Offline Scientific Analysis

OSA 5.0/5.1

Stéphane Paltani

on behalf of many people...

OSA in context

Science data
Housekeeping
Auxiliary data
Calibration data

Instrument Configuration
Catalog(s)
User inputs

Technical Processing
telemetry decoding, auto-calibration, time stamps, (corrections, GTIs, deadtimes)

Scientific Analysis
(corrections, GTIs, deadtimes), catalogs, background, binning, image deconvolution, source search, source spectra, source lightcurves

Images, Source Lists, Spectra, Light Curves

FTOOLS, XSPEC, XRONOS, private S/W
further analysis, model fitting, ...

Scope of OSA

What OSA software is supposed to do:

- Corrections, if needed
- Dead-time calculation, good-time interval selections
- Image reconstruction (including mosaics)
- Source identification and extraction
- Count extraction (spectra, light curves)
- Handling of Instrument Characteristics (IC)

What OSA software does not do:

- Image analysis
- Spectral fitting
- Timing analysis (period search, FFT, ...)

OSA constraints

Data are complex (detectors, coded-mask, ...)

- No true imaging, except for OMC
- Analysis of Integral data requires many steps, algorithms
- Scripts are needed to ease the analysis

Data sets are huge, cut into a large number of pointings, many different files

- Size is already a serious difficulty (disk space, CPU time,...)
- Users must be able to group pointings arbitrarily
- Users must be isolated from the data complexity

Many calibration files, instrument model data, complex versioning

- Must be handled through a (kind of) "database"

OSA components

Generic tools

- Developed and maintained by ISDC
- FTOOL-like

```
dal_list dol=my_file.fits[1]
```

- Written in C

Instrument-specific executables

- Developed and maintained by Instrument Teams
- FTOOL-like
- Some components written in F90

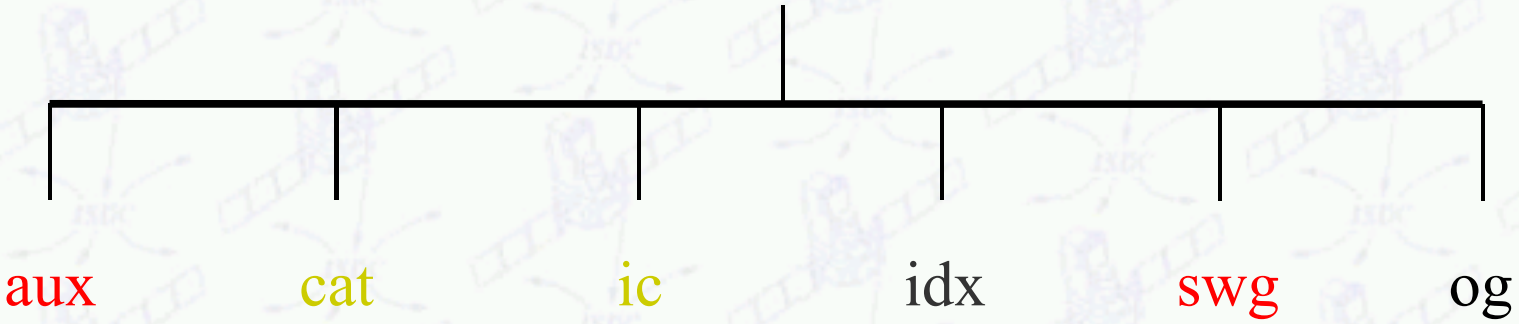
Analysis scripts

- Developed and maintained by ISDC/Instrument Teams
- C++ scripts based on ROOT and `isdcroot`
- Work in script, command-line or GUI modes

OSA installation & beyond

- Binaries of OSA >= 5.0 are available for i386/Linux, SPARC/Solaris and PowerPC/MacOSX. OSA is now very easy to install
- Sources for the rest of the Unix world. More involved.
- Does not require any commercial software anymore
- 30 libraries / 180 executables with documentation
- File repository with precise structure must be created by the user

File repository



- aux: Auxiliary data; attitude, orbit, program, ... (downloaded from the Archive)
- cat: Source catalogue (downloaded with OSA)
- ic: Instrument characteristics; calibration, instrument models, ... (downloaded with OSA)
- idx: Index tables; sort of databases...
- swg: Science Window data; instrument data (downloaded from the Archive)
- og: Observation data; products generated by OSA

OSA commonalities

ISDC has defined several concepts used for all instruments

- Data formats (~ 1'300 "templates")
- Support software libraries
  - PIL: Parameter Interface Layer (IRAF-like parameters)
  - RIL: Reporting Interface Layer (Logs, error messages)
  - DAL: Data Access Layer (plus DAL3XXX, ISDC-specific avatars)
- Groups
  - Science Window Groups
  - Observation Groups
  - Index tables (plural of "index")
- Scripts and GUIs

Parameter Interface Layer

PIL essentially reimplements IRAF/FTOOL parameter syntax

- Each component (executable/script) has a parameter file

```
dol,s,q,"",,,"Input DOL"
extname,s,h,"",,,"Extension name, wild cards allowed"
exact,b,h,no,,,"Exact?"
```

- Parameters can be queried or hidden

- Parameters can be set on the command line

```
dal_list dol=my_file.fits[1]
```

- Try:

```
dal_list --help
```

Reporting Interface Layer

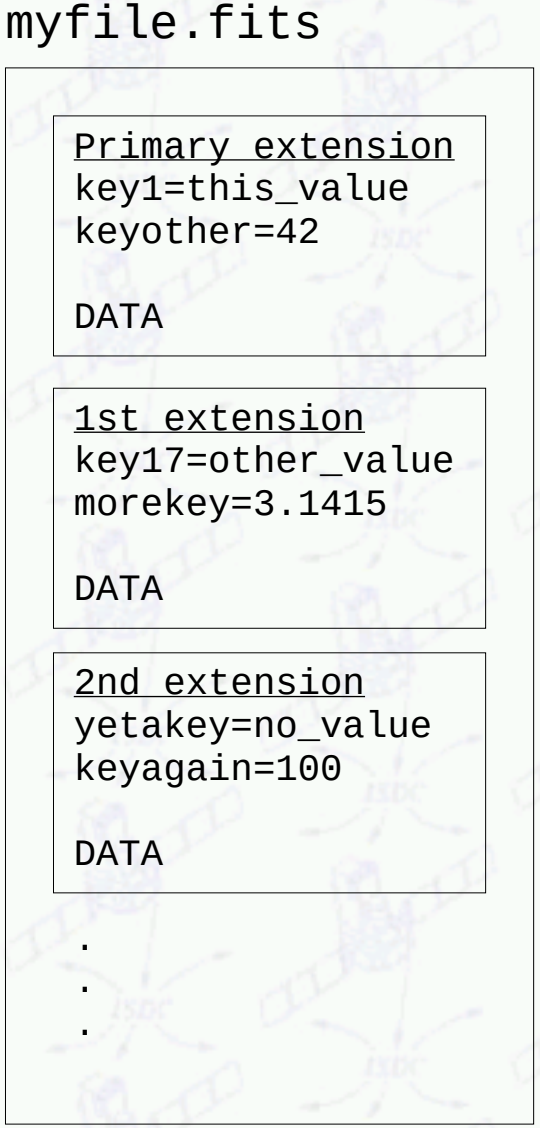
- RIL provides a common logging/error messaging interface to the developpers

- For the users, allows easy control of the output (screen vs log file), whether in interactive or in batch mode

```
isdcprv18:~ 119 > dal_verify crab_pha2.fits
Log_1  : Verifying crab_pha2.fits[JMX1-PHA2-PE,1,BINTABLE]
Error_1:    input date string has illegal format:
Error_1:    UTC_format
Warn_3 : DATE-OBS keyword values (UTC_format) is not valid
Warn_3 : DATE-END keyword values (UTC_format) is not valid
Error_1: Total number of errors: 2, Warnings: 2
Warn_2 : About to flush all DAL buffers, an error has occurred.
Warn_2 : There may be data loss, and the program may crash!
Error_2: there is at least an error
Error_2: Task dal_verify terminating with status -35560
```

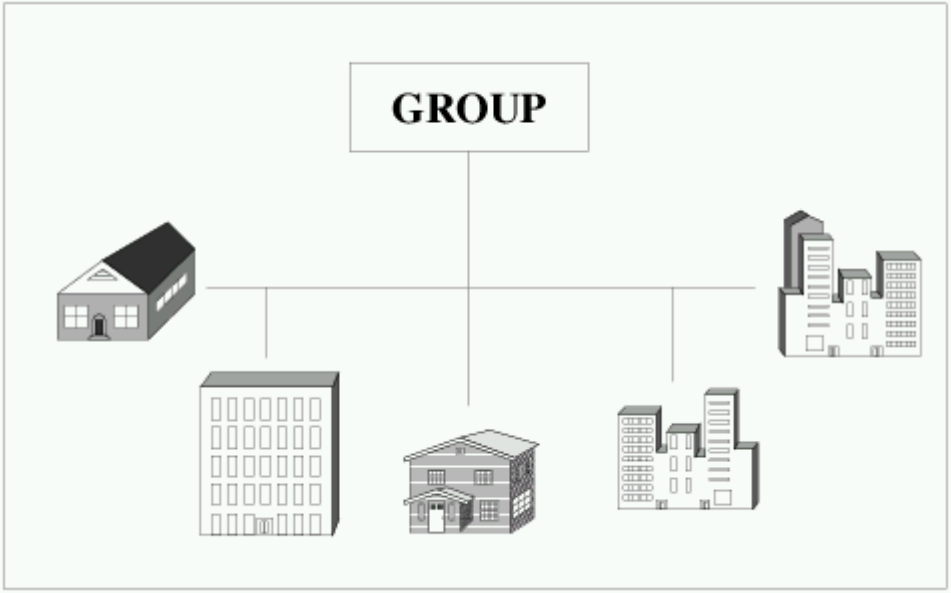
Data structures & Data Access Layer

- DAL is a layer on top of `cfitsio`
- FITS allows several "extensions" to be part of the same file: Data structures
- DAL/OSA always require data structures
  `myfile.fits ---> myfile.fits[2]`
- (actually `...[1]` is the default)
- DAL implements hierarchical grouping
- DAL3 implements index handling



Groups

- FITS allows special data structures that do not contain any data, but only points to other data structure



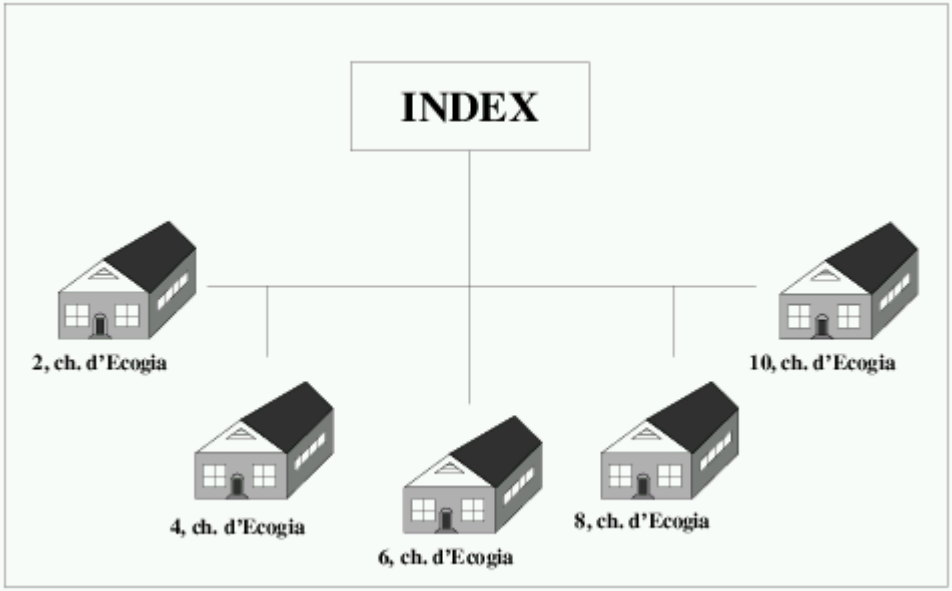
- Best analogy: directory

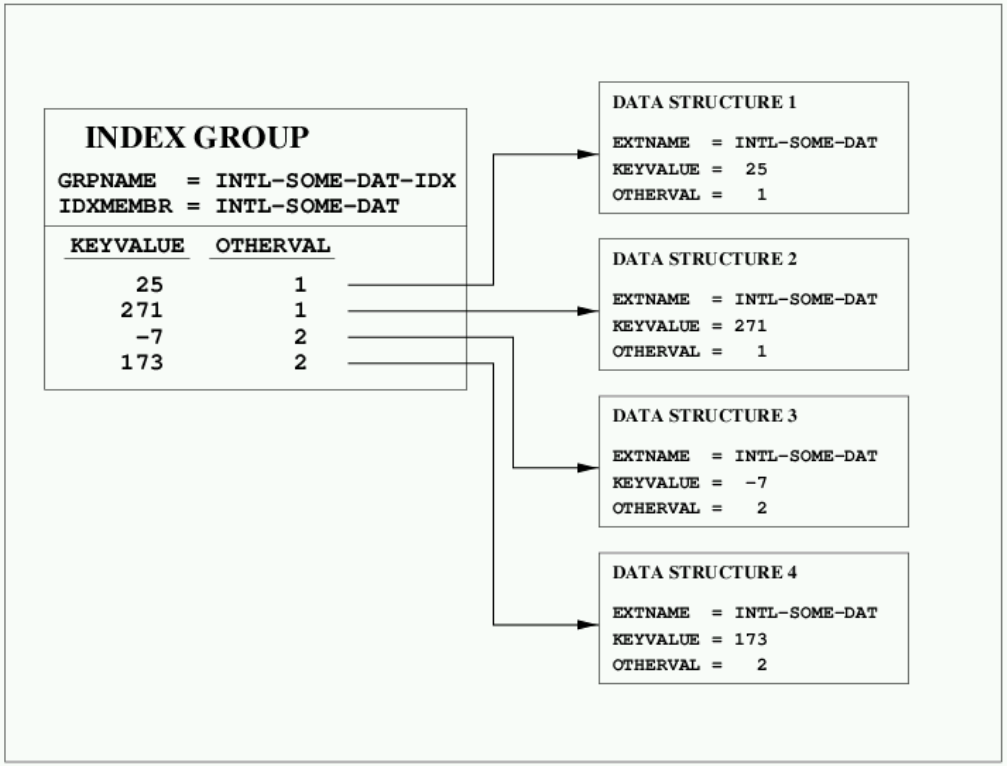
- Science Window groups contain all the data generated by INTEGRAL during a Science Window (pointing, slew,...)

- Observation groups group Science Windows on the basis of user-defined criteria and all the results from OSA. These are the basic blocks of an OSA analysis

Index tables

- Index tables are special that contain several instances of similar data structures with different "metadata"
- Best analogy: databases
- IC files valid for different periods are "stored" in an Index
- Spectra of all sources found in a pointing are "stored" in an Index





Scripts

- The analysis of the data from a single INTEGRAL instrument requires a few tens of executables

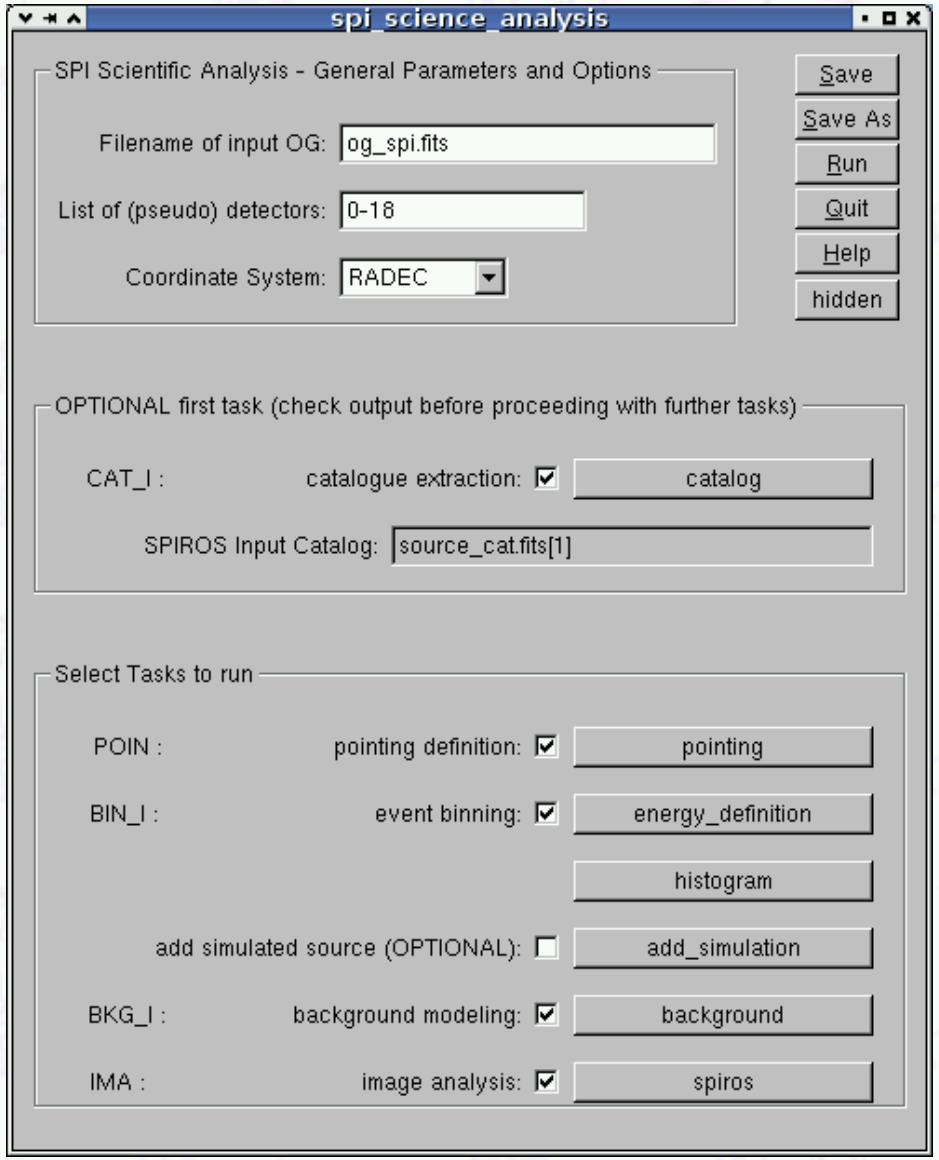
- Scripts are necessary to help users to run an entire analysis in the correct order
  - Steps are called "Level": `COR`, `DEAD`, `GTI`, `IMA`, `SPE`, `LCR`, ...
  - There is complete flexibility in the Level sequence, except order

- ISDC scripts are actually C++ code with `isdcroot`/ROOT classes

- C++ is not mandatory... Advanced users can develop their own scripts in shell, Perl, Python, tcl, ..., but, for the moment, the need to provide scripts in more user-friendly languages that users can reuse, modify is not that obvious...

Graphical user interfaces

- GUIs exist for all instruments
- Simplify parameter filling for interactive sessions, with little loss of capabilities
- Hides "hidden" parameters for the faint of heart
- With OSA 5.1, GUIs really become parameter editors
  - Save command lines
  - Load parameters from scripts
  - Can also be used as regular GUI's!



What next?

- OSA 5.1 is almost there
  - Important SPI bug fix
  - Additional ISGRI ARFs
  - Important improvement of Jem-X responses
  - More clever GUIs
  - Several bug fixes

- OSA 6.0 is in discussion stage
  - Release planned early next year
  - Jem-X spectrum extraction software
  - Other imaging methods for SPI?
  - Phase-resolved spectroscopy
  - Improvements foreseen in calibration of most instruments
  - Bug fixes, usability and miscellaneous improvements