Software Architecture for FRIB Diagnostics

Terence Brown
Diagnostics Engineer
Outline

- Introduction
- Service-Oriented Architecture
- Java Server Faces (JSF)
- ADF Faces Capabilities
- Application Development
- EPICS Connectivity
- Conclusion
Introduction

- **Goals and Objectives**
  1. Provide beam instrumentation data to support Physics, Commissioning, and Operations
  2. Maintain supervisory layer access to critical diagnostics devices
  3. Provide processed data to all users and data loggers

- **Overview of FRIB**
  - Nuclear science user facility that utilizes a sc heavy-ion linac to achieve multiple charge state acceleration of 200 MeV/u ions to produce isotopes.

» Devices and Systems
- Emittance scanner
- Beam Current Monitor (BCM)
- Beam Position Monitor (BPM)
- Beam Loss Monitor (BLM)
- 3-D Profile Monitor
- Transverse Profile Monitor
- Video Profile
- Fast Faraday Cup
Fall 2010 – Investigated Java Enterprise Edition (EE) platform to identify methods for rapid application development

Java EE platform provides for development of scalable, reliable, secure enterprise applications within a service oriented architecture

Identified Java Server Faces (JSF) as the user-interface technology

Applications Prototyped and Motivation
• Diagnostics: Advanced applications for remote testing, calibration, and monitoring of devices

  • Physics: Capability to browse the accelerator lattice and export beamline component tables for beam physics studies

  • Commissioning: Configuration application with beamline segment views of device parameters and an intuitive user-interface
Service-Oriented Architecture & Java Server Faces
Service-Oriented Architecture

- Distributed systems require implementation of a Service-Oriented Architecture (SOA)
- Architecture allows building a collection of reusable services
- SOA provides integration for applications across systems
- Ensures robustness, scalability, and performance
- Rapid application development
SOA Application Framework

- Application Framework
  - Java EE 6 web services
  - Application messaging
  - Relational database
  - Model-View-Controller (MVC)
    - View: graphical user interface
    - Model: service layer

- Building SOA Applications
  - Develop Java services
  - Create service communication
    - Messaging, data transformation, and message routing
  - Secure services
  - Optimize services
  - Build graphical user-interface

http://www.oracle.com/technetwork/developer-tools/adf/
Java Server Faces

- **Introduction to JSF**
  - Component framework that enables Java EE developers to create and deploy server-based rich internet applications (RIAs)
  - **Versions**
    - JSF 1.0 (2004) – Initial Specification
    - JSF 1.2 (2006) – Included in Java EE5
    - JSF 2.0 (2009) – Enhanced functionality with Java EE 6

- **Advantages**
  - Pure Java applications
  - Pure thin-client model
  - Applets and browser plugins are not required
  - Java EE development standard
  - JavaScript development is not required

- JSF with built-in Asynchronous Java and XML (AJAX)
ADF Faces Capabilities
ADF Faces Capabilities [1]

- JSF is the Java EE standard MVC framework
- Support for web, desktop, and mobile devices
- Oracle ADF provides extended capabilities
  - ADF Faces – feature rich JSF components for dynamic UI
  - ADF Controller – extended controller with reusable task flows
  - ADF Binding – simplified binding of view and controller to services

http://www.oracle.com/technetwork/developer-tools/adf/
ADF Faces Capabilities [2]

- **ADF Faces Rich Components**
  - Currently over 150 JSF components
  - Components are AJAX enabled
  - Provide pluggable look and feel
  - Accessibility & internationalization
  - In addition to standard components:
    » Gantt chart, Geographical map, pivot tables, calendars, gauges
  - Built-in advanced functionality:
    » Drag and drop framework
    » Dialog and pop-up framework
    » Active Data displays (push updates)
    » Page Templates

http://www.oracle.com/technetwork/developer-tools/adf/
ADF Faces Capabilities [3]

- **Layout Components**
  - Page arrangement using areas to provide advanced layouts
  - Relative positioning
  - Layout can be updated at runtime
  - Define layout of child components

- **Advanced components**
  - Decorative Box
  - Panel Collection
  - Panel Splitter
  - Panel Accordion
  - Navigation Pane
  - Panel Group Layout
  - Panel Tabbed
ADF Faces Capabilities [4]

- **Common Components**
  - Display/update information
  - Input components
    - Rich Text Editor
    - Select many List box
  - Output components
    - Output Formatted
    - Messages
  - Dynamic components (push)
  - Navigation components
  - Query components – search
  - Table components – collections
ADF Faces Capabilities [5]

- **Operation Components**
  - Add functionality and behavior to existing components
  - Drag-and-drop capabilities
  - Client listener capabilities
  - Validation capabilities
  - Data exporting and printing
  - Implement advanced behaviors
    - Collection Drag Source
    - Collection Drop Target
    - Convert Date Time
    - Show Popup Behavior
    - Auto Suggest Behavior
ADF Faces Capabilities [6]

- **ADF Data Visualization Graphs**
  - More than 50 types of graphs
  - Rendering via Flash or PNG
  - Interactive features:
    - Zoom in/out
    - Scroll
    - Time selection
    - Line and legend highlighting/fading
    - Dynamic reference lines and areas
    - Animation

- **ADF Gauges**
  - Status Meter: standard and threshold
  - Dial Meter: standards and threshold
  - LED: arrow, dot, and triangle
ADF Faces Capabilities [7]

- **Gantt Chart**
  - Track tasks and resources on a time schedule
  - Types: project, scheduling, resources

http://www.oracle.com/technetwork/developer-tools/adf/
ADF Faces Capabilities [8]

- Partial Page Rendering (PPR)
  - Enables redrawing only a portion of a page
  - Can be enabled declaratively or programmatically

- Triggering component:
  » Id = "syssel"
  » Autosubmit = "true"

- Target Component:
  » partialTriggers = "syssel"
ADF Faces Capabilities [9]

- **Page Templates**
  - Reusable ADF Faces page that contains place holders for custom page content
  - Templates are interpreted at runtime
  - Authors can change the template for an existing application without opening the application itself
  - Templates may accept parameters for passing information from the inheriting page to the template
Summary

• ADF Faces provides for rapid development of rich web user interfaces

• Easy page layout

• Advanced user interaction

• Rich dynamic data visualization

• Visual and declarative development
Application Development & EPICS Connectivity
Application Development [1]

- **Database Driven Applications**
  - Leverage existing database design
  - Build an application framework
  - Java programming infrastructure
  - Common application configuration

- **Application Development**
  - User single sign-on (security)
  - Main application menu
  - Application server for deployment
  - Each application runs as a service
  - Applications access distributed resources
Application Development [2]

- Relational Database Design
  - RDB structures for beamline devices, signal records, etc.
  - FRIB lattice imported to database
  - Database may be populated using web-based applications
  - Contains global coordinates structure

- Prototype Java Application Development
  - Device configuration
  - Lattice browser
  - System monitoring
Application Development [3]

- **Main Menu**
  - Select from available applications

- **Device Configuration**
  - Select a beamline sub-device by system or subsystem
  - Select a signal record for a specified device
  - Configure signals and related devices:
    » Signal fields
    » Signal records
    » IOCs
    » Power supplies
    » BPM
    » Port assignment
**Application Development [4]**

- **Lattice Browser**
  - Browse/configure beamline devices (BPMs, PMs, etc.)
  - Built-in nested search and sort
  - Query by system or subsystem
  - Legend for naming convention
  - Devices managed
    - Profile monitor
    - IOCs
    - Power supplies
    - BPMs
    - BLMs
Multiple Views

- Expanded view: Allows selection of system/subsystem and easy browsing

- Detached view: Provides quick access to search functionality over large systems
Application Development [6]

- Advanced Development
  - Java EE application with JSF GUI
  - Integrated Data Visualization
  - Built-in device search and sort
  - Device calibration panels
  - Online device monitoring
  - Remote device testing

T. Brown, 14 June 2011, EPICS Collaboration Meeting, Session 3, Slide 26
EPICS Connectivity [1]

- Applications
- EPICS OPI
- EPICS IOC
- Timing card
- DAQ electronics
- Primary electronics
- Detector
- HV, motion
- Network
  EPICS Channel Access
- DAQ bus (PCI/PIXI, PCIe/PXIe, VME, USB)
- MPS node

T. Brown, 14 June 2011, EPICS Collaboration Meeting, Session 3, Slide 27
EPICS Connectivity [2]

- Basic requirements for building a JSF application
  - Java Server Faces 2.1 – Download Components
  - Data Visualization Tools (DVT)
  - Flexible Integrated Development Environment (IDE)
    » Eclipse Helios
    » NetBeans 7
    » JDeveloper 11g
  - Data Service
  - Java Channel Access (JCA)

- Data Acquisition
  - Fast Digital Acquisition (DAQ) bus
  - Modular form factor preferable
  - Extensive choice of DAQ cards
  - DAQ card drivers readily available
  - Unified software interface for DAQ cards
Conclusion

- JSF technology is the standard for Java EE web user-interfaces

- Open source and commercial AJAX enabled component frameworks are available

- EPICS connectivity via Data Service and JCA

- Extremely rapid prototyping of JSF applications

- Imagine the Possibilities…