

NICOS – The Instrument Control solution at the MLZ

G. Brandl, **E. Faulhaber**, C. Felder, J. Krüger, A. Lenz, B. Pedersen

Heinz Maier-Leibnitz Zentrum (MLZ), Technische Universität München, Garching, Germany

Overview

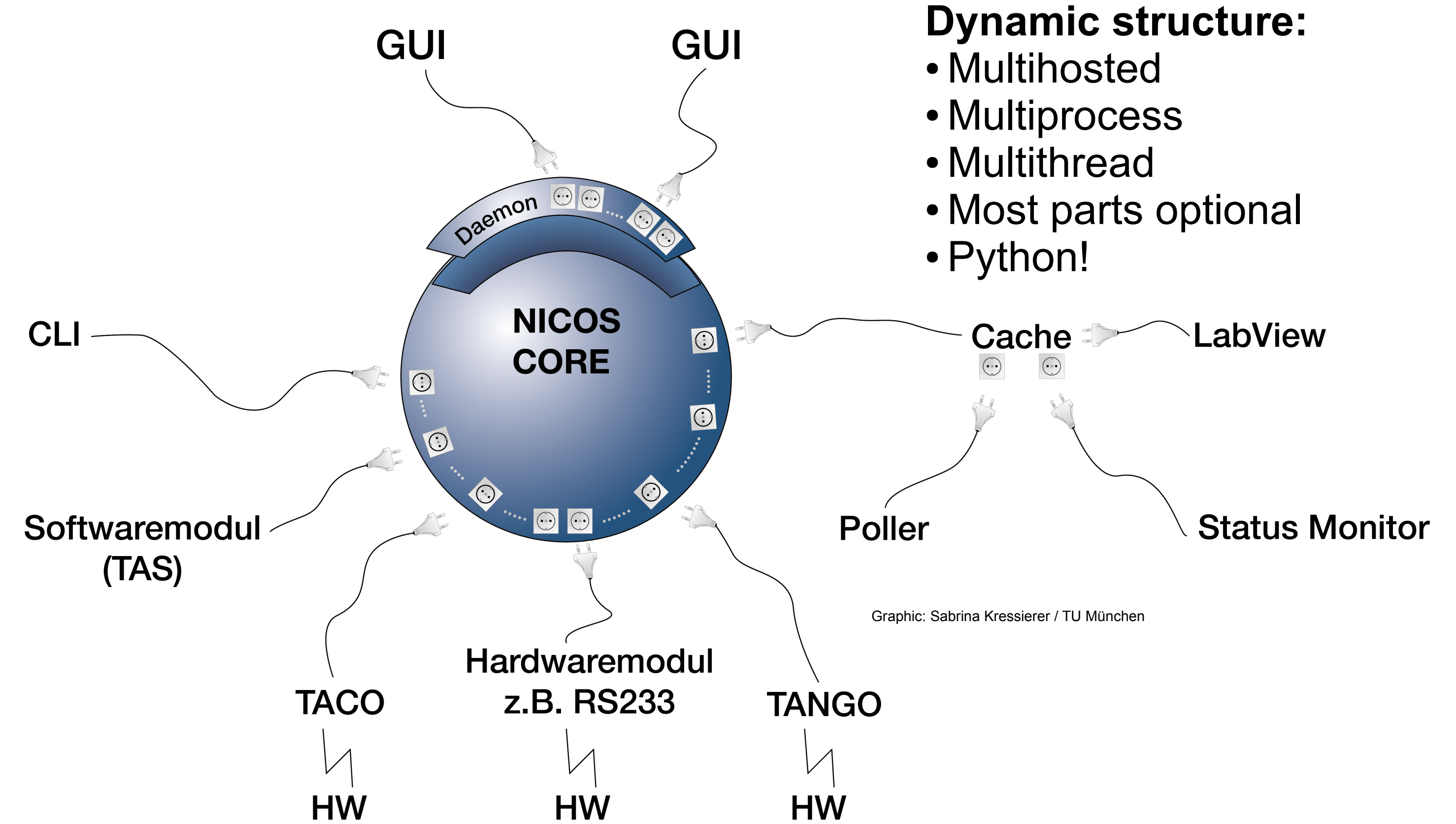
Challenges:

- Instrument control suite should support the experiments & experimenters
- High robustness and reliability
- Adaptation to individual instruments
- High flexibility & modular use of components
- Integrated, easy to use scripting language

Solution:

- Development driven by instruments needs
- co-developed with instruments
- Wide range of applications: tomography, powder diffraction, TAS, TOF,
- Support for single detectors, imaging detectors & cameras (via LIMA)
- Plug'n Play support for sample environment & instrument components
- Enhanced scripting capability, help system & configurable GUI
- Several Control Protocols already included: TACO / TANGO / MODBUS / RS232...
- Abstract device interfaces hide HW peculiarities
- Code review, git as SCM, all instruments have same code base

Structure



Dynamic structure:

- Multihosted
- Multiprocess
- Multithread
- Most parts optional
- Python!

Customers / Users

Instrument Scientist:

- Configuration files useable across instruments
- Lego-like construction of setups for more complex configurations
- Configuration change 'on-the-fly'
- Flexible configuration system to reflect hardware setup of instrument
- Console or **Graphical User Interface**
- History view of all logged data
- Instrument specific commands
- Client-Server architecture

Experimentalist / User:

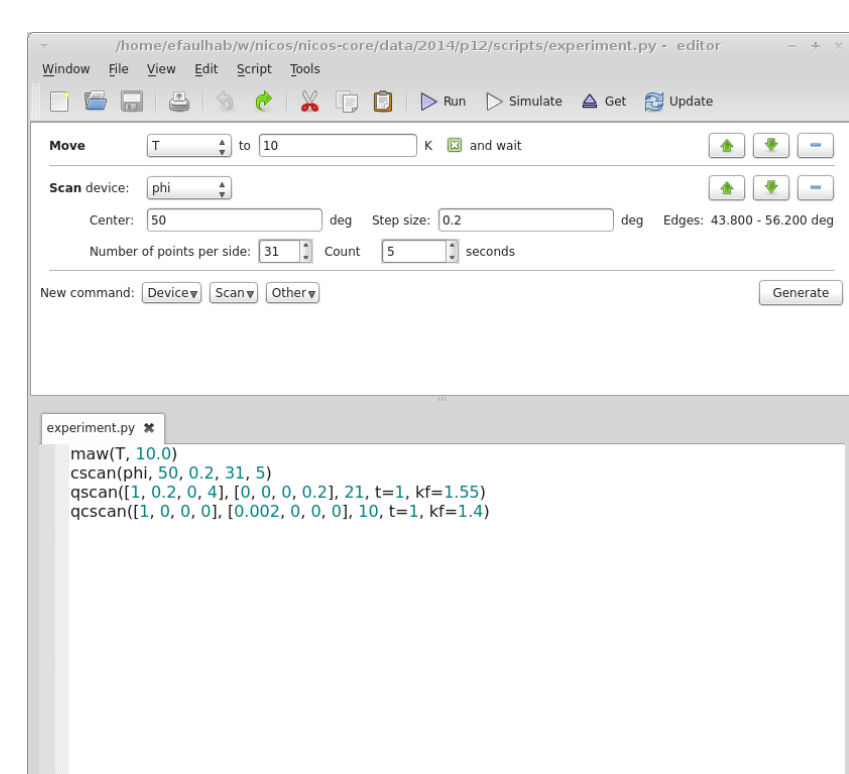
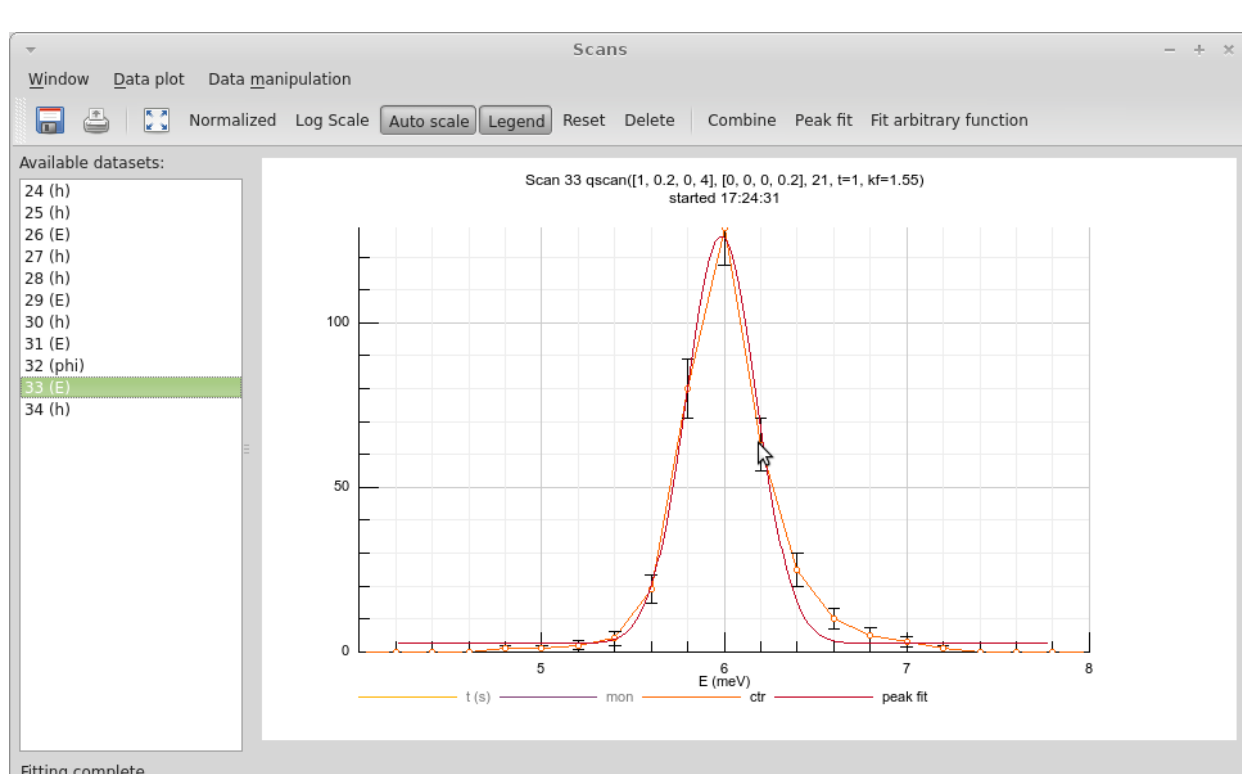
- 'Self explaining' commands & argument names
- Few basic commands needed: (move, read, count, scan, run, ...)
- Automatic logging of instrument parameters during the experiment
- Electronic logbook of the activities
- Checking scripts in a 'Dry Run' (incl. limit checks and time estimation)
- Configurable single place for experiment specific data/log files (to easily take home your data)
- Interactive online help system

Details

GUI:

- Configurable User Interface
- Connects to daemon
- Overview of scans
- Convenience functions: plot, fit, print, ...

- Graphical script editor
- Limited data treatment capabilities
- Console client available



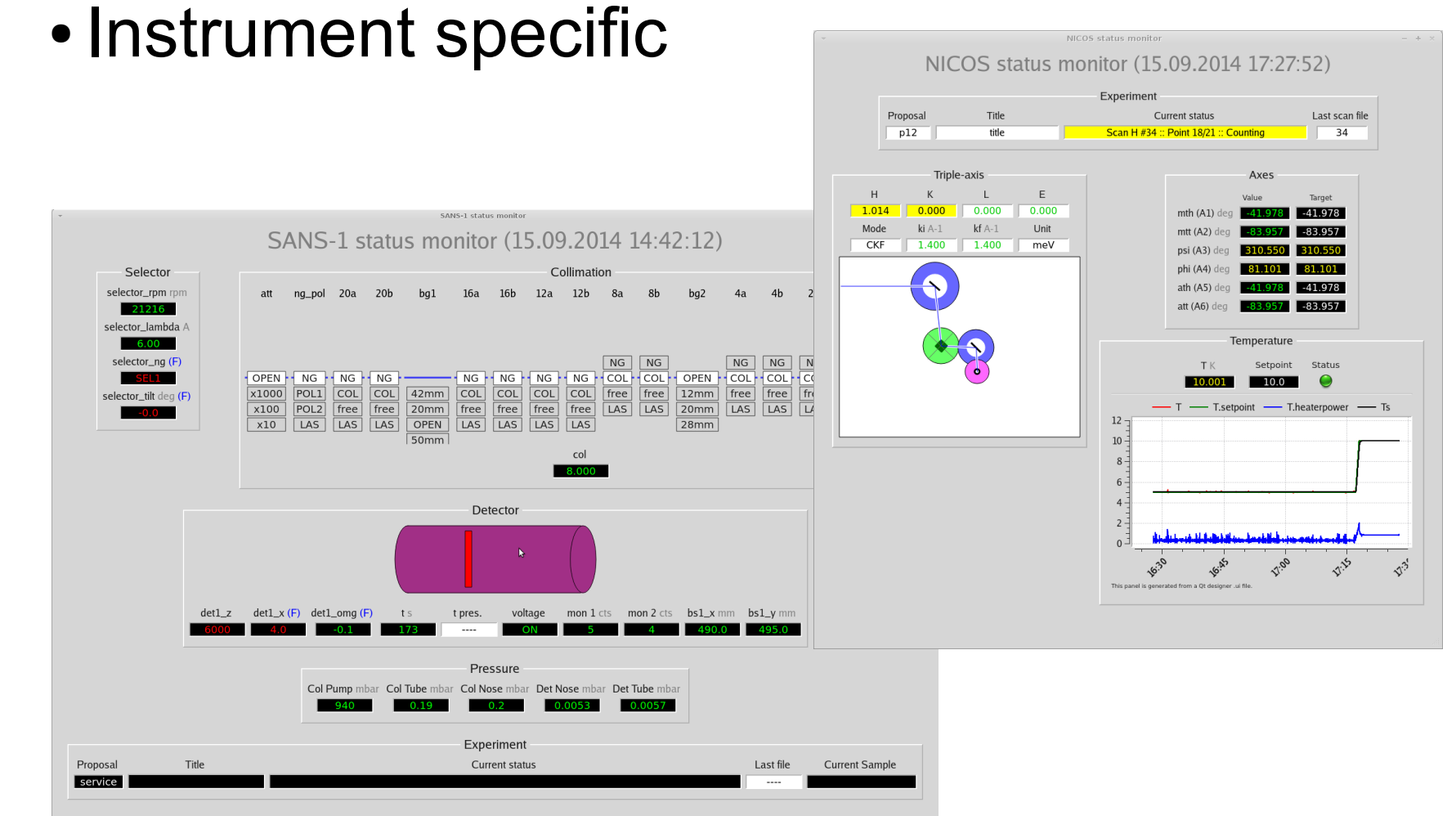
Cache:

- Central data dump
- MQTT like protocol: + timestamps + time-to-live
- Single server / Multi client
- Redistribution of updated device values
- Booking of updates
- History queries
- Data stored in daily changes files on disk
- Stand-alone tools for data analysis



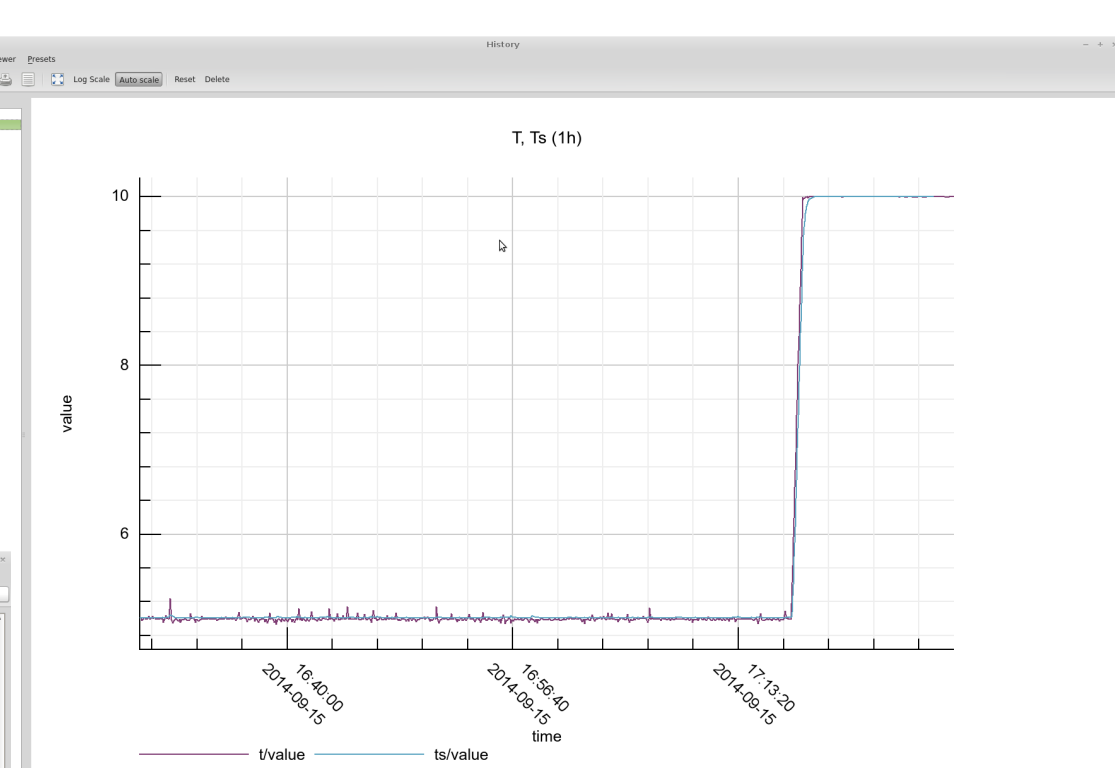
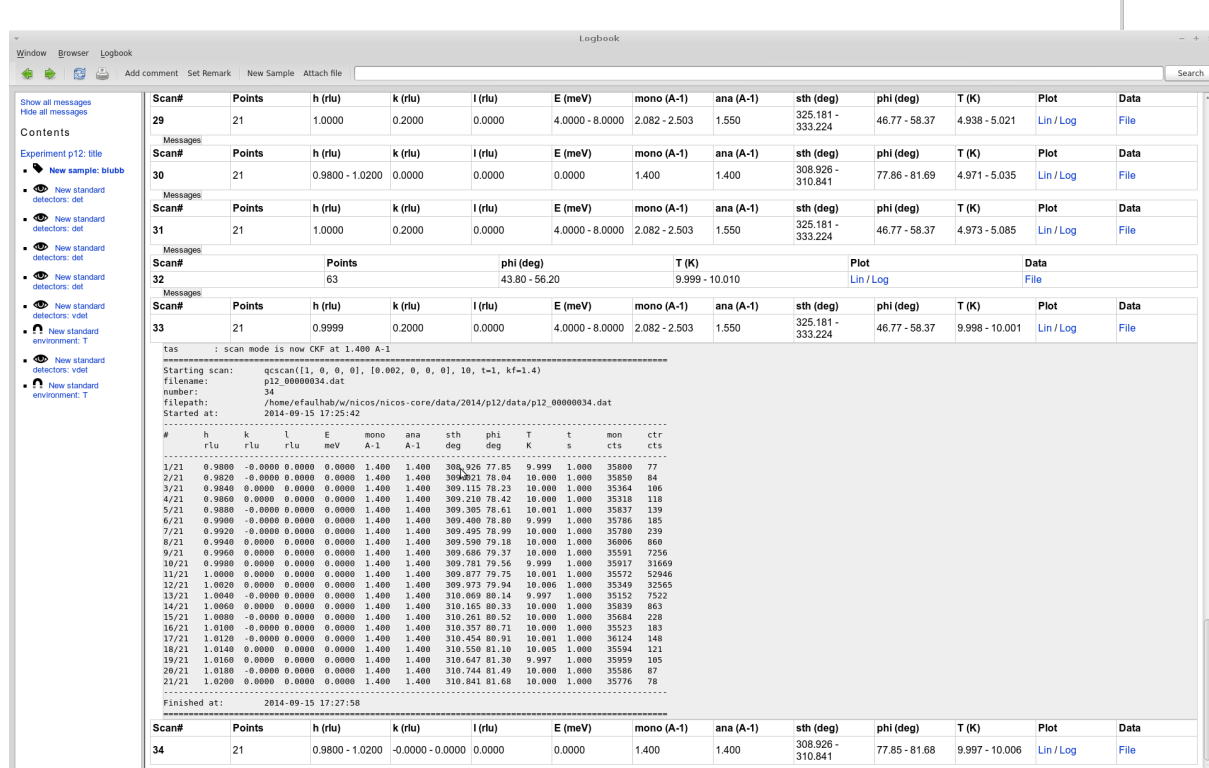
StatusMonitor:

- Configurable status display
- Updates received from cache
- Reacts on configuration change in daemon
- Supports widgets, ui-files,....
- Instrument specific



Support tools:

- Electronic logbook
- History plots
- All-in-one console client



Daemon:

- Central execution process
- Connects to HW via device servers
- Dynamic configuration loading
- Stores measurement data
- Configurable data storage
- Script execution
- Simulation

