EVLA Monitor & Control Software PDR

Operational Interface: Requirements and Design Considerations
Agenda

- Operational Requirements
- System Attributes
- Deployment
- Communications Protocols
- Recommendations
- Q&A
Requirements Document

• Operational Interface SRS
  – Revision 2.0 (document #A24101N0001), dated April 4, 2002

• Purpose of the document
  – Identify the top-level requirements
  – Will be used to design the system

• Future of the document
  – Make changes as required
  – Changes should go through a review process (formal/informal)
  – Replace “TBD”s with known values
Requirements Overview

• Supported (Client) Platforms
  – Commodity PCs/Windows/Linux (Required)
  – Sun/Solaris (Desirable)
  – Macintosh/Mac OS (Optional)
  – Requires the software to be written in a platform independent language (e.g., Java)

• Remote Observing
  – Operators may someday operate the array from the AOC
  – Engineers and technicians will need access to the system from the AOC and their homes
• Installation and upgrades
  – How will users get and install the software?
  – Client software must be available via the Internet

• Security
  – Two groups of users: those we trust and those we don’t
  – At a minimum, users must supply a username and password for authentication
  – All passwords must be encrypted
Requirements Overview (cont.)

- **Robustness**
  - System will not crash due to network glitches, broken sockets, reboots or resets of other devices

- **Reliability**
  - MTBF = 7 days

- **Availability**
  - 99.5% (unavailable 48 hours over a year’s time)

- **Maintainability**
  - Modular code
  - Standards for coding and documentation

- **Usability (Ease of use)**
  - Intuitive interfaces
  - Adhere to user interface design guidelines
  - Reduce the time it takes to train operators (currently 3 months)
# User Capabilities

<table>
<thead>
<tr>
<th>Antenna</th>
<th>VLA</th>
<th>AOC/NRAO Sites</th>
<th>WWW</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Operators</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Engineers/Technicians/Programmers</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scientists</td>
<td></td>
</tr>
<tr>
<td>Monitor/Control</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Rich Moeser
EVLA Monitor & Control Software PDR
May 14-15, 2002
System Attributes

• Loosely Coupled and Highly Adaptive
  – Changes to the core M&C system should have no affect on the client (i.e., the client will not crash or misbehave)
    • Requires a high degree of encapsulation on the core M&C system
    • Requires minimization of interface dependencies

• Discovery Based
  – Dynamic discovery of objects/services
  – Requires some form of look-up mechanism
  – The more the client can find out about the system at runtime, the more flexible and extensible the system
System Attributes (cont.)

- Screens
  - **Predefined**
    - Highly dependent on the interface
    - Higher degree of customization than auto-generated screens
  - **Automatically Generated**
    - Requires a known interface or introspection
    - Useful in the absence of predefined screens
    - Less refined than predefined screens
- **Lightweight Client**
  - **Little or no knowledge of underlying business logic**
  - **Concerned only with the presentation**
Deployment Options

- How do we get the software on the client machines?
  - CD
    - Costs time and money to burn and ship
    - Update notification via email, Web page or mailing list
  - Downloadable tar/zip file
    - User’s role: download and install
    - Update notification via email, Web page or mailing list
Deployment Options (cont.)

• Java Applets
  – Will work on essentially any OS and browser
  – Tightly linked to the browser
    • Runs within the browser’s address space
    • When the browser exits so does the applet
  – Requires Java Plug-in (most browsers use older versions of JVM)
  – Runs within “Sandbox”
    • Restricts access to system resources
      – Filesystem
      – Print capabilities
Deployment Options (cont.)

• Java Web Start
  – Web deployment mechanism for Java applications that runs on the client
  – Allows the user to launch the application from the browser or the desktop
  – Integrated into Java 2 platform as of Java 1.4
  – Allows full-featured Java applications on the client
  – Launched, deployed and updated from standard Web server
  – Allows client application to be launched while offline
Deployment Options (cont.)

• **Java Web Start (cont.)**
  
  – **How it works**
    
    • User clicks on a link to launch client-side application
    • Web browser instructed to run Java Web Start (association of JNLP file to JWS)
    • JWS connects to Web server and determines what files (if any) need to be downloaded
    • Files are downloaded to local machine
    • JWS runs the downloaded Application
    • Next time users tries to launch the application, JWS will download only the files that have changed

  – **Java Network Language Protocol (JNLP) file**
    
    • Specifies: JAR file(s) to download, run time parameters, Java version, etc.
Web page link:

<a href="demo.jnlp">DeviceBrowser Demo</a>

Sample JNLP file:

```xml
<?xml version="1.0" encoding="utf-8"?>
<!-- JNLP File for the EVLA Device Browser Demo Application -->
<jnlp spec="1.0+"
codebase="http://lorax.aoc.nrao.edu:8080/"
href="http://lorax.aoc.nrao.edu:8080/demo.jnlp">
  <information>
    <title>EVLA Device Browser Demo Application</title>
    <homepage href="docs/help.html"/>
    <description>EVLA Device Browser Demo Application</description>
    <description kind="short">A demo of the EVLA Device Browser.</description>
    <offline-allowed/>
  </information>
  <security><all-permissions/></security>
  <resources> <j2se version="1.2+"/>
    <property name="java.rmi.hostname" value="lorax"/>
    <jar href="demo.jar"/>
    <application-desc main-class="DeviceBrowser"/>
  </resources>
</jnlp>
```
Communications Protocols

• How will the clients communicate with the core M&C system?
• Some of the many options that exist
  – Java RMI
  – CORBA
  – XML-RPC
  – SOAP
Communications Protocols

- **Java RMI** (Remote Method Invocation)
  - Java’s RPC mechanism
  - Requires Java on both ends
  - Requires a centralized naming server (rmiregistry)
  - Easy to use
  - Distributed garbage collection

- **CORBA** (Common Object Request Broker Architecture)
  - Language neutral
  - Optional naming service
  - Somewhat of a steep learning curve
Communications
Protocols (cont.)

- XML-RPC
  - Uses HTTP as the transport and XML (EXtensible Markup Language) as the encoding
  - Simple, portable way to do RPC over HTTP
  - Implementations
    - Perl, Python, Java, C/C++, .NET, Tcl and many more
    - http://www.xmlrpc.org

```
<methodCall>
  <methodName>sample.sum</methodName>
  <params>
    <param><value><int>5</int></value></param>
    <param><value><int>3</int></value></param>
  </params>
</methodCall>
```
Communications
Protocols (cont.)

- **SOAP** (Simple Object Access Protocol)
  - SOAP is an XML based protocol used to exchange information throughout a distributed environment
  - Key component of the Web Services technology stack
  - An interoperability standard
  - Strong industry backing (Microsoft, Sun, IBM,…)
  - Three parts:
    - Envelope
    - Header
    - Body
<?xml version='1.0' encoding='UTF-8'>
  <SOAP-ENV:Header></SOAP:Header>
  <SOAP-ENV:Body>
    <m:DoLogin xmlns:m="urn:soapserver/soap:AuthorizationModule">
      <UserName>UserNameGoesHere</UserName>
      <Password>PasswordGoesHere</Password>
    </m:DoLogin>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
Communications Protocols (cont.)

- SOAP (cont.)
  - SOAP solutions in the MIB
    - C/C++ in the MIB
      - Embedded SOAP libraries
        » eSOAP
        » gSOAP
  - Concerns
    - Performance
Recommendations

- Use Java on the Client
  - Fulfills target platform and OS requirements
    - Commodity PCs (Windows/Linux)
    - Sun (Solaris)
    - Macintosh (Mac OS)
- Use Java Web Start for Deployment
- Use SOAP (and XML) where possible
  - Strong industry backing
  - Allows for a loosely coupled and extensible system
- Further Explorations
  - IML, AIML, IRC Project
  - LMTMC Software
  - GBT
  - DRAMA