Developers meeting at ESRF November 2018

Developers meeting in Grenoble (ESRF) between 2018-11-15 and 2018-11-16.

Present: Martin Savko (MS), Mikel Eguiraun (ME), Ivars Karpics (IK), Michael Hellmig (MH), Jan Meyer (JM), Rasmus Fogh (RF), Jakob Urbschat (JU), Bixente Rey (BR), Antonia Beteva (AB), Marcus Oskarsson (MO).

No participants absent

The aim of the meeting was to:

- Merge HardwareRepository (HR) 2.2 and current master to create a basis (3.0-alpha) for what will later become HR version 3.0.

- To identify Hardware Objects and other parts of HR that should be refactored as part of the merge process.

- Discuss and establish coding standard and routines to be used for version 3.0 of HR.

- Outline a roadmap for the release of HR 3.0 that can be given to the steering committee in February

A document with a few preparatory assignments designed to indicate where one could expect to find the biggest differences between the two branches 2.2 and master was circulated before the meeting. The document further requested all sites to create PR's with all changes to be included in the merge.

*"The **absolutely most important** step would be for each site to make a PR with their local changes before we perform the merge. It will further be **very complicated to add features** to either 2.2 or 2.3 (master before the merge) while we are in the process of creating 3.0 (merging 2.2 into 2.3 with following cleanup)"*

The complete document can be found in appendix

Day 1 (2018-11-15)

Meeting opened 9:28 by (MO), there were no additions to the already circulated agenda. The meeting agenda can be found in appendix.

There were two pull requests, #251 and #234, still remaining before the 2.3 branch could be created from the master branch. PR #234 was merged directly. PR #251 by RH required some discussion mainly because the already existing method *getHardwareRepositoryPath()* was removed and replaced by a semantically different method *findInRepository(name).* The former returning a directory and the later a file. Both AB and BR argued that the *getHardwareRepositoryPath()* should not be removed. It was decided to in anyways merge PR #251 since the method *getHardwareRepositoryPath()* currently was unused. However it was said that *findInRepository(name)* should be looked at further during the work with HR 3.0.

Branch 2.3 was created from master (commit 9496e78f...) and merging HR 2.2 into master started 9:48. The merge proceeded smoothly the biggest amount of conflicts were in queue_model_objects_v1.py, queue_entry.py, ISPyBClient2.py and Cats90.py. The method *getHardwareRepositoryPath()* that was removed by PR #251 was added back again. The merge finished at 16:19.

MO and AB presented the currently used coding conventions and tools as well as some ideas to be used as a discussion starter. The ideas presented was for improving the process for the work of HR 3.0 and included

- Clarify meaning of abstract and generic HO's
- Improving folder structure
- A way to enforce PEP-8 coding convention
- Writing tests and setting up CI
- Improving code documentation

The topic *"Clarify meaning of abstract and generic HO's"* required further discussion to explore the issue. There was otherwise a generally positive view of the ideas presented. It was decided to continue the discussion on these points at a later stage of the meeting.

AB continued to present a draft merge of MiniDiff and GenericDiffractometer to be used as a basis for discussion of the functional merge of a new common diffractometer object. It was decided that AB will continue the work on the draft merge and present a PR what will be discussed at a future developers meeting.

Day 1 ends at 18:30

Day 2 (2018-11-16)

The improvement suggestions on code convention and tools are discussed further and its decided

- To stop using the Generic prefix from HO's since it is often misunderstood and confused with Abstract. The Abstract prefix is to be used in all cases where one wants define a common API for a HO for example AbstractDiffractometer. The need for having a intermediate level of abstraction prefixed with Base is discussed along with if Abstract classes are allowed to have a certain level of implementation.

- To cleanup the Hardware Object directory structure; creating a mockup folder, moving the contents of detectors to the root folder and moving all concrete sample changer implementations to the root folder. As well as renaming the modules modules queue_model_objects_v1 to queue_model_objects and ISPyBClient2 to ISPyBClient

- To use PEP-8 for at least all new code and investigate ways of automatically converting the current code base to PEP-8 using tools like autopep8 or Black.

- To write Python3 compatible code

- To further explore means of implementing tests and continuous integration (CI)

- To use semantic versioning according to semver.org:

  "Given a version number MAJOR.MINOR.PATCH, increment the:

  - *MAJOR version when you make incompatible API changes,*
  - *MINOR version when you add functionality in a backwards-compatible manner, and*
  - *PATCH version when you make backwards-compatible bug fixes."*

MS took the initiative to try autopep8 and Black on the existing code base, the preliminary result looks promising. MS and AB expressed the wish to further look into the problem of converting the existing code base to PEP-8

AB presents some ideas and a draft merge of the AbstractCollect and AbstractMultiCollect that can be used as basis for discussion. Her proposal to introduce the concept of 'phases to collection, possibly in the form of 'pre-execute' and 'execute' commands was well received, as was her proposal to standardise access to motor objects to using a fixed set of role names. Both MO and AB expresses that it would be beneficial with an object for accessing "beamline wide" functionality to avoid duplication of logic. Such functionality could for instance be opening and closing shutters, reading certain values and accessing lims. This functionality is today often accessed via the collect object itself which often adds an

unnecessary level of logic. MO also expresses that it would be good if the collect parameters could be passed as a well defined “Data Object” preferably immutable instead of a python dictionary. It’s agreed that the draft collect presented by AB is to be continued and further discussed during a coming developers web meeting.

Other HO’s and parts of the HR to improve are identified; there is an overall wish to improve Session and especially the path related functions. There is also wishes to improve the queue related functions and IK expresses that he has some ideas that he could present during a developers web meeting.

The contents of the road map requested by the steering committee is discussed. There was general consensus that it should be possible have a 3.0.0-alpha version that runs on mockups by the next MXCuBE meeting in Lund. The exact contents of 3.0.0-alpha remains to be decided.

Day 2 ends 15:00

Summary

Actions:

- Branch 2.3 created from master (9496e78f…)
- Merged branches 2.2 and master, 2.2 into master

Work to be done/discussed before/during the next developers web meeting:

- AB to continue work on AbstractDiffractometer to create a PR that can be further discussed during a developers web meeting.
- To continue the work on the merged version of AbstractCollect and AbstractMultiCollect to be discussed during the next developers meeting.
- Session, Queue related objects, an object for “beamline wide” functions and “Pure data class”
- Converting current code base to PEP-8, status report by MS and/or AB
- Decide on roadmap and contents of 3.0.0-alpha

Decisions

- To stop using the Generic prefix from HO’s since it is often misunderstood and confused with Abstract. The Abstract prefix is to be used in all cases where one wants define a common API for a HO for example AbstractDiffractometer.

- To cleanup the Hardware Object directory structure; creating a mockup folder, moving the contents of detectors to the root folder and moving all concrete sample changer implementations to the root folder

- Rename modules: queue_model_objects_v1 to queue_model_objects and ISPyBClient2 to ISPyBClient

- To use PEP-8 compatible code for at least all new code and investigate ways of automatically converting the current code base to PEP-8 using tools like autopep8 or Black.

- To write Python3 compatible code

- To further explore means of implementing tests and continuous integration (CI)

- To use semantic versioning according to semver.org:
  “Given a version number MAJOR.MINOR.PATCH, increment the:

    - *MAJOR version when you make incompatible API changes,*
    - *MINOR version when you add functionality in a backwards-compatible manner*
    - *PATCH version when you make backwards-compatible bug fixes.”*

Appendix - Meeting agenda

Day 1: 9h-12:30h

- 09.00: Meet in ESRF mezzanine
- Perform merge
    - Create branch 2.3 from master
    - Merge master into 2.2 to create new master that will become version 3.0

Day 1: 14h-18h

- Functional merge
    - Diffractometer HardwareObjects, based on GenericDiffractometer (master) and MiniDiff (2.2)
    - Collect HardwareObject, based on AbstractCollect (master & 2.2) and AbstractMulticollect (2.2)

Day 1: 19h -

- Dinner downtown for those who wants

Day 2: 9h-12h

- (Continue with tasks from previous day, if left)
- Identify HardwareObjects to continue with for instance: Attenuators, Energy and Resolution
- Tasks to be done until next web meeting
- (Test with Mockups, if time is given)

Day 2: 13h-15h

- Discuss and clarify the points/tasks needed to be done for the roadmap that the steering committee expects
  in February
- (Brief discussion on coding standard and routines to be used for version 3)

Appendix - Meeting “Homework” document

Homework

Its necessary for each site to get in depth knowledge of what’s used on their beamlines and what will be important to consider during the merge. The **absolutely most important** step would be for each site to make a PR with their localchanges before we perform the merge. It will further be **very complicated to add features** to either 2.2 or 2.3 (master before the merge) while we are in the process of creating 3.0 (merging 2.2 into 2.3 with following cleanup)

Please complete the following steps before the meeting:

Task \ site	ALBA	BESSY	DESY	ELETTRA	EMBL	ESRF	GPhL	MAXIV	SOLEIL
Running mockup version of qt/web									
Update example xmls									
List all used hwobj classes (table on page 3)									
Submit changes to master/2.2									

We will further **take a closer look at the Collect routines and Diffractometer HardwareObjects, so please compare those** in order to get an idea what the principal differences are and what is important to you, doing for instance the following:

1. Compare AbstractMultiCollect on 2.2 with AbstarctCollect on master:
```
# Compare AbstractMultiCollect on 2.2 with AbstractCollect on master:
git diff 2.2:HardwareObjects/AbstractMultiCollect.py
master:HardwareObjects/AbstractCollect.py
```

2. MiniDiff.py on 2.2 with GenericDiffractometer on master
```
# Compare AbstractMultiCollect on 2.2 with AbstarctCollect on master:
git diff 2.2:HardwareObjects/MiniDiff.py
master:HardwareObjects/GenericDiffractometer.py
```

It’s also **strongly recommended** to perform the merge command just to get an overview where the most evident conflicts
are located and how these could affect you:

```
git checkout 2.2; git merge master; git gui
```

Do not forget to fill out the table on the following page !

List of used classes:

ALBA	BESSY	DESY 2)	EMBL	ELETTRA 3)	ESRF	GPhL 1)	MAXIV	SOLEIL
AbstractCollect AbstractDataAnalysis AbstractDetector AbstractMotor		AbstractMotor AbstractAperture AbstractCollect AbstractDataAnalysis AbstractEnergy AbstractEnergyScan AbstractFlux AbstractMultiCollect AbstractSlits AbstractXRFSpectrum	AbstractAperture AbstractAttenuators AbstractCollect AbstractDetector AbstractEnergyScan AbstractMotor AbstractSlits AbstractXRFSpectrum	AbstractMultiCollect AbstractDataAnalysis	AbstarctMultiCollect AbstractEnergyScan AbstractMCA AbstractMotor	AbstractAperture AbstractCollect AbstractDataAnalysis AbstractDetector AbstractEnergy AbstractEnergyScan AbstractFlux AbstractMotor AbstractSlits AbstractXRFSpectrum	AbstractCollect AbstractMotor	AbstractMotor AbstractCollect AbstractXRFSpectrum AbstractEnergyScan AbstractDetector AbstractAperture
GenericVideoDevice GenericDiffractometer SampleChanger		GenericDiffractometer GenericParallelProcessing GenericVideoDevice	GenericDiffractometer GenericParallelProcessing GenericSampleChanger GenericVideoDevice	GenericSampleChanger GenericDiffractometer	GenericSampleChanger	GenericDiffractometer GenericVideoDevice GenericSampleChanger	GenericDiffractometer GenericSampleChanger	GenericDiffractometer GenericSampleChanger GenericVideoDevice
BeamlineSetup DataAnalysis ISPyBClient2 InstanceServer Session QueueManager QueueModel Qt4_TangoLimaVideo Qt4_GraphicsManager Cats90 CatsMaint ParallelProcessing LdapLogin SardanaMotor (sample_centring)		CentringMathEnergy MotorWPositions Qt4_GraphicsManager SardanaMotor ApertureMockup AutoProcessingMockup BeamInfoMockup BeamlineSetup BeamlineTestMockup BeamstopMockup CameraMockup CollectEmulator CollectMockup DataAnalysis DiffractometerMockup EnergyMockup EnergyScanMockup FluxMockup GphlWorkflow GphlWorkflowConnection InstanceServer	Attenuators BeamlineSetup BeamlineTools CentringMath DataAnalysis DozorParallelProcessing ExporterMotor ExporterZoom ISPyBClient2 InstanceServer MDFastShutter MiniKappaCorrection MicrodiffLightSession QueueManager QueueModel XMLRPCServer Qt4_VimbaVideo Qt4_Graphic	MicrodiffInOut TangoShutter MicrodiffBeamstop MicrodiffInOutMockup ShutterMockup BeamlineSetup EnergyScanMockup DataAnalysis EdnaWorkflow Shapes PlottingMockup AbstractDataAnalysis XMLRPCServer QueueModel	BeamInfo BeamlineSetup Bliss BlissActuator BlissInOut BlissMotor BlissWagoCounter Camera DataAnalysis EdnaWorkflow Energy FilterAxis Frontend GphlWorkflow GphlWorkflowConnection InOut InstanceServer ISPyBClient2 ISPyBRestClient kb Lakeshore LdapLogin LimaVideo MachCurrent	ApertureMockup AttenuatorsMockup BeamInfoMockup BeamlineSetup BeamlineTestMockup BeamstopMockup CollectMockup DataAnalysis DetectorMockup DoorInterlockMockup EnergyMockup EnergyScanMockup FluxMockup InstanceServer ISPyBClient2Mockup MachineInfoMockup	BeamlineSetup ISPyBClient2 ISPyBRestClient QueueManager QueueModel Session (inherited) Cats90 CatsMaint SardanaMotor MicrodiffAperture BeamInfo(inheritted) Energy Resolution DataAnalysis TangoShutter MicrodiffLight MicrodiffMotor MicrodiffInOut TangoLimaVi	BeamlineSetup ISPyBClient2 LdapLogin InstanceServer Session QueueManager Qt4_GraphicsManager Qt4_VimbaVideo XMLRPCServer MicrodiffInOut MicrodiffMotor MicrodiffZoom MicrodiffLight Energy Resolution EnergyScanMockup XRFSpectrumMockup DetectorMockup Shapes Cats90 CatsMaint

		ISPyBClient2Mockup LdapLoginMockup MachineInfoMockup MiniKappaCorrection MotorMockup MultiCollectMockup ParallelProcessingMockup QueueManager QueueModel RedisClient ShapeHistory SlitsMockup UnitTest XRFSpectrumMockup	sManager		MD2Motor MD3UP Microdiff MicrodiffAperture MicrodiffBeamstop MicrodiffBeamstopDistance MicrodiffFocusMotor MicrodiffHolderlength MicrodiffInOut MicrodiffKappaMotor MicrodiffLight MicrodiffLightBeamstop MicrodiffSamplePseudo MicrodiffZoom MiniDiff Pilatus QueueManager QueueModel Resolution sample_changer.FlexHCD sample_changer.FlexHCDMaintenance Scintillator Session Shapes TangoLimaVideo TangoShutter XMLRPCServer XRFSpectrum	MicrodiffZoomMockup MotorMockup DiffractometerMockup CollectMockup ParallelProcessing sample_changer.PlateManipulatorMockup Qt4_GraphicsManager Qt4_VideoMockup QueueManager QueueModel ResolutionMockup ShutterMockup SampleChangerMockup Session SlitsMockup XMLRPCServer XRFSpectrumMockup *Also the following*: CommandContainer Component Container Device DeviceContainer Equipment HardwareObject HardwareObjectNode InstanceServer Procedure QueueEntryContainer	deoDevice MicrodiffZoom Shapes CentringMath ControlSystemChannel	
ALBAAutoProcessing ALBABackLight ALBABeamInfo ALBAZoomMotor ALBACalibration		Centring DigitalZoomMotor MjpgStreamVideo NanoDiff	EMBLAperture EMBLAutoprocessing **EMBLBeamlnfo** EMBLBeamlineTest EMBLBeamstop	ELETTRA/TangoEnergy ELETTRA/GenericDiffractometer ELETTRA/TangoChannel ELETTRA/GalilAxis	BlissHutchTrigger BlissRontecMCA ESRFEnergyScan ESRFSessionID232BeamCmds ID232BeamD	CollectEmulator GphlWorkflowConnection GphlWorkflow	BIOMAXEiger BIOMAXTransmission MAXIVAutoProcessing MaxIVSession BIOMAXResolution	SOLEILISPyBClient SOLEILSession SOLEILPss SOLEILSafetyShutter SOLEILMachineInfo SOLEILLdapLogin

ALBACatsMaint ALBACats ALBADataAnalysis ALBAISPyBClient ALBAEnergy ALBAFastShutter ALBAFlux ALBAFrontEnd ALBAFrontLight ALBAMachineInfo ALBAMiniDiff ALBACollect ALBAEpsActuator ALBAPilatus ALBASession ALBASupervisor ALBATransmission ALBAZoomMotorAutoBrightness ALBAZoomMotor			EMBLDoorInterlock EMBLEnergy EMBLEnergyScan EMBLExporterClient EMBLImageTracking EMBLFlux EMBLMachineInfo EMBLMiniDiff EMBLCollect EMBLPPUControl EMBLParallelProcessing EMBLSafetyShutter EMBLXRFSpectrum Marvin TINEMotor	ELETTRA/ElettraBeamInfo ELETTRA/ISPyBClient2Mockup ELETTRA/ElettraSession ELETTRA/SampleChangerElettra ELETTRA/MD2TangoMotor ELETTRA/MachCurrent ELETTRA/MD2TangoCoaxialZoom ELETTRA/__init__ ELETTRA/BCS_Valve ELETTRA/MD2Beamstop ELETTRA/MD2Shutter ELETTRA/PilatusDetectorTango ELETTRA/MultiCollectTangoMD2 ELETTRA/MD2TangoAperture ELETTRA/VUO_AuthenticateClient ELETTRA/ResolutionElettra ELETTRA/TangoDiffractometerMD2 ELETTRA/MD2TangoCamera ELETTRA/Transmission ELETTRA/TangoActuator ELETTRA/TangoCryo ELETTRA/MD2TangoLightLevel ELETTRA/BeamlineActions	efiner ID232BeamInfo ID232HutchTrigger ID232MultiCollect ID232PhotonFlux ID29BeamCmds ID29EnergyScan ID29MultiCollect ID29PhotonFlux ID29XRFSpectrum ID30A3PhotonFlux ID30BBeamCmds ID30BBeamInfo ID30BEnergyScan ID30BMultiCollect ID30BPhotonFlux ESRFMultiCollect Oxford700 Oxford700Mockup TangoKeithleyPhotonFlux Transmission		MachInfo BIOMAXAperture BIOMAXBeamInfo BIOMAXBeamlineActions BIOMAXEnergy BIOMAXMD3 BIOMAXMD3Camera MAXIVMicrodiffInOut	SOLEILUndulator TangoDCMotor TangoShutter PX1BeamInfo PX1TangoLight PX1Attenuator PX1CatsMaint PX1DetectorDistance PX1MiniDiff PX1EnergyScan PX1Eiger PX1AutoProcessing PX1Cryotong PX1Resolution SmargonAxis Smargon ChipManager PX1Configuration TangoDCMotorWPositions TangoMachCurrent PX2BeamInfo PX2Attenuator PX2MicrodiffZoom PX2Collect PX2EnergyScan PX2XRFSpectrum PX2Diffractometer PX2Resolution PX2Energy

1) These are all HardwareObjects loaded in a standard mock run, plus their superclasses. That does not mean that they are all *used*. GPhL does not do plates, video, xmlrpc, xrf, energy scan, or beamline test, for instance.
2) The same as for GPhL goes for DESY as well. Only the classes in the first paragraph are really doing something, the second paragraph came with the mockup.
3) The ELETTRA specific HardwareObjects are not yet in the GitHub repository.