Operator Interface Editor and Runtime – BOY

Xihui Chen, Kay Kasemir
chenx1@ornl.gov
Fall 2010 EPICS Collaboration Meeting
What is BOY?

- BOY (Best OPI, Yet) – An Operator Interface (OPI) development and runtime environment
- OPI - Graphical User Interface to display or control the data in a Control System
What is BOY?

• Best of SDS (Synoptic Display Studio: DESY)
  ✓ Java portability
  ✓ CSS integration
  ✓ Every property can be dynamic
  ✓ Modern Graphical Editor based on GEF (BOY reused part of the GEF related code from SDS)

• Best of EDM (Extensible Display Manager: John Sinclair)
  ✓ Simple things (Label, Textupdate, …) are simple
  ✓ Macros

• Combined with New Ideas
  ✓ Dynamic via PV-triggered scripts or rules
  ✓ Runtime works like a modern web browser (tab, CTRL, SHIFT click)
  ✓ Comprehensive types of Widgets
  ✓ Good ideas from all around the world…

Thanks to the folks who made efforts to SDS and EDM!
OPI Editor

• Your All-In-One workbench for OPI editing
OPI Editor Perspective

• Every View in the workbench can be dragged around, detached, minimized, maximized or closed.

• Recover the default perspective by resetting it.
OPI Editor

- **What You See Is What You Get (WYSIWYG)**
- **Comprehensive editing functions on toolbar and context menu**
  - Copy/Paste/Delete
  - Drag & Drop
  - Undo/Redo
  - Alignment & Distributing
  - Snap to G (Grid/Geometry/Guide)
  - Zoom In/Out
  - Copy Properties
  - Changing Orders
  - ...

![Toolbar and context menu](image)
Editing Features

• Adding Widgets
  – Select Widget on Palette
  – Draw a rectangle on Display or drag/drop to Display

• Snap to G, align, distribute

• Ctrl+Drag widgets to duplicate

• Select multiple widgets to
  – Edit common properties
  – Adjust size or move around
A simple OPI

• Create a functional OPI in a breeze
  1. Create a new OPI file
  2. Drag a widget (Knob for example) from palette to editor
  3. Enter the PV name in Properties view
  4. Click the “Run” button to run it!

• What you will get
  ✓ Show PV value on the widget
  ✓ Show PV severity via e.g. border color
  ✓ Show PV name and value on tooltip
  ✓ Use PV’s display limits as default range
  ✓ Use PV’s Hi, HiHi, Lo, LoLo limits for the ramp
  ✓ Indicate ‘disconnected’ state via a pink border
  ✓ Gray out if it is not allowed to write
OPI Runtime

- Works like a web browser
  - Display OPIs in Tabs. A tab can be rearranged by dragging it to a new place or new window
  - Open related Display in a new Tab by Ctrl+click or in a new Window by Shift+click.
  - Backward or Forward Navigation
  - Zoom In/Out
  - Top OPIs
  - Full Screen
  - Compact Mode
  - Elog and Email Screen
Demo

- Editor
- Runtime
- Virtual Linac Display
**Rules**

- Easily make widget properties dynamic

  - Condition depended property value
  - Directly output PV value to a property
  - Allow multiple rules on a widget
Scripts
- Intelligentize your OPI

- For more complex logic that Rules cannot achieved
- JavaScript grammar
- Executed whenever one of the input trigger PVs change
- Able to access the widget controller and input PVs
- Able to attach data to the widget
- Able to access the children widgets of a container
- Able to call Java code
- Allow multiple scripts on a widget
Scripts
- Intelligentize your OPI

```javascript
importPackage(Packages.org.eclipse.jface.dialogs);
importPackage(Packages.org.csstudio.opibuilder.scriptUtil);

var flagName = "popped";
if(widgetController.getExternalObject(flagName) == null){
    widgetController.setExternalObject(flagName, false);
}

var b = widgetController.getExternalObject(flagName);
if(FVUtil.getDouble(pvArray[0]) > 80){
    if(b == false){
        widgetController.setExternalObject(flagName, true);
        MessageDialog.openWarning(null, "Warning", "The temperature you set is too high!");
    }
}else if(b == true){
    widgetController.setExternalObject(flagName, false);
}
```

Access Widget
Access PV
Call Java code

Warning
The temperature you set is too high!
Actions

• Every widget can have actions on it
• Actions will appear on widget’s context menu during runtime
  – The first action could be triggered by mouse click if it is hooked to mouse click event
• Can also be executed from script
  – `widgetController.executeAction(index);`
Macro
- Input once, change everywhere

• Embedded in string based properties

• Format: $(macro_name) or $ {macro_name}

• Predefined Macro
  – Macros can be defined in preference page, Display and Container’s Macros property, actions that invoke related display
  – Macros can be inherited or overridden

• Widget Property Value Macro
  – All properties value can be accessed via $(prop_id)
Color & Font Macro
- Easy to have consistent look for your OPIs

• Predefine colors or fonts in text files
• Using Color and Font macro will help you
  ✓ Giving a consistent look to the OPIs on your site
  ✓ Reuse some particular color or fonts
  ✓ Input once and change everywhere
  ✓ Apply different theme
Font Macro
- Easy to have consistent look for your OPIs

```
//Default Title Font. It will used in case the OS specified
Title = Arial-bold-18

//Title Font for Linux GTK
Title(linux_gtk) = Sans-bold-16

//Title Font for MacOS
Title(macosx) = Lucida Grande-bold-18

Header1 = Arial-bold-16
Header1(linux_gtk) = Sans-bold-16

Header2 = Arial-bold-14
Header2(linux_gtk) = Sans-bold-14

Text = Arial-regular-10
Text(linux_gtk) = Sans-regular-10
Text(macosx) = Lucida Grande-regular-10

LinkText = Arial-italic-10
LinkText(linux_gtk) = Monospace-italic-10

LinkText2 = Arial-bold italic-10
LinkText2(linux_gtk) = Monospace-bold italic-10
```
Interaction with other CSS tools

- As most of CSS tools, CSS context menus connect BOY with other CSS tools
Drag & Drop

- Drag Source could be text or PV
- Drop to OPI editor to create widgets
- Drop to widgets on OPI runtime will change its PV
URL Path Support

• Access files on web
  – http://... ftp://...

• Access file in a plugin:
  – platform:/base/plugin/<plugin name>/<path to file>

• Support URL Path at every place that needs file path input
  – Preference Page
  – Image Widget
  – Related Display
  – Linking Container
Border

• Every widget can have border!
• Especially useful for Grouping Container
Simulation PV

• Provides simulation PV, local PV, System PV and constant PV inside CSS
  – Simulation PV is useful for test purpose or used as a timer
  – Local PV is useful for inner communication in OPI
  – System PV can give information about system, such as memory, user name, time…

• For Example:
  – sim://noise(-10,10,0.2) A noise PV
  – loc://demo(1.23) A local writable PV in memory
  – sys://time Local date and time
  – See Online Help ->CSS Applications ->Configuration->Process Variables

• Belongs to Utility.PV layer
Widgets

• Support various type of data
  – Double, Integer, String, Enum, Boolean, Waveform...

• Allow plugging in customized widgets to BOY
Widget - XY Graph

- line chart, scatter chart, bar chart, step chart, area chart...
- Five Zoom Types, Panning, Auto Scale, Add/Remove Annotations, Undo/Redo, Take snapshot.
- Configure properties during runtime
- multiple axes
- log scale, date time format
- Annotations

- Could be reused for any SWT based Eclipse Applications.
Widget – Intensity Graph

- Display 2D data
- Can be used for video display
- Generate Profile data
- Zoom In/Out
Widget – Tab Container

- A real Tab widget
- Allow different settings on each tab
- Context Menu
  - Remove Tab
  - Duplicate Tab
  - Add Tab After
  - Add Tab Before
Widgets (selected)

- Image
  - Support GIF Animation
- Image Boolean Button
  - Simple way to get customized looking button
- Linking Container
  - Reuse an OPI with Macros
  - Make animation with Rules
Performance

• A benchmark test
  – Windows Vista, 2.6GHz Quad Core CPU, 4G Memory
  – 2 OPIs, each has 500 TextUpdates with PVs update at 10Hz
  – 28% CPU usage, 88M memory usage

• Configurable GUI Refresh Cycle
  – Increase to 1000ms will reduce CPU usage to 10%
Technical View

BOY Framework

Abstract Widget
- Properties
- Macro
- Actions
- Color & Font Macro

OPI Editor
- Navigator
- Palette
- Outline
- Toolbar
- Console
- Context Menu
- Properties Sheet

OPI Runtime
- Toolbar
- Console
- Context Menu
- Script Engine
- Utility
- PV

GEF
- CSS Platform

Eclipse

XML Reader & Writer

Welcome to the Technical View of the BOY Framework. This diagram illustrates the components and features of the framework, including OPI Editor, OPI Runtime, GEF, and CSS Platform. The abstract widget is central to the design, encompassing properties, macro, actions, and color/font macro functionalities. The OPI Editor and OPI Runtime sections highlight navigators, palettes, toolbars, consoles, and context menus, along with script and utility engines. The GEF and CSS Platform components integrate with Eclipse, providing a comprehensive toolset for development and customization.
Technical View

• Coding in Java and based on Eclipse
• Eclipse Graphical Editing Framework (GEF)
• BOY is a set of Eclipse Plugins
• Integrated with CSS Platform natively, but it’s possible to integrate it with other RCP
• Reused part of the code from Synoptic Display Studio (SDS)
• The BOY widget figures and XYGraph library are standalone plugins so that they can be reused for other SWT or Draw2D applications
BOY At SNS

- SNS still primarily on EDM
- BOY used for special displays
BOY At SNS

- Operators start to create top-level displays
Example: “Steering” Tool

Try to get spot into the green, at least into orange

Tim Southern, Nick Luciano
Beam Loss Monitor

Tim Southern, Nick Luciano
Collaborators

- John Hammonds (APS)
  - MEDM to BOY Converter
  - Byte Monitor

- Ralph Lange (BNL): Many suggestions

- Feedback and suggestions from all around the world

- Nadine Utzel (ITER): Created many OPIs for ITER
Planning

• Data browser widget
• Polishing online help
• Replacing Utility PV layer with PV manager
• EDM2BOY Converter (no rush)
• Continually improve BOY by listening to users’ feedback
Summary

• BOY is an integrated OPI Editor and Runtime
• OPI Editor is a modern WYSIWYG graphical editor with comprehensive functions to facilitate your OPI editing
• OPI Runtime works like a web browser
• Rules and JavaScripts can be used to dynamic everything and giving logic to your OPI
• BOY is a set of RCP plugins written in Java based on Eclipse and GEF
• We have several collaborators and users on BOY
Thank you!

• BOY Home Page

• Download
  – Unpack and Run CSS, no installation needed.

• Install BOY Examples
  – Menu: CSS->Display->Install OPI Examples

• Online Help