ISPyB Developers Committee

Wednesday 20180131 - minutes

JSON fields in Database

Agreed to introduce JSON fields in the database to allow preliminary or site/program-specific data to be stored. This would presumably hold across all techniques, and be decided by develiopers in each case. Ellipsoidal statistics and other STARANISO output would be one candidate for JSON. It is agreed that data that are both generic and stable should NOT be put into JSON, but should be modelled in (or moved to) the main schema. Some data types, like images or complex program output could be stored either compacted within the JSON or in linked-to files, depending on circumstances. According to Karl there is a JSON data type in new database versions, e.g. MariaDB > 10.1 (current ISPyB version is 10.2), which allows querying and indexing on subfields inside the JSON and can be accessed through virtual columns.

Modelling of multisamples

The introduction of multi-sample pins and in-situ crystallography requires remodelling of the sample handling in ISPyB. The concepts used so far have been

- SampleChanger - which allows you to change between Samples

- Container – which contains samples, but cannot be nested into another container (e.g. a Puck).

- Sample – which is the thing you mount on a goniostat

- BeamlineSample – which is the smallest unit that contains material of a consistent composition, but may contain multiple crystals (e.g. a drop).

- SubSample – characterised as a location (in pixels) within a BeamlineSample.

All of which does not fit well with having multipins or plates mounted. ESRF has implemented a plate in software as a sample-changer, so that you conceptually change samples when switching from one well/drop to another. According to Karl a drop is a BeamlineSample, and plates are treated as an organised combination of drop locations, so that each drop has an integer position ID (and wells as such have no place in the model).

The consensus is towards adding an additional column with a BeamlineSample ID. This, together with the (existing) LoopType field should allow identifying and addressing each BeamlineSample uniquely.

Alternatively one could add an extra layer of objects, so that a mountable plate or multipin corresponded to a MultiSample. One issue is whether sample changing etc. might be simpler with the second alternative, where a puck containing multipins was treated as a container of objects

which could be individually mounted, rather than a complex container of individual samples. Another is whether mesh scanning etc. is always limited to a single BeamlineSample, or whether there might sometimes be scanning across e.g. areas of a plate that contained different BeamlineSamples of different composition, which would need to be tracked.

It was decided to go over known and imagined use cases in detail before making a final decision.