

## EDNA November 20-22 Workshop Minutes:

### Participants:

Mark Basham (Diamond)  
Sandor Brockhauser (EMBL-Grenoble)  
Jose Gabadinho (ESRF)  
Marie-Francoise Incardona (ESRF)  
Peter Keller (Global Phasing)  
Karl Levik (Diamond)  
Romeu Pieritz (ESRF)  
Olga Roudenko (Synchrotron-Soleil)  
Olof Svensson (ESRF)

### Summary: What happened in the EDNA workshop in a few words:

- To give an introduction to AALib to the edna developers (edna members were also invited). Romeu gave an overview of the overall system and a few specific AALib examples.
- To start the discussion around the Use Case Implementation. It was agreed to implement first a configurable application skeleton that will sequentially execute the main 3 steps of the characterization use case. Another priority should be to implement the MOSFLM plugin and its related local data model (that will be translated to a more generic data model within a translater wrapper).
- To work on the prototype planning

### Minutes: What happened in the EDNA workshop more precisely:

#### 1/ AALib Presentation: (Romeu)

Romeu presented an overview of AALib. See link  
<ftp://ftp.esrf.fr/pub/scisoft/EDNA/Workshop-20071120/EDNA-AALib.pdf>

#### Main questions regarding AALib:

- Data propagation between plugins not so obvious
- The errors are caught within AALib and a warning messages are logged in log files: some developers found this as a potential weakness since an application based on AALib wouldn't be able to handle the different potential errors in order to react accordingly. A suggestion was to return the exception object instead, and the developer instead of checking if the result was null he/she would check if it was from the Exception class.

#### 2/ Use Case implementation Document: (Marie-Francoise)

Marie-Francoise presented the technical document that was sent before the workshop. The document was used as the basis for the discussions. see link

[ftp://ftp.esrf.fr/pub/scisoft/EDNA/Workshop-20071120/edna\\_implementation.pdf](ftp://ftp.esrf.fr/pub/scisoft/EDNA/Workshop-20071120/edna_implementation.pdf)

[ftp://ftp.esrf.fr/pub/scisoft/EDNA/Workshop-20071120/workshop\\_slides.pdf](ftp://ftp.esrf.fr/pub/scisoft/EDNA/Workshop-20071120/workshop_slides.pdf)

#### The main idea and approach was presented including:

- Using both Top-down and Bottom-up approaches to reach a working prototype:
  - Top-down approach to analyze and divide the main use case into sub-systems in order to bring an application skeleton to light.
  - Bottom-up approach to specify and implement the base elements to build the overall system.

- Having a configurable skeleton that will launch sequentially the indexing-integration-strategy. The configuration will both define which plugins should be launched (for the indexing step for example) and specific parameter that need a plugin to work properly.

Questions/comments:

Sandor suggested that the prototype should be able to start from any step (i.e from the integration step if the indexing has been already performed by any other means). It was agreed that this was a question of use case and we should focus on what has been proposed: indexing - integration - strategy.

Jose asked which plugin should decide of the result in case of a parallel execution of external programs for the indexing step for instance. Olof proposed that the generic indexing plugin should be responsible, whereas Sandor would rather see that the characterization plugin should be the controller.

- Several configuration file formats have been presented (ini-like and xml formats). The conclusion was to have an xml file format (that is object-oriented) that will define the plugins configuration in a "flat" way instead of having a nested-style one (plugin configuration into another plugin configuration) that would generate potential perpetual loops when generating automatically configuration object code.

Questions/comments:

Concern of Romeu whether we need a configuration which would be time consuming to implement.

Jose asked why we did not use attributes instead of objects within the proposed xml formats... Olof mentioned that for many reasons, it was more simple.

Sandor suggested to have a dynamic configuration at runtime instead of a static one. It was said that for the prototype, a static should be enough.

- A mechanism to check the plugin has been presented: PluginChecker (to be implemented) in order to check the plugin independently.

### 3/ Planning: (Olof)

Olof presented the objectives of the prototype in general and in terms of new science (e.g. taking into account radiation damage):

- MOSFLM should be used as the default external program for the indexing and the integration step. This is the absolute minimal configuration.
- Indexing with MOSFLM on its own wouldn't be very impressive. It would be preferable to also be able to launch (optionally) XDS. Therefore we should, if time allows, also develop an XDS indexing plugin.
- XDS shouldn't be executed in parallel with MOSFLM, as we would otherwise also have to manage the concurrent results coming from MOSFLM and XDS.

The priorities:

- We should focus on the local data model for each specific plugin that will execute an external program. The idea is to implement a local data model, and a kind of translator that will "wrap" the local plugin in order to translate the local data model to the generic one (to be defined).
- Start the MOSFLM data model and its related plugin. Marie-Francoise noticed that we should implement first a common plugin from which all the EDNA plugins should derive from. Olof said that we could implement an empty common plugin in order to implement quickly the MOSFLM one. Ideally, the deadline for the MOSFLM plugin to be implemented should be end of december. Harry could be consulted if any questions regarding the input/output of MOSFLM.
- Implement the application skeleton.

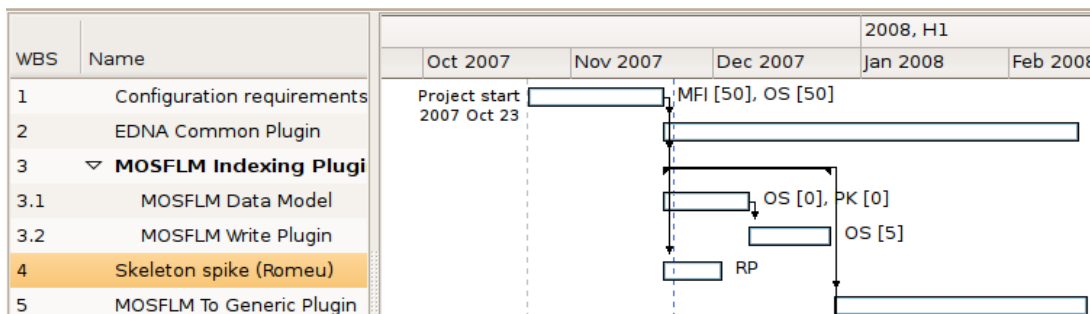
Peter asked whether we really need a local data model or should an xml output file be enough. Olof reminded that the output of MOSFLM was rather an xml file describing an html data, so at least, we should have a xsd based tool to convert this file into a real xml output file.

Romeu proposed to implement the skeleton very rapidly (~ 3 days). Marie-Francoise was doubtful with regards

to the implementation delay Romeu mentioned as in her point of view, horizontal issues should be addressed as well within the skeleton (configuration, error cases, data propagation, etc...) in order to avoid to have yet another development that should be thrown away. Olof thinks that we should focus on the success cases for the prototype and the error cases should not be a priority.

We had a quick round table to know who could participate in the prototype. Sandor proposed to see with Pierre Legrand the XDS data model. Jose will be looking on how to use AALib with tango, but won't participate directly in EDNA. Peter will participate in the local MOSFLM data model. Karl said that Alun doesn't let him participate to EDNA ( ;- ) ).

The following picture summarises the planning of the EDNA common plugin and the MOSFLM Indexing plugin:



#### General Open Questions:

- Data Model Tools:

We have to look at the XSD generation within Enterprise Architect and Umbrello. Action: Peter and Marie-Francoise. Marie-Francoise said that she had a preference for Enterprise Architect as Umbrello has a few bugs on the main functionalities (import of source code mainly). Peter mentions that other tools are available (see "gentleware")

- Development of a common Plugin:

It was agreed to develop a common plugin from which each EDNA plugin should derive. In a first step, this plugin will be empty to allow a rapid start of the MOSFLM plugin. Action: Marie-Francoise.

- Persistence:

It was agreed that this item will be addressed for the EDNA kernel but not for the prototype.

- Having a common server for the project management tools:

Olof is waiting for the computing services answer to switch a dedibox server rental. Peter mentioned another possibility to have an internet server like sourceforge... see [www.berlios.de](http://www.berlios.de). Action: Olof

#### Open questions regarding AALib:

- Should the prototype work in Jython?:

Due to a current limitation to manage external process with java, it isn't possible to launch external processes with jython. So the recommendation is to use python for the moment.

- Data propagation between plugins:

Romeu will illustrate the data propagation within the skeleton.

- Read/write xml:

Romeu's recommendation is to use xml/xsd mechanism instead of using DOM or xpath xml parsing.

- How to make the prototype a server:

Using XML-RPC mechanism to pass object instances between the server and the client. The idea is to derive the application from a ALApplicationXMLServer instead of ALApplication.

- GRID:

There was a question mark (between Jose and Romeu) regarding the definition of GRID: does it represent a distributed environment in different institutions or in a local cluster? Both definition are right according wikipedia.

4/ AALib examples: (Romeu)

Romeu presented a few examples of AALib use case and tests to illustrate the presentation he gave the day before. The examples dealt with data propagation, commandLineInterface (mapping between plugin and command parameter), XML-RPC and client-server.

The End.