WIR SCHAFFEN WISSEN – HEUTE FÜR MORGEN

Mark Könnecke, Owen Arnold

Control Systems Survey for ESS

The Context

- European Spallation Source (ESS) being built in Lund, Sweden
- Switzerland contributes with money and work, In Kind
  - Survey of existing experiment control systems
- Aims
  - Choose an experiment control program (ECP) for ESS
  - Gather knowledge about control systems
- Participants in the survey:
  - EPICS (3,4), TANGO, GDA, Sardana, NOMAD, IROHA, NICOS-2, SECI, IBEX, SNS, pshell, NSLS-2(bluesky)

Survey Method

- A questionnaire was constructed covering an exhaustive list of aspects of control systems
- The questionnaires were filled in, discussed and validated with either the original authors of the software or experts in it
- The questionnaires can be obtained on request

Results

- EPICS, TANGO discussed as a separate class
- Common Patterns
- What was learned?
- Details on selected systems
- The Selection of an ECP for ESS

EPICS, TANGO

- Distributed Hardware Abstraction Layer (DHAL)
  - Little servers implement hardware access and functionality
  - Multiple clients can access servers through a standard network protocol and standardized interfaces
- Large: collaborations, installations, many support tools
- No free lunch:
  - increase complexity
  - new sources of bugs
- BUT: Everyone is using one of them, Exceptions: (NOMAD, IROHA)
- MIA: Collect bits and pieces and present as an instrument to the user

Subtle Mismatch EPICS/TANGO Tools for Instruments

- EPICS and TANGO are targeted towards accelerators
- Accelerators
  - **Very** static configurations
  - Trained operators
  - Standard operating procedures
- Instruments
  - Dynamic and changing configurations
  - Untrained users
  - Complex operations
- EPICS/TANGO work 90% for instruments too, the difficulties come in the last mile
- Example CSS or MEDM:
  - For accelerator displays: beautiful
  - For instruments: change code for every change at the instrument?

EPICS/Tango Future

- EPICS 3
  - Best at home on register hardware (VME, …)
  - Core developers greying
  - Not really good at transporting arrays
  - Steep learning curve: 18 months
- EPICS 4
  - Proper support for arrays and structures
  - MIA: device support
- TANGO: critical dependency: CORBA
  - More approachable: 3 days advertised
- Is there a market for a new system based on modern messaging concepts?

Common Patterns: Device, Parameter Abstractions

- Most systems have a notion of a device
  - Bunch of parameters (also an abstraction)
  - Often in hierarchical arrangements
  - Device classes:
    - Readable
    - Movable, Scannable,
    - Motors are treated special
  - Represent not only hardware but meta data etc. too

Device, Parameters: Why?

- There is a cost: Level of indirection
- Benefits:
  - Abstracts from hardware
  - Helps implement persistence
  - Helps implement change notifications
  - Helps implementing history
  - Caching
  - Fine grained access control
  - Simulation mode

Common Pattern: DataSet, DataSink

- DataSet
  - Collection of meta data and detector data for a measurement or scan point
- DataSink
  - takes a DataSet and does something with it
- Common somethings:
  - Data file writing
  - Live display
  - Online data reduction
  - Whatever you want to do with the data…

General ECP Pattern

- Containers for devices and experiment routines
- Run experiment routines (scan etc)
  - against: Devices, backed by EPICS or TANGO
  - creating DataSets
  - forwarded to DataSink

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)
- Virtual or logical motors
- Managing configuration

Common Technical Choices

- XML for configuration files
- Python for scripting and implementation
- Eclipse-RCP based UIs
- Client-Server architectures
  - Instrument server(s)
  - UI interacts with instrument server
- Linux as OS

Common UI Elements

- Command Line Interfaces
- Log viewers
- 1D or 2D online data displays with interaction
- Hierarchical parameter displays
- Device lists
- Dashboards
- NEW:
  - instrument schematics with possibility to drill down
  - 3D instrument views
- Clutter is a problem in all instrument UI’s
  - Let us ask for visibility controls

Common Network Patterns

- Direct bi-directional communication, command-response
  - write parameters
  - read parameters
- RPC-mechanisms, like CORBA are an extension of command-response
- Publish-subscribe

General Control System Lessons

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

Selected Lessons: Labview

- SECI:
  - LabView has all the features to write proper software
  - But makes it very easy to write bad software
  - ISIS had to reimplement 50% of all Labview drivers
- SNS
  - NI-Datasockets irregularly failing
  - Commodity PC were not so commodity after all: cards had to match PC
- Syntax addiction

Syntax Addiction

- In Europe, when you do a TAS, you are supposed to implement MAD syntax
- NOMAD had to implement MAD syntax
- Nearly all newer synchrotron systems had to implement SPEC syntax
- ISIS had to try to be openGenie compatible
- ==> Scientists are syntax addicted!!

Selection of Ideas Implemented

- GDA: baton system for controlling access
- NOMAD: Block programming for batch file generation
- PSHELL: git for managing configuration files and scripts, a git commit per script run
- NSLS-2
  - Use of functional programming constructs in bluesky
  - Data handling
  - The Flyer abstraction
  - More details: Maksim Raitkin’s presentation

Choosing an ECP for ESS

- The accelerator people had already settled for EPICS; we had to follow
- C-Python was to be the preferred scripting language, because of numpy and better package support.
  - This deselected all the Java based systems having jython as scripting language
- This left four candidates: NICOS, Sardana, NSLS-2, IBEX

Decision Matrix

Criterion		IBEX			NICOS			Bluesky			Sardana		
	Weight	Answer	Points	Weighted	Answer	Points	Weighted	Answer	Point	Weighte	Answer	Point	Weighte
Does support for EPICS devices	2	yes	1	2	not fully ye	0.5	1	yes	1	2	partly	0.5	1
Uses Python - Mantid integration	1	scripting test	0.5	0.5	yes	python	1	python	1	1	python	1	1
Uses Qt - Mantid integration	0.8	no	0	0	no	0	0	no	0	0	no	0	0
Uses scientific plotting lib	1	not really	0	0	matplotlib	1	1	matplotlib	1	1	matplotlib	1	1
Easy to configure GUI / creating multiple experiment views	0.8	yes	1	0.8	partial	0.5	0.4	no	0	0	taurus	1	0.8
Easy to create a Synoptic view Support for "instrument configura	0.5	yes	1	0.5	partial	0.2	0.1	no	0	0	no	0	0
Already used at other sources (prior to adoption)	2	no	0	0	no	0	0	no	0	0	yes	1	2
Size of development community / current development work	2	ISIS only	0	0	FRM2 only	0	0	NSLS-2	0	0	many	1	2
Learning time as developer	0.8	steep	0.5	0.4	moderate	0.5	0.4	moderate	0.5	0.4	some	1	0.8
Sum over dependencies * number of active authors over last 6 m	2	new	1	2	moderate	0.5	1	new	1	2	CORBA	0	0
Project is NOT vunerable to forked dependencies	2	old CSS	0.5	1	no	1	2	no	1	2	no	1	2
Uses technologies or knowledge already available at DMSC	2	yes	1	2	yes	1	2	yes	1	2	half/corba	0.5	1
Integration / synergy with other ESS ICS technologies/products/s	0.5	yes	1	0.5	not fully	0.5	0.25	yes	1	0.5	no	0	0
Multi platform client	2	yes	1	2	some hassle	0.5	1	no	0	0	some hassle	0.5	1
Mainly runs on Linux	1	no	0.5	0.5	yes	1	1	yes	1	1	yes	1	**1**
Security / authentication / authorisation model	1	no	0	0	yes	1	1	no	0	0	yes	1	1
Support for scanning CLI. Scan everything against everything.	2	in deve	0	0	yes	1	2	yes	1	2	yes	1	2
Web Interface	1	dashboard	0.2	0.2	mini	0.2	0.2	no	0	0	no	0	0
Programmatic Interface	2	at EPICSS	0.5	1	pythonic	1	2	for data	0.5	1	tango	1	2
Dry Run Mode	1	genie python	0.5	0.5	built in	1		no	0	0	no	0	0
Provides a Logging service	1	MySQL	1	1	yes	1	1	python log	1	1	yes	1	1
Provides Error handling	1	distributed	0.3	0.3	yes	1	1	yes	1	1	yes	1	1
Ease of Integration with data streaming project	1	with difficulty	0.3	0.3	add device	1	1	add devic	1	1	new dev typ	0.5	0.5
Quick fixes in production by team	1	partly	0.5	0.5	yes	1	1	yes	1	1	more difficu	0.5	0.5
Codacy project grade. For points A = 1, F = 0	2									0			
Total	**33.4**		**12.3**	**16**		**15.4**	**20.35**		**15**	**18.9**		**15.5**	**21.6**

ESS Decision

- Candidates are close together
- IBEX: lowest score, no central instrument server
- NSLS-2: no server functionality
- Sardana: critical dependency CORBA
- The winner is: NICOS

Conclusions

- There are patterns:
  - Use of a DHAL
  - Experiment routines act upon devices creating datasets being forwarded to DataSinks
- On comparison, successful systems are very close together in features and capabilities

Selection Criteria

- 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

Selection Criteria 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