Products in IA

Steve Guest (RAL)

Background

- This definition originates from the Herschel IA Working Group
  - Set up two years ago following the “Tiger Team Report”, outlining a common Herschel IA framework, and endorsed by ESA, ICC managers and PIs.
  - The IA-WG is an offshoot of the Herschel Common Software Development Team (CSDT).
- Design and implementation by Jorgo Bakker and Jon Brumfit (ESTEC)
  - It is a part of the HCSS distribution
  - Designed for use in both Java and Jython
  - Extensive use of Jython “hooks” for convenience
- See also “Specifying Data Products in the Herschel Interactive Analysis System” (SPIRE-RAL-DOC-001964)

Product

- A product is comprised of:
  - A description (an extra “type” will be added soon)
  - Metadata (a few types are required)
  - Zero or more datasets
  - A processing history
- It is independent of any file (or database) format.
- It roughly corresponds to an entire FITS file, with metadata similar to keywords in the primary header.
  - It has more flexibility than FITS
- Readers and writers are used to translate between an IA product and external formats.

Dataset

- A dataset is similar to a product, except:
  - It is always a part of a product
  - It has no required metadata
  - It has no processing history
  - It roughly corresponds to a FITS extension
- The base types of dataset are:
  - A table, with equal length columns of data
  - An array (this is just a wrapper around a numeric array)
  - A composite, i.e. any hierarchy can be constructed
- Higher level data sets can (and will) be defined, e.g.
  - Spectrum
  - Image

Data

- Data items are *arrays*
    - Single data items should be treated as metadata
- Currently supported types are:
    - Boolean (1-3D)
    - Complex (64-bit signed pair)
    - Double (64 bit signed, 1-3D)
    - Float (32-bit signed, 1-3D)
    - Integer (32-bit signed, 1-3D)
    - Long (64-bit signed, 1-3D)
    - String (Unicode, 1-D only)
    - Short and Byte types are expected to follow
- Mathematical operations are performed at the data level

Metadata

- Metadata items consist of:
  - A name (keyword)
  - A value, can be
    - String
    - Boolean
    - Double (64-bit signed)
    - Long (64-bit signed)
    - Date (including time)
  - A description
  - A quantity i.e. units (uses the nanoTITAN quantity library)
- A metadata item corresponds to a FITS header record.

Notes on FITS import/export

- Metadata keywords defined in *dictionaries* are automatically translated when reading/writing FITS files.
  - The keywords are not restricted to FITS rules
  - Keywords in IA should adhere to Java conventions
  - Don’t use the FITS keywords, it’s supposed to be independent!
  - You can add your own dictionary if you like
  - We should endeavour to use the same names for the same things across the project, Herschel-wide as well as SPIRE
- Currently only FITS files written by IA can be imported.
  - Support for reading other FITS files will be added soon

More on FITS dictionaries

- This is the “standard” dictionary:
  - AUTHOR=author
  - CREATOR=creator (*)
  - DATE-OBS=startDate (*)
  - DATE=creationDate (*)
  - EPOCH=epoch
  - EQUINOX=equinox
  - INSTRUME=instrument (*)
  - OBJECT=object
  - OBSERVER=observer
  - REFERENC=reference
  - TELESCOP=telescope
- Keywords marked with (*) above are required in a product, as are endDate and modelName
- Other existing dictionaries are HEASARC, HCSS, SPIRE