

FBNetwork list:

Algorithm Editor:


Algorithm: INIT — Structured Text

```
LED0:=(MODE>2);
LED1:=false;
LED2:=false;
LED3:=false;
```



The Algorithm editor allows for the editing of algorithms in the form of Functional Block Diagram, Java, Ladder Logic Diagram and Structured Text. The Save button would save the change into the data tree, while cancel would leave the data tree unmodified.