System Integration of High Level Applications during the Commissioning of the Swiss Light Source

Andreas Lüdeke / PSI

Application Environment
Functionality Residence
Functionality Migration
Examples
Conclusion
Application Environment

User Interfaces
Connectivity
Services

Controls - Environment

GUI (java) CDEV

Gateway CA

GUI (tcl/tk) CDEV

Gateway CA

GUI (medm) CA

GUI (IDL) CA

Beam dynamics - Environment

GUI (tcl/tk) Corba

GUI (java) Corba

CDEV CA

Model Server

EPICS records

SNL device/driver sub

"Hardware" (DSPs, Controllers, SPS, ...)

Oracle Server

Orbit FB

Tracy Server

Analysis Server

Andreas Lüdeke

ICALEPCS 2001
**Controls Environment**

- Many standard applications
  - (Channel archiver, Alarmhandler, Striptool, Save & restore, ...)
- GUI builder (medm/dm2k, EMW)
- Perfect for hardware control
- High-performance & reliable network protocol (CA)

- Good connectivity to EPICS only
- Protocol inadequate for complex data
Beamdynamics Environment

+ Functionality provided by Client/Server Applications
+ Transparent Connections to
  A. EPICS channels
  B. Oracle Database
  C. Tracy Simulation Server
  D. Other Clients & Servers ...

- Data access different from EPICS
  A. Other widgets for each access type
  B. More manpower for GUI maintenance
Functionalty Residence

**User Interface Functions**

- Rapid prototyping
- Data processing tools available
- Many existing development environments (VisualAge, ...)
- Easy comparison of Methods

- Calculated values and states are private to the application
  - no access by standard tools
  - inconsistencies between applications
- Increases maintenance workload

**BD Env. Service Functions**

- Multiple connectivity to services
- High computing power
- Still no access by standard EPICS tools

**Controls Service Functions**

- High consistency between appl.
- Use of standard EPICS tools
- Reduced ability to use services
Solution: Functionality Migration

Use functionality in GUIs for prototyping
Migrate functions to server level when ready
GUIs and BD servers writing states and computed values to EPICS softchannels.
This allows to:
- export private states and values
- transparently migrate functions from GUIs to servers
Example: Optic Control

**Idea:**
Separate theoretical optic settings from adjustment parameters

**Nominal set-up for optic**
- currents, energy, sensitivity matrix, gradients, tunes and chromaticity.

**Control by physical parameters**
- Tune- and chromatic shifts
- Energy adjustment
- Sextupole scaling factor

**Implementation:**
- IDL application
- EPICS database for optic
Example: Orbit Feedback

SVD Algorithm
- Tracy server for calculations of correction matrix
- Analysis server to access EPICS and oracle data
- Correction by matrix multiplication

Orbit correction implementation
- Start: Manual correction by GUI ``OCO"
- 1st migration: Slow orbit feedback client
- 2nd migration: Fast feedback by local DSP
  A. SVD Engine client to calculate correction matrix on change of quads
  B. Actual correction performed by DSP
Summary and Conclusion

Approach:
Test functionality in prototype high-level application
Use softchannels to export application data
Migrate functionality to low-level application
Replace high-level application

Gain:
Flexible rapid prototyping
Transparent migration by soft channels
Ease of long-term maintainability