MXCuBE meeting 12-13 June 2019

For agenda, organisation and a discussion summary see https://docs.google.com/document/d/1eWDrNS-_PpZN140aoxhDtG7n-iwGdch7I7k_fUPDNEE/, which are the source for much of the material here (thanks, Ivars!).

Participants (11) :

- Jordi, (ALBA)
- Ivars, Jean Baptiste(EMBL-HH, EMBL-GR)
- Marcus, Antonia, Daniele, Olof(ESRF)
- Rasmus (GPhL)
- Michael (HZB)
- Martin (Soleil)
- Gleb (EMBL-HH)

Minutes of previous meetings

Minutes of the Lund meeting were approved – with no positive or negative comments.

Status reports

Rasmus: Done tests at ALBA and Hamburg P14; LABA tests included successful test on real-world samples (thanks to CRIMS group).

Marcus: ESRF is currently concentrating on writing a web version of BSXCuBE for BioSaxs experiments. The old BSXCuBE was built on now deprecated Qt4 based framework called Framework 4. BSXCuBE never used the hardware server but something called Control Objects instead (a similar but different concept). ESRF is now developing a new framework for beamline control applications compatible with both web and Qt front ends. The idea is that new control applications will be built on this new framework. MXCuBE3 will still be based on hardware objects and hardware repository and there is no plan to port it to the new framework. Also, it is possible to use HardwareObjects with the new framework. (MO: it is important to note that the ESRF is fully dedicated to continue the development on hardware objects).
There is active work on serial crystallography at I29; but there are no special initiatives to make sure ISPyB is ready to deal with the result.

Jordi: ALBAis no the 2.3 branch. Engaged in upgrading to new OS, Debian 9, upograding Tango from 7.7 to 9. Optical system has been changed, and resulting problems are being fielded.

Olof: Has accepted Jordi’s changes to EDNA. Is now working on EDNA2, which will switch from XML to JSON and work in Python 3.

Daniele: Taiwan is now ready to sign in to the MXCuBE collaboration.

Ivars: Imaging-based centring using tomography (with 3-click input)

Martin: n-click optical centring installed at PX2. All beamlines now use MXCuBEm, including fxed-energy beamlines; and **now** it is possible to start refactoring. There are no news on PX2 goniostat centring precision.

API object

The pre-meeting PR (by Rasmus) is essentially accepted. It is accepted that the API object should not be a normal HardwareObject. The configuration should be in YAML. The division of objects and their organisation is agreed (see https://docs.google.com/document/d/1ZXRM3qr24LM32ySEviLS6rtOMjqXek5Vgw4Hilt6cOE/edit?usp=sharing).

Abstract classes

- We shall be using methods rather than properties; mandatory methods should be as few as possible, and should be specified using the ABC module. Whenever possible we should return a None value instead of raising NotImplementedError.
- For all objects where a ‘value’ can be reasonably defined we should use get_value / set_value (i.e. **not** get_position, get_energy, …).
- For AbstractCollect we should rewrite using pre_ececute, execute, post_execute. Ivars (**ACTION**) will prepare a PR. Issues relative to single multisweep collection, detector multitrigger, … were discussed but not resolved.

Miscellaneous

- Upgrade to Python 3 is universally desired, but Ivars cannot do it yet, as Tina is Python 2.7 only.
- A normal installer should be built, with a setup.py; Marcus notes that if you use Conda for dependencies this should not be difficult.
- Release notes are necessary – but will have to wait till we produce actual releases.
- Centring could (also?) be a procedure, and should maybe not always be shown in the queue.
- is_ready should default to True for Hardware Objects
- After some discussion it is agreed that centring motors should have role names, but that code should use axis direction definitions and should **not** depend on a given motor name having a given behaviour / direction.
- Centring_maths should be promoted and should be the standard place to store axis direction vectors.
- There should be a beamline phase tha can be set and changed, but individual objects (e.g. diffractometers) may also need phases. There is no decision on the exact organisation
- Doc strings should be Google style
- Copyright should be to MXCuBE for anything that makes an integral part of the application (as opposed to specific plug-ins)

Decisions

Tasks and scheduling

Two weeks

(web meeting in 04.07.2019 – NB since postponed to 11.07.2019)

1. Make an issue to discuss the table. (Ivars)
2. Add HardwareRepository submodule to the release, so when a release is downloaded you get all what you need. (Ivars)
3. Add link to the latest mxcube paper and copy the abstract (Ivars)

Mid September

1. Beamlines object (Rasmus, Antonia)
   - Configure beamline object via yaml?
   - configuration_example.yaml copy to configuration.yaml
   - From HardwareRepository import beamline; beamline.set_phase()
2. Use pre_execute, execute and post_execute in AbstractProcedure (Marcus, Ivars) Add analysis, conclusion steps ?
3. AbstractCollect has just pre_execute, execute, post_execute as task with Exception handling. Contains a single collection item. (Ivars)
4. HWR cleanup
   - Remove AbstractBase, abstracMulticollect, AbstractAttenuators
   - Rename isReady to is_ready (default value is True), rename connectNotify to _connect_notify, remove value_changed, consider better name for update_values (reemit_values?
   - States defined as a class of enum.IntEnum in the BaseHardwareObjects.py

End October

1. Abstract classes:
   - abc for methods expecting from beamline_config (api and gui)
   - **Snake_case** and **doc strings**
   - We will not use properties but methods
   - AbstractActuator has set_value, get_value, get_state…
   - We continue with AbstractTransmission
   - Use of private members, for example: self._value, self._state, self._limits
   - It is prohibited to use beamline.energy._value, beamline.energy.value but use **beamline.energy.get_value()**

   2. Signal handling (for next milestone)

   3. AbstractDiffractometer (Antonia):
      - Move AbstractDiffractometer.set_phase to

4. AbstractCentring rename to Centring and is the only place where motor sequence and

directions are defined

File name corresponds to the class name
Jordi opens two issues about requirements and loggers

Updated Milestones

M1 2018-12-11 to 2019-03-12 (*v3.0.0-alpha.1 - “Akka”)*

- Structural changes
- PEP-8
- Sphinx documentation with Google Style docstrings
- CI with pytest and Pylint
- Python 3 compatibility
- Working version of mockups

M2 2019-03-15 - 2019-09-01 (*v3.0.0.-alpha.2*)

- Abstract classes
- AbstractDiffractometer
- AbstractCollect
- Centering
- Beamline - Introduction of a “beamline” hardware object

M3 date 2019-10-30 (*v3.0.0.-alpha.3 - “Harmony”*)

- Session - For better “data path” and functionality
- Queue – For a more extendable queue system
- Using immutable “pure data” objects for passing data

Beamline Object contents:

Top level methods

set_phase
get_phase

Contents table :

	Abstract class	Sub hwobj	abstractmethod	description
	AbstractActuator (Maybe a better name)		get_value set_value get_state get_limits validate ?	
machine_info	AbstractMachineInfo		get_machine_current	
transmission	AbstractTransmission (AbstractActuator)			
energy	AbstractEnergy (AbstractActuator)		get_wavelength set_wavelength	
flux	AbstractActuator			
beam	AbstractBeam	Aperture Slits Crl or other bea	set_size(hor, ver, shape)	contains (aperture, slits, crls, focusing opt: depends from beamline.
hutch_interlock	AbstractInterlock (abstractNState)			
safety_shutter	AbstractShutter (abstractNState)		open close	
fast_shutter	AbstractShutter (abstractNState)			
diffractometer	AbstractDiffractometer			omega, kappa, kappa_phi, centring_x, centring_y, alignment_x, alignment_y, alignment_z, beamstop
detector	AbstractDetector	detectore_ distance		detector_distance
resolution	AbstractActuator			
sample_changer	AbstractSampleChanger			can be a sample changer, plate_manipulator

				, jets, chips
session	Session			(Session, SiteSpecificSession # including file name parameters, directories, etc.)
lims	ISPyBClient			
graphics	**AbstractGraphics**		get_snapshot (overlay=True)	camera focus zoom
queue_model	**QueueModel**			
queue_manager	**QueueManager**			
collect **gphl_workflow** **xrf_spectrum** **energy_scan** **imaging**	**AbstractProcedure**			with pre_execute, execute, post_execute with task decorator, cleanup and proper exception handling
centring	**AbstractCentring**			contains n-click optical, move_to_beam (double click on the screen), automatic optical and xray centring
offline_processing	**AbstractProcessing**			configurable via beamline_object or session
online_processing	**AbstractProcessing**			configurable via beamline_object or sessionProcessing queue item?,
data_analysis	**AbstractDataAnalysis**			DNA char
motor	**AbstractMotor** (AbstractActuator)		get_dynamic_limits	