PAUL SCHERRER INSTITUT

PSI

WIR SCHAFFEN WISSEN – HEUTE FÜR MORGEN

Mark Könnecke

Control Systems Survey

The Context

- As part of a contract between ESS and PSI in the ESS design review phase I made a survey of control systems available in 2012
  - Published through NOBUGS; gone now
- Within the ECP project I surveyed new systems which have emerged until now (2016)
- Participants in the survey:
  - EPICS (3,4), TANGO, GDA, Sardana, NOMAD, IROHA, NICOS-2, SECI, IBEX, SNS, pshell, NSLS-2(bluesky)

PAUL SCHERRER INSTITUT

Survey Method

- From experience I constructed a questionnaire covering various aspects of the control system:
  - Design, parameter handling, configuration, processing logic, UI, and many more
- These questionnaires were filled in
- The questionnaires were discussed and validated with either the original authors of the software or experts in it
  - Thus I assume that they are not altogether wrong
- https://ess-ics.atlassian.net/wiki/display/DMSC/Reviews+of+Control+Systems

Result Categories

- EPICS, TANGO are a separate class
  - Distributed Hardware Abstraction Layer (D-HAL)
- Commonalities
- Differences
- Other Notes
- An Attempt at a Selection for ESS

Distributed Hardware Abstraction: EPICS, TANGO

- Separate ideosyncracies of hardware in little networked servers
- Multiple clients can access HW through a standard network protocol
  - But they need to know what is there
- Large: collaborations, installations
- No free lunch:
  - increase complexity
  - no quick fixes
  - new sources of
- BTW: what is sample environment doing?

Common Control System Abstractions

- Everyone has a notion of a device
  - Bunch of parameters
  - Often hierarchical
  - Readable,
  - Movable, Scannable, etc.
  - Motors are special
  - Virtual motors
  - Represent not only hardware but meta data etc. too
- Control Systems implement a container of some form for devices
- In servers: context objects for handling multiple clients

PAUL SCHERRER INSTITUT

Common Features

- Scanning
- Scripting and Batch Processing
- Various forms of waiting/running:
  - wait for something to finish
  - wait for a list of things to finish
  - start without waiting
- Access control, three levels:
  - RO
  - User
  - Specialist
- Data file writing (high entropy)

Common Technical Choices

- XML for configuration files
- Python for scripting and implementation
  - C-python or jython
- Eclipse-RCP based UIs
  - good, but 10 year old choice
- Client-Server architectures
  - Instrument server(s)
  - UI interacts with instrument server
- Linux as OS

PAUL SCHERRER INSTITUT

Common UI Elements

- CLI
- Log viewers
- 1D or 2D online data displays with interaction
- Hierarchical parameter displays
- Device lists
- Dashboards
- NEW: instrument schematic with possibility to drill down

PAUL SCHERRER INSTITUT

Common Network Patterns

- Direct bi-directional communication
  - write
  - read
- Publish-subscribe
- Some: RPC

Control System Experiences

- Control systems result of evolution
- Allow for easy modification of GUIs
- Hardware standardization is a good thing
- Independent of the technical choice, having a uniform system is important
- Make parts of the system replaceable
- Do not neglect the CLI
- Take care of data format and other standards right away
- Avoid blame games
- Collaboration can have its downsides

Different Bases

- Based on EPICS:
  - IBEX, SNS, NSLS-II, pshell, GDA
- Based on TANGO:
  - Sardana, NICOS-2
- IROHA: own component model using HTTP XML-RCP as transport
- NOMAD: goes for hardware directly
- No (few) systems are exclusive

Differences and Specialities

- Repositories of Ideas
- In alphabetical order:
  - GDA
  - IBEX
  - IROHA
  - NICOS-2
  - NOMAD
  - NSLS-II
  - Pshell
  - Sardana
  - SECI
  - SNS
- With silly comments from me

PAUL SCHERRER INSTITUT PSI

GDA

- Client-server system written in Java and an Eclipse-RCP GUI
- Uses CORBA and Java-RMI for communication between server and client
- Batton system to control access
- Client talks both to the server and EPICS
  - Tight dependencies between server and client
- Writes NeXus files
- jython scripting
- Works nicely for DIAMOND
- Separate build/instrument

PAUL SCHERRER INSTITUT

GDA Continued

- But nowhere else
  - Modularisation issues
  - Eclipse SW proved to be difficult to build automatically
  - Server not OSGI: SPRING and XML
  - What can we learn?
  - Keep your stuff separate
    - core components
    - library components shared between instruments
    - instrument specific components

IBEX

- Thin layer on top of EPICS
- Main components:
  - CSS based GUI
  - blockserver to manage configurations: python CA server
  - MySQL DB for holding PV-details
  - EPICS gateway to translate short user names to PV
  - Two archivers
    - Run (streaming) data
    - instrument parameters and logs

IBEX Continued

- procServControl from Diamond, a tool to start/stop IOC via CA
- Scripting via CA, planning for a script and scan server
- As of now: no central instrument script server
- Lots of archiving and logging is done via EPIS tools
- Windows

PAUL SCHERRER INSTITUT

IROHA

- Based on Robotics Technology Middleware
  - OMG standard component model
- TC Component
  - Typed I/O interfaces
  - Control/Service interfaces
- Data flow networks
- XML-RPC over HTTP
- Tools on top of the middleware
- Not managed by a central group

PAUL SCHERRER INSTITUT

NICOS-2

- Sits on top of TANGO
- Consists of a number of servers:
  - nicos-daemon
  - nicos-cache
  - nicos-poller
  - nicos-elog
  - nicos-watchdog
- Very feature complete
- Configured via a directory hierarchy of python scripts

NICOS-2 Continued

- Implemented in python
- UI in PyQT
  - I find the UI confusing
  - PyQWT no longer supported
- Uses home grown network protocols based on python serialization
- The interface to image or other multi-dimensional detector data leaves to be desired

NOMAD

- Single, multithreaded C++ instrument server talking to the hardware directly
  - Hardware:
    - bus layer
    - driver layer
    - abstraction layer
  - Control queue (internal DSL)
- (NO)MAD scripting interpreter

NOMAD GUI

- Custom NOMAD Java GUI communicating with the server through CORBA
- Tree view of instrument
  - Different level of detail revealed depending on user privilege
- Block language style graphical batch programming
- Data Display panes
- Very tightly integrated into NOMAD

NSLS-II Software System

- Multiple components on top of EPICS
  - Bluesky, scan logic
  - ophyd, interface to EPICS
  - metadatastore
  - filestore
  - databroker
  - suitcase
- Very, very tightly integrated with python
- As of now, no central instrument server but proof of concept

Bluesky

- At a fairly high level of abstraction
- A **RunEngine** executes **Plans** producing **documents**
- **Plans** are essentially sequences of messages for the run engine
  - Something to iterate over (Generators)
- **Documents**: dictionaries with a scheme
  - Consumers listen to these for online DR, storage etc., also via 0MQ
- An application of functional programming to DAQ
- Abstraction: flyer device

NSLS-II Data Handling

- All documents go into a MongoDB, the metadatastore
- All files are indexed into the filestore
  - your register a handler for the file type
  - write file, register with filestore
- Databroker allows to search and access in python via file type handlers
- Suitcase packages data for transport off site into HDF-5 files

PSHELL

- jython and Java on top of EPICS
  - Most functionality in jython
  - Java for devices, device pool
  - jython extended to provide much functionality of numpy
- Different run modes
  - workbench
  - server with REST API
  - CLI
  - Instrument specific UI

PAUL SCHERRER INSTITUT

PSHELL Continued

- Configurable ASCII and HDF-5 file saving
- Can stream data to other programs via 0MQ
- Plugin mechanism, manage and create in workbench
- Uses git for configuration and script management
  - git commit whenever a script is run
- Configuration editor integrated in workbench
- Uses threads for running
  - normally blocks
  - parallel execution

PAUL SCHERRER INSTITUT

Sardana

- Python on top of TANGO
- Sardana servers
  - device pool
  - macro server
- CLI client
- GUI client
- GUI toolkit
- NeXus file writing
- DataRecorder abstraction for data writing
- Very feature complete
- Used at ALBA, DESY, MAX-IV, Solaris, ESRF

PAUL SCHERRER INSTITUT

SECI

- Is being phased out
- Labview for HW access
- C++ for neutron DAQ
- C# application SECI to organize Labview VI and n-DAQ
- Scripting in openGenie and python via DCOM to SECI
- Labview
  - you can write good code in Labview
  - but it is easy to write bad code, they rewrote 50% of their drivers
- Run ECP in VM

SNS

- Many failed attempts
- CSS, scanserver on top of EPICS
  - Wizards to mitigate complexity of CSS GUI
- EPICS-4 replacing ADARA for event streaming
- Initial design fell over
  - 12-15 Windows PC per beamline
  - NI-data sockets irregularly failing
  - Commodity HW was not commodity

What we do not get

- No candidate implements configuration from a DB
- Integration with the ESS logging system has to be done for each of them
- Except PSHELL, no system implements a WWW-UI
  - Well, we start with a CLI
  - But the hooks must be there
- Integration with the ESS n-DAQ has to be done by us
- Integration with Online-DR (Mantid) has to be done by us

Attempt at a Pre-Selection

- GDA, NO, SW-management issues, jython
- IBEX, TBC
- IROHA, NO, difficult collaboration
- NICOS-2, TBC
- NOMAD, NO, but look at UI for ideas
- NSLS-II, TBC?
- Pshell, TBC
- Sardana, TBC
- SECI, NO, being phased out
- SNS, NO, similar in concept to IBEX, do not cross the pond

PAUL SCHERRER INSTITUT

Candidates in Detail

- IBEX
  - Missing the scripting server component
  - Missing features
- NICOS-2
  - Needs work at protocols and data interface
  - Dependencies
- NSLS-II
  - Young project
  - No server yet

Candidates in Detail Continued

- PSHELL
  - jython?
  - Young project
  - Standardization within PSI
- Sardana
  - No one likes CORBA anymore
  - Anecdotal evidence of SW organizational issues

Summary & Discussion

- Great variety of systems out there … and … working
  - All are the results of evolution
  - Given enough Skill and manpower you get anything to work
- Let the fun begin……
- Candidate solutions after preselection:
  - Sardana
  - NICOS-2
  - Bluesky, NSLS-II
  - IBEX, newDAS

Questions to be answered?????

- How tightly is Sardana integrated with Tango?
  - Dependency on TANGO/CORBA? Plans to change that
- How willing are the bluesy people to collaborate?
  - Do they have the resources to do that?
  - Maturity level?
- TODO: fill in a matrix of criteria/candidates

Criteria by Freddie

- EPICS Support
- Support for non EPICS devices
- # community provided drivers
- Driver development time
- Ease of GUI configuration
- Support for synoptic view
- Support for "instrument configurations“
- GUI technology "looks nice", or is easy to make so
- Already used at other neutron sources / shared user base
- Size of development community / current development work / opportunities for collaboration
- Learning time
- Integration with Streaming

Criteria by Freddie 2

- Community size
- Use at other n-facilities
- Dependencies/Longevity
- Use of technology already available at ESS
- Multi platform client
- Security model
- Scan support
- Scripting support
- Remote WWW-interface
- Simulation support
- Logging/Error reporting integration
- Ease of analysis -DAQ integration