XAL Status

Thomas Pelaia II

ICALEPCS 2007
October 15, 2007
Active XAL Developers

• Christopher Allen
• Chungming Paul Chu (Stanford Linear Accelerator)
• Sarah Cousineau
• John Galambos
• Jeff Holmes
• Thomas Pelaia II
• Andrei Shishlo
• Yan Zhang
• Alexander Zhukov
What is XAL?

• Infrastructure for building accelerator physics applications and scripts
• Collaboration among laboratories
• Open Source Java packages
• Developed for accelerator physics commissioning and studies for the Spallation Neutron Source at Oak Ridge National Lab
• Independent of any Integrated Development Environment (IDE)
XAL Toolbox

• Several general packages (solver, interpreter, math, …)
• Channel access package (wraps JCA)
• Services
• Accelerator Hierarchy
• Online model for accelerator physics
• Application framework
Application Framework

• Infrastructure for document based applications

• Common Look and Feel
  – Standard and customizable toolbar and menus

• Delivers behaviors users have come to expect

• Rapid development
Document Based Application

• Application Adaptor
  – Handles application wide behavior
  – Provides application properties (name, file types …)

• Document
  – Reads and writes a data archive
  – Main controller for a document window

• Document Window
  – Main view corresponding to a document
Recent Application Framework Enhancements

• Universal Copy, Cut and Paste
  − Java drag and drop support == free and automatic copy, cut and paste support

• Improved Visual Cues
  − Menu and toolbar items gain standard icons

• Desktop Pane Support

• Bricks User Interface Construction Application
Motivation for Bricks

• Reduce threshold for creating an application
• Encourage Model-View-Controller (MVC) design pattern
• Eliminate code associated with layout
• No good Java tools for graphically constructing user interfaces
Existing Java Graphical Interface Construction Tools

• Poor support for MVC
• Force you to use a particular IDE
• Add code complexity instead of reducing it
• No support for Java based scripting languages
What is Bricks?

• Application for constructing gorgeous graphical user interfaces

• Runtime package for instantiating graphical user interfaces
Bricks Features

• Graphical layout of views

• Stores layout of views within an XML file instead of computer generated code

• Runtime package provides access to any view
  – Code assistant works with any IDE

• MVC compliance
  – You write code for the Models and Controllers
  – Bricks takes care of instantiating the Views

• Great for Java based scripts
  – No need to compile any code
Bricks Application

View Hierarchy

Preview

View Palette

View Inspector
Accessing Views

• Get a window reference to instantiate a window from a Bricks file

```java
dialogReference = document.getDefaultWindowReference("HostConfigDialog", owner);
```

• Get any view from a window reference using its tag

```java
JButton addButton = (JButton)dialogReference.getView("AddButton");
```
Bricks and XAL Application Framework

• *Bricks* and the XAL Application framework can be used together or independently of each other

• Application Framework has support for *Bricks*

• *Bricks* has support for Application Framework

• You pick the technologies that work best for your application
Applications and Scripts

• Over four dozen applications
• Numerous scripts
• Several new and enhanced applications
RTBT Wizard (recent enhancements)

• Provides tools for measuring and transporting beam through RTBT and onto the Target

• Improved target beam position projection and user interface

• Integrated target beam parameters report generation and browsing
Lossviewer II (Sasha Zhukov)

• Monitors and displays beam losses
• Rewritten from scratch
• Support for multiple types of loss detectors (currently beam loss monitors and neutron detectors)
• Consolidated display
Knobs (Tom Pelaia)

• Change multiple process variables concurrently with specified coefficients relative to knob value

• Users can define multiple knobs and group them within a document

• Knob coefficients set either manually or with automated tools
Magnet Cycling (Andrei Shishlo)

• Convenient way to cycle bend magnets to remove hysteresis effects

• Individual magnet configuration of cycling properties including number of cycles and dwell times
RF Simulator (Yan Zhang)

- Simulates both the RF controller and beam in the ring.
- Mimics beam loading and RF feed-forward and feedback processes
Quad Shaker (Andrei Shishlo)

• Measures beam position relative to quadrupoles

• Determines quadrupole misalignments

• Corrects beam position relative to quadrupole centers
Trip Monitor and Viewer (Tom Pelaia)

- Service to continually monitor trips
- Logs trips to database
- Application manages service and browses historical trips
Collaboration

• XAL has been an open source, collaborative project from the start

• Code contributions from collaborators around the world

• Fragmentation of source code among labs

• New effort to address collaboration

• http://sourceforge.net/projects/xaldev
Summary

• XAL is mature with over four dozen applications
• Several new and enhanced applications and scripts
• Bricks allows for rapid construction of user interfaces
• New features for Application Framework
• XAL collaboration is evolving