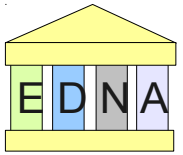
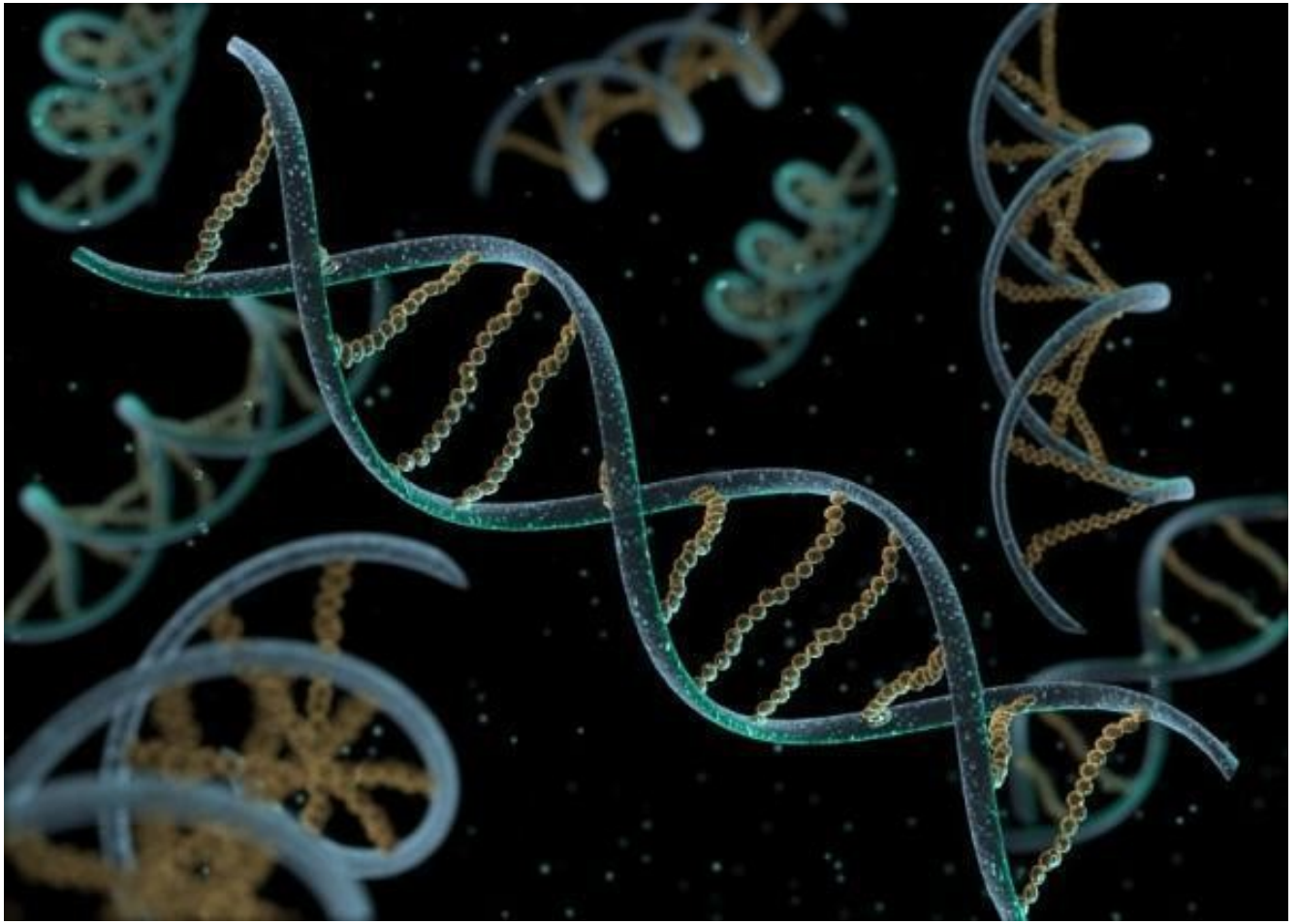
EDNA – building on the experiences from MX automated strategy determination

Olof Svensson
EDNA Project Manager
Data Analysis Unit
Instrument Support and Development Division
ESRF

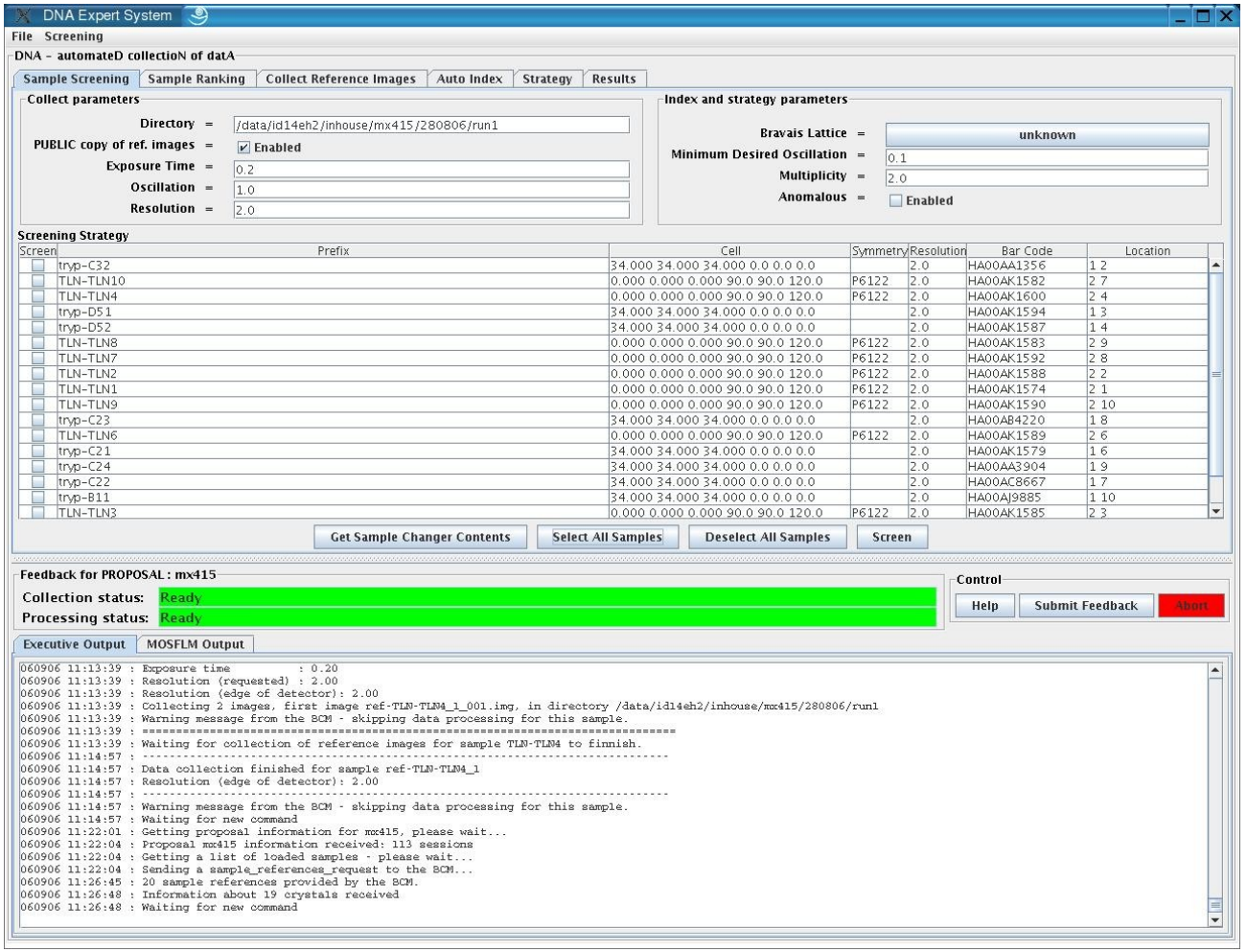


In the beginning there was DNA...



… or RNA or something else, if you speak about life...

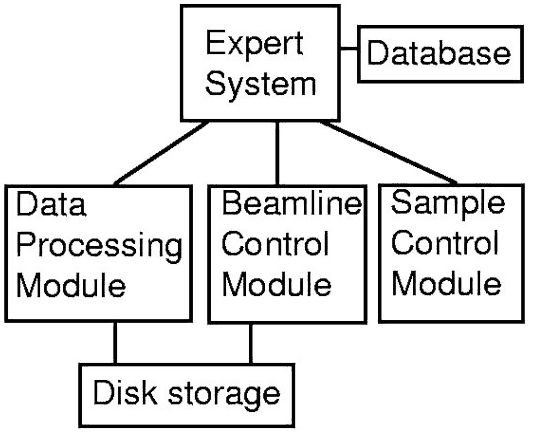
In the beginning there was DNA...

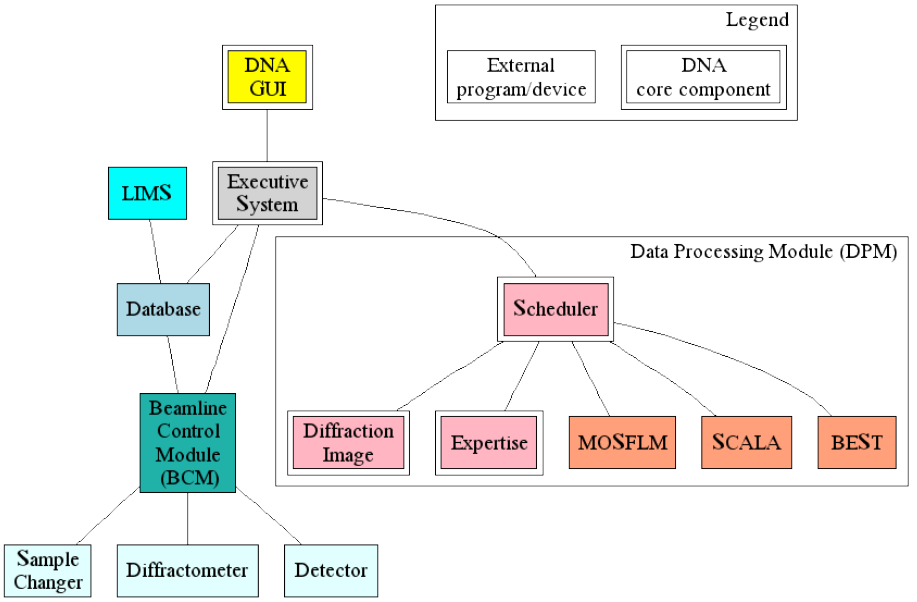


… if you speak about EDNA!

Beginning of automated MX characterisation... … or DNA in one slide

- Kick-off meeting in 2001
- Initial collaborators :
  - ESRF
  - Daresbury SRS
  - MRC LMB Cambridge
- Initially no external funding
- Meaning of ”DNA” :
  - automate**D** collectio**N** of dat**A**
- **Main development period 2001 – 2005**
- More collaborators and more developers entered the project, mainly thanks to external fundings : BioXHIT and e-HTPX.
- Since 2005 an integral part of the ESRF ”data collection pipeline”
- Installed and used at :
  - ESRF
  - Diamond
  - Recently used at NSLS (Brookhaven), now using EDNA
- **Major component of the 2008 BESSY innovation award**





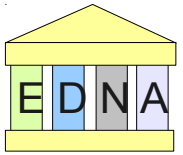
...so – why not DNA?

Good things learned from DNA:

- Collaboration :
  - Scientists involved in design and testing
  - Executive committee for setting milestones / deliverables and for resolving conflicts between developers
- Two major Use Cases implemented :
  - Characterisation + data collection with online integration and quick scaling
  - Automatic screening and ranking

Not so good things learned from DNA...:

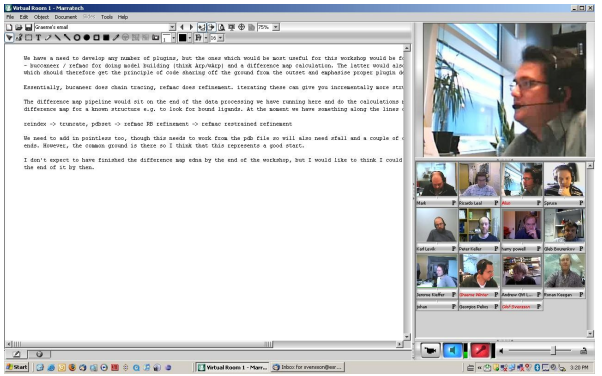
- The choice of name...
- Collaboration :
  - No project agreement
  - Minimal project management
- Not modular:
  - Too costly to change work flow
  - Poorly designed data model
  - Difficult for new developers to enter the collaboration
- MX hardwired!

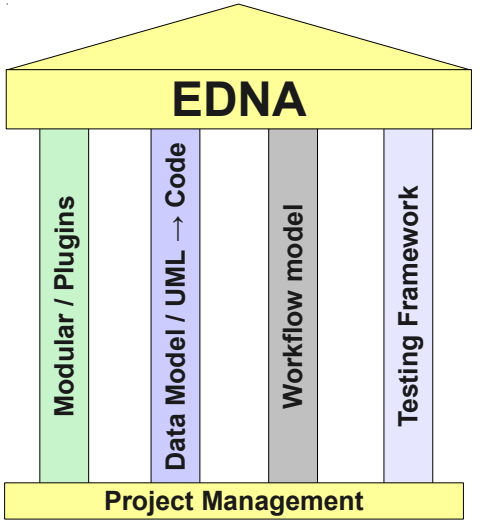


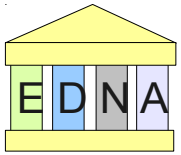
What is then EDNA?

- EDNA is about collaboration:
  - Code sharing (SVN)
  - Coding conventions
  - Code reviews
  - Open source (LGPL, GPL)
  - Bug tracker
  - Wiki
  - Memorandum of Understanding
  - Executive committee
  - Project manager / coordinator
  - Regular meetings / video conferences

- EDNA is a framework:
  - "Generic" kernel
    - Data modelling framework
    - Support for multi-threaded modules (plugins) development
    - Support for workflow development
    - Testing framework
  - "Specific" applications (MXv1, bioSaxs etc.)
  - Automatic testing and nightly builts
  - Automatic API doc generation

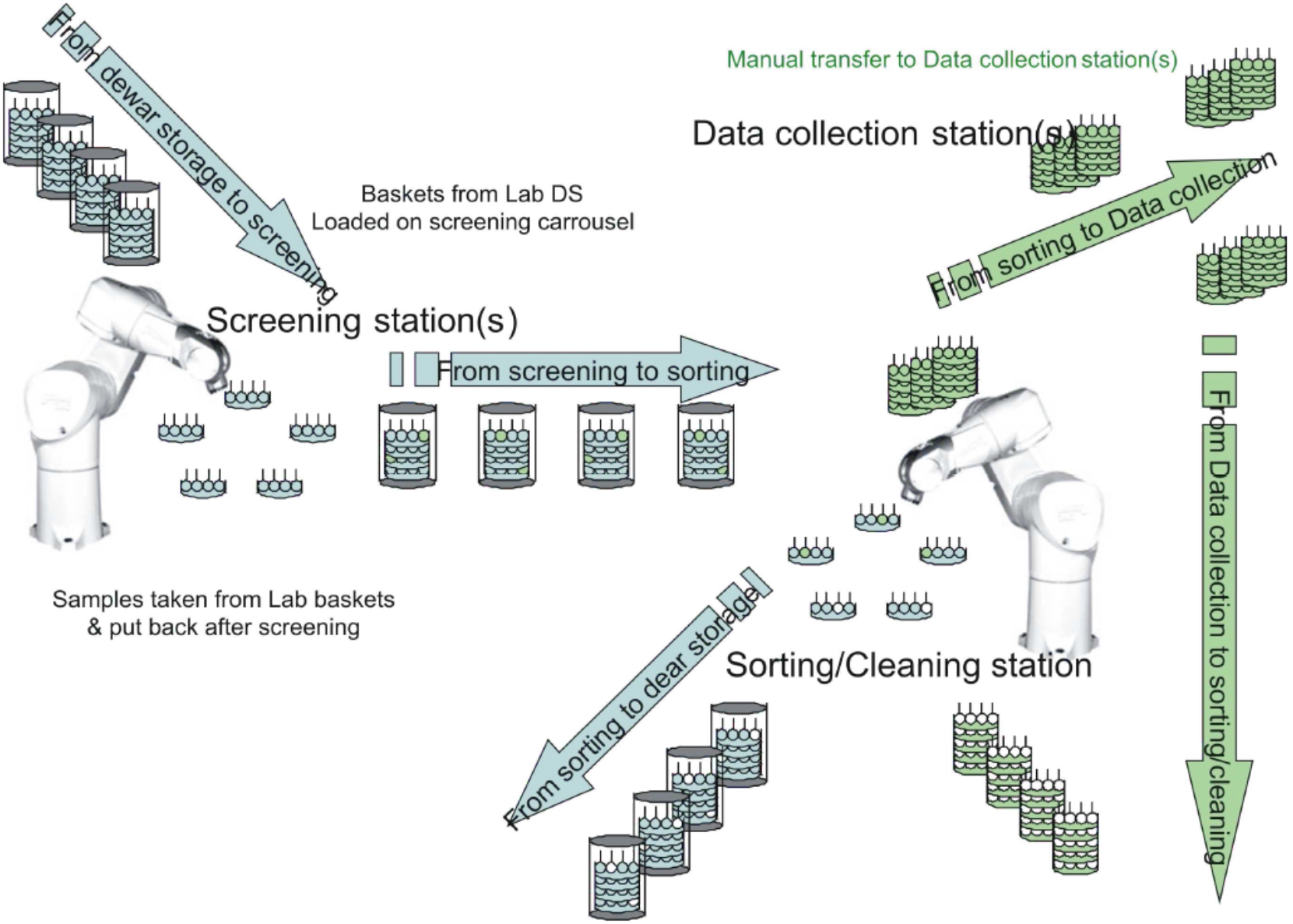






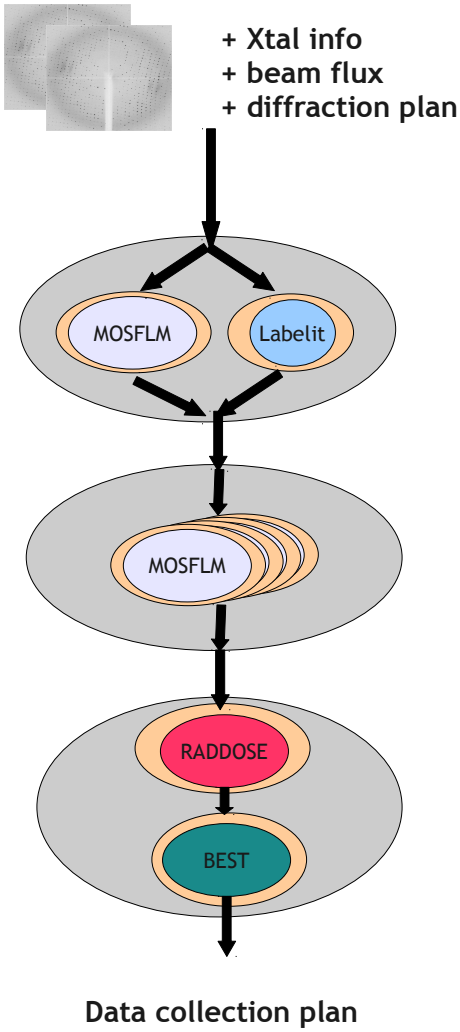
… what's mentioned above are lessons learned from the DNA collaboration

Challenge for the ESRF Upgrade :
Massively Automated Sample Selection Integrated Facility

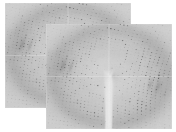


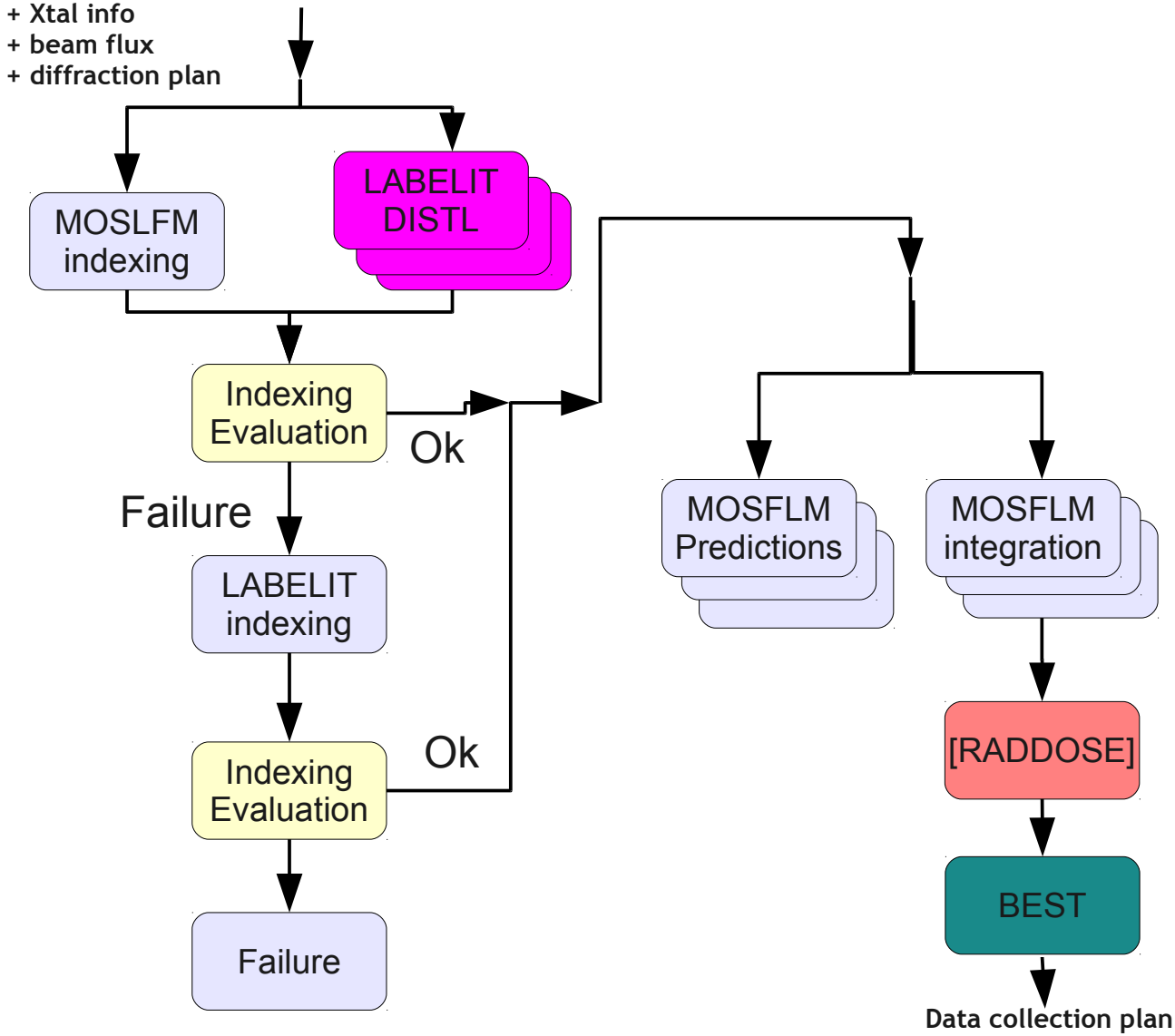
Example EDNA workflow : MXv1 Characterisation v1.1

- MX sample characterisation taking into account radiation damage
- Indexing using MOSFLM or Labelit
- Parallel integration of reference images
- If flux + beamsize : RADDOSE for estimating radiation damage (not in DNA!)
- BEST strategy calculation
  - taking into account radiation damage
  - multi-subwedge data collection strategies

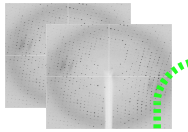


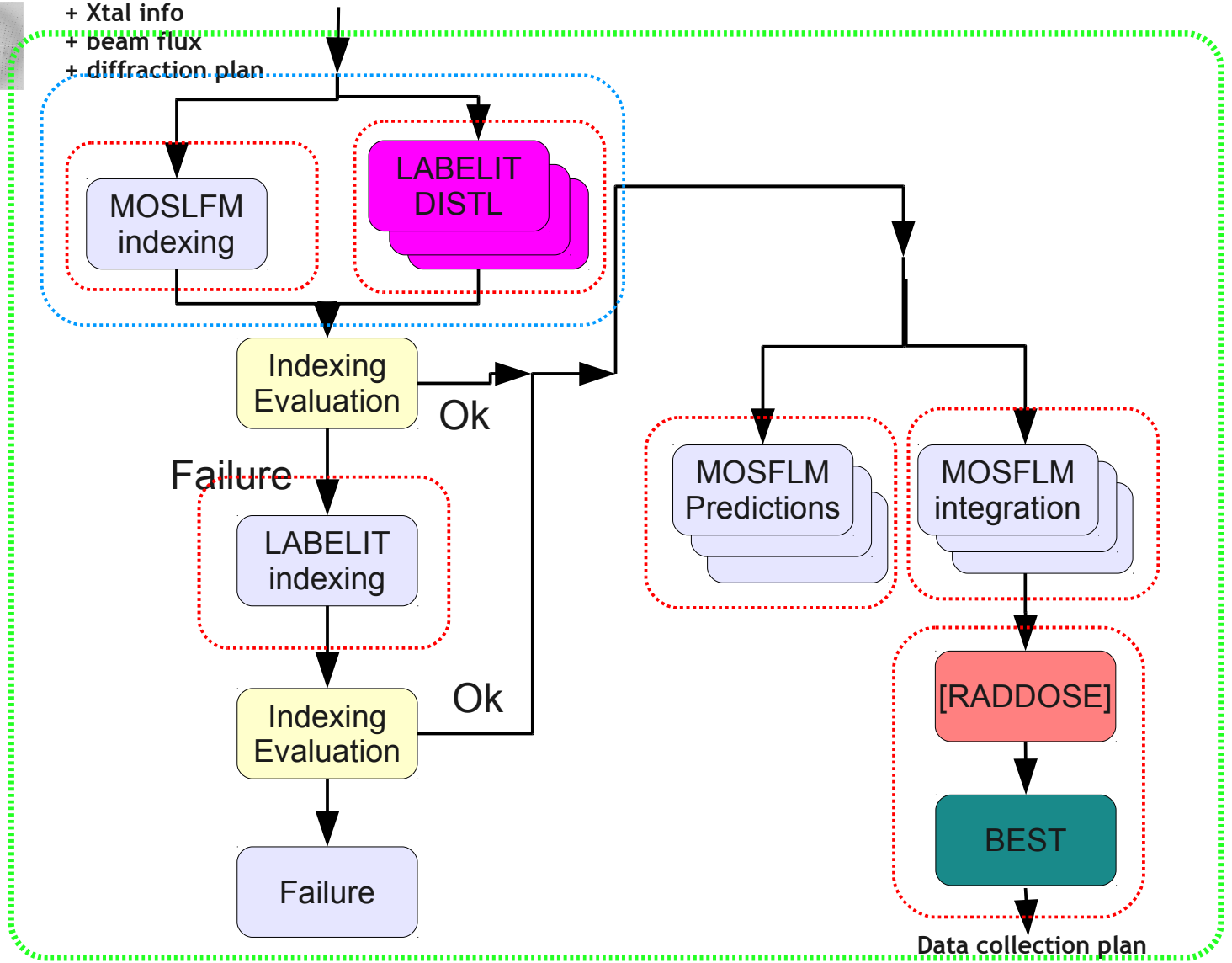
Example EDNA workflow : MXv1 Characterisation v1.2



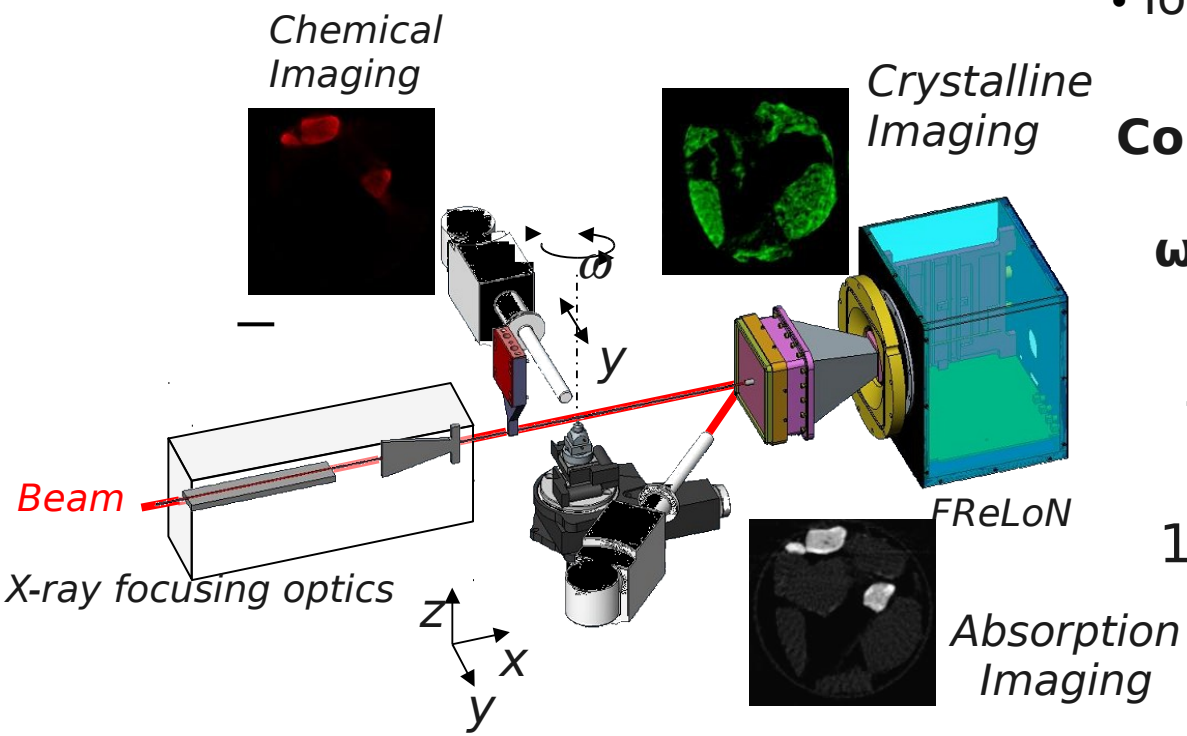


Example EDNA workflow : MXv1 Characterisation v1.2





Part of ESRF upgrade project UPBL4: Nano-imaging / Nano-analysis



Experimental setup @ ID22:

- KB optics with µ-/nano-focused beam
- FReLoN 2k taper: diffraction data
- Fluorescence (single element) detector
- IonChamber (I_o and attenuated I)

See e.g. Bleuet et al., Nature Mater. 2008

Commonly N ≈ 160 scanning steps/y-line,
ω ≈ 60 angular steps/tomographic scan

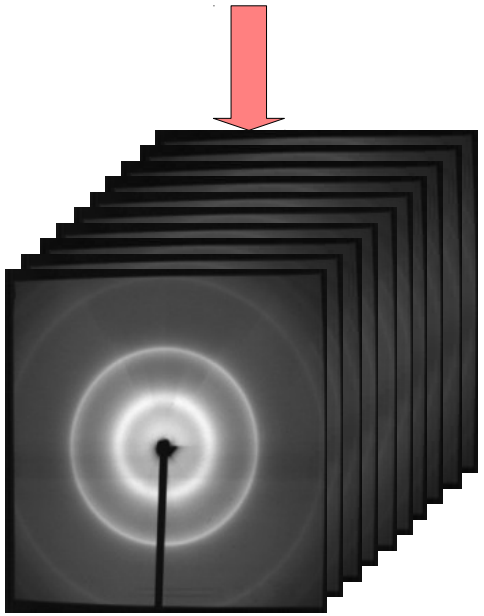
10-15h beamtime required/tomo scan
100 Go data generated/tomo scan

Slide from Pierre Bleuet

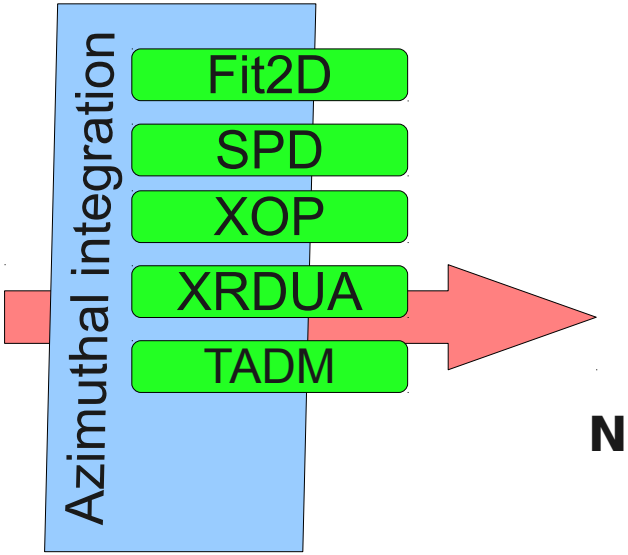
Online data Analysis for NINA

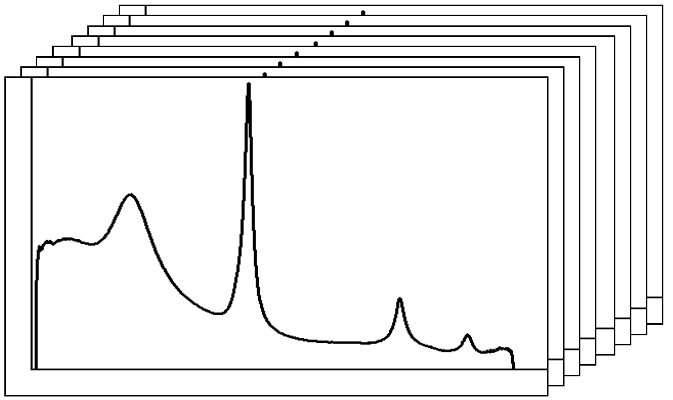
- Focus on most CPU demanding application: DiffractionCT

FReLoN2k
(EDF, 16bit UInt, LittleEndian)



N x **ω** 2D Diffraction Images

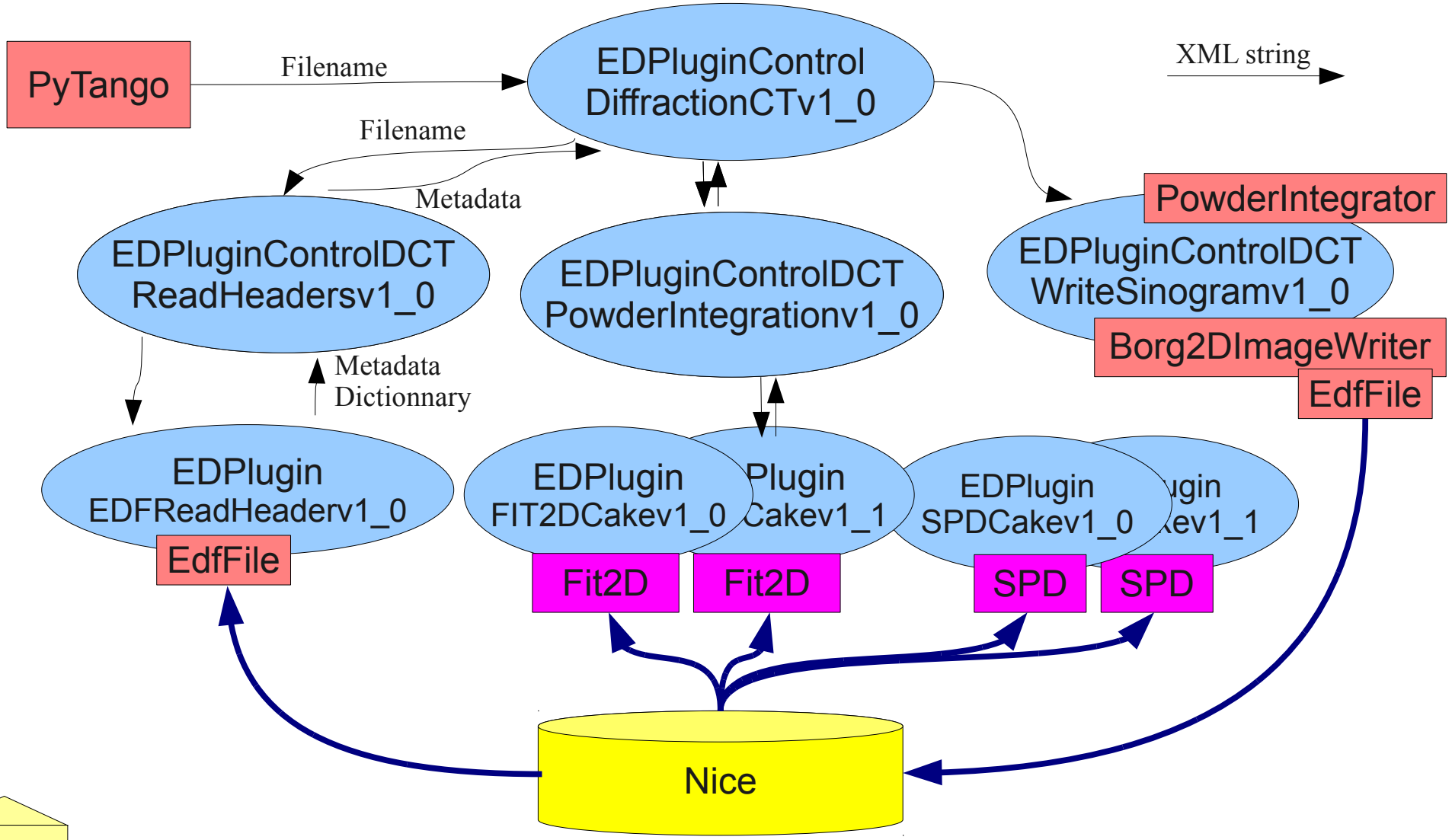




N x **ω** 1D diffraction patterns

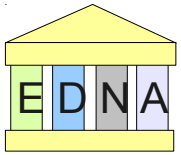
Integration of many tools
Need plugin gestion

Diffraction CT implementation in EDNA



Current development status

- Execution plugins:
  - MxExecPlugins: 24 exec plugins
  - Exec plugins: 21 exec plugins
  - Saxs(4), SPD (4), FIT2D (2), EDF (2), HDF5 (2), thumbnail, video, …
- Control plugins:
  - MX v1 26 control plugins, 7 exec plugins
  - MX v2 3 control plugins
  - DiffractionCT v1 6 control plugins
  - BioSaxs 5 control plugins
- Other projects (CCP4, Dimple, Darc ...):
  - Managed mainly by Diamond and CCP4
  - 27 plugins: 8 control plugins & 19 execution plugins
- Total: 119 – 71 Execution plugins, 48 control plugins



EDNA Testing Framework

- The EDNA testing framework consist of three layers :
  - Kernel Unit tests
  - Plugin Unit tests
  - Plugin Execution tests

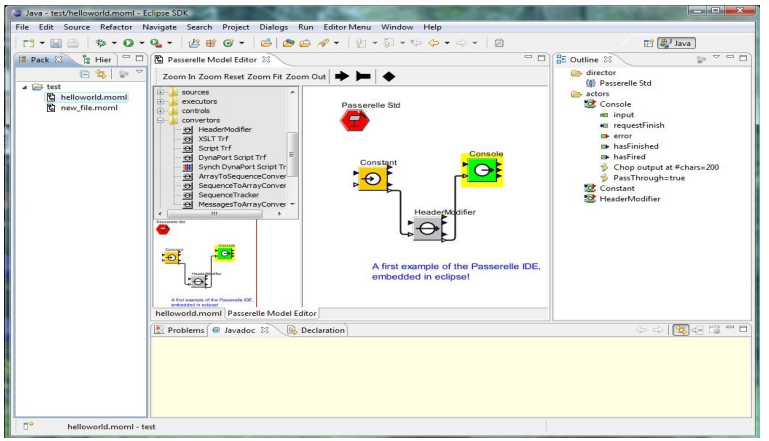
- Example of EDNA Plugin Execution tests result:

```
[UnitTest]: ###################################################################
[UnitTest]: Result for EDTestSuiteAll : SUCCES
[UnitTest]:
[UnitTest]:  Number of executed test suites in this test suite : 2
[UnitTest]:
[UnitTest]:
[UnitTest]:   Total number of test cases executed with SUCCESS : 175
[UnitTest]:   Total number of test cases executed with FAILURE : 0
[UnitTest]:
[UnitTest]: Total number of test methods executed with SUCCESS : 273
[UnitTest]: Total number of test methods executed with FAILURE : 0
[UnitTest]:
[UnitTest]:                                              Runtime : 630.138 [s]
[UnitTest]: ###################################################################
```

- Automatic testing every night + builds

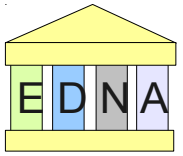
Future challenges for the EDNA framework

- Replacement of Enterprise Architect with an Eclipse based modelling tool
- Replacement of generateDS for data binding code generation
- Workflow editor:
  - Implicit documentation of workflow
  - Implicit parallel workflows
  - Possibility to “easily” modify / construct new workflows
  - Possibility to debug workflows
  - Possibility to restart a stopped workflow
- Improved logging



Upcoming events

- EDNA Full Meeting on Thursday September 23rd at the ESRF:
  - Overview of status of EDNA projects
  - MXv1 → MXv2 transition
  - Updated Memorandum of Understanding (MoU), separate kernel, MX and other MoUs
- Introduction to EDNA plugin development at the ESRF (not yet scheduled)
- Code camp / kernel developments (not yet scheduled)

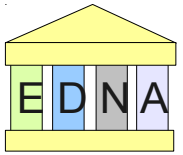


Conclusions

- The experiences of MX automated crystal characterisation (DNA) were used as the foundations for the EDNA collaboration

- The EDNA framework allow rapid and robust scientific application developments thanks to built-in support for four “pillars” of software engineering techniques:
  - Data modelling
  - Multi-threaded module/plugin development
  - Workflow developmet
  - Testing framework

- EDNA is now used for applications in MX and non-MX scientific data analysis developments

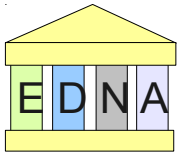


EDNA Collaborators 2010

Alexander Popov(e)
Alun Ashton(b)
Andrew Leslie(h)
Andrew McCarthy(c)
Andrew Thompson(k)
Clemens Schulze(j)
Clemens Vonrhein(f)
Darren Spruce(e)
Elspeth Gordon(e)
Ezequiel Panepucci(j)
Gérard Bricogne(f)
Gerrit Langer(c)
Gleb Bourenkov(c)
Gordon Leonard(e)
Graeme Winter(b)
Harry Powell(h)
Jérôme Kieffer(e)
Johan Turkenburg(m)
Johan Unge(g)
John Skinner(i)
Karl Levik(b)
Katherine McAuley(b)
Lucile Roussier(k)
Marie-Farnçoise Incardona(e)
Mark Basham(b)
Meitian Wang(j)
Michael Hellmig(a)
Olga Roudenko(k)
Peter Keller(f)
Peter Turner(l)
Pierre Legrand(k)
Robert Sweet(i)
Romeu Pieritz(e)
Sandor Brockhauser(c)
Sean McSweeney(e)
Takashi Tomizaki(j)
Thomas Schneider(c)
Uwe Mueller(a)

(a) BESSY, Berlin, Germany
(b) Diamond Light Source, UK
(c) EMBL, Grenoble, France
(d) EMBL, Hamburg, Germany
(e) ESRF, Grenoble, France
(f) Global Phasing, Cambridge, UK
(g) MAX LAB, Lund, Sweden
(h) MRC LMB, Cambridge, UK
(i) NSLS, Brookhaven, U.S.
(j) SLS, Villigen, Switzeland
(k) Synchrotron Soleil, France
(l) University of Sydney, Australia
(m) University of York, UK

EDNA developers
Executive committee



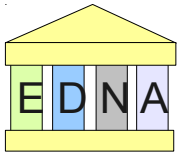
EDNA Data Model Framework

- What is a data model? From wikipedia:

  A data model in software engineering is an abstract model that describes how data are represented and accessed. Data models formally define data elements and relationships among data elements for a domain of interest.

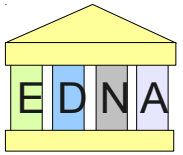
  Communication and precision are the two key benefits that make a data model important to applications that use and exchange data.

- Since we want to make workflows → communication between programs → data modelling is important



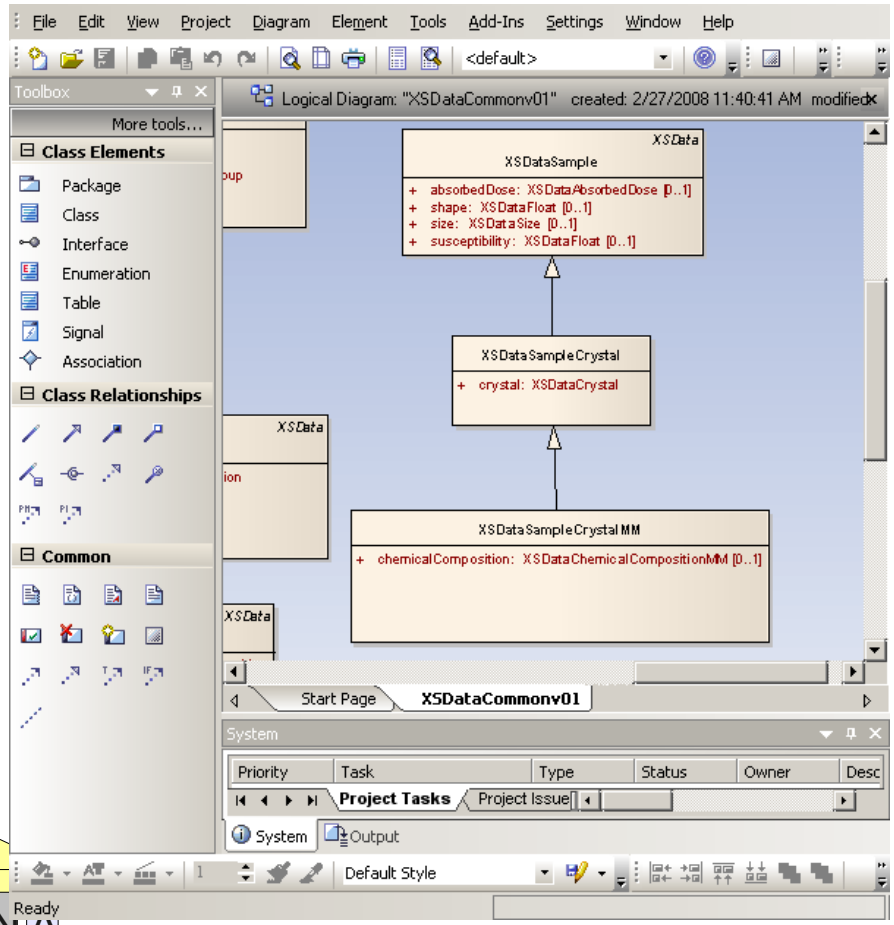
How are Data Models used in EDNA?

- The “common” data model :
  - This data model defines a set of simple basic types (e.g. double, string etc) and some more complex (3x3 matrix) which can be used by all other EDNA data models.
  - The common data model is a part of the EDNA kernel.
- The “specific” data models :
  - Data models which are specific for a certain task or program, e.g. data models for MOSFLM, XDS, FIT2D etc
  - The specific data models are typacilly used only by a few EDNA plugins (modules)
- The “generic” or “project” data models :
  - These data models should not be dependent on a single program but rather be developed for a certain scientific area, e.g. MX, tomography etc.



The EDNA Data Model Framework

- From UML diagrams to generated code (data binding) :




```
<xs:element name="XSDataSample" type="XSDataSample
<xs:complexType name="XSDataSample">
     <xs:complexContent>
          <xs:extension base="XSData">
               <xs:sequence>
                    <xs:element name="absorbedDose
                    <xs:element name="shape" type=
                    <xs:element name="size" type="
                    <xs:element name="susceptibili
               </xs:sequence>
          </xs:extension>
     </xs:complexContent>
</xs:complexType>
```

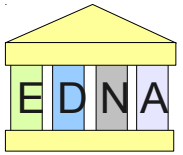
Why do we want modules / plugins ?

- Again from wikipedia:

  In computing, a plug-in is a set of software components that adds specific capabilities to a larger software application.

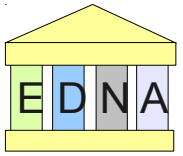
  Applications support plug-ins for many reasons. Some of the main reasons include:

  - *to enable third-party developers to create capabilities which extend an application*
  - *to support easily adding new features*
  - *to reduce the size of an application*
  - *to separate source code from an application because of incompatible software licenses.*



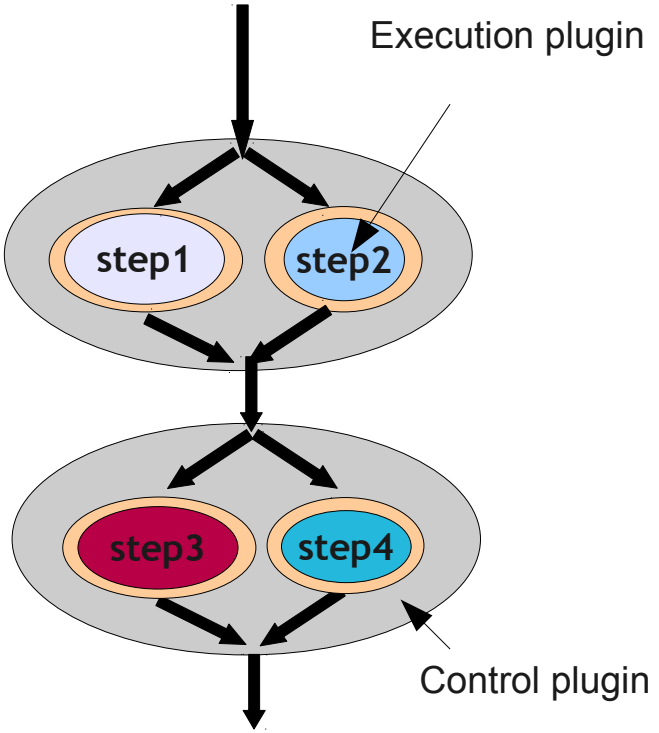
EDNA Plugins Features :

- Self-contained plugin structure:
  - Data model(s)
  - Plugin source code
  - Data binding objects
  - Unit and execution tests
  - Data for tests
  - Documentation
- Fast dynamic plugin loading (cache)
- Multi-threaded plugin execution and synchronisation (threadsafe)
- Plugin configuration
- Persistence of data in XML using data bindings



EDNA Modularity : Plugins and their hierarchy

- Plugin base class :
  - Configuration, working directory, etc.
- Execution plugins :
  - Execution of external programs, e.g. (bash) scripts
- Controller plugins:
  - Control of execution plugins
  - Parallel execution
  - Synchronisation

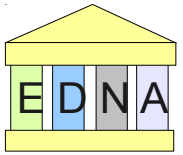


Conclusions

- The experiences of MX automated crystal characterisation (DNA) were used as the foundations for the EDNA collaboration

- The EDNA framework allow rapid and robust scientific application developments thanks to built-in support for four “pillars” of software engineering techniques:
  - Data modelling
  - Multi-threaded module/plugin development
  - Workflow developmet
  - Testing framework

- EDNA is now used for applications in MX and non-MX scientific data analysis developments



EDNA Collaborators 2010

Alexander Popov(e)
Alun Ashton(b)
Andrew Leslie(h)
Andrew McCarthy(c)
Andrew Thompson(k)
Clemens Schulze(j)
Clemens Vonrhein(f)
Darren Spruce(e)
Elspeth Gordon(e)
Ezequiel Panepucci(j)
Gérard Bricogne(f)
Gerrit Langer(c)
Gleb Bourenkov(c)
Gordon Leonard(e)
Graeme Winter(b)
Harry Powell(h)
Jérôme Kieffer(e)
Johan Turkenburg(m)
Johan Unge(g)
John Skinner(i)
Karl Levik(b)
Katherine McAuley(b)
Lucile Roussier(k)
Marie-Farnçoise Incardona(e)
Mark Basham(b)
Meitian Wang(j)
Michael Hellmig(a)
Olga Roudenko(k)
Peter Keller(f)
Peter Turner(l)
Pierre Legrand(k)
Robert Sweet(i)
Romeu Pieritz(e)
Sandor Brockhauser(c)
Sean McSweeney(e)
Takashi Tomizaki(j)
Thomas Schneider(c)
Uwe Mueller(a)

(a) BESSY, Berlin, Germany
(b) Diamond Light Source, UK
(c) EMBL, Grenoble, France
(d) EMBL, Hamburg, Germany
(e) ESRF, Grenoble, France
(f) Global Phasing, Cambridge, UK
(g) MAX LAB, Lund, Sweden
(h) MRC LMB, Cambridge, UK
(i) NSLS, Brookhaven, U.S.
(j) SLS, Villigen, Switzeland
(k) Synchrotron Soleil, France
(l) University of Sydney, Australia
(m) University of York, UK

EDNA developers
Executive committee