Karlsruher Institut für Technologie

Concert control system

M. Vogelgesang, T. Farago, T. Rolo et al.

What are we talking about?

- We designed and implemented a new beamline control system based on Tango and native device access
- First usage in a real tomography experiment last week
  - Camera control via libuca
  - Shutters and motors via PyTango
  - Aerotech controller directly
  - Setup processing with the UFO framework

Why not Sardana?

- Pros of Sardana
  - Already in use at ALBA and presumably at some beamlines at DESY
  - Automatic GUI generator saves some work
- Cons
  - Hard to get running, thus a huge blocker for deployment
  - Open development stopped some time ago
  - Tries to replicate SPEC with all its shortcomings

Requirements

- Asynchronous device access and processes
- Prioritize usability for user over beamline scientist over developers
- Provide a session mechanism to separate different experiments
- Integrates all technologies that we developed so far

Technology

- Python 2.6 and up
- Use of high quality packages such as Quantities, Futures, Logbook, ...
- Quality assurance with
  - Jenkins
  - 48 unit tests
  - Pylint
- Documentation right from the beginning
- pip and virtualenv friendly

Examples: device access

Spin up the rotary stage to 5° per second

```
aero = Aerorot()
aero.velocity = 5 * q.deg / q.s     # this blocks
aero.stop()                         # this not
```

Calibration processes

Scan and analyze camera response

```
camera = UcaCamera('pco')
scanner = Scanner(camera['exposure-time'], feedback)
scanner.minimum = 5 * q.ms
scanner.maximum = 1 * q.s
x, y = scanner.run().result()
```

Autofocusing

```
elya = TangoMotor(get_device('iss/tomotable/m_elyafoc'))

def feedback():
    camera.trigger()
    return np.sum(np.gradient(camera.grab()))

focuser = Focuser(elya, 0.01 * q.mm, feedback)
focuser.focus(0.5 * q.mm)
```

Show information in a shell

```
>>> ring = StorageRing()
>>> ring

 Parameter  Value
 current    83.7565078735 mA
 energy     2.50469827652 MeV
 lifetime   36.5215942048 h
```

Session handling

```
usage: concert [-h] [--version]  ...

optional arguments:
  -h, --help  show this help message and exit
  --version   show program's version number and exit

Concert commands:

    log       Show session logs
    show      Show available sessions or details ...
    edit      Edit a session
    start     Start a session
    init      Create a new session
    rm        Remove one or more sessions
```