EVLA Overall Software Design
Final Internal Review

System-wide Services:
User Interface

Topics

• Design Goals
• Languages and Technologies
• Application Types
• Deployment
• EVLA Subsystem User Interface Needs
• Active Development
• Demo: Device Browser
Design Goals

• Where possible, prefer general purpose UIs
  – Not hard-wired to a known interface
  – Resistant to changes in the EVLA System
• Thin (lightweight) rather than thick clients
  – Use thick (or rich graphic) applications only when necessary
• Simple and Intuitive
  – Easy to learn
  – Reduces the time it takes to train users
• Cater to both expert and novice users

Design Goals (cont.)

• Consistent Look and Feel
  – Adhere to prescribed set of user interface guidelines
• Easy to deploy
• Common EVLA Communication Infrastructure
  – Communicate with all EVLA devices in the same manner
  – Will be determined by the recently established Distributed Object Communication team
Languages and Technologies

• Most of our development will be done using the Java programming language
  – Platform Independent, “Write once, run anywhere”
  – Object-oriented
  – Extensive tool and library availability
  – Strong industry support
  – Many years combined experience of EVLA developer staff

Languages and Technologies (cont.)

• Java environments downloadable from Sun
  – J2SE (JRE)
    • Includes JVM, Java Plug-in and Java Web Start
    • JDBC, JFC/Swing
  – J2SE (SDK)
    • Same as JRE, adds compilation and debugging tools
  – J2EE (Enterprise Edition)
    • Adds JSP, Servlets and XML tools
Languages and Technologies (cont.)

- Castor
  - Java/XML data binding framework
- HTML
- Jython
  - A Java Python interpreter
- Thinlet
  - Lightweight GUI library (< 38KB)
- Maven
  - Project management and comprehension tool

Application Types

- Web-based Applications
  - Ubiquitous
  - Most often used for static form-based applications
  - Easy to deploy
  - Supporting Technologies: Java Applets, JSP, Servlets, JDBC
- Standalone Java Applications
  - Typically run on desktop or laptop PCs
  - More dynamic than Web-based applications
  - Provide richer graphics and tend to be more full featured than Web-based applications
Deployment

- Web-based Applications
  - Need only deploy HTML files, Servlets to Web server
- Java Web Start (for standalone Java applications)
  - A Java application deployment mechanism
  - JWS applications can be launched from a browser or from the desktop, but unlike Applets, JWS applications do NOT require a browser
  - Automatically checks and downloads any changes to the application

Subsystem UI Needs

- Proposal Construction, Submission and Management
  - Users: Astronomers and NRAO personnel
- Program and Observation Preparation
  - Users: Astronomers
- Observation Scheduling
  - Users: Operators (Astronomers)
- Observation Executor
  - Users: Operators
- Monitor and Control of AMCS/CMCS
  - Operators, Engineers, Techs, Developers
- Observation Status Screen
  - Users: Astronomers
Active Development: LabVIEW Adapter

- Provides a means of monitoring and controlling EVLA devices via LabVIEW screens
- The first version of the LabVIEW Adapter was released the first week of June
- LabVIEW screens are currently being developed for T304 module
- The same code used by the Device Browser to connect to the MIBs is also used by the LabVIEW Adapter
- Deployed using Java Web Start. See http://www.aoc.nrao.edu/asg-internal/maven-test/jnlp/jnlp.html

Active Development: Device Browser

- The Device Browser is a general-purpose GUI that displays hierarchical information about the device to which it is connected
- Being used for Ant 13 testing
- Capable of connecting to any MIB and the CBE. In the future it will connect to any EVLA device
- Provides both monitor and control capabilities
- Default screens exist for Devices, Monitor Points and Control Points
- Custom screen generation
- Deployed using Java Web Start
- Runs on Desktop and PDAs
Demo: Device Browser

14 June 2004