Meeting Minutes - MXCuBE Developers code camp 2nd-3rd and 28th of April 2020

Introduction

The meeting was held virtually at whereby.com/mxcube between the 2nd and 3rd of April 2020. The meeting was originally scheduled for the end of April with Soleil or ESRF as host. ESRF management had due to the EBS upgrade discouraged participation to, or hosting of, events not directly associated with the upgrade program. Soleil therefore proposed to host the meeting as it would mean less time spent at ESRF for organising the meeting and involve only a short travel. The meeting was at ESRF's initiative then rescheduled to the first week of April and held virtually because of the COVID19 pandemic. The information at the time of the decision indicated that most sites in Europe would close for several weeks, meaning that most people would be back at work again for the original date and then most likely be too busy with other activities to attend the meeting.

Participants (Day 1, 2 and 3)

- Jordi (Alba)
- Lais (LNLS)
- Rasmus (GPhL)
- Ivars (EMBL-HH)
- Martin (Soleil)
- Michael (HZB)
- Jean Baptiste (EMBL-GR)
- Marcus, Antonia and Daniele (ESRF)

Minutes: Marcus and Rasmus

Organisation:

The aim of the meeting was to write and update documentation, mockup classes and tests for classes added and updated as a result of the recent refactoring work. The meeting was held over three days, two initial consecutive days and a final day for summarising the work done, leaving the participants some time to review the work done and finish the tasks they had been assigned.

Day 1 and 2:

The meeting took place on whereby.com/mxcube where the participants meet at twice a day. Each day had a specific topic, Day 1 - instrumentation and Day 2 - Procedures. The participants meet twice a day to assign and review/discuss the work being done. The abstract classes to be worked on were assigned an implementer responsible to write the mockup, test and documentation and one or several reviewers to review the work. A document with a list of abstract classes and the possibility to volunteer as implementer or reviewer for a given class were circulated before the meeting.

Day 1 started with an introductory session, agenda below.

Day 1 (Instrumentation)

09h - 10h	Introduction ● Review of contributing guidelines and examples and other questions ● Workflow ● WIP Abstract classes clarifications - for discussion at code camp #496 ● To what HardwareObject should beam location/center belong ● Should Energy implement AbstractMotor ● Discussing release of a 3.0.0(-alpha) version ?
10h - 14h	Session 1
14h- 15h	Review of Session 1 and assignment for Session 2
15h - 17h	Session 2

Introduction

A very brief review of the current contributing guidelines were performed, AB summarized the Sphinx Google style docstrings and a link to the GitHub pages “contributing guidelines” were given.

Workflow

The participants agreed on a workflow where a PR were to be created for each AbstractClass with the name [CODE CAMP] - <abstract class name> and possibly with the [WIP] prefix to indicate that it was being worked on (not ready to be reviewed). The branch to be used was cc-<abstract class name> .

WIP Abstract classes clarifications - for discussion at code camp #496

RF had before the meeting provided a PR (#496) to be discussed during the introduction. The idea of the PR was to discuss a possible implementation to enforce the access to certain members of AbstractActuator and BaseHardwareObject. The intent was to facilitate the implementation of subclasses, reduce code duplication and mistakes. The idea was appreciated by the participants. However a big concern regarding the limitations of this approach was expressed, by MO, AB and MS. The proposed implementation supports a majority of the possible cases but there are a few cases where the proposed access restriction would be too restrictive and make implementation complicated, including the occasional need to get a value without actual hardware access. It was therefore decided to **not** make certain members of AbstractActuator and BaseHardwareObject private by using __ (double underscore).

Decision: To not use __ to make _value, _state private.

To what HardwareObject should beam location/center belong

RF noted that “beam center” exists on both BeamInfo and Detector and wonders to which object it should belong. IK explained that there are two beam centers one for the X-Ray detector and one for the on axis viewer (sample view). MS states that it would be better if beam center for the X-Ray detector was kept on the Detector object as this would give the possibility to support the use of multiple detectors (something that could be interesting).
As part of the discussion it is confirmed that beam position/centre should be given in pixels.

Decision: Beam center for the X-Ray detector should belong to detector and beam center for the on axis viewer to SampleView.

Should Energy implement AbstractMotor

RF further wanted to discuss if the Energy object should inherit AbstractMotor.

Decision: Energy should inherit AbstractActuator

Discussing release of a 3.0.0(-alpha) version ?

A discussion regarding a release of a 3.0.0-alpha version to indicate that the refactoring work is stabilizing is discussed. JA raises the idea that the signals emitted from a particular class also should be discussed and clearly documented as a part of the API. It's generally agreed that a documentation of the signals should be part of the API and discussed at a later moment, perhaps at the Barcelona meeting. IK also notes that it might be interesting to use some kind of marshalling library to validate the data sent in the signals, and specifically mentions the Marshmallow library. MO indicates that there currently is a similar proposal for the AbstractProcedure using the library Pydantic.

Decision: Signal documentation as part of the API to be further discussed and hopefully settled for the Barcelona meeting.

Any other business

Specific states

The question was raised (by RF) whether specific states should cover all possible states, or whether the specific state might be None when the HardwareObject were in an uninteresting state (such as READY or UNKNOWN already covered by the generic states. It was agreed not to take a decision on this point and leave both possibilities open for individual cases.

AbstractTransmission

The question was raised (by JA) whether there was a need for an AbstractTransmission class, or if AbstractMotor would be sufficient. It was agreed that AbstractTransmission should be written and used. Review of Session 1 and assignment for Session 2
The abstract classes worked during the morning were:

AbstractBeam - Ivars

AbstractDetector - Martin

BaseHardwareObject - Rasmus

AbstractActuator, AbstractNState and AbstractMotor - Implementation: Antonia, Review: Rasmus

AbstractSampleChanger - Marcus, Jean Baptiste

AbstractTransmission - Jordi

Most of the work were still ongoing when the second video meeting of the day was held, and most PR's were thus marked WIP. Apart from these there were a few minor PR's made with various bug fixes done just before and during the code camp which could be merged directly. A status round where each person presented their work and the PR's were discussed were conducted.

Discussion

It was confirmed that transmission should be given in % (from 0 to 100) (raised by JA).

It was agreed that the detector get_beam_position function should have both detector distance and wavelength as input parameters, and to add additional functions of AbstractDetector, handling e.g. energy thresholds. Point raised by MS.

Day 2 (Procedures)

09h - 10h	Review of Session 2 and assignment for Session 3
10h - 14h	Session 3
14h- 15h	Review of Session 3 and assignment for Session 4
15h - 17h	Session 4

Review of Session 2 and assignment for Session 3

The work from the previous day continued during session 3. IK states that it would be a good idea if we could test all the base class functionality from i.e. AbstractMotor in a generic way. RF volunteers to create a set of tests that test all common functionality from AbstractActuator, Motor and NState. MO makes the remark that pytest has the concepts of fixtures that could be interesting to look into. It was decided to postpone the discussion regarding AbstractProcedure to the afternoon as there is still quite a lot of work to finish with the current PR's.

Discussion

The concept of a generic propertiesChanged' signal was discussed and approved (raised by IK).

It is agreed that AbstractFlux is probably needed, since flux has functions to do with dose rate that are not covered by AbstractMotor.

The need for AbstractMCA is discussed, but the decision is postponed.

The question is raised whether there is a need for AbstractSlits. The consensus is that there is. AB proposes to introduce AbstractBeamDefiner; the consensus is that this would be useful but should not be done at this meeting.

Review of Session 3 and assignment for Session 4

MO presents his view of the AbstractProcedure and how it could fit with what's currently done in MXCuBE3 and how certain parts, like displaying dialogs on demand are common with for instance the workflows. MO also explains that there is an idea to make it possible to provide an easy way to write custom procedures that could be dynamically loaded into MXCuBE from for instance a certain folder on the file system. MS makes the remark that he thought the Procedures were for Collect, Energy and XRF scan. MO explains that that is also the case and that those are included in the presented concept. MS states that it would be interesting to also consider an "analysis" and "conclusion" functionality in addition to "pre execute", "execute" and "post execute", which is met with general agreement. A discussion regarding the current queue implementation is conducted as MS also states that the way characterisation is currently implemented is quite good. MO explains, with clarification from RF, that a "depth first" execution pattern is used on the task nodes. Each task node has a list of child tasks that are executed before the parent task and the result of the child node execution is attributed to the parent which is then executed. MO asks if this kind of model would be sufficient, which also seems to be meet with general agreement. AB, MS and (IK in his absence but due to previous expressed interest) is assigned to investigate if the current AbstractProcedure could be used for Energy and XRF Scan.

Decision: AB, MS and IK to investigate the possibility of using the current proposal for AbstractProcedure for Energy and XRF scan.

MO asks when the third day of code camp should take place and suggests the last week of April. It's decided that MO will create a doodle for the third day of meeting.

Decision: MO to create doodle for third day of code camp

Discussion

It is agreed that AbstractActuator.value may be (set to) None. (point raised by RF).

Day 3 (Concluding discussion)

The third and final day of the code camp was held on the 28th of April. This would give the participants some time to finish the work and discussions that were started during the first phase, day 1 and 2, of the codecamp.

The day was divided into two sessions, the first intended to conclude ongoing activities and the second to discuss what to do next.

Session 1

9h-10.30h:	- Status update - Review pending PR's - Discuss "APRIL CODE CAMP" tagged issues - AOB

Session 2

14h-15.30h:	- Discussion on procedures and how to proceed with that work - Draft possible solution - List HardwareObjects that are affected by this change - Discussing creating a pre release with what we consider to be stable API-wise - Listing HardwareObjects that we consider have a stable API - AOB

Session 1

Status update

There were no immediate blocking situations related to the code camp reported from the status round. Some sites, EMBL-HH and HZB Berlin, are carrying out covid-19 related research and have been busy with users. The rest of the sites as well as GPhL are also working in parallel with other activities.

Review pending PR's

All the pending PR’s were reviewed:

- [CODE CAMP] AbstractMachineInfo issue #448 (Review 2):
- [CODE CAMP] Added file containing .py header text
- MicrodiffAperture inherit from ExporterNState instead of ExporterMotor.
- [WIP] Snake case to BaseHardwareObjects
- [WIP][CODE CAMP] AbstractSampleChanger - Temp PR (added new test)
- [CODE CAMP] AbstractDetector
- [WIP] AbstractDiffractometer issue #276

The pull requests that had points that could benefit from everybody's presence or had blocking issues were discussed, the rest was left for discussion on GitHub.

[CODE CAMP] AbstractMachineInfo issue #448 (Review 2)

LC made a revised version of the previous AbstractMachineInfo PR, nothing blocking.
It was agreed that AbstractMachineInfo should not be an AbstractActuator.

[CODE CAMP] Added file containing .py header text

Nothing blocking. It was agreed that there was no need for an initial ‘#!’ line. Copyright statement can be from 2010 or current year, as you prefer.

MicrodiffAperture inherit from ExporterNState instead of ExporterMotor:

AB asks whether it’s a good idea to combine the translation in and out of the beam into the aperture hardware object and implement those as set_in and set_out. JA states that the “in beam” position could be implicit with a selected aperture and that the “out of beam” position could be added to the list of apertures.

IK had raised a point through a comment on the PR regarding using units consistently in all classes inheriting the same base class. The matter is discussed and there are some concerns that it would be a time consuming task. MO suggests creating an issue so that the details on how units could be used/implemented could be discussed.

Action: MO to create issue for unit discussion

[WIP] Snake case to BaseHardwareObjects:

MO suggest that a day is dedicated to the snake case conversion as its important to share the knowledge on difficult cases and benefit from the change being done in “one moment”

IK explaines that he thinks that the to snake case refactoring should be made in three steps

- Method definitions are renamed
- Deprecation warnings are added for the old definitions
- The site specific code is updated to use the snake case definition

RF states that a three stage process like the one suggested by IK would mean unnecessary work as the deprecation warnings need to first be added and then removed. It would then be, according to RF, be likely that the removal of the deprecation warnings are not done completely or at all.

It's decided that the approach will be further discussed in an issue opened by IK. There is a slight favor for using MO's suggestion of dedicating a day for the refactoring together with RF's suggestion of not adding deprication warnings.

IK asks if it would not be good to decide a date for when the came to snake case conversion would take place and suggests sometime in June.

MO states that June is probably in the middle of vacation for many people but that how to proceed on the matter could be discussed on the June monthly web meeting

Action: IK to open issue for camel- to snake case conversion
Decision: To not use deprecation warnings but to perform the conversion in one go
Decision: To add discussion on camel to snake case to June meeting agenda

[WIP][CODE CAMP] AbstractSampleChanger - Temp PR (added new test):

Nothing blocking

[CODE CAMP] AbstractDetector

There is an ongoing discussion on the PR there is nothing blocking

[WIP] AbstractDiffractometer issue #276

IK asks for the state of AbstractDiffractometer and AB comments that it's outdated because of the recent changes to HWR.

Discuss "APRIL CODE CAMP" tagged issues

The issues that could benefit from the presence of all meeting participants were discussed,

- AbstractVideoDevice
- HardwareObject Specific state
- Replace all CamelCase with snake_case
- Replace codacy with something else
- Consider adding attributeChanged signal
- Consider renaming def.update_values to def.emit_values
- Add licence and attributes as header to all code files
- AbstractCollect
- AbstractProcessing or AbstractOfflineProcessing and AbstractOnlineProcessing
- Create AbstractResolution
- Create AbstractMachineInfo
- Centring

- AbstractProcedure

A few of the issues, *Replace all CamelCase with sanke_case*, *Add licence and attributes as header to all code files* were discussed during the pull request review.It was further decided that the issues related to AbstractProcedure: *AbstractProcessing or AbstractOfflineProcessing and AbstractOnlineProcessing, AbstractCollect, AbstractProcedure* would be further discussed during the afternoon.

The review of the issues amounted in the following actions and comments:

Comment: MO states the it would be good if all sites could list the classes used for video streaming in issues #549 AbstractVideoDevice.

Decision: The implementation of a possible attributeChanged signal should be done, if at all, after the refactoring work has been completed.

Decision: The update_values method of BaseHardwareObject should be renamed to re_emit_values

Comment: IK comments that AbstractCollect seems to be in an intermediate state and that he made some fixes to make it work. MO remarks that an incomplete version of the new AbstractCollect proposal has been merged and that it can either be reverted to its old form or that IK make a PR with his fixes (whatever IK thinks is the best). The work on AbstractCllect will continue as the work on AbstractProcedure continues.

Decision: To create an AbstractResolution

Any other business

JA states that it's not entirely clear how the specific states are to be used even if he thinks the concept is good he would like to see a different implementation. JA made a comment on a PR which in some way was lost and would have liked to discuss the topic further as he believes its not been discussed enough.

RF states that it has been discussed on various issues and PR’s and that we need to be able to continue, even if we are not completely satisfied with the current implementation, as the feature is implemented and working.

MO agrees that the topic has been discussed in both issues and PR’s but that it would be no problem for JA to open a new issue to outline his proposal but that it would be complicated to consider something that completely changes the way that the states are handled.

Action: JA to open issue on specific states

Session 2

MO explains that the idea of the session is to discuss how to proceed with the work on the AbstractProcedure and that it would be good to be able to release a “pre” release with the parts of HWR that are considered to be stable.

Discussion how to proceed with AbstractProcedure

MS states that one way to look at the problem would be to ask what benefits AbstractProcedure would bring to the application. MO explains that from a MXCuBE3 perspective it would unify the way custom beamline scripts (Beamline Actions) and experimental methods like XRF and Energy scan are executed. These could if they have the same interface further be enqueued and executed by the MXCuBE queue system. MO also notes that the QueueEntry and the AbstractProcedure have a very similar API and one could perhaps be replaced by the other. JA comments that maybe AbstractProcedure could be phased in for the experimental methods like XRF, Energy scan and Collect to then be interfaced with the queue. Both MS and RF notes that it might be quite complex and maybe not even desirable to replace QueueEntry with AbstractProcedure. JA notes that it would maybe be beneficial to separate the two for when/if the queue is reworked. It's agreed that AbstractProcedure would be introduced for all the experimental methods such as Collect, Energy scan and XRF. MO will create an issue where the affected classes can be listed and MO and AB will look at using AbstractProcedure for XRF and Energy scan as a start.

Decision: MO to update the AbstractProcedure issue with a list of affected classes
Decision: AbstractProcedure does not replace QueueEntry but will be used by it
Decision: MO and AB will start with using AbstractProcedure for XRF and EnergyScan

Discussion on creating a pre release

MO wonders what the participants think about creating a release with the features that is considered to be stable. Both to communicate within the community which parts that can start to be used but also to parties interested in the project. MO further asks what such a version could be called and RF suggests “pre-alpha”. RF comments that it would be desirable with a stable version of AbstractDetector and Diffractometer before the list of stable classes are made.

It's agreed that it would be beneficial with a release containing a subset of API wise stable classes. It’s decided that the list will be established for the June meeting and that the camel to snake case conversion would be the last activity done before releasing the “pre-alpha” release.

Decision: To release a version containing a subset of classes with stable API. The list of classes to be included is to be made in association with the June meeting.