JavaFX and CS-Studio: Benefits and Disadvantages in Developing the Next Generation of Control System Software

Claudio Rosati, ESS, Lund, Sweden – Kunal Shroff, BNL, Upton, Long Island, New York – Kay Kasemir, ORNL, Oak Ridge, Tennessee

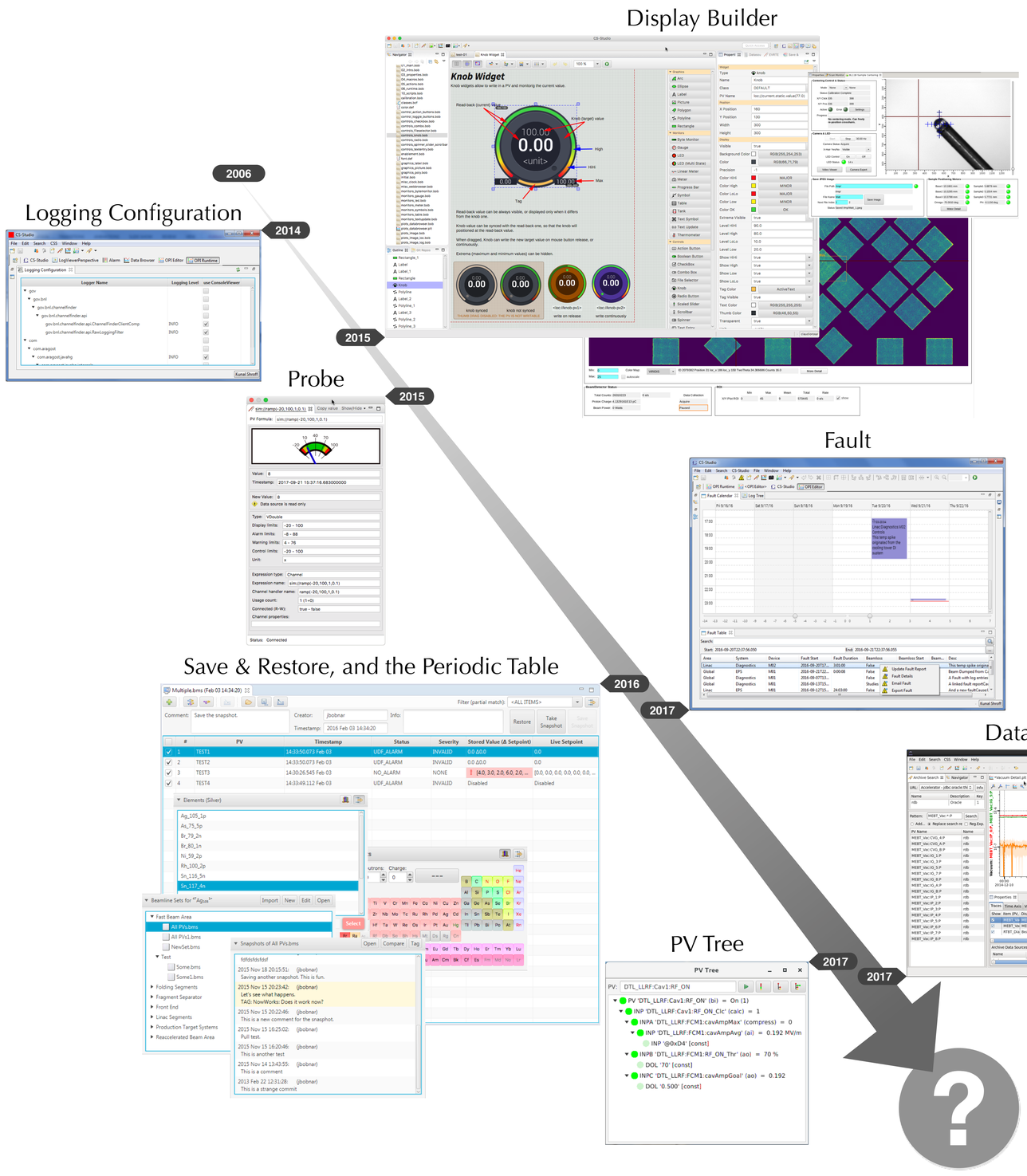


Figure 1: JavaFX features in CS-Studio

Phoebus Application Framework

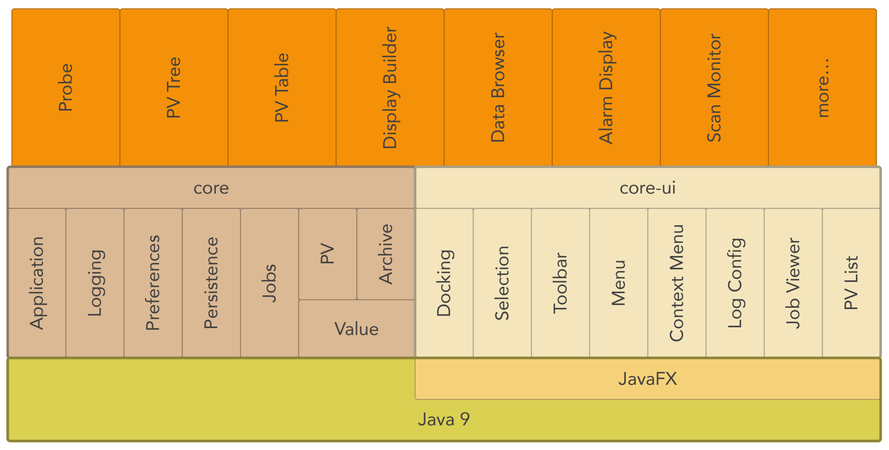
- Project started few months ago to explore the feasibility of using the new Java 9 JPMS and the standard SPI as means for porting the existing codebase into a new JavaFX-only application;
- Allow easy migration of the functionalities of key CS- Studio tools, specifically the *Display Builder*, *Data Browser*, *PV Table*, *PV Tree*, *Alarm UI*, and *Scan UI*, supporting their original configuration files with 100% compatibility;
- Provide full control of windows placement, free from RCP restrictions;
- Use JavaFX as the graphics library to overcome limitations of SWT;
- Prefer core Java functionality over external libraries whenever possible (JavaFX for UI, Java 9 Platform Module System for bundling, SPI for locating and loading services and extensions, `java.util.logging` for logging and `java.util.prefs` for preferences, …);
- Reduce build system complexity, fetching external dependencies in one initial step, then supporting a fully standalone, reproducible build process;
- Provide core shared modules and services (Actions, Menu bar, …);
- Available at https://github.com/shroffk/phoebus.

Eclipse RCP Benefits

- OSGi packaging, control of exported packages, dependencies;
- Extension points mechanism;
- Cross platform (Mac OS X, Linux, Windows);
- Configuration of site-specific products via plugins and features;
- OSGi console with ‘telnet’ access;
- Jetty for services that have web interface;
- Hierarchical preferences;
- Workspace persists window layout and preference changes;
- Object contribution mechanism for context menu;
- Support for CVS, Subversion and Git;
- PyDev;
- XML Editor;
- WikiMedia Editor;
- Maven (in theory).

Eclipse RCP Disadvantages

- ‘Editors’ part stack doesn’t participate in Perspective handling, opening up when any ‘Launcher’ is invoked;
- No control over initial location of newly opened parts, with restrictions on moving parts around (this being one of the bigger overall issue);
- API has grown in too many directions: PopupMenu, ActionSets, Action & Handler, …;
- JavaFX in SWT FXCanvas is sluggish;
- Workspaces and associated restrictions running on multiple machines;
- Somewhat heavyweight application with slow initial launch phase;
- Maven (in practice).



ICALEPCS2017
Barcelona · Spain, October 8-13 · Palau de Congressos de Catalunya

claudio.rosati@esss.se
shroffk@bnl.gov
kasemirk@ornl.gov

europeanspallationsource.se
bnl.gov
ornl.gov