Data Analysis Sequencing - EDNA

Olof Svensson
Data Analysis Unit
ISDD ESRF

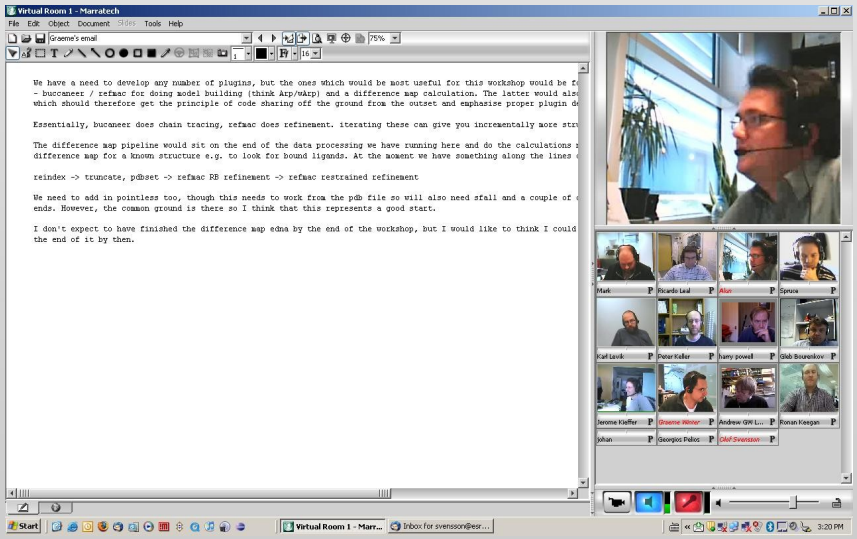
Why do we need EDNA?

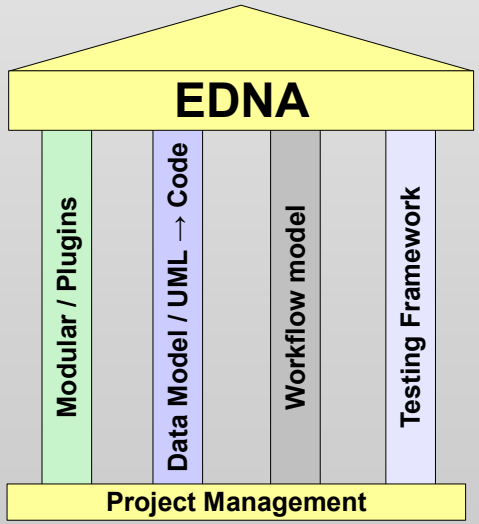
- EDNA is the best answer we (developers) have come up with so far for answering these questions :
  - How can we automate data analysis workflows?
    - “pipeline” existing scientific software for (online) data analysis workflows
    - abstract certain calculations to be “generic”, e.g. indexing of a diffraction pattern
    - “flexible” workflows, rapid changes depending on the scientific needs
  - How can we make these workflows robust?
    - easily adapted to new versions of scientific software packages
  - How can we collaborate efficiently?
    - re-use of code without breaking existing functionality

What is EDNA?

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

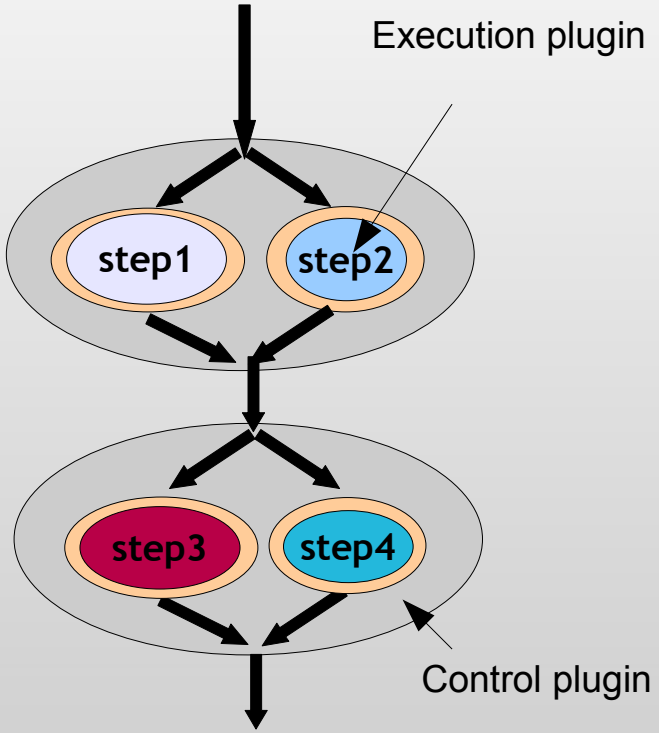
- EDNA is a framework:
  - "Generic" kernel
    - Data modelling framework
    - Support for multi-threaded modules (plugins) development
    - Support for workflow development
    - Testing framework
  - "Specific" applications (MXv1, bioSaxs etc.)
  - Automatic testing and nightly builds
  - Automatic API doc generation
  - No GUI





EDNA Modularity : Plugins and their hierarchy

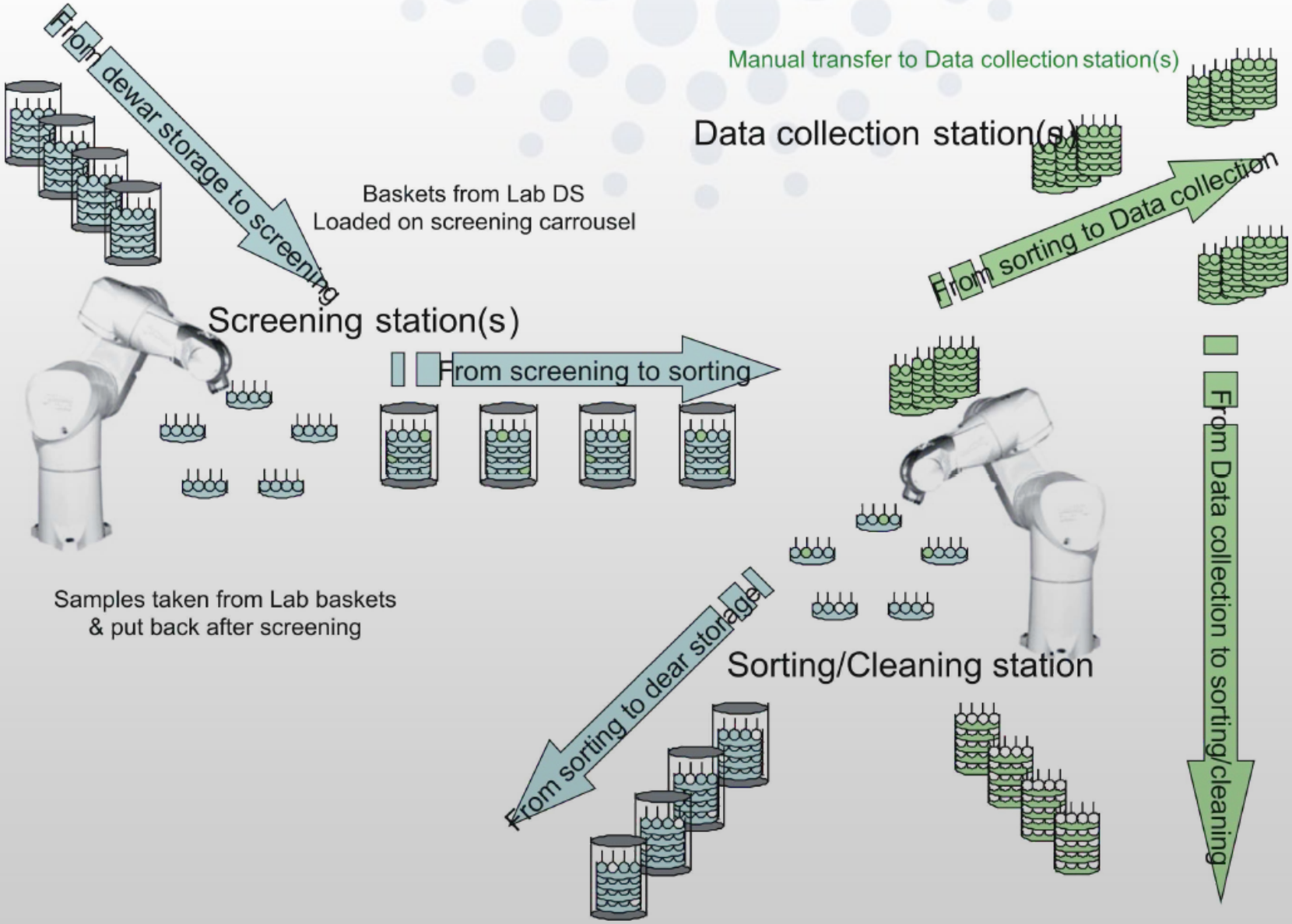
- Plugin base class :
  - Configuration, working directory, etc.
- Execution plugins :
  - Execution of external programs, e.g. (bash) scripts
- Controller plugins:
  - Control of execution plugins
  - Parallel execution
  - Synchronisation



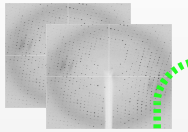
Existing scientific EDNA workflows

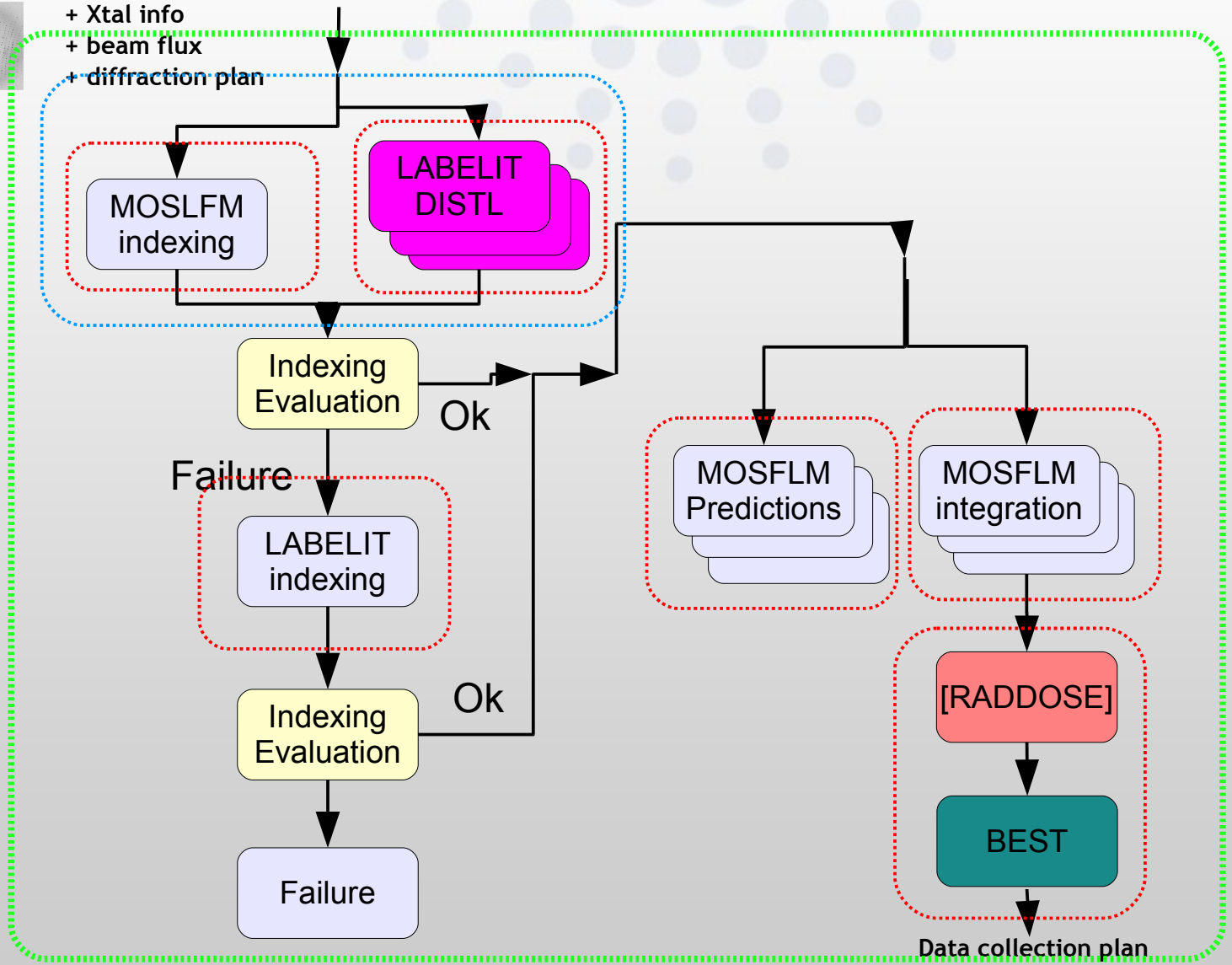
- Macromolecular crystallography:
  - Characterisation taking into account radiation damage (MOSFLM, Labelit, RADDOSE, BEST)
  - Connection with experiment data base (ISPyB)
  - Parallel execution of characterisation (GRID data processing)
  - Parallel creation of image thumbnails
- Diffraction Computed Tomography
  - SPD: Image correction, fast azimuthal integration
  - Sinograms saved in HDF5 format
- Small Angle Scattering
  - Image correction and fast azimuthal integration
- Full Field XAS
  - Image correction (dark, flat)
  - Image alignment (offset measurements by FFT)
  - HDF5 output

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



MXv1 Characterisation v1.2





EDNA Testing Framework

```
[UnitTest]: ##########################################################################
[UnitTest]: Result for EDTestSuiteKernel : SUCCESS
[UnitTest]:
[UnitTest]:  Number of executed test suites in this test suite : 2
[UnitTest]:
[UnitTest]:
[UnitTest]:   Total number of test cases executed with SUCCESS : 14
[UnitTest]:   Total number of test cases executed with FAILURE : 0
[UnitTest]:
[UnitTest]: Total number of test methods executed with SUCCESS : 48
[UnitTest]: Total number of test methods executed with FAILURE : 0
[UnitTest]:
[UnitTest]:                                    Runtime : 25.847 [s]
[UnitTest]: ##########################################################################
```

EDNA Testing Framework

```
[UnitTest]: ##########################################################################
[UnitTest]: Result for EDTestSuitePluginUnitAll : SUCCESS
[UnitTest]:
[UnitTest]:  Number of executed test suites in this test suite : 5
[UnitTest]:
[UnitTest]:
[UnitTest]:   Total number of test cases executed with SUCCESS : 83
[UnitTest]:   Total number of test cases executed with FAILURE : 0
[UnitTest]:
[UnitTest]: Total number of test methods executed with SUCCESS : 202
[UnitTest]: Total number of test methods executed with FAILURE : 0
[UnitTest]:
[UnitTest]:                                    Runtime : 122.892 [s]
[UnitTest]: ##########################################################################
```

EDNA Testing Framework

```
[UnitTest]: ######################################################################
[UnitTest]: Result for EDTestSuitePluginExecuteAll : FAILURE
[UnitTest]:
[UnitTest]:  Number of executed test suites in this test suite : 5
[UnitTest]:
[UnitTest]:
[UnitTest]:   Total number of test cases executed with SUCCESS : 124
[UnitTest]:   Total number of test cases executed with FAILURE : 8
[UnitTest]:
[UnitTest]:
[UnitTest]: OBS! The following test methods ended with failure:
[UnitTest]:
[UnitTest]:   EDTestCasePluginExecuteControlCharForReorientationv2_0_noKAPPA_ :
[UnitTest]:     testExecute :
[UnitTest]:       Plugin failure assert: should be False, was True FAILURE: Expected different from obtained - identifier
/mntdirect/_scisoft/users/svensson/tmp/EDTestSuitePluginExecuteAll_20110114-093729/tmpPW7olu
[UnitTest]:
 ...
[UnitTest]:
[UnitTest]:
[UnitTest]: Total number of test methods executed with SUCCESS : 129
[UnitTest]: Total number of test methods executed with FAILURE : 8
[UnitTest]:
[UnitTest]:                                   Runtime : 1108.997 [s]
[UnitTest]: ######################################################################
```

EDNA Testing Framework

Checking out EDNA Repository... to revision 2758
Tests using the default python under linux: Python 2.6.5
Launching EDTestSuiteKernel with /usr/bin/python .....SUCCESS! 16.411 [s]
Making EDNA-kernel tarball distribution.....
Launching EDTestSuiteCCP4v0 with /usr/bin/python .....SUCCESS! 16.466 [s]
Making CCP4v0 tarball distribution.....
Launching EDTestSuitePluginExecPlugins with /usr/bin/python .....SUCCESS! 920.778 [s]
Making ExecPlugins tarball distribution.....
Launching EDTestSuiteBioSaxs with /usr/bin/python .....SUCCESS! 49.655 [s]
Making BioSaxsv1 tarball distribution.....
Launching EDTestSuitePluginUnitMXPluginExec with /usr/bin/python .....SUCCESS! 57.916 [s]
Launching EDTestSuitePluginExecuteMXPluginExec with /usr/bin/python .....SUCCESS! 365.850 [s]
Launching EDTestSuitePluginUnitMXv1 with /usr/bin/python .....SUCCESS! 73.398 [s]
Launching EDTestSuitePluginExecuteMXv1 with /usr/bin/python .....SUCCESS! 3040.207 [s]
Making MXv1 tarball distribution.....
Tests using jython under linux: Jython 2.5.2rc2
Launching EDTestSuiteKernel with /home/tester/bin/jython .....SUCCESS! 80.371 [s]
Tests using the default python under : WinePython 2.7.1
Launching EDTestSuiteKernel with /home/tester/bin/pythonw .....SUCCESS! 44.725 [s]

How EDNA will evolve in the future

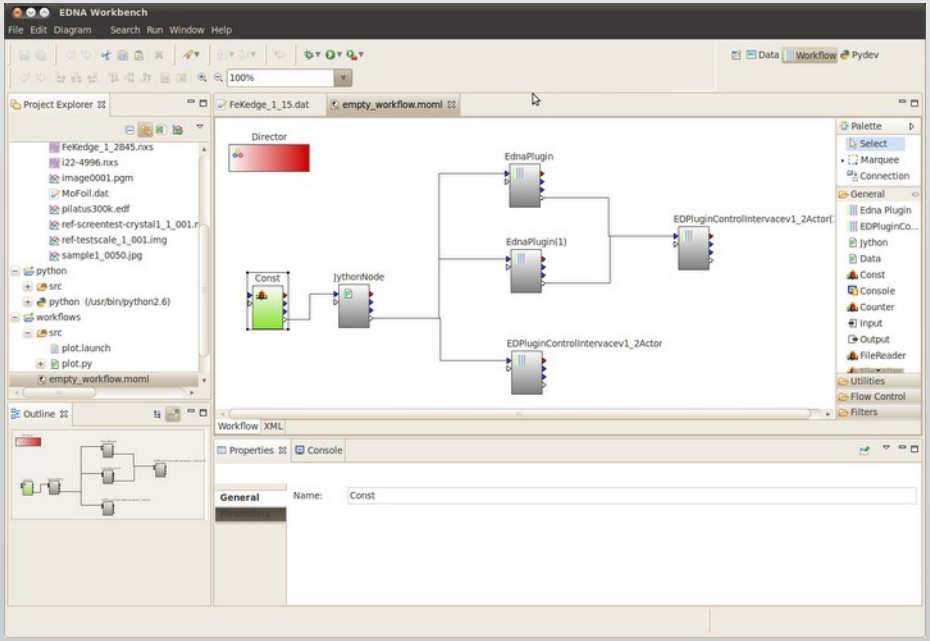
- Common EDNA developments (Kernel):
  - Improvements of the data model framework (in progress)
  - Improvements of logging (in progress)
  - Full support of Windows and MacOS (in progress)
  - Enhanced support of grid engines / job schedulers
  - Improved documentation (plugin use cases)
  - Graphical workflow editor (Data Analysis Workbench)
- Scientific developments:
  - MX further enhancements of characterisation (kappa, XDS etc)
  - MX auto processing wrappers
  - Biosaxs data analysis (EMBL Hamburg software suite)
  - Tomography
  - More to come...

How to run EDNA on a cluster with a job scheduler

- EDNA is optimised for parallel execution of processes/plugins:
  - Thread safe parallel execution of plugins
  - Automatic synchronisation
  - Automatic workload limitation
- Implemented : EDNA TANGO server
  - No job scheduler → no load balancing
- Today the EDNA kernel provides a limited support for running workflows on a cluster with a job scheduler:
  - Works only if the program is started by EDNA in a script
  - Call to grid engine submission must be blocking (sun/oracle grid engine "-sync y")
- Future : support for any job scheduler (sun/oracle, torque, condor, pbs, oar etc)
  - Synchronisation through sockets?
  - Persistent EDNA plugin launchers?

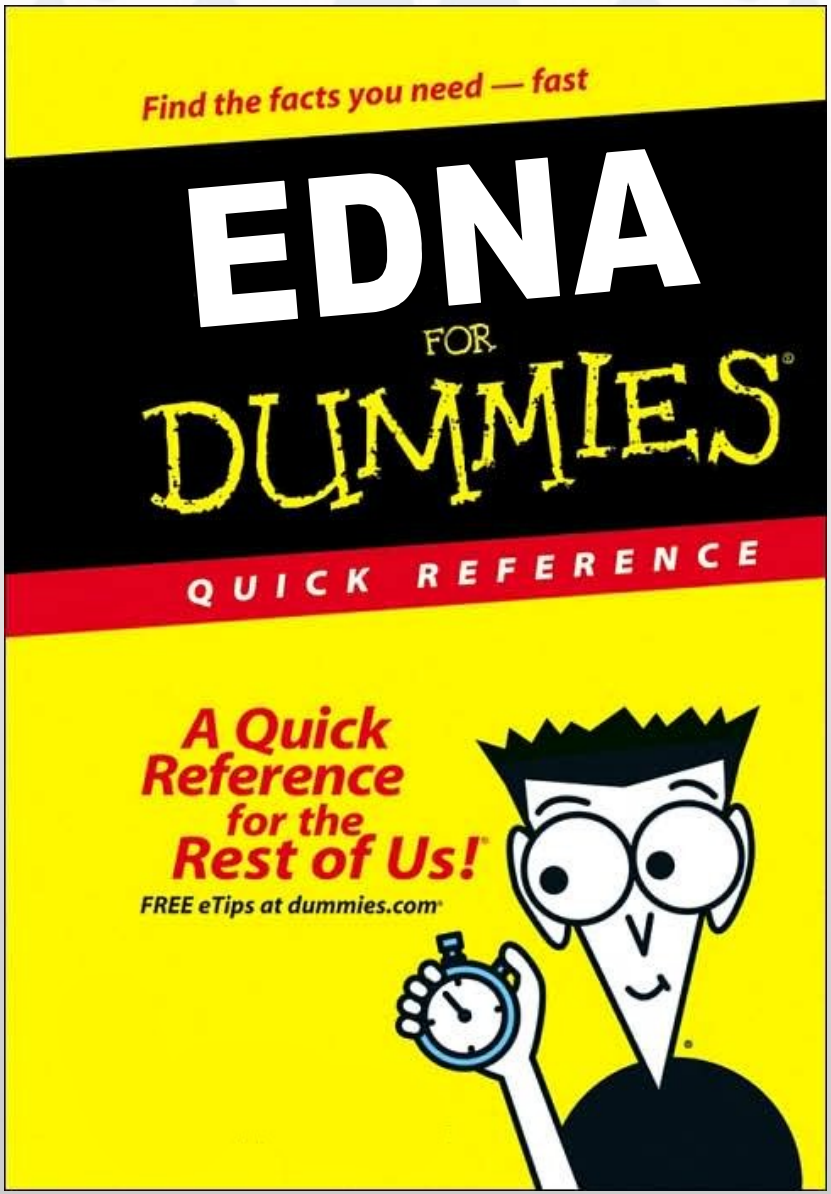
What the EDNA GUI could look like

- EDNA has no GUI framework
- All EDNA workflows must be executable on the command line
- Possible solution for a generic EDNA GUI:



- 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

Documentation!



EDNA Documentation

- Available today :
  - Data models (png)
  - Automatic API doc generation
  - Wikipages with developers' “How-to”s
  - Minutes / presentations of previous meetings, code camps etc
- Planned :
  - Automatic plugin documentation repository (use cases etc)
  - Workflow documentation (workflow tool)

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