ISPyB Strategy Meeting

EMBL-HH, February 12, 2020

Meeting Minutes

Participants

ALBA: Daniel Sanchez (via Web)

DESY: Jan Meyer,

DLS: Dave Hall, Neil Smith, Martin Walsh

ESRF: Alejandro de Maria Antolinos, Solange Delagenière, Stuart Fisher, Andy Gotz, Gianluca Santoni, Olof Svensson

EMBL-Hamburg: Gleb Bourenkov, Ivars Kapics, Marina Nikolova, Thomas Schneider

Elettra: Annie Heroux

GPhL: Gérard Bricogne, Rasmus Fogh

HZB: Manfred Weiss

Max IV: Jie Nan, Alberto Nardella (via Web)

SOLEIL: Tatiana Isabet (via Web)

Some names are clearly missing (or maybe even wrong?) – please help fill them in

Presentations

Overview Presentations

Introduction: Gleb Bourenkov

ISPyB Strategy: Gianluca Santoni

SynchWeb/DLS overview: Dave Hall

ISPyB at EMBL-HH MX: Gleb Bourenkov

Global Phasing' s viewpoint on the ISPyB Collaboration: Gérard Bricogne

ISPyB at Max IV: Jie Nan

ISPyB at SOLEIL: Tatiana Isabet

Technical Presentations

ISPyB Collaboration: Alejandro de Maria Antolinos

ISPyB Back end evolution: Neil Smith

There were some changes in presentation order etc. from the agenda, so this may be slightly inaccurate or incomplete.

For details of the talks please see the relevant presentation slides.

ISPyB strategy (Gianluca Santoni) – discussion highlights

Clarifications: EXI2 and MXCuBE3 will share some display modules, which is possible because the underlying technology is shared. Among the drivers for the development of EXI2 are better handling of the very numerous data collection items sometimes generated by Cryo-EM, better display of Cryo-EM quality metrics, better display of calculations for SAXS. Off-line data analysis is done by stand-alone applications, which can be triggered either externally or through EXI2.

Thomas Schneider strongly recommends using EXI2 for result tracking and monitoring only, and not for control or triggering applications.

Gérard Bricogne asks how multi-sweep processing (a long-standing GPhL request) can be handled as part of first line auto-processing, rather than as reprocessing.

Alejandro de Maria notes that SynchWeb can handle reprocessing, but because results are stored in ISPyB they are limited by the ISPyB data model - and Gleb Bourenkov raises the question of how much improvement over auto-processing the SynchWeb reprocessing will deliver. ESRF wants to support completely unlimited reprocessing applications with heterogenous data, and therefore have decided to use Mongo-DB for schema-free data storage.

SynchWeb/DLS Overview (Dave Hall) – discussion highlights

Clarifications Diamond Dewar identifiers are (e.g.) DLS-MX-0001, DLS-IN-0001, DLS-EM-0001 etc. They are tracked and guaranteed unique by the Diamond shipping system while passing through DLS. At present a Dewar can have multiple identifiers, if contents are used for more than one technique, but DLS is now reviewing whether Dewars, like pucks, should have just one ID and be available across proposals. (Dave Hall) . DLS will laser-etch identifiers on pucks, for free. It is possible for users to have Dewars kept on site between sessions, but this facility is not available for pucks and individual samples.– for logistical reasons. Crystal, sample and location information is kept in the ISPyB database and so is in principle accessible; populating is always done through an API. DLS supports data entry from Formulatrix format, while entry of e.g. CrystalDirect data would require writing a new API. The database itself is the only public general-purpose interface. Fault information, as well as RobotActions, are pushed to ISPyB from GDA; MXCuBE could do the same if desired. In response to a question from Gleb Bourenkov, it was clarified that beam energy history is populated from data collections only, and so does not track all energy changes.

Annie Heroux notes that Elettra get their samples by personal delivery (Dewar shipping is very expensive in Italy) and that the mandatory shipping information required in SynchWeb is both inappropriate and very cumbersome in that context. Stuart Fisher notes that there is a mechanism to bypass this on the Visits page (“Now you tell me!”).

Technical presentations and discussion

Initial presentations were given by Neil Smith and Alejandro de Maria (see slides for details). The discussion was general and wide-ranging; what follows is an attempt at restructuring.

Micro-services and databases

There was much discussion on the meaning and implications of using microservices. In the most obvious implementation you would split your database and have separate partial databases for each kind of microservice. Neil Smith argued strongly (and was never contradicted) that separate microservices could function even on top of a single database, and that there was no technical need to split the current database. Indeed the complex web of links and foreign keys between tables (which follow from the nature of the data to be stored, rather than from potentially reversible design decisions), as well as the need for combining data from multiple tables into a single view, suggested that split databases might require some non-trivial optimisation to avoid significant slow-downs.

The canonical purpose of microservices is to allow dealing with parallel access by large numbers of users (in the tens of thousands) that go beyond what a single database can handle; this is not a problem for ISPyB (Stuart Fisher). One technical reason for using microservices (within DLS) is to separate out the parts of the operation that do not depend strongly on the file system (shipping, registration and statistics, mainly) so that these can continue to function in the face of file system problems. For the ISPyB collaboration as a whole, the more urgent problem would be the lack of modularisation and of clear interfaces, and the difficulty of sharing components. In this respect microservices, while congenial to modularisation, are neither necessary nor sufficient to provide a solution to that problem.

There was a separate discussion about whether the database in its current state would serve as a basis for continuing collaboration. The general conclusion seemed to be ‘yes’, but there were some problems. It was agreed that the database contained a lot of dead and unused structure, as well as inconsistencies in naming and approach, and that a clean-up of these would be a clear improvement. It was more contentious to what extent the improvements would repay the necessary work. One view of recent work in the developers’ group was that discussions were going well, and that small and many larger changes had been agreed (or rejected) and dealt with. Another view was that the two sites were blocking each others’ proposed changes. A key instance of this was the recent proposal (from DLS) of renaming and reorganising the Sample tables in order to make them cleaner and more consistent, and to move away from the MX-specific naming that was quite misleading when used for non-MX techniques. This proposal would have entailed a very significant amount of work in renaming and refactoring existing code, and the benefits would accrue mainly to work with

non-MX techniques, which is not relevant for the ESRF. Neil Smith (for DLS) considered that this change proposal had now been discussed, judged unrealistic, withdrawn, and so could be considered as having been successfully dealt with. Nevertheless, the discussion shows that the two main development sites do not in practice share even their database schema. Possibly the biggest problem is not the differences in the database requirements (which could be dealt with), but the lack of shared code between the development sites. Making local changes to stay synchronised with the other partners can easily seem futile, if there is little prospect of your local changes ever feeding through into actual use.

Synchrotron harmonisation

Harmonisation of procedures between synchrotrons was discussed but seen as a difficult problem. Some categories of users (pharma companies and CROs) would like to ship a simple spreadsheet with sample codes and information on desired experiments, then download the final results by a uniform procedure (Gérard Bricogne). Gleb Bourenkov adds that sample logistics are less of a problem than data logistics. Thomas Schneider made the point that experimental control must remain with the synchrotron, so that it would not be acceptable to pass in actual experimental instructions (as opposed to information on preferred protocols, with associated parameter values) from an external source. Various obstacles to harmonisation were mentioned: User Office procedures are outside ISPyB control; data quantities are large; users differ between wanting to download everything at once, or to leave the data at the synchrotron; and there are also legal and commercial obstacles to harmonisation on both the client and synchrotron side.

Notebooks and processing protocols

DLS store Jupyter notebooks; calculations are only repeatable while data are still in store, so that once data are archived they must be unarchived first. For non-MX techniques there are fewer standardised calculations, but all experiment types use the general processing job tables (calculation, input files, output files, …). Martin Walsh notes that many techniques are moving towards standardised experimental protocols and mail-in services; the truly bespoke experiment types will simply ignore these facilities. As noted by Annie Heroux there are conflicting pressures between the need for standardisation in order to support (semi)automated protocols and structured data on the one hand, and the continuous changes in the underlying science on the other hand. Thomas Schneider notes that there are other efforts to standardise scientific vocabularies (ICAT, …) that could be used instead of *de novo* data modelling.

Strategic choices

Alejandro de Maria (for the ESRF) points out that, subsequent divergences notwithstanding, the Memorandum of Understanding had some fairly precise specifications of what should be covered by the collaboration (Database and back-end API), which techniques should be supported (MX, SAXS, and Cryo-EM), and even technical choices to be complied with until otherwise agreed (Java, JBOSS). The ESRF puts a high priority on making precise up-front agreements on the scope of future collaboration. A user survey reports that for the field of MX, SynchWeb and EXI are comparable in scope and quality, though each has some specific strong points. ESRF has some

reservations about the technologies that underlie SynchWeb. It has been estimated that, were the ESRF to switch to SynchWeb, it would require 18 person-months for the ESRF to reach the level of service that EXI provides at the moment, which is considered unrealistic. (Gleb Bourenkov opines that most of this time would be spent on SAXS, which SynchWeb does not support at the moment). It is proposed that a way forward would be for each group to continue development of its own implementation, free from the obligation to conform to the other, while a pilot proposal should be made on an entirely new, future application for both sites to use. In order to work, this would require:

- Precise up-front agreement on the scope of the project, major technical choices, and decisions about the adoption of existing standards (such as ICAT)
- Inclusion of the full code stack, including the user interface, within the collaboration.
- Empowering the developers to take decisions on technical questions
- Sufficient resources

Martin Walsh comments that different sites will unavoidably have different priorities, pressures, and requirements, and that any agreement must allow each site to cater for those.

Neil Smith (for DLS) proposed a model where the two main development sites try to work together on a limited area (to start with), coupled with splitting the monolithic backend and the adoption of microservices (see above). DLS was particularly interested in the possibility of basing new developments on GraphQL. The idea is that EXI and SynchWeb can then gradually change towards a more modular, common, and shareable architecture while keeping their existing applications working, thus avoiding a ‘big-bang’ solution that would take a very large effort and require parallel development and support for old and new applications for a considerable time to come. This also matches the approach currently taken at the DLS to develop away from older technologies (such as Marionette). The development should start with agreement on not just software technology, but also specifications and coding standards. It is expected that the first of the new separated-out services could be in production after around two years.

Participation from additional contributors

Several participants made the point that in order to have a successful collaboration you need to have multiple contributors that share both the effort and the resulting code. Gleb Bourenkov pointed out that his group had wanted for some time to contribute resources to ISPyB to deal with urgent needs for support of serial crystallography, but that the barriers to entry (at the time and still today) made this impossible. It was made clear that the crux of the problem was not in a lack of help or openness, and that minor changes and fixes could indeed be made without too many problems. However, (1) any major effort would require changes that might potentially break things for other participants, and (2) ISPyB was at the moment too complex, monolithic, and opaque to make this kind of work realistic outside the two main development sites. The database schema is highly complex, interconnected, and contains many unused items. Some constraints are explicit, some are implicit; the two main sites use different sets of fields, possibly in different ways; and there is no

single set of documentation (except for the actual code) that would make it simple to decide which changes you would need to make, and what their likely consequences might be. As things stand, a major new development by third parties would require a prohibitive amount of effort in order to develop a detailed understanding of working practices in both of the existing implementations, and/ or would run a high risk of ultimately not being adopted. Andrew Gotz underlines that both EXI and Synchweb share the same problem of being complex, monolithic, with too much undocumented code, and thus too hard to understand or start contributing to for any group without major resources and experience - which means that even standardising on one of the two existing implementations would not solve the main problem, so that some major new developments were therefore necessary. He saw the top-priority need as the creation of a new prototype for the very different applications of the future, and welcomed microservices and GraphQL as possibly being part of the solution.

Final discussion and conclusions

- All participants agree to go forward together and to try and make the collaboration work.
- The Collaboration intends to work towards a common framework, with as much shared code as possible, and try to modularise.
- The two main development centres will stay with their separate implementations and work towards increased sharing; there are no plans to abandon either implementation or to start a completely new alternative for now.
- The immediate target is to develop a shared backend based on the existing database.
- The developers are tasked with making a working prototype of a shared back-end API for one precisely defined domain. In the process they will pilot a framework for collaboration and a set of technology choices, design rules, good practices, and specifications. This will demonstrate that the process works, that the result is useful, and establish a model that can be used (possibly after some tuning) to proceed to a complete back-end.
- The prototype should be demonstrated at the next ISPyB/MXCuBE at ALBA in June 2020.
- The technology choices should be based on agreement between the participants. Based on the poll of participating groups (and in the absence of major new arguments) that would mean MariaDB, Python, and REST / web services.
- The choice of domain to work on initially is still to be decided. It should preferably be something relatively simple, and it would be highly desirable with a topic (or topics) of equal interest and usefulness for both main development sites. Shipping was mentioned as a possibility by the ESRF, but this is less interesting to DLS; changing the sample part of the model is less interesting to ESRF (and anyway that proposal has been withdrawn by DLS); serial crystallography is both topical and interesting to many groups, but is rather complex and would require non-trivial scientific thinking in addition to the software development.

- One view of the upcoming work, promoted by Martin Walsh, emphasised producing useful results that would add value at both ESRF and DLS, and exercising managerial oversight and external review. Another view, promoted by Gleb Bourenkov, emphasised quality and team- and process-building over immediate results, concentrating on making a good base for the continuing joint development of ISPyB, and leaving the developers to deliver a working prototype to be evaluated.
- Gleb Bourenkov said that EMBL-HH would base its decision as to whether to dedicate additional resources to ISPyB on the outcome of this prototype-building exercise.