cosylab

CONTROL SYSTEM LABORATORY

Data Aaccess Layer for CSS

Igor Križnar

igor.kriznar@cosylab.com

DAL Goals

- Common API for accessing data from different CS sources
  - Based on interfaces rather then abstrat classes
  - Pluggable implementation of DAL interfaces
- Consistent OO and wide API design
  - simple access to the data (JavaBean style)
  - programming aids (auto-completion)
  - compile time error checking
  - easy debugging
- Narow style access with introspection
  - metadata
  - for generic applications
- CSS ready

Terms Used

- Device
  - Container for properties and commands
- Property
  - Container for dynamic remote data channel
  - Channel in EPICS
- Characteristics
  - Key-value pairs
  - Rather static description of device or property (min,max,units,etc.)

DAL Device Features

- Access to commands
  - As objects in generic device
  - As methods for particular device API
  - Synchronous/asynchronous execution
- Access to properties
  - By name in generic device
  - By API for particular device
- Access to characteristics
  - Synchronous, asynchronous, group
- Connection state (connected, disconneted, failed)
- Property grouping

DAL Property Features

- get/set for values
  - Synchronous and asynchronous
- Get for characteristics
  - Synchronous, asynchronous, as a group
- Data quality (condition), enum set of codes: error, alarm, timeout, etc.
- Events
  - Value update, change
  - Data quality
  - Custom events (blue beam)
- Concious data types
  - Double, Long, String, BitSet, Object and sequences of these
  - multi-type-casting

Implementing DAL

- Interfaces, no abstract classes
- Implementation from scratch
  - Full control
- By using common plug (glue) classes
  - More conveniente
  - Easier to ensure DAL compliance

Plug Design

- Provides common plug (glue) code
- device/property proxies
- Thin (minimalistic) as possible
  - Easier mentainance
  - Less performance overhead
  - Set of helper classes, must not be framework of it's own
- Features
  - Connection catching
  - Lifecycle management

DAL Implementations

All in prototype stage at the moment

- Simulation
  - Side efect of desing testing
- EPICS
  - Presented by Matthias
- GSI
  - Virtual accelerator number
  - PSPanel demo (JFrame, CosyBeans widgets)

Open Points

- Finish common plug implementation
- Define/extend common constants (characteristic names, alarm/error codes)
- Naming service and directory
  - Must provide meta-data (introspection)
- What does it mean “blue beam”?