ICAT Metadata Ingest with python-icat

Rolf Krahl
Helmholtz-Zentrum Berlin für Materialien und Energie

Abstract

- Metadata catalogues are the central element in a data repository used by PaN facilities for providing FAIR data.
- Facilities need to implement custom workflows to ingest the data collected at the experiment into their metadata catalogue.
- python-icat is a Python client library for the ICAT metadata catalogue that may be used to implement facility specific workflows.
- Recent releases of python-icat (version 1.1, with some improvements in 1.2 and 1.3) added a dedicated module for the ingest of metadata from the experiment.
- Leveraging XML techniques, the method provides fine grained control of the metadata that get into the metadata catalogue. At the same time, it is higly customizable to the need of the facility.

ICAT Metadata Catalogue

ICAT is a metadata catalogue to support large facility experimental data, linking all aspects of the research chain from proposal through to publication. It is designed to be used as an index and access portal to a scientific data repository.

Technically, it is based upon a relational database to store the metadata and to allow searching for datasets. The ICAT schema is structured around the central entities *Investigation*, *Dataset*, and *Datafile*. *DatasetParameter* may be attached to *Dataset* to store the physical parameters of a measurement.

python-icat – Python interface to ICAT

python-icat is a Python package that provides a collection of modules for writing programs that access an ICAT service via its API. It is intended to implement the facility workflows around ICAT. Most important features include:

- Clients for ICAT and IDS,
- Define native Python classes to represent the entity object types from the ICAT schema,
- Read configuration from various sources, such as command line arguments, environment variables, and configuration files,
- Build JPQL expressions to search the ICAT server,
- Dump and restore ICAT content to and from a flat file.

ICAT Data Files

python-icat provides command line scripts `icatdump` and `icatingest` that dump and restore ICAT content to and from a flat file respectively.

Originally, the scripts were intended as a debugging tool, to help the admin to verify the exact content in the ICAT server more easily or to populate ICAT test instances with content. Meanwhile, the scripts turn out to be handy in many different situations.

A specific file format, ICAT Data File, has been defined for the scripts. The files are basically a faithful dump of the content according to the ICAT schema: a list of objects with their attributes as defined in ICAT. Variants exist based on XML and YAML. The format provides sophisticated features to serialize and deserialize the relations between the objects.

Use ICAT Data Files to Ingest Metadata from the Experiment?

Facilities need to move the datafiles created during the experiment into their data repository and at the same time ingest the corresponding metadata into the metadata catalogue.

- Obvious question: could we use python-icat's `icatingest` tool for that purpose? E.g. could we write a dedicated ICAT Data File at the experiment and ingest that into ICAT?
- Answer: yes, in principle. But ...
- ICAT Data Files are just too powerful for that use case. They could contain arbitrary ICAT content.
- We want to create new *Datasets* and *DatasetParameters* from the metadata collected at the experiment. We certainly don't want it to create new *Instruments* or *Users* in ICAT. And we want to have control to which *Investigation* newly created *Datasets* are going to be added.
- It would be rather difficult to control the power of the input format if we would use plain ICAT Data Files here.

Metadata Ingest Files

The answer that metadata ingest issue is: Metadata Ingest Files. They are a restricted variant of ICAT Data XML Files, strictly limited to the content that we want to allow in this use case. Most relevant limitations:

- The allowed object types are restricted to *Dataset*, *DatasetInstrument*, *DatasetTechnique*, and *DatasetParameter*.
- The attributes in the object definitions for *Datasets* are restricted to *name*, *description*, *startDate*, and *endDate*.
- Object definitions for *Datasets* can not include references to the related *Investigation* or *DatasetType*. These relations will be added with prescribed values during ingest.
- Object definitions for *Datasets* can reference a related *Sample* only by *name* or by *pid*. A relation of the *Sample* with the *Investigation* will be implied during ingest.
- *DatasetInstrument*, *DatasetTechnique*, and *DatasetParameter* can only relate to *Datasets* defined in the same ingest file.

python-icat `IngestReader` class

python-icat version 1.1 introduces the `icat.ingest` module, providing class `IngestReader`. The class takes an *Investigation* as argument and a Metadata Ingest File as input.

The processing steps can be summarized as:

1. Validate the input against a XML Schema Definition (XSD). This will enforce the restrictions in the Metadata Ingest File format.
2. Apply an Extensible Stylesheet Language Transformation (XSLT) to transform the input into generic ICAT Data File format. This will add the relations and attribute values that should be prescribed during ingest. In particular, it will add a reference to the *Investigation* provided as argument to all *Datasets* in the input.
3. Feed the result of the transformation into the standard ICAT Data File input reader.

Facilities may provide their own XSD and XSLT files to customize the input format and the result.

Implementing Ingest Scripts

python-icat does not provide a command line script for the metadata ingest:

- Usually, the ingest of the metadata is integrated into the workflow that also moves the datafiles into the storage of the repository.
- This may even be required, because a particular order of the steps may need to be observed in order to avoid race conditions in the access to the storage.
- How the datafiles are to be moved into the storage is very specific to the facility. This cannot be implemented in a library package.
- But python-icat provides documentation and example scripts to explain how the `IngestReader` class can be integrated into a typical facility ingest script.

References

- A. Götz et al (2024). Extending the ICAT Metadata Catalogue to New Scientific Use Cases. ICALEPCS 2023. https://doi.org/10.18429/JACoW-ICALEPCS2023-WE3BC007
- R. Krahl and The ICAT project (2024). python-icat – Python interface to ICAT and IDS. Zenodo. https://doi.org/10.5281/zenodo.7564751
- The ICAT project: https://icatproject.org/
- python-icat documentation: https://python-icat.readthedocs.io/