The JCOP Framework
– A Practical Overview –

Sascha Schmeling
IT/CO • JCOP FW Team
Overview

- The JCOP Framework in a Nutshell
  - Aims
  - Architecture
  - Ingredients
  - Devices and Tools
  - Developments

- A Practical Example
Aims of the JCOP Framework

- Reduce the development effort
  - Reuse of components
  - Hide complexity
  - Facilitate the integration
  - Help in the learning process

- Reduce resources for maintenance
  - Homogeneous control system
  - Operation
  - Maintenance

- Provide a higher layer of abstraction
  - reduce knowledge of tools
  - interface for non experts

- Customize & Extend industrial components

- Modular/Extensible
  - core
  - mix & match components

- As simple as possible

- Development driven by the JCOP FW Working Group
System Architecture

- **JCOP FW (Supervisory)**
- **SCADA (PVSS)**
- **Supervisory Application**
- **FSM, DB, Web, etc.**
- **PC (Windows, Linux)**
- **FE Application**
- **Device Driver**
- **Commercial Devices**
- **UNICOS FW**
- **PLC**
- **Front-end**
- **Communication**
- **Supervision**

- **OPC, PVSS Comms, DIM, DIP ...**
- **Other Systems**
JCOP FW Ingredients

User Framework

PVSS

Commercial Supervisory Application

Event Manager

Data Manager

Controls Hierarchy

Device Editor Navigator

 OPC client

 OPC server

 FW Custom FE

 C/C++ code

 External System

 User specific FE

 Equipment

 GEH

 EAH

 OPC client

 OPC server

 Devices

 February 26th, 2003

 Sascha Schmeling, IT/CO • The JCOP FW Team
Devices

- **Generic Analog-Digital devices**
  - Analog Input/Output
  - Digital Input/Output
  - Process Monitor
- **CAEN power supplies**
  - Crates SY127, SY403, SY527, SY1527
  - Plans to include SY2527
- **Wiener power supplies**
  - OPC server being developed by the company
  - CAN interface. One board currently supported (NI)
- **Wiener Fan Tray**
- **ELMB**

- **ISEG power supplies**
  - OPC server being developed by the company
  - CAN interface. One board currently supported (Peak)
- **PS and SPS machine data server**
  - Common server provided for all experiments
  - SPS needs customization for each beamline
- **Logical Node/View**
  - Composite device
  - A means to build hierarchies of devices
Tools

- **Device Editor/Navigator**
  - Main interface to the Framework
  - System management: installation, login, etc
  - Configuration and operation of devices

- **Controls Hierarchy**
  - High level view of experiment
  - Includes FSMs

- **External Alarm Handler**
  - Receive alarms in PVSS from an external system

- **Trending**
  - Simplify & extend PVSS trends (templates, tree, etc)

- **Device Support Extension**
  - Template to incorporate new devices

- **Generic External Handler**
  - To incorporate C++ code to panels and ctrl manager
  - Easier to use than standard PVSS C++

- **Mass Configuration**
  - Initial release including creation and deletion

- **Component Installation**

- **Tree Browser**
  - Tree widget for Windows and Linux
Other Ingredients

- Guidelines to produce a coherent control system
  - Look and feel
    - e.g. colors, fonts, layout
  - Alarm classes
  - Naming convention
  - Exception handling
  - File organization
  - and so on

- Libraries
  - Setting of address, alerts, archiving, etc
  - List manipulation
  - Exception handling

- Examples
  - Panels
  - Buttons
  - Scripts

- Tutorial
  - Use of several FW tools
  - Connection to real hardware/simulator
  - Can be run in the SCADA lab or in your own lab
  - Available as a one day course from the Training Service
Current Version

- Version 1.2.1
  - Released in January 2003
  - Runs on Linux and Windows
  - PVSS 2.12

- Internal intermediate releases
  - To meet user specific needs
  - Small improvements/bug fixes
Current Developments

- **General**
  - Simplification of the addition of devices
  - Rewrite some of documentation

- **Mass Configuration**
  - Working on UC modify devices (alerts, archiving, etc)

- **Configuration**
  - Requirements defined within FW WG
  - 1st version of prototype foreseen for April-03, including:
    - ELMB, CAEN, Wiener LV, Siemens PLCs
  - Finished evaluation of XML, trying ODBC

- **Data Storage and Retrieval**
  - Requirements defined
  - Writing use cases
  - 1st version of prototype for Conditions DB by April-03

- **Access Control**
  - First implementation from UNICOS, based on SASG discussions
  - Integration in JCOP Framework in progress

- **Tests in a distributed environment**
  - Plain PVSS
  - PVSS + Framework

- **Trending**
  - Histograming
  - Prototype of integration with ROOT

February 26th, 2003
Sascha Schmeling, IT/CO • The JCOP FW Team
CosmoALEPH’

An example for a DCS based on PVSS and the JCOP Framework
A “Small” Experiment

Distances:

- Gallery - Trolley: 18 m
- Trolley - HCAL: 36 m
- Bypass A - Bypass C: 95 m
- Cavern - Bypass A/B: ~260 m
- Cavern - Alcove: ~925 m

Layout of CosmoAleph
The Hardware
Control System Architecture

- DCS
  - Station 1
    - Tracking
    - Trigger
    - Infrastr.
    - HV
    - Cool.,…
    - Gas
  - Station 2
    - Tracking
    - Trigger
    - Infrastr.
    - HV
    - Cool.,…
    - Gas
  - Station 3
    - Tracking
    - Trigger
    - Infrastr.
    - HV
    - Cool.,…
    - Gas

February 26th, 2003
Sascha Schmeling, IT/CO • The JCOP FW Team