Minutes EDNA developers meeting
February 18^{th} - 19^{th} 2008 at Synchrotron Soleil, France

Presents: Alun Ashton, Gleb Bourenkov, Sandor Brockhauser, Gerard Bricogne, Marie-Françoise Incardona, Peter Keller, Pierre Legrand, Karl Levik, Andrew Leslie, Ezequiel Panepucci, Romeu Pieritz, Harry Powell, Lucile Roussier, Olga Roudenko, Bill Shepard, Olof Svensson, Andrew Thompson and Meitian Wang

Agenda:

Monday February 18^{th}, 14:00 – 1800 :
- Demo
- MOSFLM indexing plugin and data model
- XDS plugin and data model
- Generic data model

Tuesday February 19^{th} 9:00 – 13:00 :
- Generic data model, continuation
- RADDOSE use case
- Project agreement
- Next meeting

1. Introduction and Demo

Olof started the meeting by giving a short recapitulation of the current status of the EDNA development together with his view of the goals of this meeting, see Appendix 1. He then made a demo of the current implementation of the testing framework.

2. MOSFLM specific data model and plugin

Olof then briefly presented the current status of the MOSFLM specific data model and how the Python data bindings are created from the data model. Andrew asked if it was easy to redo the Python bindings if the data model was changed. Olof demonstrated this by changing the name of an attribute of the data model, regenerating the data binding and demonstrated that the change did make the EDNA test suites to fail.

3. XDS plugin

Sandor then presented the work made by him and Pierre during the two last days of the week preceding this meeting. As this work was based on existing python code written by Pierre no specific data model for XDS was developed. Sandor presented a working example of the code running XDS. 4 XDS specific plugin has been created (1 abstract and 3 final plugins as below):

- XDSv01: generic abstract class for specific XDS plugins
- Img2XDS: using Pierre's PyXDS code as external application to be started up, it convert the "generic indexing input DataModel - namely a list of image files" into XDS

specific indexing input data object.
- XDSIndexing: using the XDS specific indexing input data object, runs xds as an external application and converts its output to an XDS specific indexing output data object.
- XDSIntegration: using the XDS specific integration input data object (that is the same as the XDS specific indexing output data object), runs xds as an external application and converts its output to an XDS specific integration output data object.

For all 3 XDS specific final plugins, separate TestCaseUnit python code has been created with:

- Separate test configuration for each
- Separate test input
- Separate test reference output
- Each test case code starts up the corresponding plugin with its own test input, loads the reference output object, and compares with the actually generated output object (Note that optionally, the test code is also able to store the actually generated output as a reference output for later runs.)

Even the XDS specific DataModel has not been expressed in Enterprise Architect generated XSDs, its elements are created and used within XDS specific indexingInputObjects, indexingOutputObjects = integrationInputObjects, and integrationOutputObjects. The comparison of both the binary and String elements of such complex objects are done by the Testing codes.
Both Sandor and Pierre pointed out that it was not evident how to write a plugin for the current EDNA implementation due to lack of documentation. They have therefore written a report with a list of suggestions for improving the EDNA infrastructure. This report was welcomed but was not discussed further as the main topic for this meeting was the EDNA data models. The report will be transmitted to Olof and Marie-Françoise. Olof acknowledged the lack of documentation but pointed out that due to lack of resources, and due to the fact that we are in a very early stage of development of the plugin hierarchy which hence is a moving target, it is difficult to develop documentation at this stage since this would inevitably remove resources for development, threatening the timescale of the prototype. He also pointed out that if Sandor and Pierre had been listed as developers, and participated in the discussions on the edna-dev@esrf.fr mailing list, many of their difficulties would have been if not removed at least reduced. Olof therefore asked Sandor if he would like to join the EDNA development team however Sandor again declined.

Gerard and Olof agreed that in spite of all the difficulties it was after all a significant achievement that Sandor and Pierre managed in only two days to write a plugin that can run XDS. Olof also suggested that if firm commitments of new developers is confirmed much more effort will be devoted to documentation and training. However, with the current resources and the goal of a working prototype for June 2008 documentation will be kept at an essential but minimal level.

4. Generic data model : Indexing required input

The rest of Monday afternoon and a part of Tuesday morning was devoted to the discussions around a generic indexing required input data model. See Appendix 2 for the status of the model after these discussions. The most important conclusion of these discussions is that we will try to use as much as possible the IMAGE-CIF definitions in general and in particular for the detector and goniostat geometries.

5. Generic data model : Indexing required output

It was again agreed to use as much as possible the IMAGE-CIF definitions. See Appendix 3 for the results of these discussions which were started on Monday afternoon and finished on Tuesday morning.

6. Generic data model : Integration required input

As we felt that we had gained a certain momentum and well advanced the generic data model, we took the decision to deviate from the agenda and continue with the integration part of the model instead of discussing RADDOSE and the project agreement.

The discussion was initially very smooth, resulting in a first suggestion as presented in Appendix 4. We agreed that if possible this data model would not only be used for integrating the reference images, but also for the collected data set. Therefore the input could be lists of images, where the integration of each list of images would result in a single resulting file of integrated reflections.

At this moment Olof, Marie-Françoise and Romeu pointed out that the data model started to resemble the “experimental data model” (see spike report page 26). Olof therefore suggested that all the experimental data was attached to each image individually, see Appendix 5.

This suggestion sparked a vivid discussion which had to be interrupted at 13:00 without any clear consensus. On one hand Olof, Marie-Françoise, Romeu and Gleb argued that this is a true generic data model which would allow us to model future experiments where, for example, the wavelength varies from image to image. On the other hand Pierre, Gerard and Andrew argued that this proposition is too generic. They feel that this data model will be difficult to use in practise and that there's no need for EDNA to implement anything else for modelling data collections as they are made today, i.e. as oscillation wedges with constant wavelength, detector and gonio-meter settings. This discussion was a déjà vu of the many discussions we had during the spike and since we never managed to find an agreement during the spike it was not surprising that we didn't find any agreement in this meeting.

Olof concluded the discussing by deciding that the data model for the prototype will be based on the one suggested in Appendix 4 and not the one in Appendix 5. Although he feels that this decision could have impacts on the modularity of EDNA in the long term he recognises the need of EDNA members to have a data model to which they feel familiar.

After the meeting, in the discussions during the lunch, Olof suggested that a possible solution could be to introduce more layers, including a completely generic data model layer:

- 3rd party programs layer (e.g. MOSFLM specific)
- User-friendly Generic MX layer (wedges, sub-wedges etc., see BioXDM style data model in spike report)
- Generic SR experiment layer (e.g. experimental points, see Experimental-Strategy Points data model in spike report)

The motivation for the completely generic layer is not only for not limiting EDNA to future, today unknown, experiments but would also allow EDNA to be applied to experiments in other synchrotron radiation scientific domains than macromolecular crystallography.

However, **for the moment this layer is just a thought and will not be used in the EDNA prototype**.

7. Next meetings

Due to lack of time, this discussion took place at the coffee after the lunch with a subset of the meeting participants:

We agreed that there's no time for arranging a developers workshop before June 2008. Although such a workshop would be very useful, the preparation of it would remove development resources necessary for accomplishing the EDNA Characterisation prototype for June 2008. Marie-Françoise said that we can't expect the prototype (by definition) to be of a high quality in terms of development and documentation. She suggested that developers working on the prototype are more than welcome to give any constructive feedbacks in order to prepare a user guide to be written after the prototype demo.

We also agreed that we should arrange an EDNA full meeting in June 2008. The goal with this meeting would be to present the prototype and take decisions concerning the developments post-prototype. No decision was taken concerning the date, which could either be around June 15th or June 30th. The week of June 23rd is ruled out due to a BioXHIT training course to which Andrew, Olof and Sasha will participate. The venue should probably be in the UK, either in Cambridge or at Diamond.

Appendix 1 : Presentation

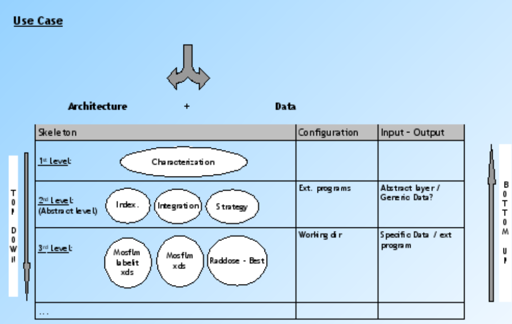
EDNA

Status 20080218

November 21st - 22nd workshop

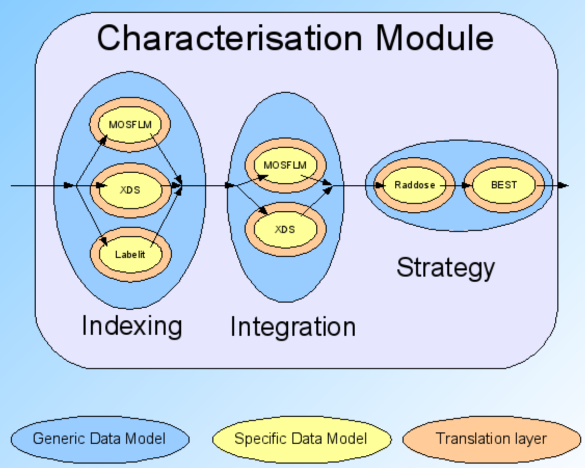
- Introduction to the AALib framework :
  - Plugin architecture
  - Thread management and thread safety
  - Running both in Python and in Java (via Jython)
- How to go from Use Case to code :
  - Top down changed to bottom up
  - We first concentrate on developing modules / plugins for running MOSFLM, BEST, XDS etc
  - Data models will be program specific
  - We then combine these modules in more complex modules, for example indexing using MOSFLM or XDS, characterisation etc.
  - Data models will be generic

From Use Case to Module



Development status

- EDNA server www.edna-site.org:
  - Wiki
  - Bugzilla
  - Subversion : still on SourceForge – to be moved to edna-site.org
- EDNA kernel:
  - Configuration
  - Plugin class hierarchy
- Testing framework:
  - Test cases (unit tests, plugin execution)
  - Test suites
- Python / Jython
- Data model:
  - Enterprise Architect
  - generateDS



Goals with this meeting

- Use cases / data models :
  - Validate MOSFLM indexing
  - Advance XDS indexing
  - Advance MOSFLM integration
  - Advance RADDOSE
- Global data model :
  - Generic indexing: merging of MOSFLM and XDS indexing results (non-compatible parameters, e.g. mosaicity)
  - Generic characterisation : required input
- Project agreement – licence for prototype
- Next meetings :
  - Workshop ?
  - Full meeting June 2008 : presentation of prototype

Appendix 2 : Generic indexing required input :

- Definition of the laboratory coordinate frame :
  - http://www.iucr.org/iucr-top/cif/cifdic_html/2/cif_img.dic/Caxis.html
- Detector :
  - Same IMG-CIF definition as above
    - Detector coordinate frame
    - Distance
    - Beam centre
    - Two theta
- Wavelength
- Goniostat
  - Orientation : http://www.iucr.org/iucr-top/cif/cifdic_html/2/cif_img.dic/Cdiffrn_scan_axis.html
- List of images:
  - Directory / filename UTF8
  - Oscillation start + end

Appendix 3 : Generic indexing required output :

- Solution container :
  - Orientation matrix
    - Orthonormal U-matrix in imagecif coordinate / reference frame
    - Brookhaven orthonormalisation convention – used by PDB
    - Reciprocal lattice units 1/Ångströms
    - Matrix type (Peter)
  - Unit cell parameters
    - Ångströms – degrees
    - MMCIF naming convention
  - Bravais lattice
    - IT Bravais lattice symbol
  - Distortion factor
    - The Polish one

- Plugin ID / ... info to be retrieved from EDNA persistence .... after prototype...
- Plugin
  - id
  - Name
  - Input
  - Working directory
- Rms error for the triclinic solution
- Selected solution :
  - Solution
  - Parameters to judge success :
    - Rms error
      - Spot positions mm
      - Phi angle degrees
    - Percentage of spots indexed
      - Definition whether a spot is considered indexed
    - Shifts true / false
      - Detector geometry
      - Goniostat geometry
- List of all possible solutions
  - Solutions
- List of warnings / errors
  - Errors
    - Images not found
    - Images not readable
    - No spots found
    - Too few spots found
    - Indexing failed – no solution found
  - Warnings

Appendix 4 : Generic Integration Required Input   first suggestion

- Lists of images
  - For each list of images :
    - Goniostat geometry
    - Beam properties :
      - Divergence
      - Polarisation
      - Wavelength
    - Detector geometry
    - For each image :
      - Directory / filename UTF8
      - Oscillation start + end
      - Exposure time
- Generic indexing solution
- Mosaicity

Appendix 5 : Generic Integration Required Input   second suggestion

- Lists of wedges
  - Wedge with constant wavelength
- For each image :
  - Directory / filename UTF8
  - Oscillation start + end
  - Exposure time
  - Beam properties :
    - Divergence
    - Polarisation
    - Wavelength
    - (Optional : Intensity)
  - Detector geometry
  - Goniostat geometry
- Generic indexing solution
- Mosaicity