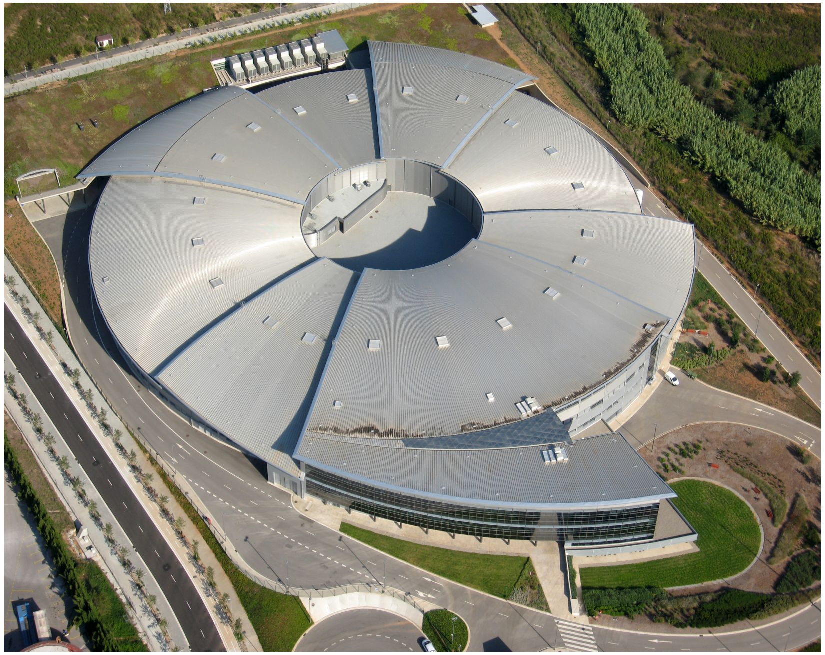
Development of the Continuous Scans in Sardana

Zbigniew Reszela

on behalf of the Alba Controls Group

Tango Meeting 2015

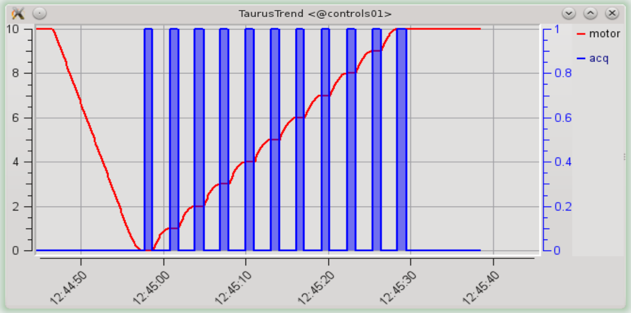


Alba Synchrotron Light Facility

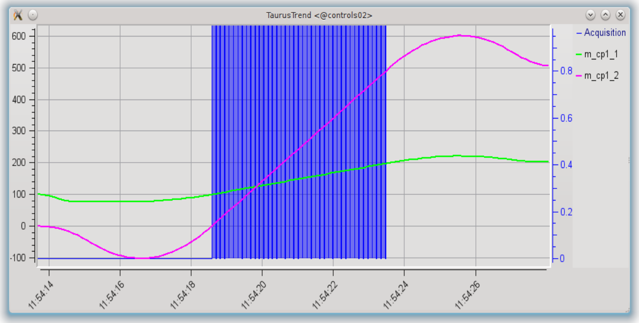
Sardana logo

Continuous Scans

- Scan – process of measurement signal(s) from exp. channel(s) while varying the actuator(s) set point
- Simultaneous (continuous scan) vs. sequential (step scan) movement and acquisition:
  Reduction of scan time – higher throughput of the system, reduces variation of the conditions, etc.



Motion & acquisition during the step scan.



Motion & acquisition during the continuous scan.

Continuous Scans - challenges

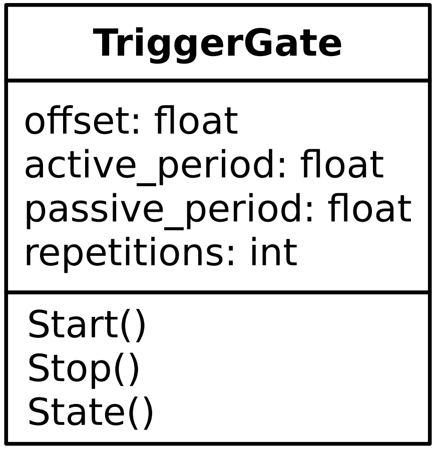
- **Scan at the constant velocity of the actuator**:
  - accel. and decel. space are out of the scan range
  - non-linear pseudomotors require trajectory control
- **Synchronized on time/position scan points**:
  - complex configuration of generators and receivers of the trigger/gate signals: trigger/gate, non-equidistant synchronization
  - support of hardware and software synchronization
- **Data collection, buffering and merging**:
  - detectors with or without internal memory
  - optional interpolation of data in case of missed triggers
- **Flexibility in configuration**: arbitrary number of actuators and detectors, number of intervals and integration time
- **As close as possible to step scan** (in terms of configuration and execution aspects)

Sardana Scans – history

- *Step scan* – just single acquisition but with software synchronized controllers
- *Software continuous scan (ascanc)* – “best effort” approach – consecutive execution of single acquisitions while actuators are at the constant velocity
- *Hardware continuous scan (ascanct)* – unique solution applied at 3 Alba's beamlines, but tied to the Ni660X trigger!
- *Generic continuous scan* – in progress
  - synchronization thanks to TriggerGate (TG) elements
  - MeasurementGroup associates and configures TG elements
  - software synchronized ExpChannels

Synchronization – TriggerGate

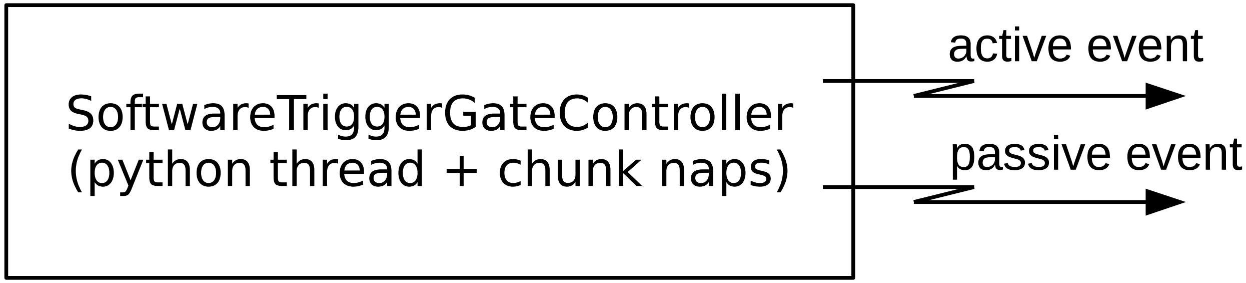
- Basic API – just equidistant configuration
- New controller type allows to plugin custom hardware
- Was only verified in the time domain:
  - Ni660X “PulseTrainGenerator”
  - Software Trigger/Gate



TriggerGate

offset: float
active_period: float
passive_period: float
repetitions: int

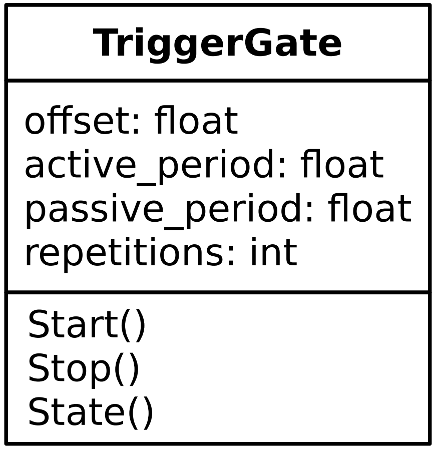
Start()
Stop()
State()



SoftwareTriggerGateController
(python thread + chunk naps)

Synchronization – TriggerGate

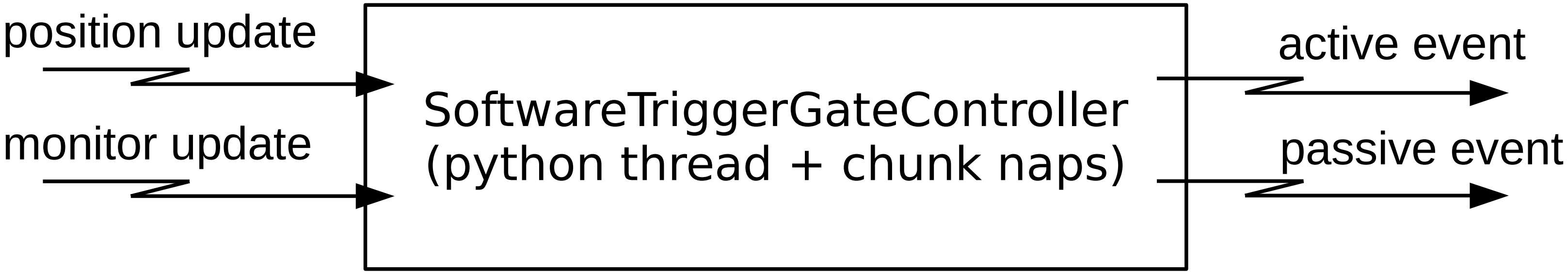
- Basic API – just equidistant configuration
- New controller type allows to plugin custom hardware
- Was only verified in the time domain:
  - Ni660X “PulseTrainGenerator”
  - Software Trigger/Gate



TriggerGate

offset: float
active_period: float
passive_period: float
repetitions: int

Start()
Stop()
State()



position update

monitor update

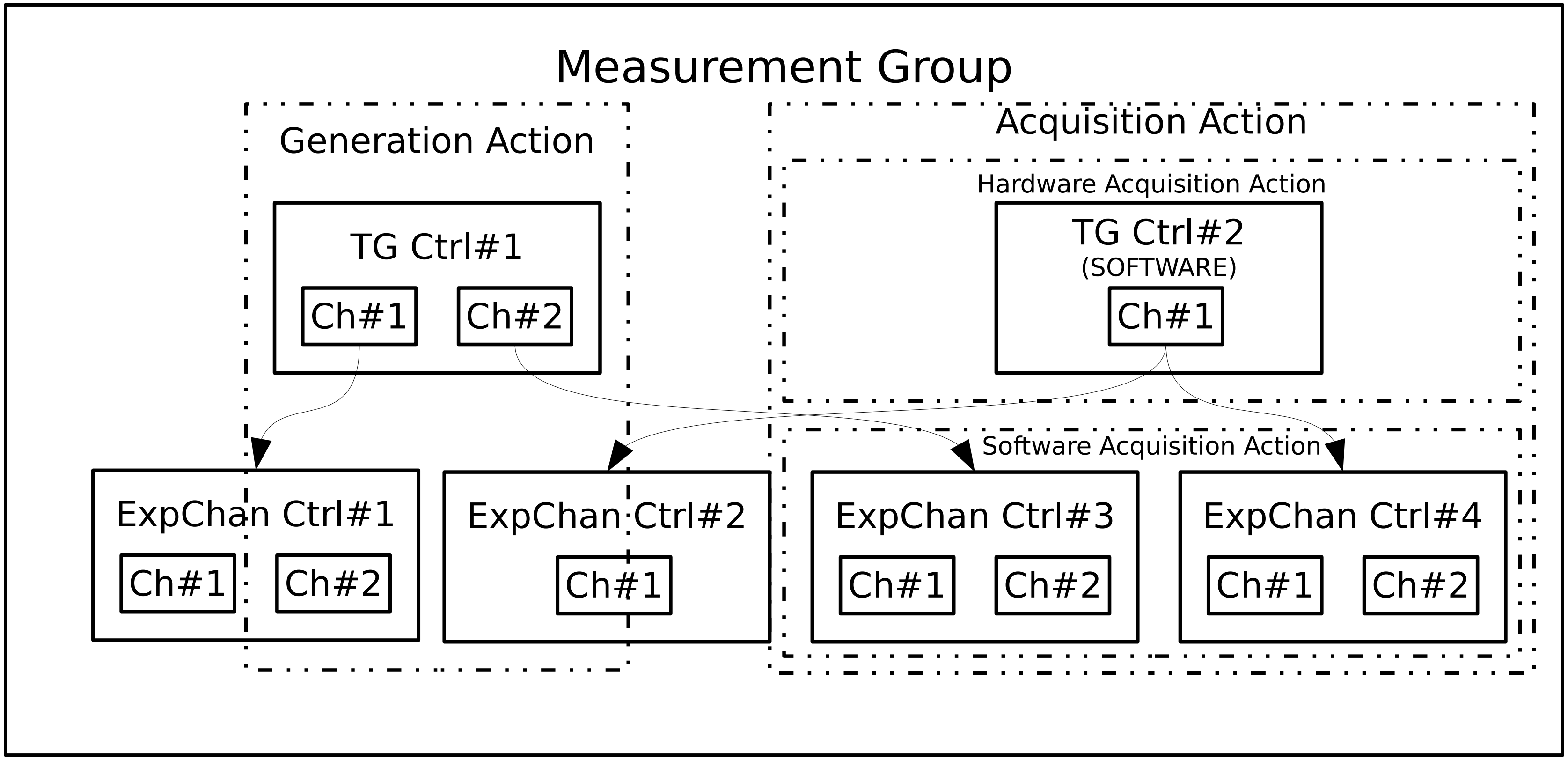
SoftwareTriggerGateController
(python thread + chunk naps)

active event

passive event

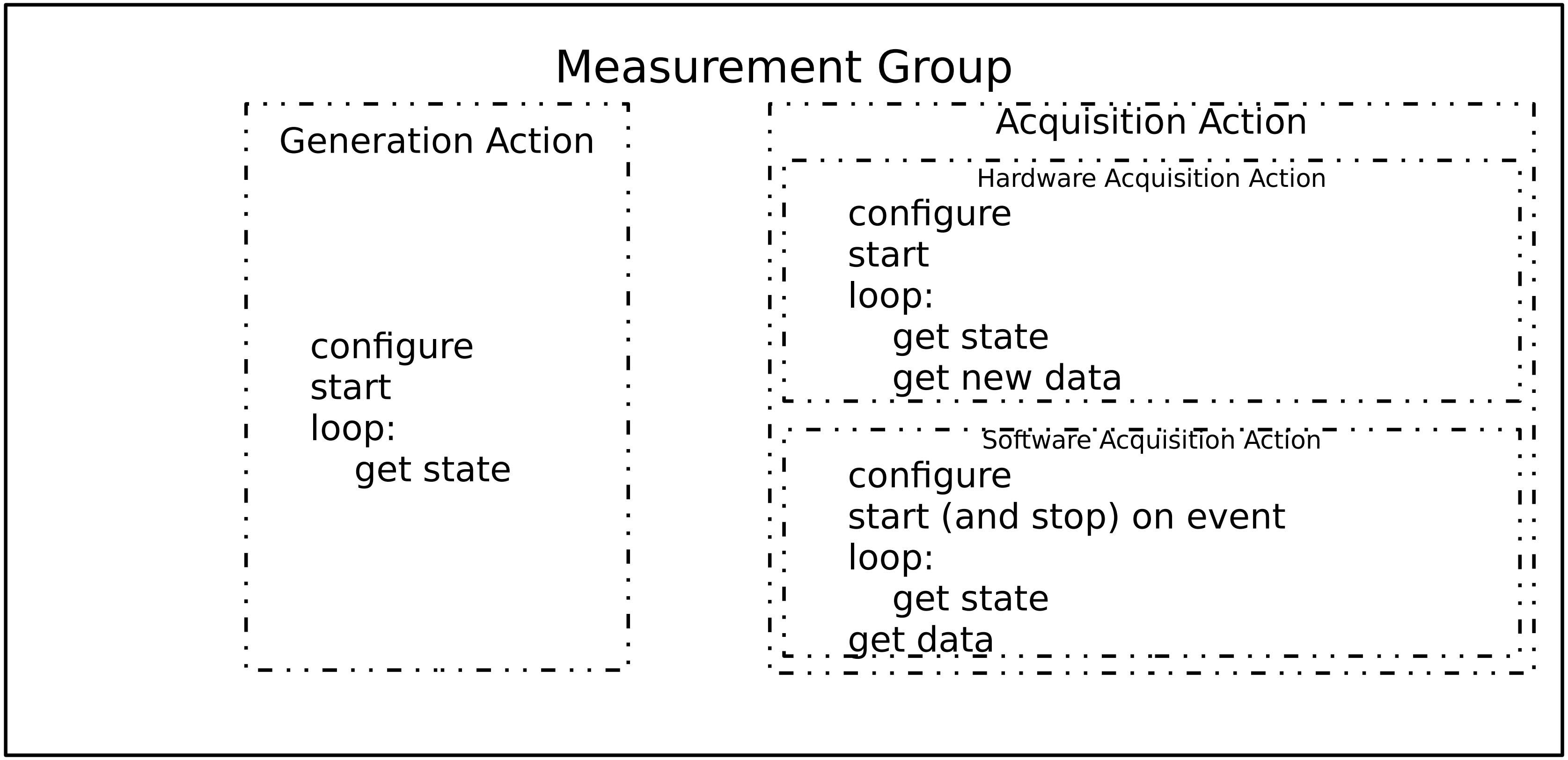
Coordination – Meas. Grp.

- Group of experimental channels and synchronization elements
- Coordination using *acquisition* and *generation* actions



Coordination – Meas. Grp.

- Group of experimental channels and synchronization elements
- Coordination using *acquisition* and *generation* actions



Measurement Group

Generation Action

```
configure
start
loop:
    get state
```

Acquisition Action

Hardware Acquisition Action

```
configure
start
loop:
    get state
    get new data
```

Software Acquisition Action

```
configure
start (and stop) on event
loop:
    get state
get data
```

Configuration – Meas. Grp.

- *Static* dictionary – hardware connections/configuration

dict <str, obj> with (at least) keys:

- 'timer': the MG master timer channel name / id
- 'monitor': the MG master monitor channel name / id
- 'controllers': dict <Controller, dict>:
  - ctrl_object: dict<str, dict> with (at least) keys:
    - ~~'units': dict <str, dict>:~~
      - ~~unit_id: dict<str, obj> with (at least) keys:~~
        - ~~'id': the unit ID inside the controller~~
        - 'timer': the timer channel name / id
        - 'monitor': the monitor channel name / id
        - 'trigger_type': 'Trigger'/'Gate'
        - 'trigger_element': the trigger/gate name / id
        - 'channels': dict<str, obj> with (at least) keys:
          - 'id': the channel name (channel id)

Legend:
<text> - new
~~<text>~~ - to be removed
<text> - to be discussed

Configuration – Meas. Grp.

- *Dynamic* attributes – acquisition parameters:
  - **integration time** → applies to experimental channel controller & trigger/gate channel
  - **repetitions** → applies to experimental channel controller & trigger/gate channel
  - **offset** → applies to trigger/gate channel
  - **dead time / slow down** → affects trigger/gate channel configuration
  - **synchronization source**:
    - moveable → trigger/gate channel
    - monitor → trigger/gate channel & experimental channel controller

Development approach

- Developed iteratively and incrementally by the CTGENSOFT Scrum Team (Guifre Cuni, Carlos Pascual-Izarra, Carlos Falcon, Marc Rosanes, Zbigniew Reszela) + David Fernandez-Carreiras

- Spent time:
  - core development: 8 sprints (2 weeks) – 440 h
  - NI controllers + BL04 migration: 3 sprints – 285h
- Work on the Sardana core level
- Strong emphasis on the automated testing, TDD whenever possible

Future steps

- Update the SEP6 document and probably reduce its scope
- Look for the opinions and contributions from the Sardana Community and others (in particular the ESRF)
- Validate the TriggerGate API in the position domain
- Refactor the Measurement Group configuration
- Implement the optional interpolation of the missed data
- Implement the timescan

Questions

```
Door_BL04 [75]:



Door_BL04 [76]:

Door_BL04 [77]:



Door_BL04 [78]:


Door_BL04 [79]:

Door_BL04 [80]:
Door_BL04 [81]:
Door_BL04 [82]:

Door_BL04 [83]:
Door_BL04 [84]:

Door_BL04 [85]:
Door_BL04 [86]:
Door_BL04 [87]:
Door_BL04 [88]:
Door_BL04 [89]:
Door_BL04 [90]:
Door_BL04 [91]:
Door_BL04 [92]:
Door_BL04 [93]:
Door_BL04 [94]:
Door_BL04 [95]:
Door_BL04 [96]:
Door_BL04 [97]: madscan 11.24 11.44 1
```