

Application Support

T. Birke

- RDB structure @ BESSY
- Applications @ BESSY
 - Requirements
 - Current State
 - Problems
- Plan / Ideas

RDB structure @ BESSY

- device-classes configured by RDB:
power supplies/magnets, RF, WLS, vacuum system, timing/triggers, misc. I/O, interlock systems, temperature sensors, embedded controllers, few GPIB devices, stepper motors
- Set of tables per device-class
(ca. 4 – 15 tables)
- Each set has anchor-table *device-names* according to naming convention
- Sets are structured differently to meet special characteristics of device class
- No central structure, spanning all device-classes
- No common structure for table-sets

Applications @ BESSY

Save/Restore

Requirements

- which channels to save/restore at all
- hierarchy/grouping of channels
- save/restore conditions of a device

Current State

- script contains “hierarchy” of sophisticated SQL-queries and raw name-lists
- selection is controlled by devicename-patterns
→ [naming convention](#)
- hierarchy on target system is file system based
→ [directories/files](#)
- config-files are plain lists of PV names
(from SQL-queries)
- save/restore conditions maintained by hand

Problems

- e.g. for PS, RDB has no information about the PVs that model the device (e.g. [Q1PD4R](#) ↔ [Q1PD4R:set](#))
→ [assumptions in script](#)
- raw name-lists tend to be hard to maintain
- if complete channel-names were in RDB, it would still lack certain attributes ([roles](#))

Applications @ BESSY

Archiver

Requirements

- which channels to archive in which way
- Retrieval: hierarchy/grouping of channels
- dynamic archiving conditions

Current State

- some configs created by simple scripts
- some hand-written configs
- no dynamic archiving at all

Problems

- e.g. for PS, RDB has no information about the PVs that model the device (e.g. `Q1PD4R` ↔ `Q1PD4R:set`)
- raw name-lists are hard to maintain (typos, missing names, names not existing anymore...)
- if complete channel-names were in RDB, it would still lack certain attributes (*roles*)
- *static archiver configuration currently created by hand (with help of RDB-queries)*
- *archiving switched on/off manually*

Conspicuous: same requirements as Save/Restore !!!

Applications @ BESSY

Alarm Handler

Requirements

- which channels to monitor alarms of
- hierarchy / grouping of channels
- qualification of alarms / conditional alarms
- proper aliases for channels
- actions when alarms occur...

Current State

- some configs created by simple scripts
- some hand-written configs

Problems

- device names – channel names – PV names
- raw name-lists ...
- missing *roles* of devices

Again the same requirements!!!

Applications @ BESSY

Orbit Correction / Optics

Requirements

- magnets with function, length and position
- mapping magnets ↔ power supplies
- conversion factors $H = f(I)$
- device-interdependencies
e.g. kicker has no effect as long as trigger is off
- BPMs to use (position)...

Current State

- all infos in RDB and retrieved on startup
- model of machine optics created on startup
- meaning/functionality of devices is solved by naming convention
- library cares for DB-queries and handles mappings, conversions and cdev interface
- cdev and CA-gateway configuration created by hand (cdev with little help of RDB)

Problems

- device-interdependencies and status of current HW-installation (BPM) is solved, but *not* in RDB
- so is cdev and CA-gateway configuration

Plan / Ideas

- Set of tables/views to hold
 - all existing channel-names
 - *meanings* of channels
 - application-specific attributes
 - generic roles of devices/channels
 - dependencies between devices
- No application-specific tables (possible?)
- Syntax of config-files in RDB (possible?)

→ creation of configuration completely from out of the RDB