Drag and Drop Controls Display and Builder
(Synoptic Display)

Timofei Bolshakov, Andrey Petrov
Fermilab Accelerator Controls Department
March 26, 2007
Motivation

- Operations of complicated colliders (Tevatron, the LHC, and the ILC) require sophisticated control systems.
  - Large interlocking sub-systems
  - Security
  - Data pool management
  - Alarms
  - Logging

- The people who must build, operate, and maintain these accelerators
  - Operators
  - Engineers
  - Accelerator physicists

require rapid development of control displays and application programs.
Motivation

- For rapid development, the system expert (operator, engineer, or physicist) should be the one to develop the displays or applications.

- These advanced control systems can seem overwhelming to non-controls experts.
  - This is why Lab View is so popular
  - However Lab View offers little of the benefits of an advanced control system.
Drag and Drop is an environment that gives non-control system experts the ability to quickly build controls displays which operate inside the control system framework.

Drag and Drop:
- is easy to use
- sophisticated enough to handle complex displays
- operates inside the control system framework.
- uses web browsers and/or Java Web Start (requires no extra software installation)
- is easily extendible
- is a mature application
  - Developed in 2001
  - Fermilab Cryogenics department are heavy users
  - Well documented
Drag and Drop Controls Display and Builder

- Drag and Drop a consists of two parts:
  - Display
  - Builder

- The Drag and Drop Display can be run from a web browser (readings only) so it can viewed anywhere in the world.
  - Files are stored on a web server (well organized and secure)
  - Displays are extremely quick because it uses Scalable Vector Graphics (SVG) so that the screen does not constantly have to be re-drawn.

- The Builder has a simple graphical user interface that offers a rich set of graphical components
  - Display can be built and deployed in a matter of minutes
  - The builder is easily extendible
  - can be run on any machine because it is based on Java
- Cryo/Meson/CompressorRoom.xml
- florian/linac6.xml
- Cryo/A0/A0PiCryo9cell.xml
- test/Example.xml
Demo – Empty Builder
Demo – Builder, Meson Compressor Room
Demo – Klystron, Step 2, Project Properties
Demo – Klystron, Step 3, Adding a Device
Demo – Klystron, Step 5, Adding Indicator
Demo – Klystron, Step 6, Adding Graphics

The image shows a software interface for designing and customizing display elements. The interface includes tools for selecting and modifying shapes, such as squares and arrows. A panel for setting properties of a selected shape is open, allowing for customization of fill color, stroke color, and stroke width. The user interface is designed to be user-friendly for creating and modifying technical diagrams.
Demo – Klystron, Step 9, More Devices
Demo – Klystron, Step 10, Setting Up Device
Demo – Klystron, Step 11, Connecting
Demo – Klystron, Step 13, Saving Project 2
Drag and Drop is an environment that gives non-control system experts the ability to quickly build controls displays which operates inside the control system framework.

Drag and Drop:
- is easy to use
- sophisticated enough to handle complex displays
- operates inside the control system framework.
- uses web browsers and/or Java Web Start (requires no extra software installation)
- is easily extendible
- is a mature application
  - Developed in 2001
  - Fermilab Cryogenics department are heavy users
  - Well documented
Fermilab – Requirements

No requirements –

System is working in production for 5 years,
actively used in Cryogenic Department
(some enhancements possible).
LHC – Requirements

- **Administrator's point of view:**
  - 1 computer with Tomcat / Jetty web server inside Technical Network – Windows, Linux, Mac or Solaris.
  - One proxy – record from department web-server.
  - File system for storing XML files (component description and projects) and / or class files – 100 Mb.

- **Software Engineer point of view:**
  - Some refactoring needed to utilize Generics & java.util.concurrent.
  - Device-type Components for JAPC.
  - Runtime Engine and Builder have to use RBAC and to be standard LSA application.
  - Visual Components with CERN DataViewer and other libraries.

- Porting can be done in days or in weeks (in case of refactoring).
Some software details.

- **Components:**
  - Simple interface: *init, start, stop* (lifetime), **properties** (how many inputs and outputs etc),
  - I/O pipes handling – read and write standard objects.
  - Visible components have to be AWT.
  - Some restrictions to be effectively rendered on SVG.
  - All properties are described in XML document – used in Builder and during Runtime.

- **Runtime Engine:**
  - Just standard (LSA for CERN) application with standard authorization (RBAC for CERN, Kerberos for FNAL).
  - Reads XML project description from web or from file.
  - Read only if started from web.

- **Builder** – produces projects XML, reads components XML descriptions.