MXCuBE Code Camp meeting

September 22-23, 2022

Participants:

Marcus Oscarsson, Antonia Beteva, Olof Svensson (ESRF)
Michael Hellmig (HZB)
Gruzinov Andrey (DESY)
Bo Yi (NSRCC)
Jacob Oldfield, Daniel Eriksson (ANSTOM)
Merghdad Yazdi (MAXIV
Nicolas Moliterno (LNLS)
Martin Savko (SOLEIL)
Rasmus Fogh (Global Phasing)

Apologies: Roeland de Boer (ALBA)

Agenda

The agenda shared before the meeting

Day 1 (22nd of September)
10h - 12h Introduction and presentations
13h - 15h Open discussion on testing - identify tests

Day 2 (23rd of September)
10h - 12h Discussion / Writing tests
13h - 15h Writing tests

Day 1

Presentation by MO -Introduction to hardware objects and testing

MO gave a presentation in two parts, where the first part gave a historical background of the MXCuBE project, a brief introduction to writing HardwareObjects and how the queue execution works. The second part was about testing.

XML to Yaml

MO mentioned that HardwareObjects can with mxcubecore be configured via YAML files instead of XML files. RF added that it should be relatively easy to switch to YAML configuration by inheriting the ConfiguredObject base class. JO and RF suggested that all new HardwareObjects should be written using ConfiguredObject.

AB later discovered that changing to YAML configuration would not be as easy as first thought as changing commonly used base classes such as AbstractActuator to ConfiguredObject would mean changing all objects inheriting AbstractActuator objects to YAML.

MO noted that a tool that converts XML configuration files to YAML could be written and the introduction of ConfiguredObject could be done in one go. It was also mentioned that this has to be done with care to consider backward compatibility.

DECISION: New HardwareObjects to be written using YAML configuration
ACTION: Write tool for XML -> YAML configuration file and find good moment for switch

Other Topics

Other topics that arose during the presentation were; Processing (OS) and VideoStreaming (MH, MY). It was decided these topics would be handled outside of the code camp most suitably between interested parts.

Testing

MO gave an introduction to testing meant to be used as a basis for discussion. The main focus was on where to best put testing effort. JO and DE pointed out that the current test coverage is quite low (around 6%). JO also mentioned that test coverage does not indicate what's being tested and that achieving a high coverage should not be a goal in itself. JO continued to explain that a higher coverage is of course desirable but reaching very high figures towards 100% might involve testing data models and might give little added value.

Discussions continued around continuous integration (CI) and MO mentioned that the old Travis based routine now has been replaced by GitHub Actions.

JO brought up that Liniting should be part of the CI and explained how they are using Black and Flake8 at ANSTO. The linting routines used for mxcubecore up until now have been using black (with the –safe option) and pylint. It's generally agreed that strictening the rules using a routine similar to that of ANSTO would be an improvement. However, this change needs to be done with care and a good moment to introduce this has to be chosen as it “touches” a large number of files.

ACTION: Usage of Flake8 and Black to be investigated

Docker images and python packages can be created and pushed to the GitHub docker registry and PyPI in CI (JO). JO shared their .rc and project files that contain a more up to date way of for instance packaging and managing dependencies. It was agreed that JO would make a PR with these.

ACTION: JO to make PR with project related files

Day 1 ended with identifying areas to test which resulted in the following categories:

- Control system abstractions in mxcubecore/Command for instance Tango, Bliss, EPICS and so on as its a central part that is frequently used.

- AbstractClasses that are untested or have little coverage

- Site specific code

An issue (https://github.com/mxcube/mxcubecore/issues/704) was created for communicating after the video session ended, especially intended for questions regarding getting pytest to run against the latest code base.

Day 2

The day started with a quick recap of the results from Day1. MO made a quick tutorial about getting pytest up and running and JO made a presentation about the pytest plugin for visual studio code.

The discussion around what to test continued somewhat and unitest.mock module was mentioned as a tool that could make otherwise hard to test code more accessible for testing.

OS spoke a bit about how he performs testing for the workflow engine using the various development servers available. It was agreed that the approach is interesting from a site perspective or for a more complete simulation environment and could be presented at a future meeting. However it was considered too much overhead/complicated for testing mxcubecore and would also go outside the boundaries of what we want to test.

The day ended with writing tests and a few people volunteered to write some tests,

- AB: AbstractFlux
- JO: EPICS motor related code
- RF: AbstractResolution
- MO: AbstractDetector and investigation of using the unittest.mock module for testing for instance ISPyB.