

Treepath Based Instrument Control

Mark Könnecke¹ Nick Hauser² Ferdi Franceschini² Tony
Lam² Markus Zolliker¹

¹Paul Scherrer Institute
Switzerland

²ANSTO
Australia

November, 2, 2008



- The quest for a GUI for SICS
- SICS: client server DAQ system
 - SICS server: ANSI-C and Tcl which does all the work
 - Clients: implement UI
- Instrument scientists: parameter editor
- Communication problem
- Our solution has the potential to be generic



- Map the aspects of instrument control:
 - Parameter editing
 - Command execution
 - Online data display
 - Batch processing
- into a tree structure and operations on this tree.
- Protocol to communicate with the tree.

- Tree of nodes
- Nodes have types: text, int, float, int and float arrays.
- Nodes have properties: key-value pairs
- Node properties give hints to the UI
- Property examples:
 - values
 - priv
 - type
 - viewer
- Notion for properties: (key=value)

- Commands are mapped to tree nodes with child nodes representing the parameters.
- Example:
- count (type=command)
 - countmode
 - preset

- Example: a powder diagram
- powder-diagram(type=graphdata)
 - rank
 - dim
 - two_theta (type=axis) (dim=0)
 - counts (type=data)
- Scheme extends to n-dimensional data

- Stored sequences of tree operations
- Execute in client
- Translate and execute on server

Tree Access Through the Network

- Nodes identified by unix path strings.
- Example: `/instrument/monochromator/wavelength`
- `getgumtreexml path`, returns tree structure in XML SDO format
- `hset path value value ..`
- `hget path`
- `hnotify path`
- `hlist path`
- `hlistprop path`
- `hgetprop path key`



- Errors prefixed with ERROR:
- Wrapped into asynchronous completion token scheme (ACT)
- contextdo 133 hset path value
 - 133 ::>response<::
- Other implementation: use json formatted messages
- All I/O asynchronous
- Messaging protocol

- Client operates on the tree
- Underlying control system (Treepath server)
 - Provides the tree and information on it
 - Implements Treepath protocol
 - Maps tree operations on device operations and measurement procedures
 - Optionally: updates parameter values in the tree
 - Optionally: updates data for online graphics

- SICS has been extended to become a treepath server
- Some parts of this, i.e the tree node code, is reusable
- This is a working system!
- Configuration commands in SICS to configure tree
- 2 days to configure for an instrument.

- Tree commands is already CLI!
- Generic wrapper: widget command syntax
 - object par value value ..
 - object par
 - object method par par ...
 - function par par ...
- Code generation from SDO or generic wrappers can map to an object hierarchy in the OO language of your choice. Been done for python!
- With a dictionary: code generation to functional or procedural languages
- Maps also to other concepts like ReST WWW-services



- The concept prefers tree interfaces
- But does not limit!
- Just map UI gestures to tree operations.
- GUI can introspect the tree and adapt itself
- Generic GUI's possible

- Eclipse-RCP based
- Shares many ideas with Gumtree ANSTO
- Instrument control only application
- Reproduce traditional SICS command set as closely as possible
- Supports 9 SINQ instrument now
- Had to be as generic as possible
- Still in development



GTSE Tree Display

The screenshot shows the MOUNTAINGUM SICS interface. At the top, there is a menu bar with 'File', 'Mountaingum', and 'SICS'. Below the menu bar are icons for 'New Window', 'Tree', 'Graph', 'Terminal', 'Batch', and a large red 'STOP' button. The main window is divided into three sections:

- Tree View:** A hierarchical tree structure with 'name' and 'value' columns. The 'experiment' node is expanded, showing sub-nodes: 'title' (UNK), 'user' (UNK), 'datafilenumber' (997), 'batchpath' (/hor), 'sinq', 'velocity_selector', 'collimator', 'attenuator', 'sample', 'beam_stop', 'detector', and 'commands'. Each node has a lock icon.
- Table View:** A table titled 'experiment' with columns 'Name', 'Current', and 'Target'.

Name	Current	Target
title	UNKNOWN	Read-Only
user	UNKNOWN	Read-Only
datafilenumber	9979	Read-Only
batchpath	/home/sansinsg/rother/ok	Read-Only
- Log View / Console:** A text area showing log output: 'nampos = ps list of known named positions', 'nampos = dt list of known named positions', and 'nampos = dt list of known named positions'. Below this are buttons for 'Log View', 'Console', 'Make It So', 'Enqueue', and 'Interrupt'.

At the bottom of the window, a status bar shows 'Spy @ sans.psi.ch' and 'Driving'.

File Mountaingum SICS

New Window

Graphics **detector** radial_average

detector
temperature
radial_average
quickview

detector

Colour Mapping
Rainbow

Data Mapping
 Normal
 log

Z Mapping
 Automatic
 Manual

x,y,l
-89,-43,16

COG x,y, Sum(l)
-3, -1, 761201

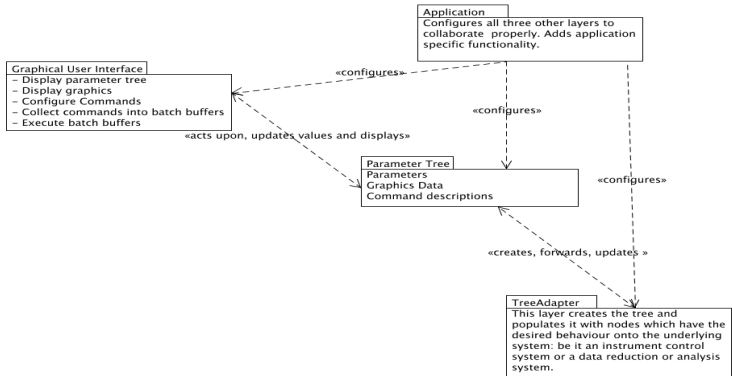
Reset Zoom

Refresh

Mapping Range
Lower Limit Upper Limit

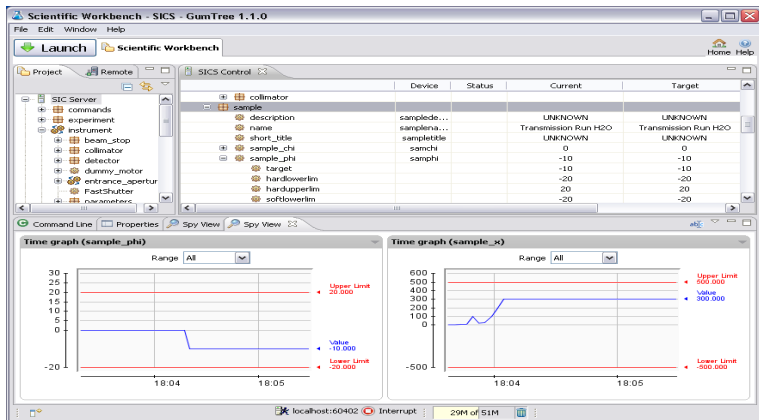
Spy @ sans.psi.ch Driving

Gumtree SE Package Structure



- Sophisticated integrated scientific workbench to cover DAQ, data reduction and data analysis
- Uses the tree in a more OO-way to generate different UI's
 - Python API
 - Dashboard interface
 - Graphical representations of devices

Gumtree ANSTO Instrument Perspective



Gumtree ANSTO Dashboard Display

Scientific Workbench - Dashboard/ics.gdb - GumTree 1.1.0

Launch Scientific Workbench SCS Instruments Status Monitor

SCS Control Automator Editor Dashboard 02

Velocity Selector

	Status	Current	Target
LandRad	0	0	0
LandRadRefWH	0	0	0
V5dg	0	0	0
V5gen	0	0	0

Sample Aperture

	Status	Current	Target
EndFaceFormIn	20095	20095	0
SAPPosIn	0	0	0
SAPPosYIn	0	0	0
SAPPosZIn	0	0	0
SAPShape	UNKNOWN	UNKNOWN	0
SAPIn	0	0	0
SAPOut	0	0	0

Beam Stop

	Status	Current	Target
BSShape	UNKNOWN	UNKNOWN	0
BSDIn	0	0	0
BSDOut	0	0	0
DefPostOffIn	0	0	0

E Aperture

	Status	Current	Target
EAPPosIn	0	0	0
EAPIn	0	0	0

Sample

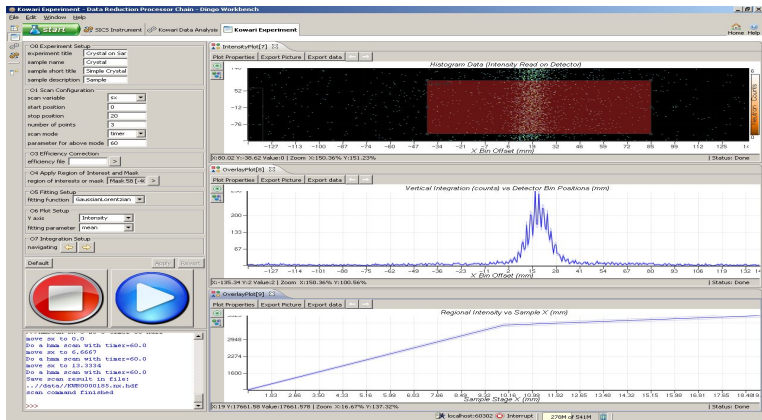
	Status	Current	Target
SamplePosIn	0	0	0
SamplePosOut	0	0	0
SamplePosZIn	0	0	0
changer_position			

Beam Position

	Status	Current	Target
BeamCenterC	0	0	0
BeamCenterY	0	0	0

localhost:60003 Internet 29% of 51M

Gumtree ANSTO Instrument Integrated DA



- Treepath maps parameters, commands and online data into a tree and operations on the trees
- Only abstraction covering all aspects of DAQ
- Scalable to a huge number of parameters
- Instrument becomes discoverable
- Tree layout can vary to a large degree
- Adaptable, generic and generative UI's possible
- CLI and scripting access well supported
- Generic if an agreement on details can be reached
- Possibly useful for data analysis, too