One year of development with EDNA

Focus on non-MX applications
Layout

• My vision of EDNA as a developer
  • Strength
  • Weaknesses

• Experience of diffraction tomography
  • Description of the experiment
  • Solutions based on SPD & HDF5
  • Spreading of the solution (ID11, ID13)

• Other application: Saxs / BioSaxs

• Versatility in developments
  • EDNA tool box
  • Documentation, training sessions, code camps
One year of development with EDNA

Strength & Weaknesses
Strength of EDNA

• EDNA is a robust pipe-lining tool for on-line data analysis
  • It has been tested with thousands of tasks at once
• EDNA allows hi-performances
  • Multi-threaded implementation
• EDNA relies on data-models
  • Visual communication with scientists
  • Automatic bindings with the code
• EDNA has a strong testing framework
  • Unit & Execution tests
  • Non regression test before nightly builds
• EDNA is efficient to program
  • Plugin generator for execution plug-ins based on the data-model
  • Re-use of plug-ins already written by others: EDNA-Toolbox
• EDNA is an international collaboration (with DLS, CCP4, ...)
Weaknesses of EDNA

• Technical weaknesses:
  • Learning curve is (too ?) steep
    • After one year, most of it is justified
  • Multi-threading limited by the GIL in C-Python
    • Issue only in pure python plugins
    • No problem if the GIL is released (Numpy, external programs ...)
  • Databinding does not (yet) allow the transmission of large arrays
    • Will be addressed in next code camp.

• Communication weaknesses:
  • Few scientists know EDNA, mainly IT staff.
  • Too much MX-related for most people not involved in the project
  • Lack of documentation (even if better since last code-camp)
  • Lack of manpower in the project working on the kernel
Diffraction tomography

Online pre-processing for azimuthal integration

Generation of sinograms with (live) ROIs
ID22 – Fluorescence-Diffraction Tomography

Beam

Diffraction

CCD

Sample

Fluorescence Detector

1cm

 Courtesy of V.A. Solé

Jérôme Kieffer - EDNA non-Mx
Original idea: 20h data acquisition, 30h treatment

Azimuthal Integrations
Fit2d software

Ny×Nω Diffraction Images

Ny×Nω Diffraction Patterns

Reconstruction
- Capillary
- Ferrite
- sp3
- Calcite
- Cubic

PyMca software

Phase Sinograms

Sum Pattern

Sum Sinogram

Acknowledgements: Pierre Bleuet CEA - Grenoble
First non-MX application of EDNA @ESRF

Diffraction tomography on ID22

• Mesh 60 x 170 (10,000 images)

• All data processed on the fly:
  • Meta-data extraction
  • Dark-noise & flat field intensity correct.
  • Tilt & distortion corrections
  • Azimuthal integration
    • With fit2D (2 plugins)
    • With SPD (4 plugins)
  • Sinogram generation as 3D objects in HDF5,
    • live ROI with PyMca

• Performances: full treatment in 40 mn
  • 1.5 sec / image with SPDCakev1.5
  • EDNA processed 10k images @ 4.3 images/second (8-core, 7x speed-up)
Online data Analysis for NINA

• Focus on most CPU demanding application: DiffractionCT
  FReLoN2k
  (EDF, 16bit UInt, LittleEndian)

Integration of many tools
Need plugin gestion

\( \mathbf{N} \times \omega \) 2D Diffraction Images

Azimuthal integration

- Fit2D
- SPD
- XOP
- XRDUA
- TADM
- Saxs Angle
- SciPy

\( \mathbf{N} \times \omega \) 1D diffraction patterns
Evolution of the SPD integration

- **SPDCake v1.0:**
  - Lookup-table re-generated for each image, no tilt ...
  - 9 sec per 4M image

- **SPDCake v1.1:**
  - Lookup-table re-generated for each image, no tilt

- **SPDCorrect v1.0:**
  - Many SPD processes under control (configured)
  - Tilt is integrated in the distortion of the detector
  - 7 sec for initialization, 0.5 sec for each new image to correct

- **SPDCake v1.5:**
  - Based on SPDCorrect v1.0
  - 7 sec for initialization, +1.5 sec for each image to integrate
  - Compatible with many detector in a single experiment
Tilt as distortion of image ...

...That can be stored in a spline file

$B'$ (x,y)

$\phi$

$\begin{bmatrix}
  x \cos \phi \\
  y \\
  -x \sin \phi
\end{bmatrix}$

$d'x \cos \phi / (d - x \sin \phi)$

$d'x / (d - x \sin \phi)$

0

$d = \text{Distance sample to detector}$

Fit2D definition not SPD

x (to the center of the ring)

y (to the source)

z (to the source)
Spreading of the SPD-solution

• **Soft Condensed Matter Nanofocus beamline (ESRF ID13)**
  - SPD plugin tested with the online mode (EDNA Parallel Execute)
  - 10 frames per second (512 x 512 image)
  - 2 SPD processes under control on a 4-core computer (Coral)
  - Image correction (dark, flat, distortion) + azimuthal integration

• **Materials science beamline (ESRF ID11)**
  - SPD plugin tested with the offline mode (EDNA Parallel Execute)
  - Images coming from 3 different detectors
  - Selection of the right process for correction and azimuthal integration
Image writer issue with multi-threading

EDNA is massively multi-threaded !!!
  • Many instances of a single plugin writing at the same time
  • Some writing are missed due to consequent access

• Solution found:
  • Modify only a class attribute in memory → only one object
  • Then write it down on disk using a class method (with lock).

• HDF5 implementation:
  • Write the sinograms as map of spectra:
    • Store in a 3D dataset (x, y, spectrum)
  • Allows online ROI selections and sinograms creation
Visualization of HDF5 files for DCT

• Images stored as a stack:
  3D: (Image, x, y)

  Visualization with PyMca

• Map of Spectra:
  • 3D: (x, y, spectrum)

Live ROI selection for the sinograms with PyMca

• Data visible with HDFview
Other Application
Small Angle Scattering:
Application to proteins in solution
EDNA-Preprocessing
Overview of the preprocessing pipeline

- BSxCube
  - Normalization
  - Azimuthal integration
  - Averaging of frames

ESRF ID14-eh3
Focus on EDPluginBioSaxsNormalizev1_0.py

- Raw Image + Metadata (Pilatus)
- WaitFile
- SaxsMac
- Add Metadata
- Normalized Image
Focus on EDPluginBioSaxsAzimutIntv1_0

Normalized Image (+ Metadata)

WaitFile

(Get) Metadata

SaxsAdd

(Set) Metadata

SaxsAngle

SaxsCurves

Masked Image

Integrated Image

Integrated Spectrum
Focus on EDPluginBioSaxsAveragev1_0

Integrated Images + Metadata

WaitMultiFile

(Get) Metadata

SaxsMac

(Set) Metadata

SaxsCurves

Averaged Image

Averaged Spectrum
Edna toolbox

• General Execution plugins: 70 plugins
  • Generic command line execution
  • HDF5 writers for stack of images, map of spectra
  • Image conversion, movie making …
  • Conditional branching, accumulator of information
  • Synchronization of tasks

• 5 real applications available from repository
  • BioSaxs
  • Ccp4 (DEMO)
  • DiffractionCT
  • Dimple
  • MX v1 & v2
  • Raw photography development (DEMO)
Coming next: Tutorial & Code Camp in October

• New tutorial
  • Easier subject: Raw digital photography development pipeline
  • Testable and runnable during the training (just need a digital camera)
  • Making use of the EDNA-toolbox (rely on existing code)
  • Tutorial for scientists with skills in Python and developers interested in ODA

• Code Camp about:
  • Datamodels (go away from Enterprise Architect)
  • Data-bindings (allow the transfer of matrices, units)
  • Per plugin logging
  • Execution under Windows
Acknowledgements

• ID11 (Materials Science)
  • Jon Wright
• ID13 (NanoBeam)
  • Manfred Burghammer
• ID14-eh3 (BioSaxs)
  • Louiza Zerad, Adam Round, Thomas Boeglin
• ID22 (NiNa)
  • Alexander Rack & Remy Tucoulou
• SPD developers
  • Peter Boesecke & Rainer Wilcke
• Online data analysis workgroup
  • Cyril Guilloud, Armando Solé & Olof Svensson
Diffraction CT implementation in EDNA

- **PyTango**
- **EDPluginControlDiffractionCTv1_0**
- **EDPluginControlDCTReadHeadersv1_0**
- **EDPluginControlDCTPowderIntegrationv1_0**
- **EDPluginControlDCTWriteSinogramv1_0**
- **EDFReadHeaderv1_0**
- **EDF**
- **Metadata**
- **Filename**
- **XML string**
- **PowderIntegrator**
- **Borg2DImageWriter**
- **EdfFile**
- **FIT2DCakev1_0**
- **FIT2DCakev1_1**
- **SPDCakev1_0**
- **SPDCakev1_1**
- **Fit2D**
- **SPD**
- **Nice**