JÜLICH
FORSCHUNGSZENTRUM

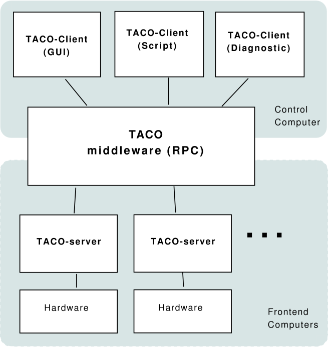
New Developments for the JCNS Neutron Scattering Instruments

M.Drochner, A.Erven, L.Fuss, J.Heinen, H.Kleines, D.Korolkov, S.Mattauch, M.Monkenbusch, M.Wagener
FZ Jülich

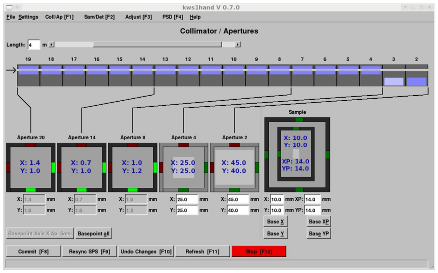
Since the Juelich 'Dido' research reactor was shut down 5 years ago, the larger part of its neutron scattering instruments was moved to the new FRM2 neutron source at Munich, and some newly constructed.
While we used a set of standard components as far as feasible, we had to accept some compromises due to the time pressure to get things going. In particular, we have a number of different graphical and command-line user interfaces now which is difficult to handle for users who work on multiple instruments.
After the initial rush, we are now working towards some consolidation, both on a common GUI philosophy and a high-level scripting language.
The design of the new user interface will be presented.

Existing software:

- Control system based on TACO middleware (ESRF)
- Using simple request / response scheme and database
- GUI clients using Qt3 (Trolltech / Nokia)
- Python clients using bundled TACO bindings
- Python clients using abstraction libraries
- Historical and special solutions (psh, Ruby)



General software structure



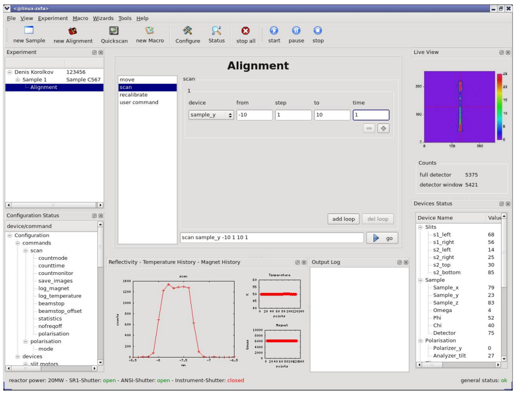
Length: 4 m

Aperture 20: X: 1.4 Y: 1.0 — X: 1.4 mm Y: 1.0 mm
Aperture 14: X: 0.7 Y: 1.0 — X: 0.7 mm Y: 1.0 mm
Aperture 8: X: 1.0 Y: 1.2 — X: 1.0 mm Y: 1.2 mm
Aperture 4: X: 25.0 Y: 25.0 — X: 25.0 mm Y: 25.0 mm
Aperture 2: X: 45.0 Y: 40.0 — X: 45.0 mm Y: 40.0 mm
Sample: X: 10.0 Y: 10.0 XP: 14.0 YP: 14.0 — X: 10.0 mm XP: 14.0 mm Y: 10.0 mm YP: 14.0 mm

KWS GUI screenshot (manual control program)

Ongoing work and future plans:

- Replacement of historic "psh" as low level scripting language for maintainability (standardize on Python)
- New common high level scripting / macro language
- No immediate plans to replace TACO, but better caching might be needed if many clients access status variables
- Integration with other control systems (eg. EPICS)
- Common look & feel of GUIs (ongoing discussion)



Denis Korolkov 123456
Sample 1 Sample C567
Alignment

scan
1

device	from	step	to	time
sample_y	-10	1	10	1

scan sample_y -10 1 10 1

Counts
full detector 5375
detector window 5421

device/command
Configuration
commands
scan
countmode
counttime
countmonitor
save_images
log_magnet
log_temperature
beamstop
beamstop_offset
statistics
nofreqoff
polarisation
polarisation
mode
devices
slit motors

Device Name	Value
Slits	
s1_left	68
s1_right	56
s2_left	14
s2_right	25
s2_top	30
s2_bottom	85
Sample	
Sample_x	79
Sample_y	23
Sample_z	83
Omega	4
Phi	52
Chi	40
Detector	75
Polarisation	
Polarizer_y	0
Analyzer_tilt	27

reactor power: 20MW - SR1-Shutter: open - ANSI-Shutter: open - Instrument-Shutter: closed
general status: ok

Proposed new GUI layout

New high level scripting language:

- Classical, simple command + arguments, no variables and control structures needed (known from other facilities)
- Abstract notion of devices (can be axes, temperatures, ...)
- Only few primary commands, help function
- Only one or two-dimensional scans
- Integration with Python to allow extension by experts
- "Access levels" to restrict potentially harmful commands to experts
- Not immediately required but "nice to have":
  - Context sensitive command line completion
  - Collaborative work with external expert
  - Logging
  - Arithmetics / calculator function

Scripting language examples:

```
move omega 7.5; move detector 5; wait time 10
```

(move axes successively)

```
move omega 7.5 detector 5; wait time 10
```

(move axes simultaneously)

```
scan <device> <from> <steps> <to> <time>
```

Mitglied der Helmholtz-Gemeinschaft