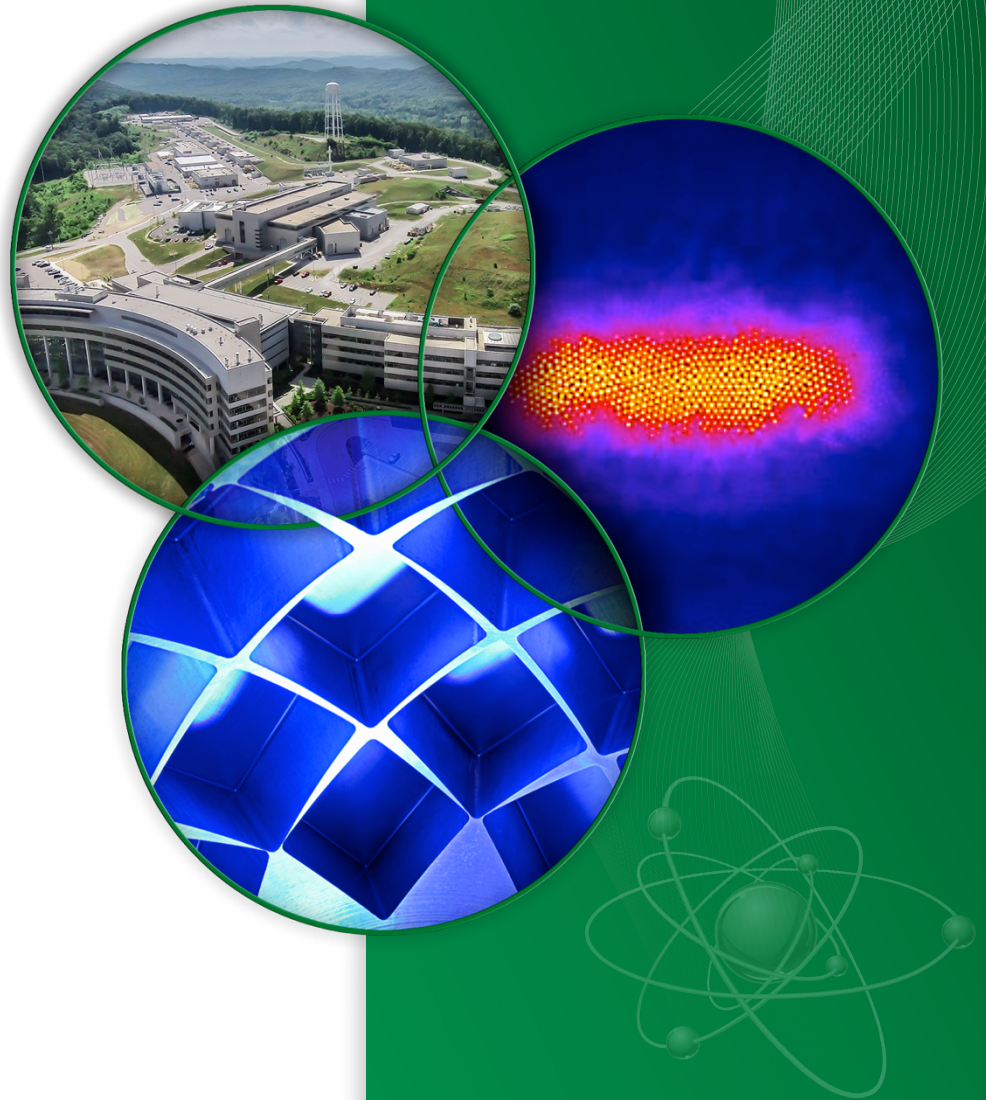


display.builder Update Project

Oct. 2015

Kay Kasemir

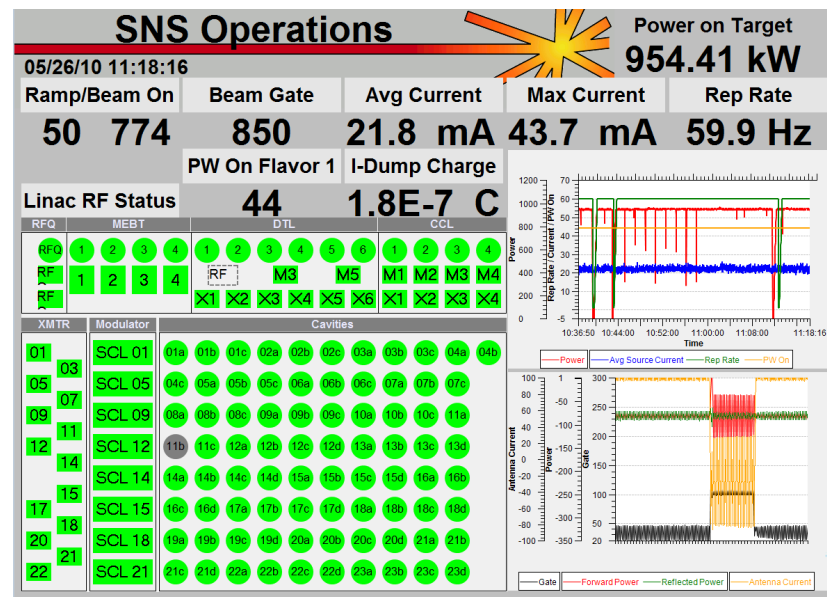


Why?

BOY is the control system user interface on SNS beam lines

.. also used at NSLS-II, KEK, FRIB, ITER,

DLS invested in good EDM translation



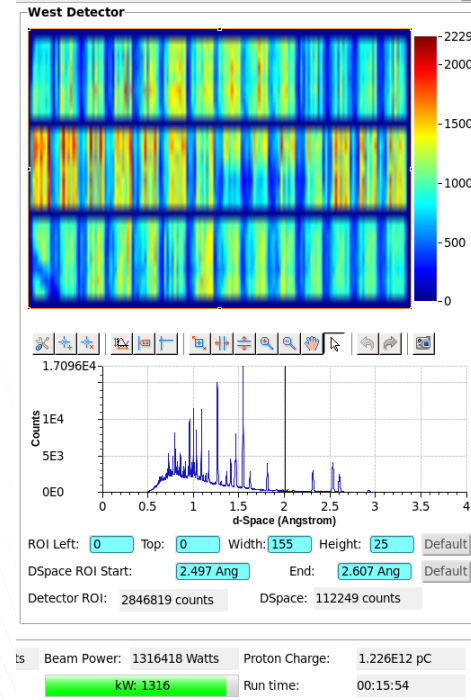
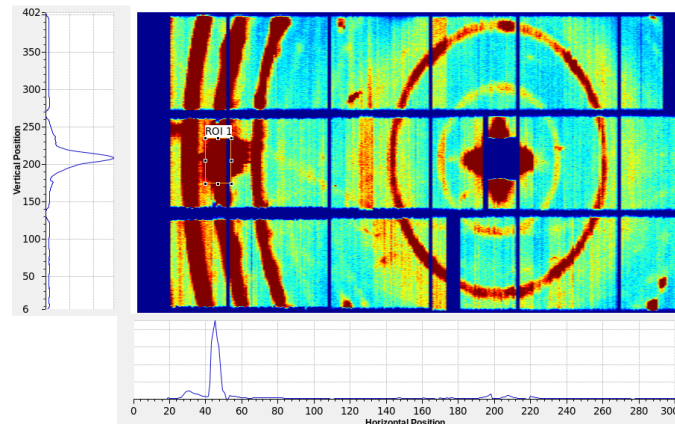
Motor BL9:Mot:Slit1:Left

S1 Left

2.000 mm STOP

- 2.000 mm +

Tweak 0.100 mm More



What's Good

- Editor
 - Various Alignment options
 - Edit properties of multiple widgets
- Widgets describe meaning
 - LED with PV and alarm-sensitive border, *not* Circle with border and background color dynamically linked to PVs
 - Group, *not* Rectangle that happens to be around other widgets



Accelerator	
Mode:	Target
Power:	1331.05 kW
Charge:	2.3622E-5 C
Energy:	939.500 Mev
Rate:	59.9 Hz

Scripts

- Are evil and should be avoided
- .. but still better than separate python/Qt, tcl/tk, perl/
wxWindows, .. tools

Model's XML

- .. is dump of widget property map.
- “Save As” without change results in different file; impossible to track real changes.

→ Save elements in consistent order

Model Loading

- Is on UI thread, freezing application.
 - Keeps warning “..was created with newer version..” because of compile time, not actual version changes
- Model load/save in background threads.
- Let widgets handle version changes.

Data Flow

- .. moves early on to UI thread,
- .. including scripts!

→ Rewrite.

In background thread:

PV updates → optional scripts → model changes

On UI thread:

UI displays model changes

Why don't we just change it?

- Model tied to SWT/Draw2d widgets.
- Runtime depends on *GEditorF*.

→ Rewrite.

Model: Hierarchy of Widgets w/ Properties.
Representation: Maps model to SWT, JavaFX.
Runtime: Connect to PVs

Could have different web repr&run

display.builder Idea

To end user, very similar to BOY.

Internally, independent...

1. Model

- Widgets and properties (no SWT/Draw2D/RCP)

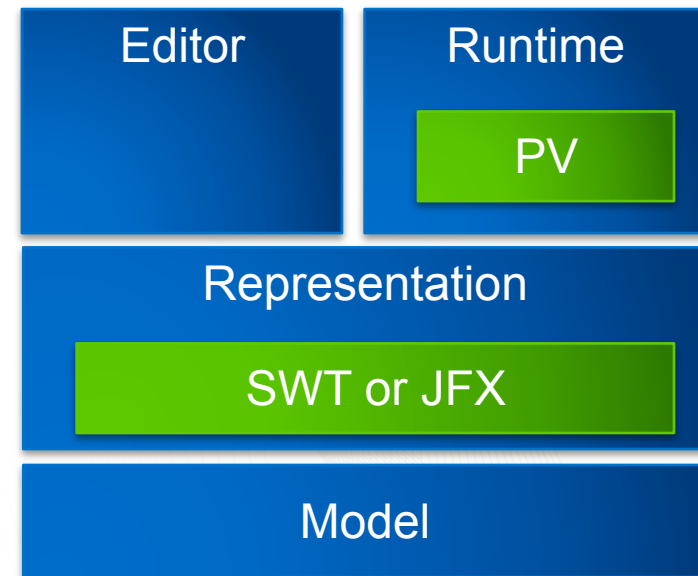
2. Representation

- SWT, JavaFX: Replaceable

3. Runtime

- Handles PVs, Scripts

4. Editor



Model

- Widgets
 - Rectangle, Label, TextUpdate, LED, ProgressBar, Image, XYPlot
... for now, implemented only essential properties
 - Group (contains other widgets)
 - Embedded Display
 - ActionButton
- Properties
 - Change notification
 - Well defined category & order
 - Colors, Fonts
 - Structures, Arrays (XYPlot “axis”, “traces”)
- Persistence
 - Loads existing *.opi files
 - Saves in defined order
- Macros
 - For text as well as ‘x’, ‘visible’, ..

TODO

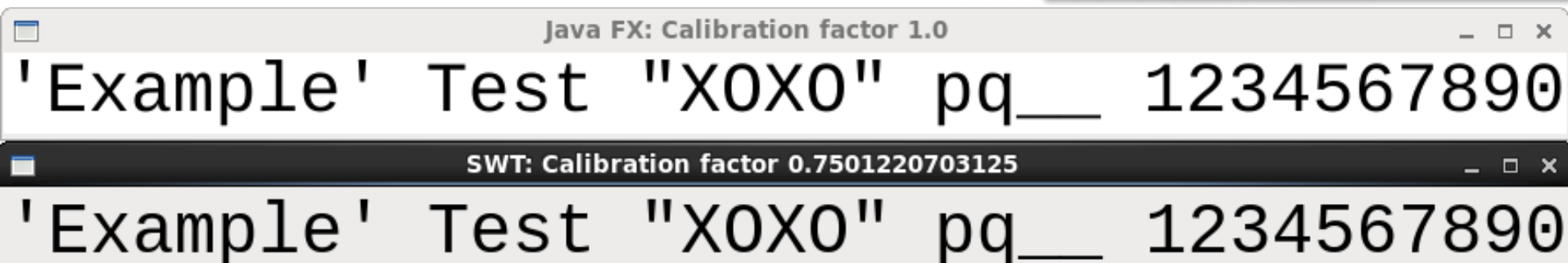
- Many more properties and widgets
- Defaults based on ‘class’ property

Representation

- SWT
 - Incomplete, just to show it's possible
- Java FX
 - For all widgets
- Fixed the font size problem!
 - BOY didn't distinguish between "point size" and pixel

TODO

- Many more widgets
- Many properties: Alarm sensitive, ..
- Web Representation



Runtime

- Common Widget Runtime
 - Loads model in background
 - Creates representation on UI thread
 - Handles PVs and scripts in background
 - One Jython interpreter & thread per window
 - Throttles UI updates
- EmbeddedDisplay Runtime
 - Keeps hosted display private
- ActionButton Runtime
 - Replaces current or opens new window
 - Writes PVs

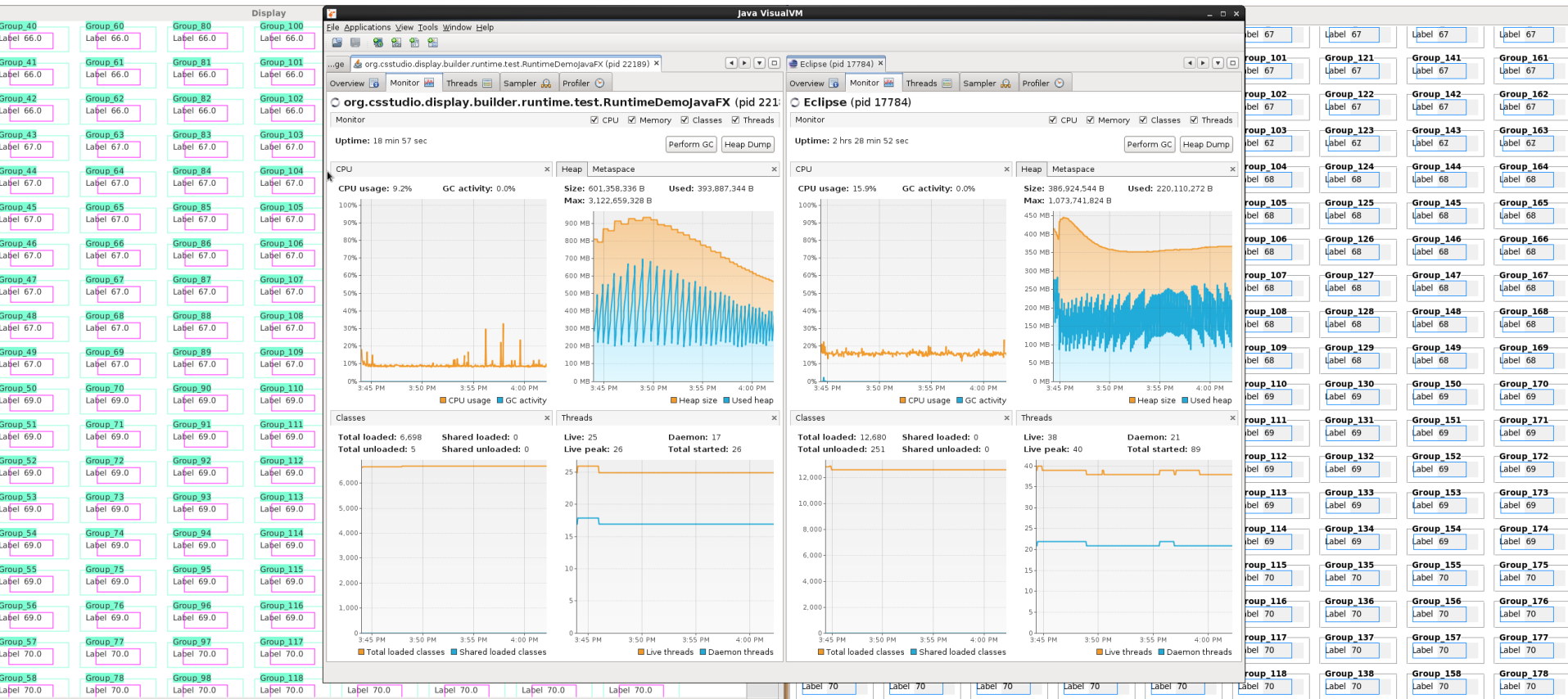
TODO

- Navigate back/forward



CS-Studio/SWT vs. display.builder/JavaFX

- 200 { Group, Label, TextUpdate (10Hz), Rectangle (10 Hz Script) }
- Scripts, PVs, Model updates off the UI thread
- Windows: Same CPU Load.
Linux: Lower CPU Load for JavaFX 😊😊



Compare BOY and display.builder

SNS Instruments Overview: very similar!

The image displays two side-by-side screenshots of instrument control software interfaces, comparing the BOY (left) and display.builder (right) systems. Both interfaces show a grid of instrument panels for various beamlines at SNS.

BOY Interface (Left):

- BL-1A USANS:** Shutter (green), Main (button), T0 Chopper (red), IPPS (red).
- BL-1B NOMAD:** Shutter (green), Vacuum (red), T0 Chopper (green), IPPS (red).
- BL-2 BASIS:** Shutter (green), IPPS (red).
- BL-3 SNAP:** Shutter (green), T0 Chopper (red), IPPS (red).
- BL-4A MRef:** Shutter (green), IPPS (red).
- BL-4B LRef:** Shutter (green), IPPS (red).
- BL-5 CNCS:** Shutter (green), IPPS (red).
- BL-6 EQ-SANS:** Shutter (green), IPPS (red).
- BL-7 VULCAN:** Shutter (green), Main (button), Detector (green), Choppers (green), IPPS (red).
- BL-9 CORELLI:** Shutter (green), Main (button), Vacuum (red), T0 Chopper (green), IPPS (red).
- BL-11A POWGEN:** Shutter (green), Main (button), Vacuum (red), T0 Chopper (green), AGES (green), Agilent PS (green), IPPS (red).
- BL-11B MANDI:** Shutter (green), Main (button), Vacuum (green), IPPS (red).
- BL-12 TOPAZ:** Shutter (green), Main (button), IPPS (red).
- BL-13 FNPB:** Shutter (green), IPPS (red).
- BL-14B HYSPEC:** Shutter (green), Main (button), IPPS (red).
- BL-15 NSE:** Shutter (green), Main (button), IPPS (red).
- BL-16B VISION:** Shutter (green), Main (button), IPPS (red).
- BL-17 SEQUOIA:** Shutter (green), Main (button), IPPS (red).
- BL-18 ARCS:** Shutter (green), Main (button), IPPS (red).

Accelerator Panel (Right):

- Mode: MEBT-BS
- Power: 0.00 kW
- Charge: 0.0000E0 C
- Energy: 939.500 Mev
- Rate: 1.0 Hz

Summaries Panel (Bottom Right):

- Gateways (button)
- ODH (button)

display.builder Interface (Right):

The display.builder interface is a duplicate of the BOY interface, showing the same grid of instrument panels and accelerator/summary information. The layout and controls are identical to the BOY interface shown on the left.

Compare BOY and display.builder

- Loads via http://
- Groups update macros of contained widgets

TODO: 'alarm sensitive' color/border/..

CA Gateways						CA Gateways					
	WebOPI	CMF	USANS	NOMAD	SNAP		WebOPI	CMF	USANS	NOMAD	SNAP
PV Count:	1125	10	2	74	2	PV Count:	1125.0	10.0	2.0	74.0	2.0
Connected:	1025	10	2	74	2	Connected:	1025.0	10.0	2.0	74.0	2.0
Active:	860	10	2	74	2	Active:	860.0	10.0	2.0	74.0	2.0
Not connected:	100	0	0	0	0	Not connected:	100.0	0.0	0.0	0.0	0.0
Received Events:	37.62 Hz	1.80 Hz	0.00 Hz	2.80 Hz	0.00 Hz	Received Events:	37.62 Hz	1.80 Hz	0.00 Hz	2.80 Hz	0.00 Hz
Sent Events:	65.93 Hz	2.30 Hz	0.50 Hz	3.30 Hz	0.50 Hz	Sent Events:	65.93 Hz	2.30 Hz	0.50 Hz	3.30 Hz	0.50 Hz
Loop Rate:	111.65 Hz	100.95 Hz	99.15 Hz	101.35 Hz	99.25 Hz	Loop Rate:	111.65 Hz	100.95 Hz	99.15 Hz	101.35 Hz	99.25 Hz
Gateway CPU:	0.028 %	0.005 %	0.006 %	0.001 %	0.006 %	Gateway CPU:	0.028 %	0.005 %	0.006 %	0.001 %	0.006 %
Host Load:	0.000	0.000	0.000	0.160	0.070	Host Load:	0.000	0.000	0.000	0.160	0.070
	VULCAN	CORELLI	POWGEN	MANDI	VISION		VULCAN	CORELLI	POWGEN	MANDI	VISION
PV Count:	92	335	142	25	176	PV Count:	92.0	335.0	142.0	25.0	176.0
Connected:	92	335	142	25	176	Connected:	92.0	335.0	142.0	25.0	176.0
Active:	61	318	142	25	144	Active:	61.0	318.0	142.0	25.0	144.0
Not connected:	0	0	0	0	0	Not connected:	0.0	0.0	0.0	0.0	0.0
Received Events:	0.30 Hz	11.91 Hz	4.90 Hz	1.70 Hz	3.20 Hz	Received Events:	0.30 Hz	11.91 Hz	4.90 Hz	1.70 Hz	3.20 Hz
Sent Events:	1.10 Hz	16.31 Hz	5.40 Hz	2.20 Hz	3.70 Hz	Sent Events:	1.10 Hz	16.31 Hz	5.40 Hz	2.20 Hz	3.70 Hz
Loop Rate:	99.85 Hz	102.65 Hz	101.55 Hz	100.25 Hz	100.45 Hz	Loop Rate:	99.85 Hz	102.65 Hz	101.55 Hz	100.25 Hz	100.45 Hz
Gateway CPU:	0.003 %	0.001 %	0.003 %	0.004 %	0.001 %	Gateway CPU:	0.003 %	0.001 %	0.003 %	0.004 %	0.001 %
Host Load:	0.000	0.440	0.000	0.040	0.020	Host Load:	0.000	0.440	0.000	0.040	0.020

Compare BOY and display.builder

Nested groups, embedded displays.

Resolves file locations 'relative' to parent,
passes macros

The screenshot displays the 'Slits S1' interface, which is organized into nested groups and embedded displays. The main window is titled 'Slits S1' and contains two primary sections: 'Virtual Motors' and 'Real Motors'.

Virtual Motors Section:

- Motor BL9:Mot:Slit1:X:Center:** S1 Horizontal Center. Value: 0.0002. Tweak: 1.0000. Buttons: STOP, -, +, More.
- Motor BL9:Mot:Slit1:X:Gap:** S1 Horizontal Gap. Value: 4.0005. Tweak: 1.0000. Buttons: STOP, -, +, More.
- Motor BL9:Mot:Slit1:Y:Center:** S1 Vertical Center. Value: 0.0007. Tweak: 0.1000. Buttons: STOP, -, +, More.
- Motor BL9:Mot:Slit1:Y:Gap:** S1 Vertical Gap. Value: 59.9996. Tweak: 1.0000. Buttons: STOP, -, +, More.

Real Motors Section:

- Motor BL9:Mot:Slit1:Left:** S1 Left. Value: 2.000 mm. Tweak: 0.100 mm. Buttons: STOP, -, +, More.
- Motor BL9:Mot:Slit1:Right:** S1 Right. Value: 2.001 mm. Tweak: 0.100 mm. Buttons: STOP, -, +, More.

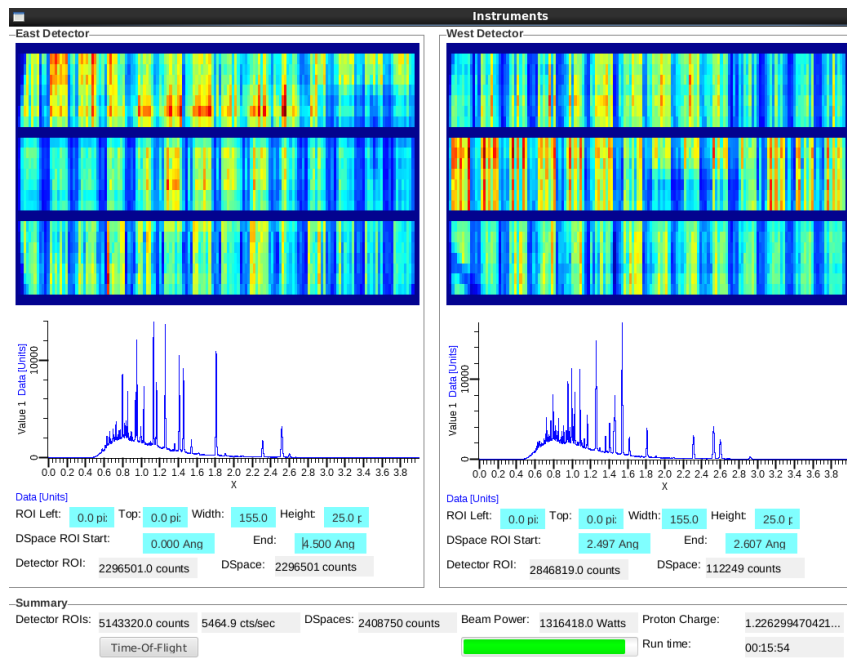
Slits S1 Section:

- Virtual Motors:**
 - Motor BL9:Mot:Slit1:X:Center:** S1 Horizontal Center. Value: 0.0003. Tweak: 1.0000. Buttons: ST..., -, +, More.
 - Motor BL9:Mot:Slit1:X:Gap:** S1 Horizontal Gap. Value: 4.0006. Tweak: 1.0000. Buttons: ST..., -, +, More.
 - Motor BL9:Mot:Slit1:Y:Center:** S1 Vertical Center. Value: 0.0003. Tweak: 0.1000. Buttons: ST..., -, +, More.
 - Motor BL9:Mot:Slit1:Y:Gap:** S1 Vertical Gap. Value: 60.0006. Tweak: 1.0000. Buttons: ST..., -, +, More.
- Real Motors:**
 - Motor BL9:Mot:Slit1:Left:** S1 Left. Value: 2.000 mm. Tweak: 0.100 mm. Buttons: ST..., -, +, More.
 - Motor BL9:Mot:Slit1:Right:** S1 Right. Value: 2.001 mm. Tweak: 0.100 mm. Buttons: ST..., -, +, More.
 - Motor BL9:Mot:Slit1:Bottom:** S1 Bottom. Value: 30.000 mm. Tweak: 5.000 mm. Buttons: ST..., -, +, More.
 - Motor BL9:Mot:Slit1:Top:** S1 Top. Value: 30.001 mm. Tweak: 0.100 mm. Buttons: ST..., -, +, More.

A 'Slit System Drawing' button is located at the bottom right of the interface.

Plotting

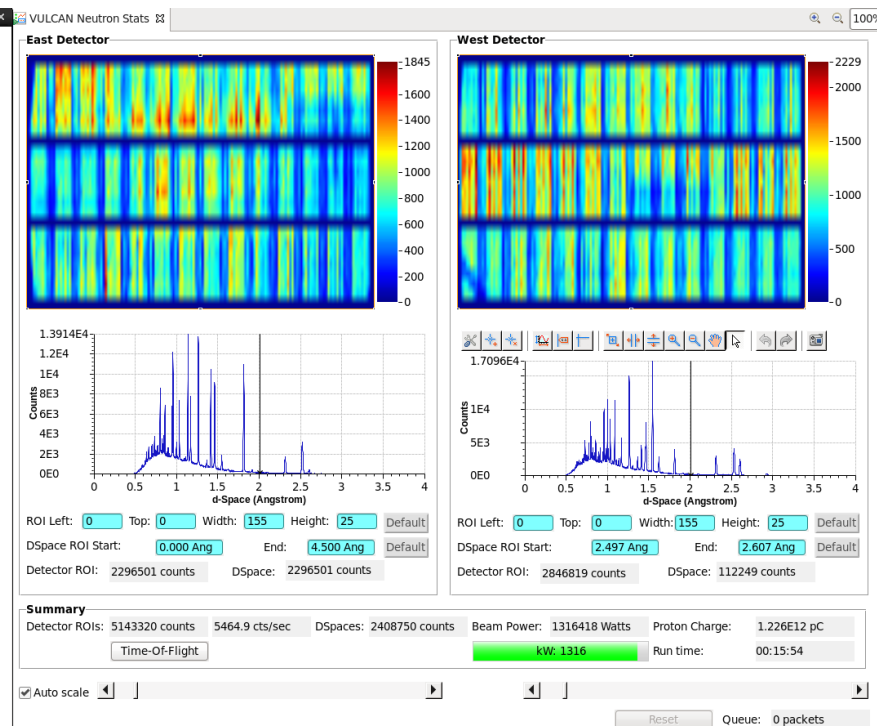
Basics of Image and XYPlot



Display Builder

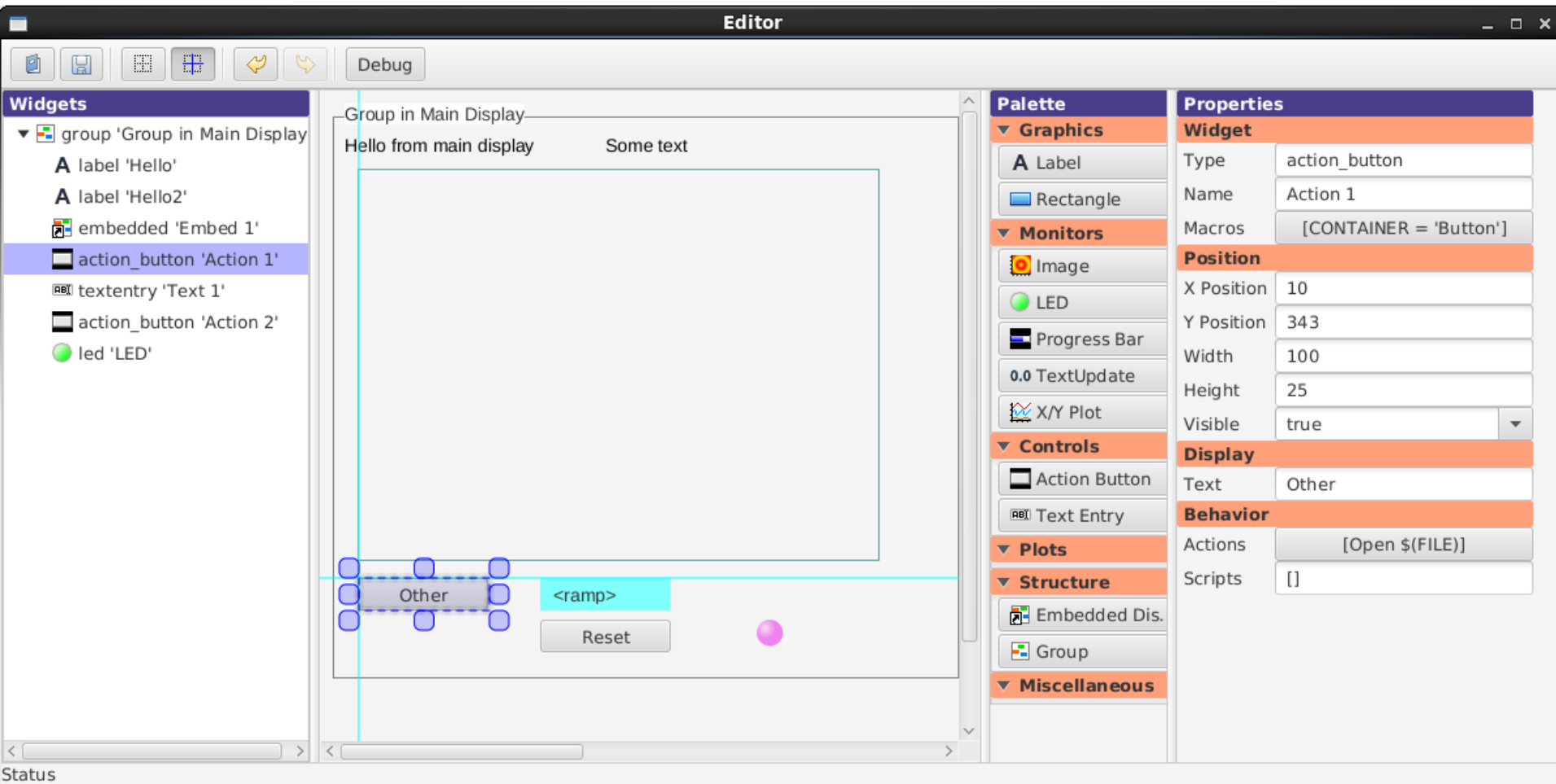
Reset Queue: 0.0 packets

Process Name	Status	% CPU	Nice	ID	Memory	Waiting Channel	Session
gnome-screensaver	Sleeping	0	0	10413	2.4 MIB	poll_schedule_timeout	
java	Sleeping	12	0	9901	553.5 MIB	futex_wait_queue_me	
java	Sleeping	8	0	10258	1.3 GiB	futex_wait_queue_me	
gnome-system-monitor	Running	4	0	9949	5.3 MIB	0	



BOY

Editor Exploration



- Grid, Guidelines
- Common Properties

Editor Exploration

- Palette of available Widgets
- Show current model: Display and Widget-Tree Outline
- Editable Properties
 - In defined order
- Tracker to move/resize
 - Snap-to-Grid
 - Snap-to-Geometry (parallelized)
- Toolbar
- Selection:
Rubberband, multi-widget,
in and out of 'Group'

TODO

- Complex properties
 - Scripts
 - XYPlot “Axis”,
“Trace” structure
- Rulers
- Copy/Paste

Summary

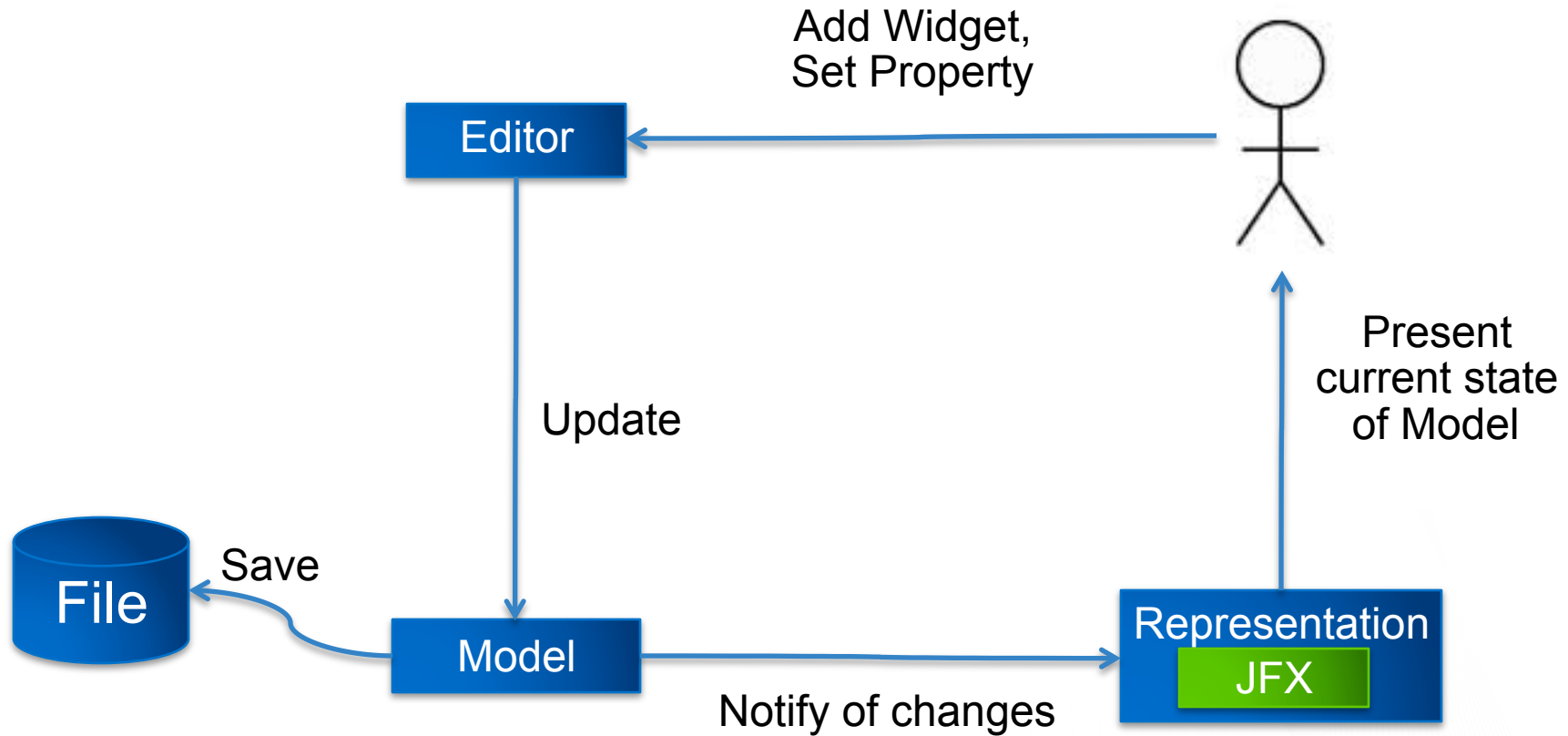
Reads existing files, looks the same

- Separate Model, Representation, Runtime
- Background threads whenever possible (less freezing)
- Typically faster

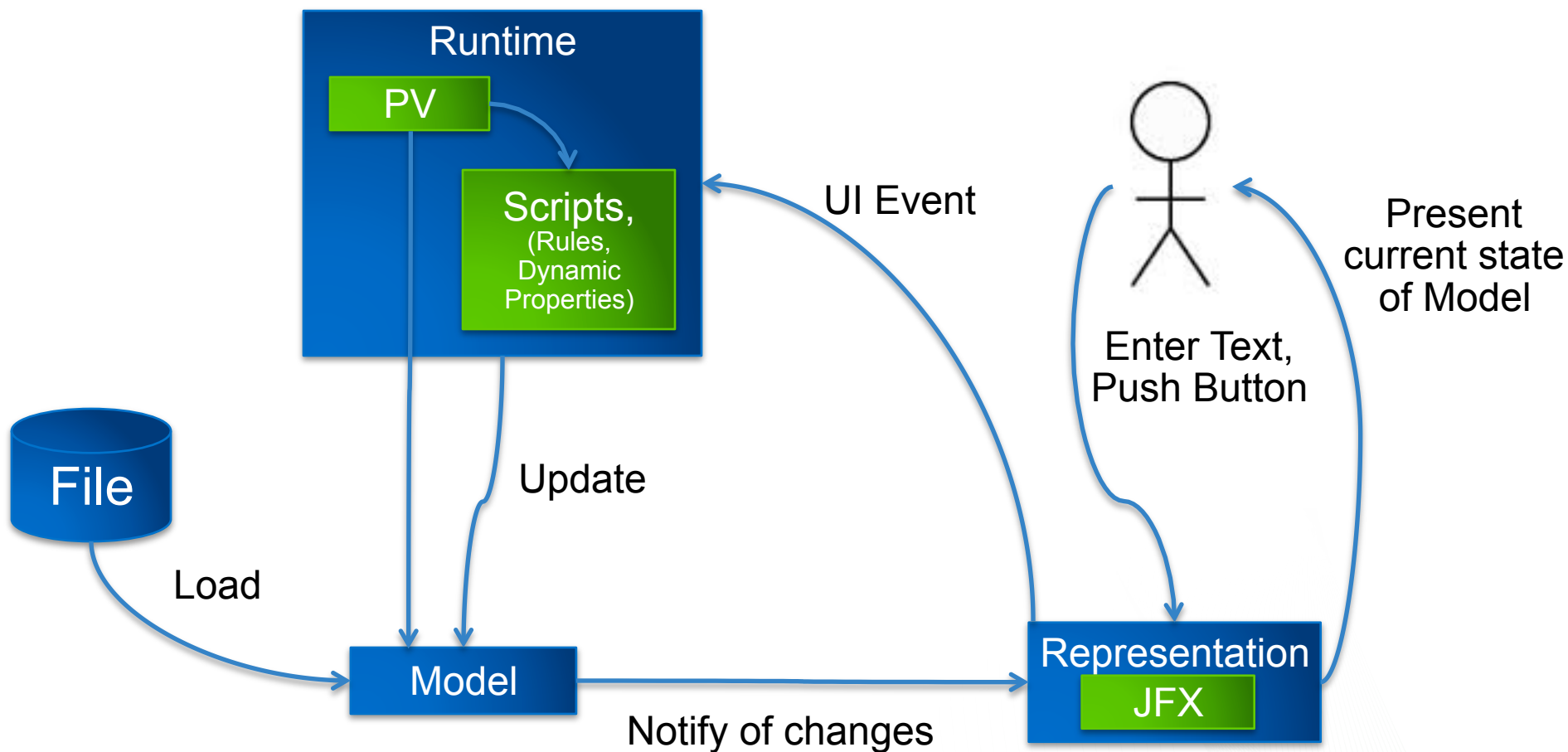
To Do

- Many widgets, gazillion properties
- Editor smoothness
- Eclipse RCP integration (FXCanvas, context menus)

Design Time Data Flow



Runtime Data Flow



JavaFX Quirks ☹️

- Java API vs. Style sheets
 - LED uses API for basic color, css for ‘highlight’
 - css limited to single –fx-effect, while API can chain multiple.
 - ProgressBar
 - Default css is nice, but blue.
 - API for setting color would get flat, solid rectangle.
- ToolTip only for ‘Control’, not ‘Shape’
- Mouse clicks unaware of Shift/Control/Alt/.. State
 - Additional onMouseDown(), which needs to ‘arm()’ button
- Custom-drawn Widgets
 - ‘Canvas’ draws on UI thread
 - WritableImage only offers set(x, y, color)
 - Using AWT to draw image in background, then display in Canvas

Linking Container

- Model merges linked content into one large model, then uses “instanceof AbstractContainer” to hack around the result.

→ Rewrite.

Container has *internal* model of linked content.

‘Connectors’ to widgets within container can’t be supported because content may change.

Editor

- GEF?
 - “Will not be developed further”
 - Requires Draw2D representation for every widget
- GEF4?
 - Provisional API
 - GEF4 MVC = Demo of Bezier curve editor?
 - No ‘Palette’, no Context Menu