

Synchrotron SOLEIL

Alain BUTEAU : Software for Controls and Data Acquisition group manager
Majid OUNSY : Responsible for « High Level Applications » development

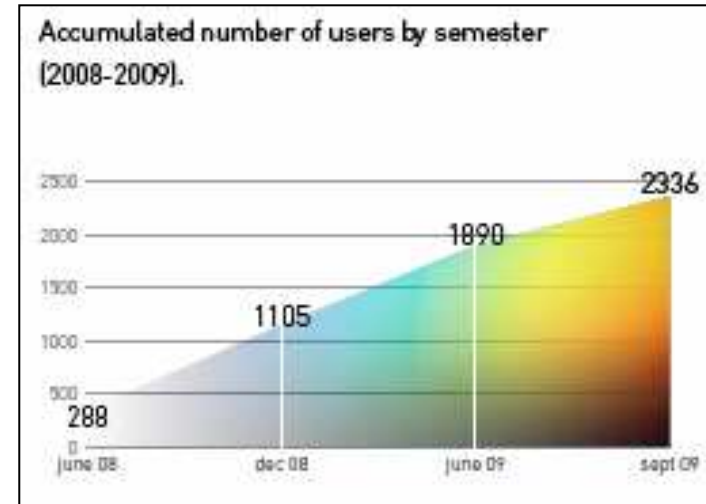


- ❖ NeXus at SOLEIL
- ❖ Limitations and solutions
- ❖ Next steps of software development
- ❖ Conclusion

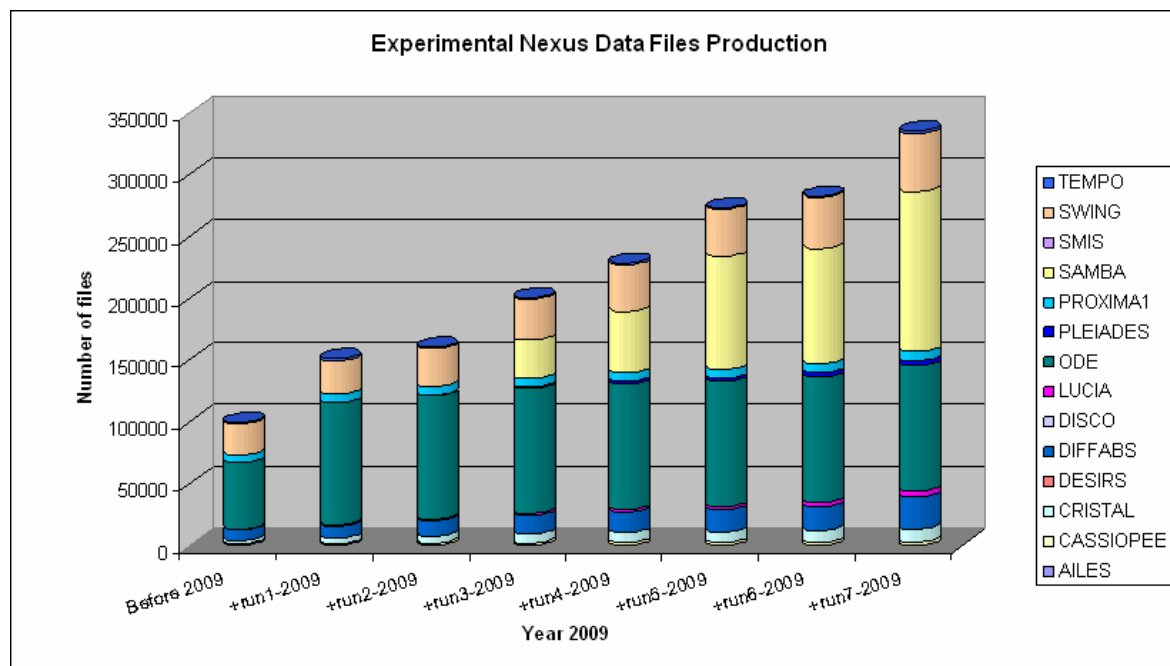
How SOLEIL faces the Data Format challenge using NeXus on a large scale basis

A few reminders on SOLEIL experimental data files production

- ❖ SOLEIL is **in operation since 2007**
- ❖ About 20 beamlines in operation
 - *14 of them open to external users*
- ❖ For each beamline, the daily volume of files ranges from a few Mbytes up to 100 Gbytes
 - This volume is increasing with:
 - *2 dimensional detectors used instead of punctual detectors*
 - *Continuous (i.e without motor stops) scans*
 - *Soon: 2D pixel detectors (XPAD, PILATUS)*
- ❖ A great diversity of scientific applications:
 - *physics, chemistry, new materials, environmental science, biology, ...*
- ❖ Which means diversity of detectors, acquisition process, data volumes, data lifetime policies



- ❖ Using NeXus as the SOLEIL standard Data Format was an early decision taken in **2003**
- ❖ It is nowadays a “de facto” standard on most beamlines
 - ✓ 10 of the 14 open Beamlines are storing NeXus files
 - ✓ Mid-November 2009 : about **350 000 NeXus files**



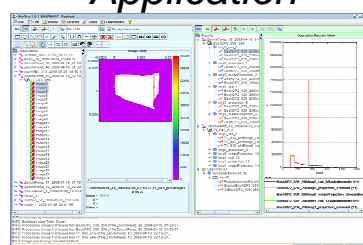
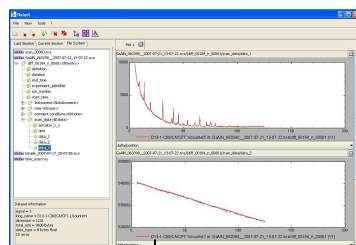
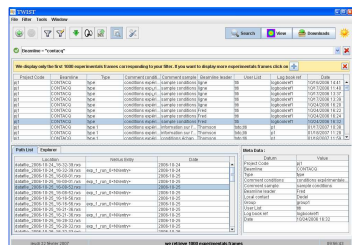
NeXus Files choice : Are we happy ?



File retrieval

File browsing

SAXS Data Analysis Application

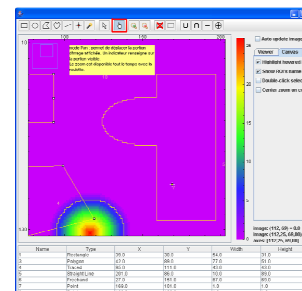
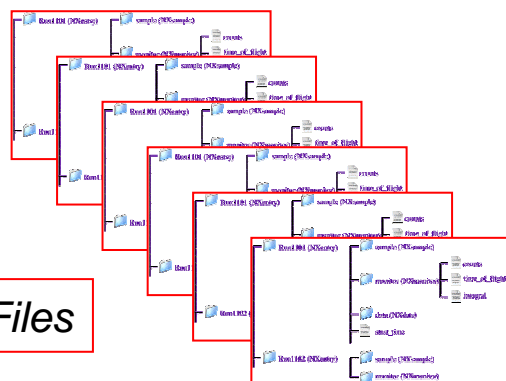


NeXus Application Interface

- NeXus is a good and efficient storage format
- Thanks to a unique API and a « SOLEIL standardized internal data organization », we could :

- ✓ develop common software solutions
- ✓ Decouple the development of Acquisition softwares from Data Analysis software

SOLEIL NeXus Files

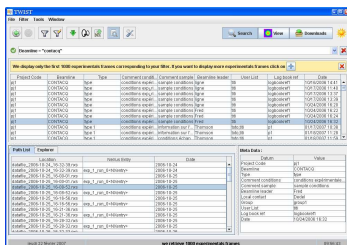


The COMETE library of data visualization components

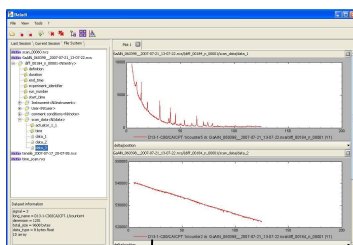
<http://comete.sourceforge.net>

NeXus Files choice : Are we happy ?

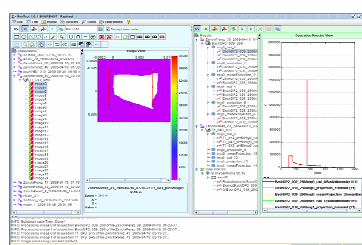
File retrieval



File browsing



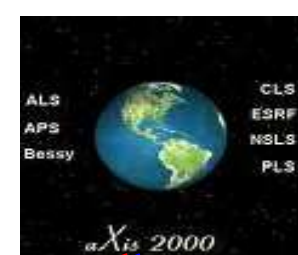
SAXS Data Analysis
Application



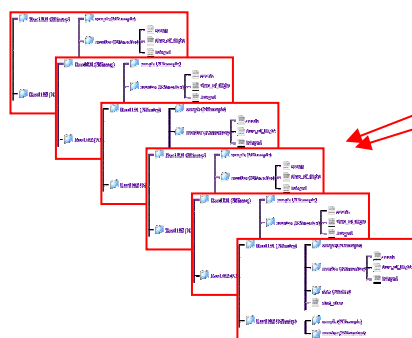
Data Analysis
Application B



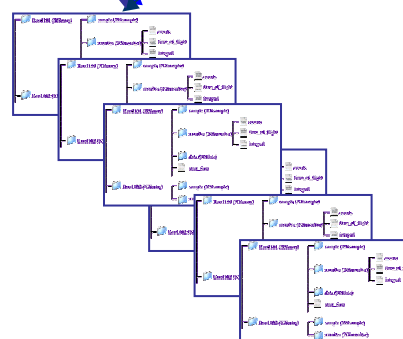
Data Analysis
Application C



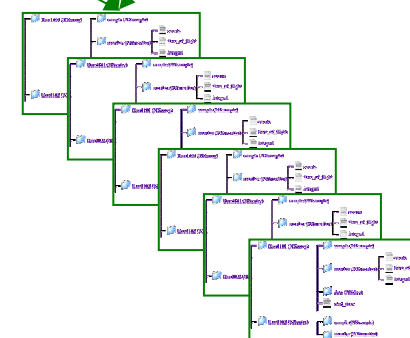
NeXus Application Interface



SOLEIL NeXus Files



ESRF Files



DESY Files

Which solutions are possible ?

❖ The MAHID group asked :

- ✓ Find solutions to data format issues from the **data analysis point of view** ?
- ✓ Put in **common** different **algorithms** for analyzing data ?
- ✓ Find the most suitable ways to **exchange data** ?

❖ The foreseen solutions :

- ✓ Choose Nexus/HDF5 data format

Why not ? But it's not enough

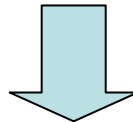
- ✓ Define a standard internal data file structure for experimental data storage

✓ **It's a complex process, involving :**

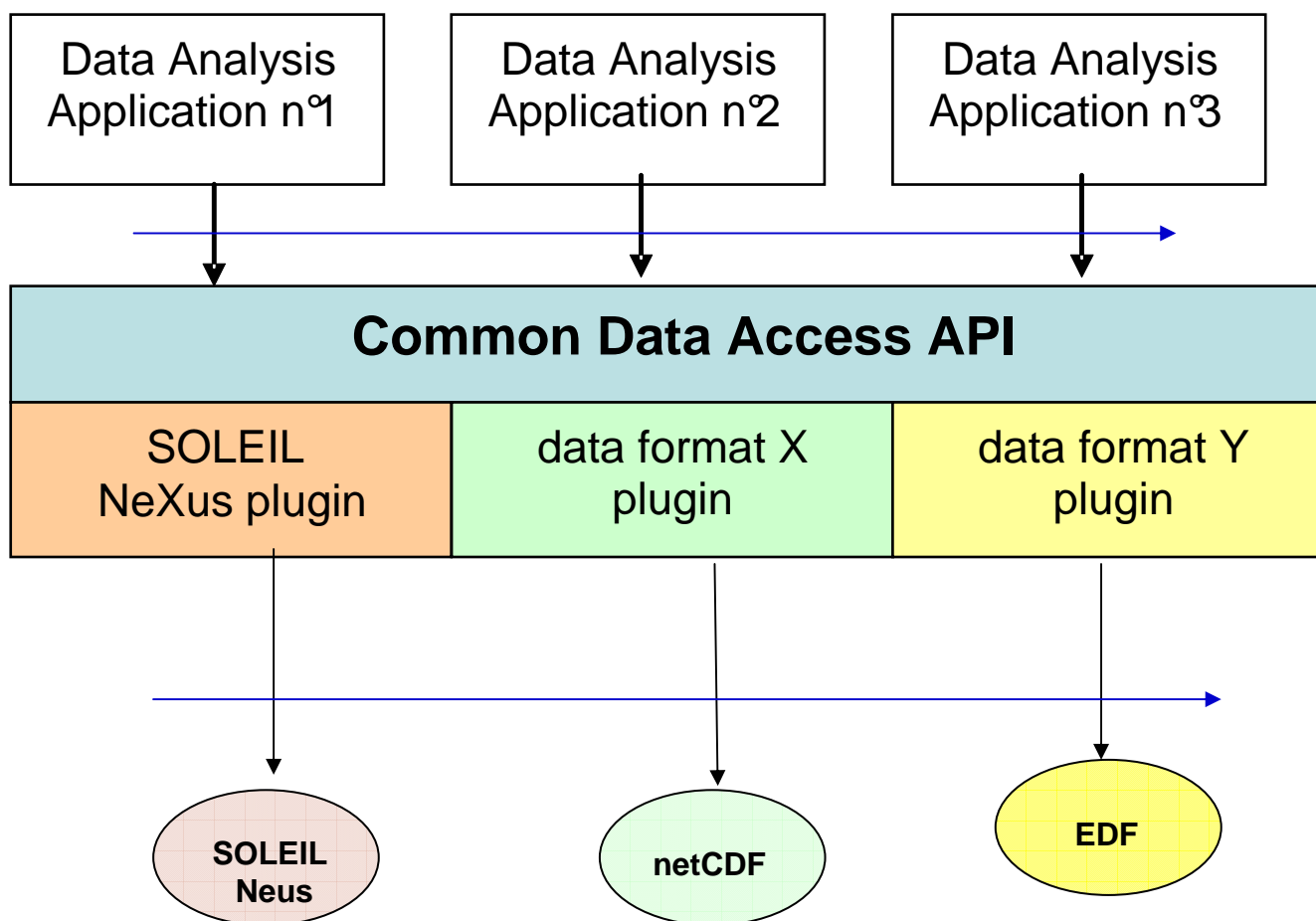
- ✓ **Many institutes**
- ✓ **many software developers**
- ✓ **many existing data format and files**

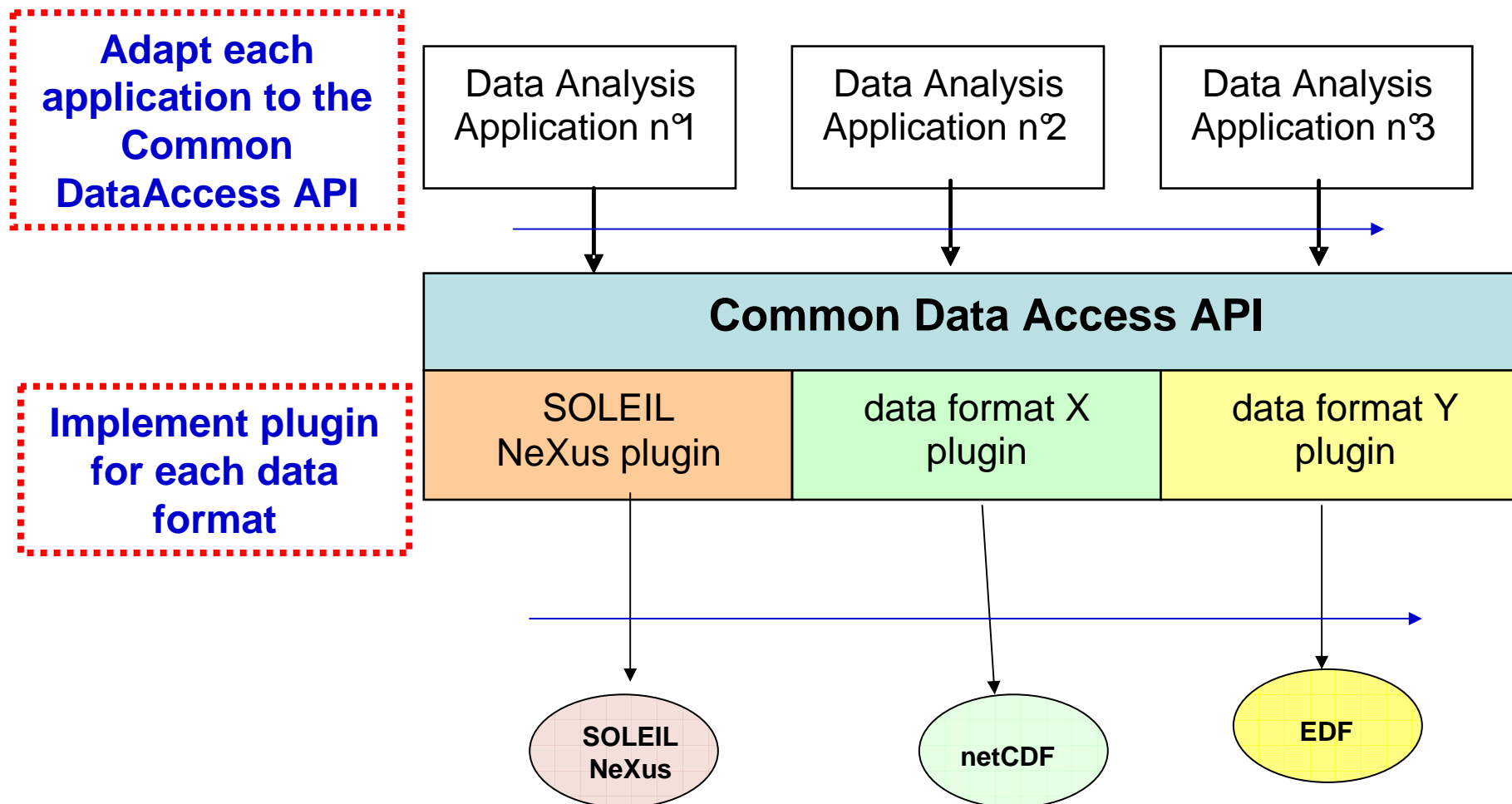
❖ Proposal from Gerd Wellenreuter (MAHID Group) :

- ✓ Access data thanks to **attributes/tags**
- ✓ Define a standard way to access synchrotron experimental data



- ❖ We only agree on an “**Abstract Software Interface**” defining a Data Access Model
- ❖ We let **Institutes implement this Software Interface** to deal with their own data files





- ❖ It is a light process
 - ❖ *Developing a plugin costs a few weeks of work*
 - ❖ *Adapting an application costs a few weeks of work*
- ❖ It allows to deal with existing files
- ❖ It is an open process
 - ❖ *Newcomers have only to implement the standardised interface*

Next steps for SOLEIL

How will it look like ?

❖ Our aim is to be able to write data analysis source code like this

```
Filehandle file_handle = Factory.getDataset(new File(filename));
```

```
// Get root group
```

```
Group rootGroup = file_handle.getRootGroup();
```

```
// Get sample group
```

```
Group sampleGroup = rootGroup.findGroup("sample");
```

```
// Get sample name
```

```
String sampleName = sampleGroup.findDataItem("name")}
```

```
}
```


Current status and next steps for SOLEIL on the « Common Data Model API » project

- ❖ ANSTO developed a first version of a “Common Data Model API”
 - ❖ SOLEIL enhanced this version to obtain a V 0.1 “demonstration version” of this “Common DataModel API” in java with
 - Plug-in for SOLEIL Nexus Data Format
 - Plug-in for ANSTO HDF5 Data Format
- } Version 0.1 will be demonstrated on Wednesday
- ❖ V 0.1 is a demonstration version which is open to enhancement/extensions by of other interested institutes
 - ❖ *SOLEIL will continue with ANSTO on the API development*
 - ❖ After feedback on the V0.1, SOLEIL will develop plugins for external data format which are interesting for SOLEIL’s scientists

- NeXus choice has been the right one for experimental data storage
 - allowing to address in **an uniform way the diversity** of detectors and experimental techniques
- The “Common Data API” is an opportunity
 - to address in a **uniform way the diversity** of Data Analysis applications

SOLEIL is ready to collaborate with Data Analysis developers to adapt them to the “Common Data Access” architecture

Annexe: code example

From the plugin developer point of view

❖ “Attribute-tag” mechanisms is to be implemented in the plugin to ease the application development

```
// Read file
FileHandle file_handle = Factory.getDataset(new File(filename);

IDataContainer container = new DataContainer(dataset);

// Programmatically code up the dictionary (key and path)
Map<String, String> dictionary = new HashMap<String, String>();

dictionary.put("sampleName",    "/entry1/sample/name");
dictionary.put("monitor1Counts", "/entry1/monitor/bm1_counts");
IPathResolver pathResolver = new DictionaryPathResolver(dictionary);

container.setPathSolver(pathResolver);
```