

DRAFT

EDNA Prototype Report

Abstract

EDNA will be a new generation environment for ODA (Online Data Analysis), in particular for the automation of the collection and processing of X-ray diffraction data from macromolecular crystals. This report describes the EDNA prototype that will serve as a base for future EDNA developments. The prototype implements the Use Case “Crystal characterisation taking into account radiation damage” by using MOSFLM for indexing and integration, RADDPOSE for radiation damage estimation and BEST for data collection strategy calculations.

The EDNA prototype described in this report was demonstrated in a BioXHIT meeting held at the ESRF (2008-06-26 - 2008-06-27). The minutes of this meeting are attached to this report.

Table of Contents

1 Introduction.....	2
2 The EDNA Prototype.....	3
2.1 Goals.....	3
2.2 Prototype features.....	3
2.3 Data model.....	4
2.4 Plugins and architecture.....	4
2.5 Configuration.....	4
2.6 Prototype execution.....	4
2.7 Distribution and licence.....	4
3 Development tools and testing framework.....	5
4 Conclusions of the prototype development.....	5
4.1 Implementation.....	5
4.2 Development.....	5
4.3 Project structure and collaboration.....	5
5 Minutes EDNA Prototype Demonstration meeting	
June 26th – June 27th 2008.....	6
5.1 Overview of the meeting.....	7
5.2 Prototype requirements (Deadline end of July).....	7
5.3 After the prototype (August - October).....	8
5.4 New plugin submission / development procedure.....	8
5.5 Next meeting – Soleil.....	8
Appendix 1 : The EDNA project and the prototype history.....	9
Before February 2007.....	9
February 2007 – June 2007.....	9
June 2007 – October 2007.....	9
October 2007 – June 2008.....	9
Appendix 2 : The EDNA prototype generic data models.....	10
Appendix 2.1 : The Common data model.....	10
Appendix 2.2 : The Characterisation data model.....	14
Appendix 3 : How to run the prototype.....	16

1 Introduction

The EDNA project is an international collaboration between Synchrotron Radiation (SR) facilities, laboratories and companies. For more information about the EDNA project and its history see Appendix 1.

The EDNA prototype described in this report is an important mile-stone in the EDNA development plan, defined in the “spike” report¹. The deadline for the prototype has been set to end of June 2008 in order to make this report a BioXHIT² deliverable.

This report contains a description of the EDNA prototype and a summary of the developments behind it. The first part of this report is a short overview of the prototype. The second part consists of a list of Appendices that contain more detailed information.

1 See http://www.edna-site.org/wiki/index.php/EDNA_Documents

2 See <http://www.bioxhit.org>

2 The EDNA Prototype

2.1 Goals

The main goal of the EDNA prototype is to implement the Use Case “Crystal characterisation taking into account radiation damage”³. In short this means running the following steps:

- Indexing with MOSFLM
- Integration with MOSFLM, produce input files for BEST
- Estimation of radiation damage using RADDDOSE
- Strategy calculation using BEST resulting in multi-subwedge strategy

Initially the goal was to perform indexing with both MOSFLM and XDS, however the objectives were reviewed (See Appendix 1).

2.2 Prototype features

The prototype has the following features implemented:

- Execution plugins : BEST, RADDDOSE, MOSFLM indexing, MOSFLM integration and MOSFLM generation of image with predicted spots.
- Control plugins : Indexing, Generation of images with predicted spots (parallel execution), Integration (parallel execution), Strategy and Characterisation.
- The characterisation plugin can be launched with one or several reference images, compared with DNA which can only be used with two reference images.
- It is possible to provide all relevant parameters needed for radiation damage estimation by RADDDOSE and BEST in the XML input for characterisation.
- In case of radiation damage parameters available or in case of anisotropy, the strategy produced by BEST consists of a list of sub wedges. The BEST plugin can be configured to produce single sub wedge strategies.
- All plugins create a separate directory where the XML input, XML output and working files are stored. If the plugin is a control plugin, the execution plugins create separate directories in the control plugin working directory. This hierarchy of working directories make it easy to find for example log files from the third-party programs, see Figure 1.

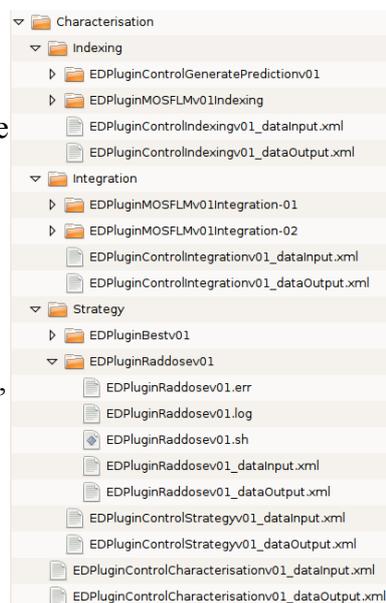


Figure 1: Log file hierarchy

³ See <http://www.edna-site.org>

2.3 Data model

An important part of the EDNA project, if not the most important, is the development of a generic data model for passing results from one program as input to other programs, for example using XDS for indexing and MOSFLM for integration. The prototype does contain three levels of data models:

- Common : data model containing the most generic definitions like detector, space group etc.
- Characterisation : this data model contains definitions of (generic) indexing, integration and strategy inputs and results.
- Specific : each plugin in charge of the execution of third-party programs (i.e. MOSFLM, RADDPOSE and BEST) has it's own specific data model.

The EDNA prototype data models are described in detail in Appendix 2.

2.4 Plugins and architecture

EDNA is a modular system based on a plugins factory mechanism and multi-thread management (AALib). It uses a programming language that allows integration with the python and the java worlds (Python/Jython).

EDNA consists of a kernel and a set of plugins organized in a classes hierarchy. This logical organization warranties the possibility to introduce easily new plugins that will naturally inherit the main existing plugins. More precisely: execution plugins that are responsible for the execution of 3rd party softwares and control plugins that are responsible for the data flow and the invocation of the execution plugins mentioned above.

2.5 Configuration

It is important to be able to configure the EDNA software. The aim is to set the required technical parameters to be able to launch properly all the plugins which are part of EDNA including the controller plugins and the execution plugins mentioned above.

For instance, it should be possible to define the 3rd party software that is to be launched for a particular step (e.g. choosing MOSFLM and/or XDS for the indexing when this feature will be available) and in a more general way to set the required parameters for an execution plugin to be launched (working directory, 3rd party software paths, etc...)

The current way to configure EDNA is an editable xml configuration file. This could be wrapped up in a GUI to be presented and updated by a user.

2.6 Prototype execution

The EDNA prototype is implemented as a Python class called EDApplicationPrototype. In order to facilitate the startup of the prototype a script is also provided called edna-prototype. See Appendix 3 for instructions on how to run the prototype.

2.7 Distribution and licence

The prototype will be distributed to EDNA members who are willing to test it. The deadline for the prototype package is the same as for the final prototype report : before the end of July 2008.

The licence for the EDNA prototype is GPL v3.

3 Development tools and testing framework

In order to have a fast and reliable development process, a set of development tools have been put in place in addition to the development of a specific framework to develop and test EDNA. This will help to ensure the robustness of the package when adding or re-factoring any EDNA feature:

- Development Environment: Eclipse
- UML Data Modeling: Enterprise Architect
- Source and Bugs managements: Subversion, Bugzilla
- Dedicated web server
- Data classes code generation from UML (XSD data binding)
- EDNA Test Framework: EDNA kernel tests, plugins unit tests, plugins execution tests (for both control plugins and execution plugins). See Appendix XX

4 Conclusions of the prototype development

4.1 Implementation

The implemented prototype has fulfilled its goal : given one or several reference images, it returns a full characterisation that takes radiation damage into account and provides a multiple sub-wedge data collection strategy. The implementation based on AALib seems to be robust.

4.2 Development

The EDNA development environment (EDNA kernel + development tools, see Section 3) is now sufficiently mature to develop the first EDNA version. The data model development using Enterprise Architect and the corresponding API generation works very well. Thanks to the plugin structure and testing framework new developers should be included in the collaboration without any difficulties. The Use Cases as defined on www.edna-site.org were very useful for developing the prototype.

4.3 Project structure and collaboration

The EDNA project structure still needs to be finalised by finishing the project agreement. We also need to agree on a license for distributing future EDNA developments.

5 Minutes EDNA Prototype Demonstration meeting

June 26th – June 27th 2008

Presents :

Alun Ashton (Diamond, UK), Gleb Bourenkov (EMBL Hamburg, Germany), Sandor Brockhauser (EMBL Grenoble, France), Mirek Gilski (Center for Biocrystallographic Research, Poznan, Poland), Elspeth Gordon (ESRF, France), Marie-Françoise Incardona (ESRF, France), Andrew Leslie (MRC LMB, UK), Karl Levik (Diamond, UK), Andrew McCarthy (EMBL Grenoble, France), Sean McSweeney (ESRF, France), Harry Powell (MRC LMB, UK), Olof Svensson (ESRF, France), Takashi Tomizaki (SLS, Switzerland), Johan Unge (MAX lab, Sweden)

Partly presents :

Romeu Pieritz (ESRF, France), Sasha Popov (ESRF, France), Darren Spruce (ESRF, France)

Agenda :

Thursday June 26th :

14:00 - 18:00 : Prototype demonstration and discussion

- Demonstration
- Data model
- Implementation
- Configuration

18:00 - : Executive committee meeting with project manager

Friday June 27th :

9:00 – 12:00 : Future developments

- Improvements within the scope of the prototype
⇒ final requirements for the prototype, deadline end of July
- Generic data model + XDS
- Deploying EDNA on beamlines : MOSFLM post refinement, POINTLESS, MTZUTILS and SCALA plugins
- HTML pages
- Project agreement, licence, distribution
- Next meeting (December 2008?)
- Longer term developments (if time allows) : KAPPA (through STAC), GRID, MAD / SAD data collection strategy optimisation

5.1 Overview of the meeting

Olof opened the meeting with a short presentation describing the meeting agenda and the goals with the meeting. He then made a short demonstration of the prototype by running it on different sets of input images, illustrating successful runs with single or multiple images, including parallel indexing of several images, and explaining the organisation of the generated output files. This will be remembered as a very hot event due to broken air conditioning and the unusually hot weather in Grenoble that day...

After coffee the meeting moved to a new meeting room with air conditioning. Various aspects of the prototype were discussed (data model, logs, error handling etc). We started on Thursday to discuss as well the future of the project, and the meeting was closed by noon on Friday after having agreed on the points outlined in the following sections.

5.2 Prototype requirements (Deadline end of July)

We agreed on the following set of items that should be implemented in the prototype :

- **Add default RADDPOSE structure** : If no structure information is available, then EDNA should use a default structure (Gleb will provide one).
- **Generation of prediction images** : demonstrated but not fully implemented, should be a part of the final prototype.
- **Executive summary** : The prototype should give the essential results in the form of an executive summary. Elspeth will provide help on what should be present in this summary.
- **Integration I / sigma** : The integrated intensity I and I/sigma as function of resolution should be a part of the integration result.
- **Missing items in data model** : The following parameters were missing in the data model presented and should be part of the final prototype data model :
 - Ranking resolution, ring current
 - Diffraction plan: Resolution, I/sigma, multiplicity and completeness as input to strategy
 - Max phi speed, min exposure time, min osc width (should also be part of configuration)
- **Prototype report** :
 - Add why new development (from spike report)
 - Deployment / installation instructions
 - Annotate the UML diagrams
- **Licence** : It was suggested to release the prototype under LGPL3 with the same modifications that were introduced for the CCP4 licence to make it valid under British law..

5.3 After the prototype (August - October)

- **Tests of the prototype on beamlines** : Sandor and Alun agreed to make sure that the prototype will be tested on ESRF and Diamond beamlines respectively and would coordinate feedback.
- **Small XDS plugin meeting in September** : As Gleb and Sandor would like to help developing the XDS plugin, it was suggested that a small meeting is organised at the ESRF in the period mid-September till the end of September. The developers documentation (see below) will not be ready for this meeting.
- **Developers' documentation** : It was agreed that before developing any new functionality in EDNA, Marie-Françoise and Olof should write up developers' documentation, in particular :
 - “How to write an EDNA plugin” documentation
 - Data model implementation and development
 - Plugin hierarchy etc.
 - Coding convention, development tools etc.
- **Plugin Workshop by mid October** : Limited to max 10 people who are expected to be contributing to EDNA.
 - How to develop MOSFLM post refinement, LABELIT (ask Brookhaven to do this plugin), POINTLESS, SCALA, ISPyB (it was suggested that Karl could do this) and DENZO (Mirek).
 - As the BioXHIT funding has stopped travel money must be taken from individual institutes budgets.

5.4 New plugin submission / development procedure

We discussed briefly a work flow for submitting a new plugin for EDNA :

- Specifications – related Use Case
 - ⇒ Validation
- Coding including tests
 - ⇒ Code review

5.5 Next meeting – Soleil

We agreed to suggest to Andrew Thompson if Soleil could organise the next full EDNA meeting. Possible dates (in December):

- 8th 14:00 -> 9th 12:00 or
- 15th 14:00 -> 16th 12:00

Appendix 1 : The EDNA project and the prototype history

Before February 2007

The DNA (automated collection of data) collaboration has successfully developed a software system capable of automatic screening, automatic ranking and automatic data collection with on-line data reduction (integration and scaling). For more information about the DNA project see the following links: [DNA web site](#)⁴ and [DNA article](#)⁵.

The initial collaboration divided the DNA system into a set of modules, where one developer was responsible for one module. This way of collaborating was initially successful but has made it difficult for new developers to enter the project. Moreover, the overall goals of DNA have gradually been extended so that the initial inflexible design had to be “stretched” in order to implement new features. This has led to code which is difficult to maintain and new features became more and more costly in terms of development efforts to implement. Therefore a decision was taken to start a completely new collaboration with a new way of collaborating and a completely new design.

February 2007 – June 2007

The EDNA project started under the name of “DNA 2.0” in the DNA full meeting at Diamond in February 2007. For more details about the period up to June 2007 see http://www.edna-site.org/wiki/doc/DNA20SpikeReport_20070920.pdf.

June 2007 – October 2007

The main activities during this period were the preparation of the Use Cases and the Project Agreement. On October 23rd a “Project Agreement Meeting” in which the name of the project was changed to “EDNA”. For more details about this meeting see http://www.edna-site.org/wiki/doc/FinalProjectAgreementMinutes_20071023.pdf.

October 2007 – July 2008

After the Project Agreement meeting the work on the EDNA prototype (as defined in the Spike report) started.

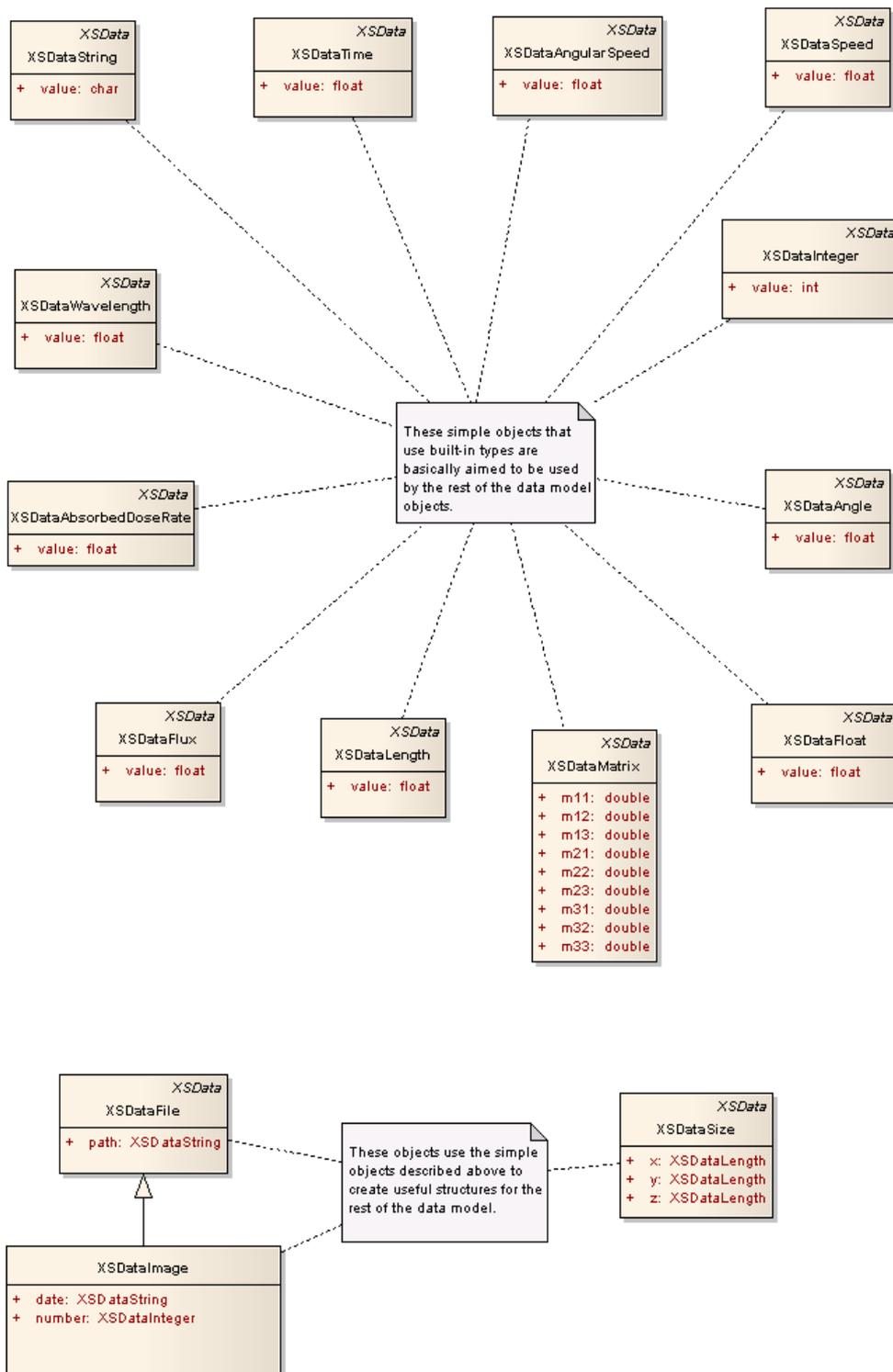
- In November the first EDNA developers workshop was organised with the main goal of being an introduction to Aalib. There are four documents available from this workshop, see http://www.edna-site.org/wiki/index.php/EDNA_Documents.
- In February a developers meeting was organised at the Soleil synchrotron, France. The main goal of this meeting was, a part of a general discussion around the current status of the project, the elaboration of a generic data model, see : <http://www.edna-site.org/wiki/images/4/40/MinutesMeetingSoleil20080218.pdf>.
- Olof was absent from March 5th till April 10th. Therefore the decision was taken to reduce the scope of the prototype by removing the implementation of XDS plugins and a fully generic data model.
- The prototype demonstration meeting which will take place at the ESRF from June 26th to June 27th.
- The prototype was released to EDNA members on 2007-07-31.

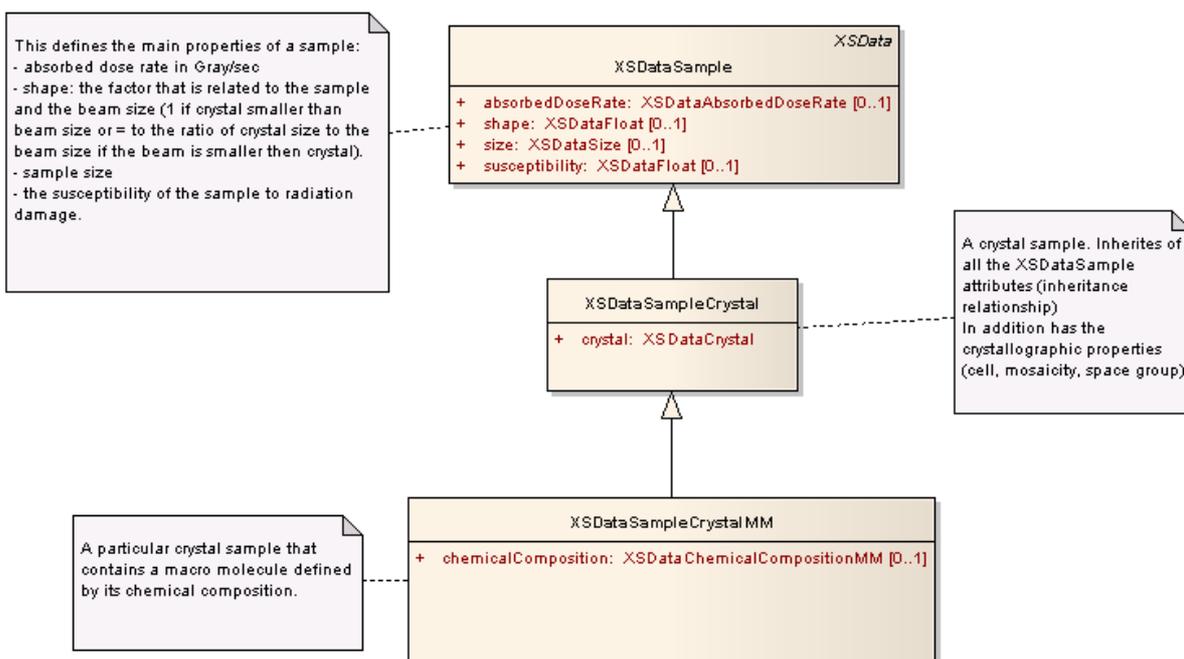
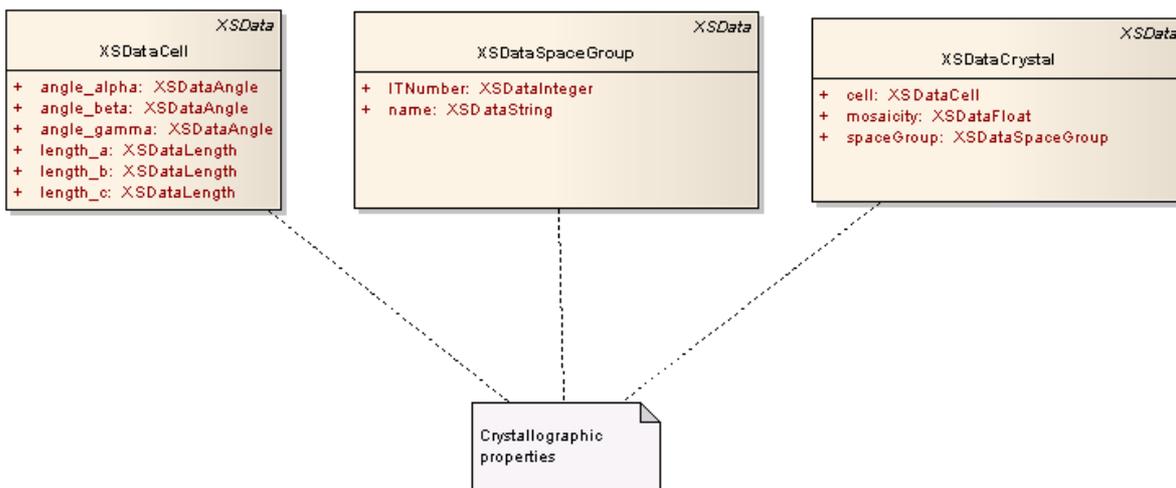
4 <http://www.dna.ac.uk>

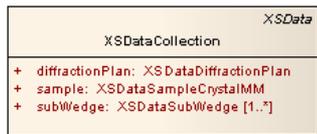
5 <http://journals.iucr.org/d/issues/2002/11/00/ba5026/index.html>

Appendix 2 : The EDNA prototype generic data models

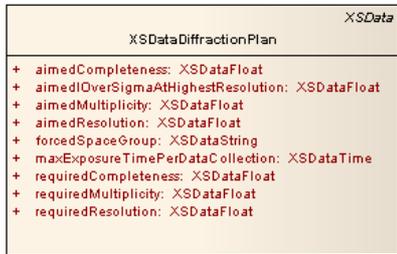
Appendix 2.1 : The Common data model







The data collection carried out or to be carried out of a particular sample with specific user inputs defined by the diffraction plan.



This object contains the main properties a user can parameterize for a crystal characterisation.

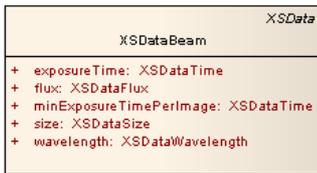
- the aimed* parameters are the parameters that a user would like to reach for a Best run.
- the required* are not yet used
- the maxExposureTimePerDataCollection should be as the maximum time to be reached for a data collection duration.
- the forcedSpaceGroup: option to force the space group of the indexing solution



A subwedge is defined as a list of images that has been collected or to be collected with some particular experimental condition. If it concerns some images to be collected, the image list is empty.



This object encapsulates all the physical properties of an experiment instrumentation. i.e.: Beam, detector, Goniostat

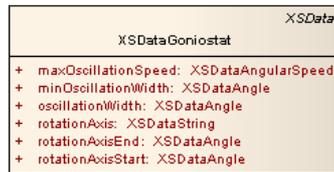


This object contains all the properties related to the beam:

- the exposure time (sec)
- the flux (photons/sec)
- The minimum exposure time that could be carried out on a beam line (sec)
- The size of the beam (mm x mm)
- The wavelength (Å)

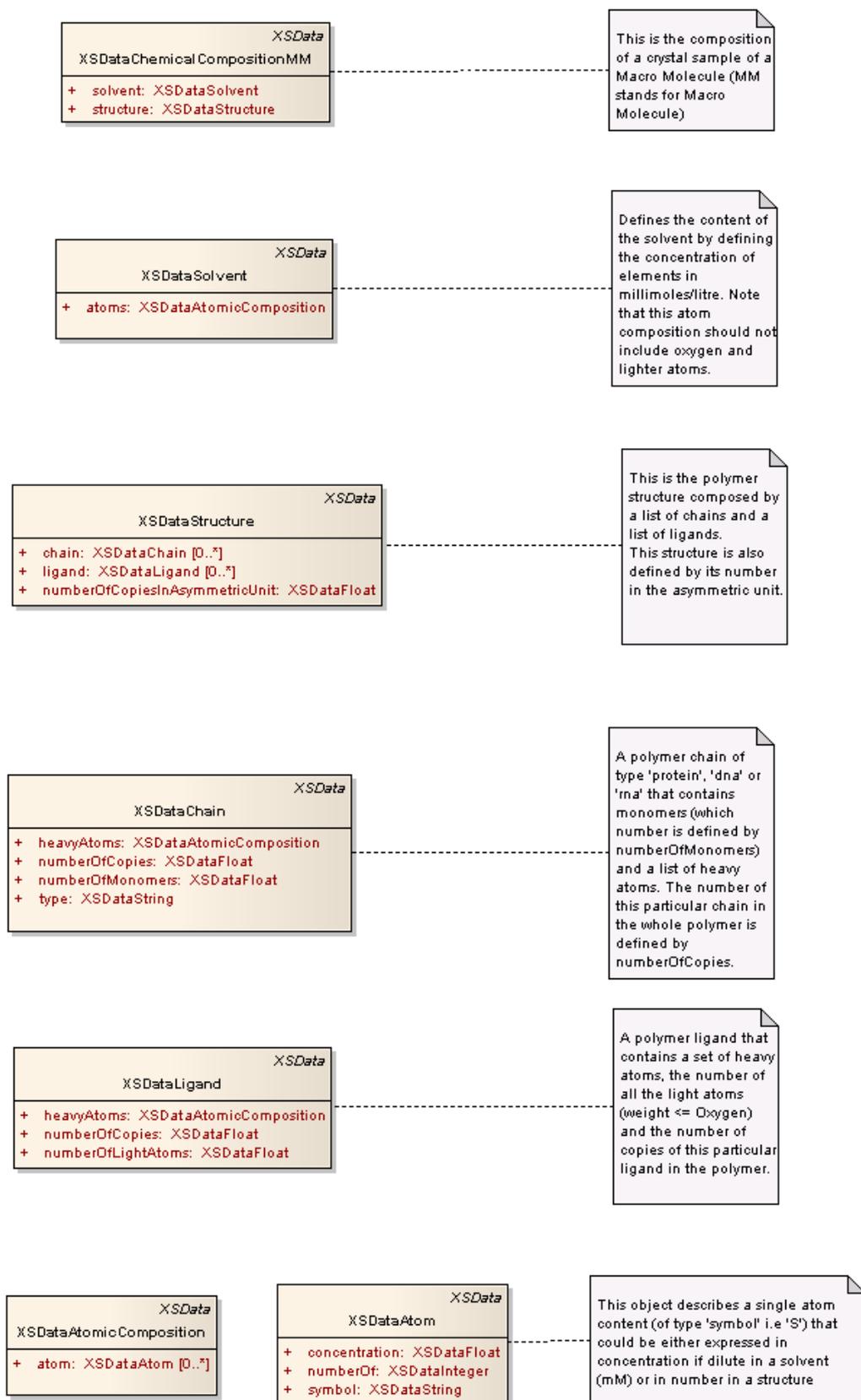


The properties of a detector. The attributes of this object should be splitted up

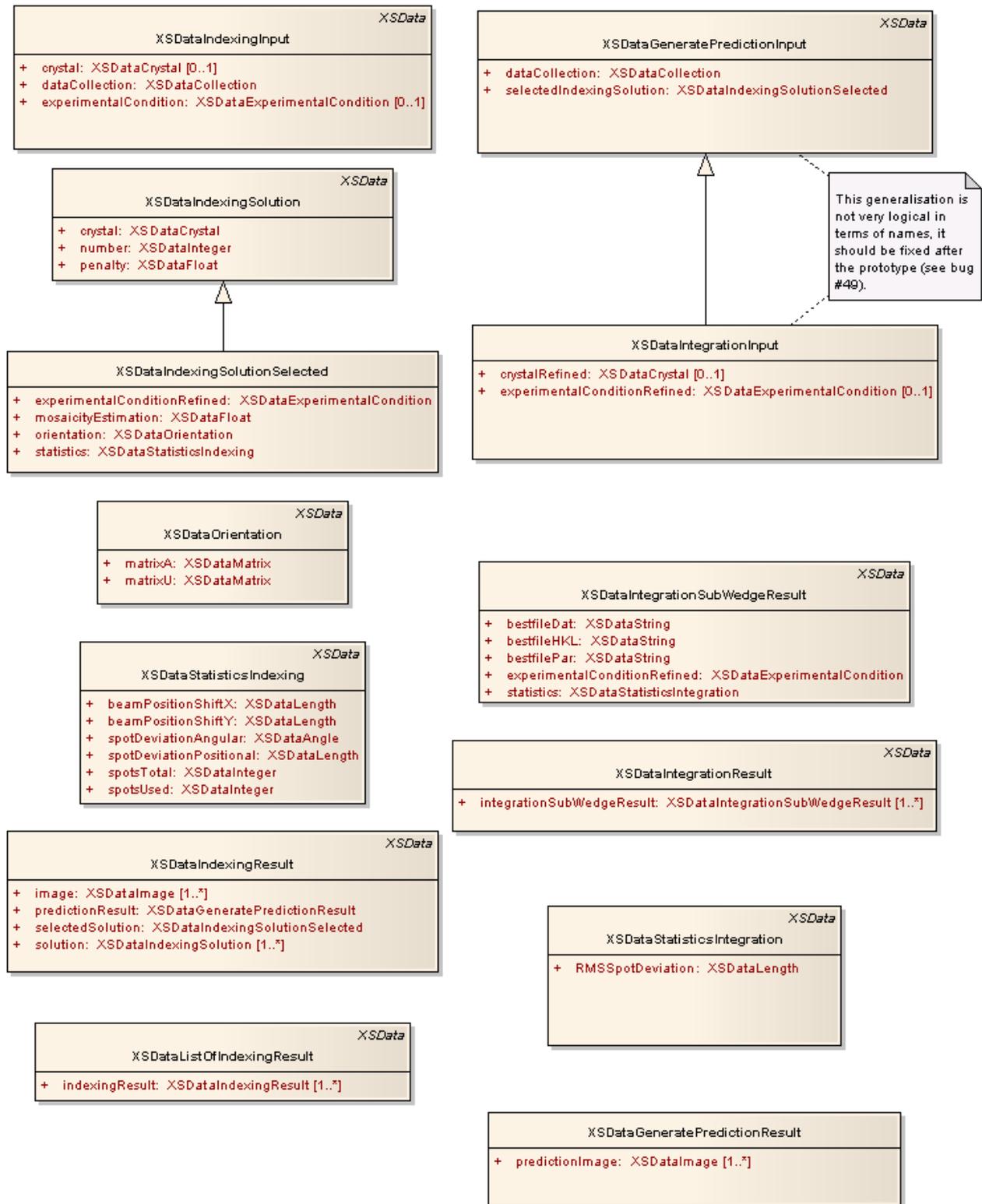


The properties of a goniostat:

- the maximal rotation speed that can be carried out
- the minimal width for an oscillation angle
- the oscillation width of subwedge
- the name of the rotation axis (typically phi)
- the rotation start angle
- the rotation end angle



Appendix 2.2 : The Characterisation data model



<i>XSData</i>	
XSDataCharacterisation	
<ul style="list-style-type: none"> + dataCollection: XSDataCollection + indexingResult: XSDataIndexingResult [0..1] + integrationResult: XSDataIntegrationResult [0..1] + strategyResult: XSDataStrategyResult [0..1] 	

<i>XSData</i>	
XSDataStrategyInput	
<ul style="list-style-type: none"> + bestFileContentDat: XSDataString + bestFileContentHKL: XSDataString [1..*] + bestFileContentPar: XSDataString + crystalRefined: XSDataCrystal + diffractionPlan: XSDataDiffractionPlan + experimentalCondition: XSDataExperimentalCondition + sample: XSDataSampleCrystalMM 	

<i>XSData</i>	
XSDataStrategyResult	
<ul style="list-style-type: none"> + collectionStrategy: XSDataCollection + statistics: XSDataStatisticsStrategy + strategySummary: XSDataStrategySummary 	

<i>XSData</i>	
XSDataStrategySummary	
<ul style="list-style-type: none"> + attenuation: XSDataFloat + completeness: XSDataFloat + iSigma: XSDataFloat + rankingResolution: XSDataFloat + redundancy: XSDataFloat + resolution: XSDataFloat + resolutionReasoning: XSDataString + totalDataCollectionTime: XSDataTime + totalExposureTime: XSDataTime 	

<i>XSData</i>	
XSDataResolutionBin	
<ul style="list-style-type: none"> + averageIntensity: XSDataFloat + averageSigma: XSDataFloat + chi2: XSDataFloat + completeness: XSDataFloat + IOverSigma: XSDataFloat + IOverSigmaChi: XSDataFloat + maxResolution: XSDataFloat + minResolution: XSDataFloat + percentageOverload: XSDataFloat + redundancy: XSDataFloat + rFactor: XSDataFloat 	

XSDataStatisticsStrategy	
<ul style="list-style-type: none"> + resolutionBin: XSDataResolutionBin [0..*] 	

Appendix 3 : How to run the prototype

- Make sure that EDNA_HOME points to the root directory of edna, if the EDNA prototype is installed in /usr/local/xtal/edna:

BASH shell: export EDNA_HOME=/usr/local/xtal/edna

TCSH shell: setenv EDNA_HOME /usr/local/xtal/edna

- Make sure that EDNA_SITE is set to for example "mySite"
- Copy the edna/prototype/conf/XSConfiguration_ESRF.xml to edna/prototype/conf/XSConfiguration_mySite.xml (or whatever you chose as EDNA_SITE). Edit this file, you need to set the paths for the following programs/packages:
 - MOSFLM version 7.0.1 or later
 - BEST version 3.1.0.d (2007.07.16) or later
 - RADDOSE version 4.2 of 20070705
 - CCP4 version.6.0.2 or later
- Execute the script \$EDNA_HOME/prototype/bin/edna-prototype with either :
 - --data + path to a data collection XML file.
 - --image + some images paths as arguments (not just the file name, the full path is needed). In case of using the --image keyword the following arguments are optional:
 - --flux : beam flux in photon/s.
 - --beamSize : beam size in mm (assuming a square beam)
 - --maxExposureTimeBeforeRadiationDamage : the longest time a crystal can be exposed to the X-ray beam before being destroyed by radiation damage. This value can also be used to optimise data collection in case of lack of time (e.g. end of beamtime).
 - --minExposureTimePerImage : the shortest exposure time at which the shutter of the beamline can work reliably.