A meta-data layer for astronomical archives

Fabien Chéreau, Bruno Rino, Diego Marcos - Virtual Observatory Project Office, ESO
ADASS 2009

The problem

As astronomical archives are getting larger, so is the demand for smarter and richer services from the community. That is how the Virtual Observatory came about: to enable scientists to find and interpret the data they want, without knowledge of how it was obtained, without requiring them to read through the instrument documentation to understand archival data.

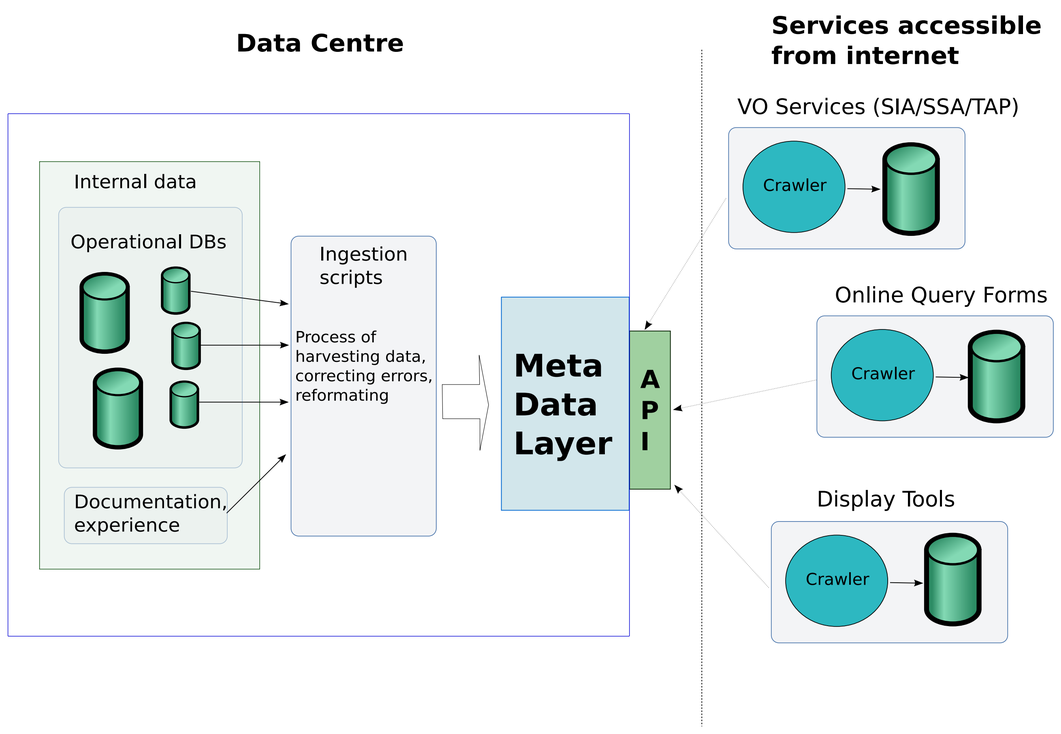
Meta-data is at the heart of the VO. Correct and precise meta-data enable the value of query services, and VO services in particular. However, the design of many large astronomical archives is still primarily focused on supporting acquisition and storage of data. This results in meta-data being spread across a complex mesh of sources (databases, text files, personal knowledge), with varying degrees of quality (in terms of completeness, correctness and precision), and with many operational constraints (one cannot risk locking the database with large queries, incorrect meta-data cannot always be corrected).

This complexity of the data sources alone makes building innovative services a daunting task. And as of 2009 the creation of true cross-archive science query services still remains an unreachable dream.

This leads to the following 2 conclusions: quality assurance should be a reusable effort, from which all (current and future) query services benefit; operational constraints suggests that queries services should be based on separate systems than the observatory operations.

Proposed solution

The proposed solution is built around a meta-data layer that shields the query services from the complexity of the data sources, and shields the data sources from the demands of the query services.



The roles in the data flow are clearly identified and decoupled:
• Ingestion scripts fill the meta-data layer from the operational databases.
• Crawlers update service-specific databases taking information only from the meta-data layer changed since the last crawl.
• Clients perform queries on the service-specific databases.

Query services never hit the operational databases; only the ingestion scripts do. This single access point makes it feasible to define rates of ingestion that do not impact operations.

Part of the ingestion scripts correct and add information not found in operational databases. Concentrating these tasks makes inconsistencies between the operational databases and the meta-data layer clear and manageable.

The data model of the meta-data layer should be flexible. It must cope easily with the rapidly evolving requirements of existing and emerging query services.

Changes in the meta-data itself must be kept. While operational databases might only keep the latest version of metadata, for query services it can be important to easily return to earlier versions.

Service interface

The meta-data layer service API is kept deliberately simple: it allows only to get one meta-data file encoded as JSON and to list the content of the layer (with support for versioning). These 2 features are the strict minimum needed for a crawler to go through the whole content of the exposed meta-data. It is important to notice that there is no query capability at this level.

REST API

/	Return the list of ids of all documents of the meta-data layer
/?op=size	Return the number of elements of the meta-data layer
/?changedsince={ts}	Return the list of ids of documents wich were modified since date ts
/id	Return the document for this id or an HTTP 404 error if there is no document for the id
/id?v={ts}	Return the document for this id at version v or HTTP 404 error if there is no document for the id/version e.g. /id?v=2009-06-16T09:07:05
/id?op=timestamp	Return the last version number of the document or an HTTP 404 error if there is no document for this id
/id?op=history	Return the list of previous versions numbers of the document or an HTTP 404 if there is no document for this id

JSON : Query result format

The content of the meta-data files are returned as JSON files. The files are semi-structured, based on schemas carefully defined for common types of documents (images, spectra, etc...) but also allowing the flexibility to add custom items specific to a subset of documents, e.g. for storing information items specific to a given telescope or institution.


```
{
   "id": "LP175C0685.WFI.2005-12-09T05:12:15.441",
   "type": "image",
   "title": "WFI.2005-12-09T05:12:15.441.fit",
   "publisher": "ESO SAF",
   "collection": "175.C-0685",
   "license": "ESO Data License",
   "creator": "Dupondt",
   "characterization": {
      "spatialAxis": {
         "footprint": {
            "worldCoords": [[[122.17098758933295, -48.631020973582558], [123.02688788...
                            [123.02688788655796, -49.176814585277484], [122.1709875893329...
         },
         "centralPos": [122.59955700853416, -48.90469867250895]
      },
      "temporalAxis": {
         "boundingBox": [53713.216845379997, 53713.218233320003],
      ...
```


A multi-archive use case

The meta-data layer allows to completely decouple the roles of the meta-data providers and the one of the services providers (query service, display service etc..). This unleashes the possibility of building cross-archives services using exactly the same principles (API+crawler) as for the single archive use case, provided that the various data centres adopt the same conventions for the service interface API.

The design and adoption of this API might well be the currently missing corner stone on which to build the rest of the VO.