Science & Technology Facilities Council

ICAT Ideas

20/21 Nov 2017

Steve Fisher
<dr.s.m.fisher@gmail.com>

Warning

- I am presenting a set of ideas:
  - I won't have to implement them
  - They are not necessarily consistent with each other
  - Some are very disruptive - but not necessarily crazy

Problems to be addressed

- “Everybody” is happy with Datafiles but some find Datasets inconvenient
  - Hierarchies: bad, trees: good
  - DLS and CLF mainly want a file system
- DataCollections are important for DOIs and provenance but don’t have unique names
- Currently ICAT accepts almost any JPQL query and the authz makes it worse
  - RDBMS should be able to cope but sometimes on huge tables Oracle decides to scan the entire table
  - Need to add ad-hoc indices to cope with problems
- Information in lucene duplicates that in RDBMS

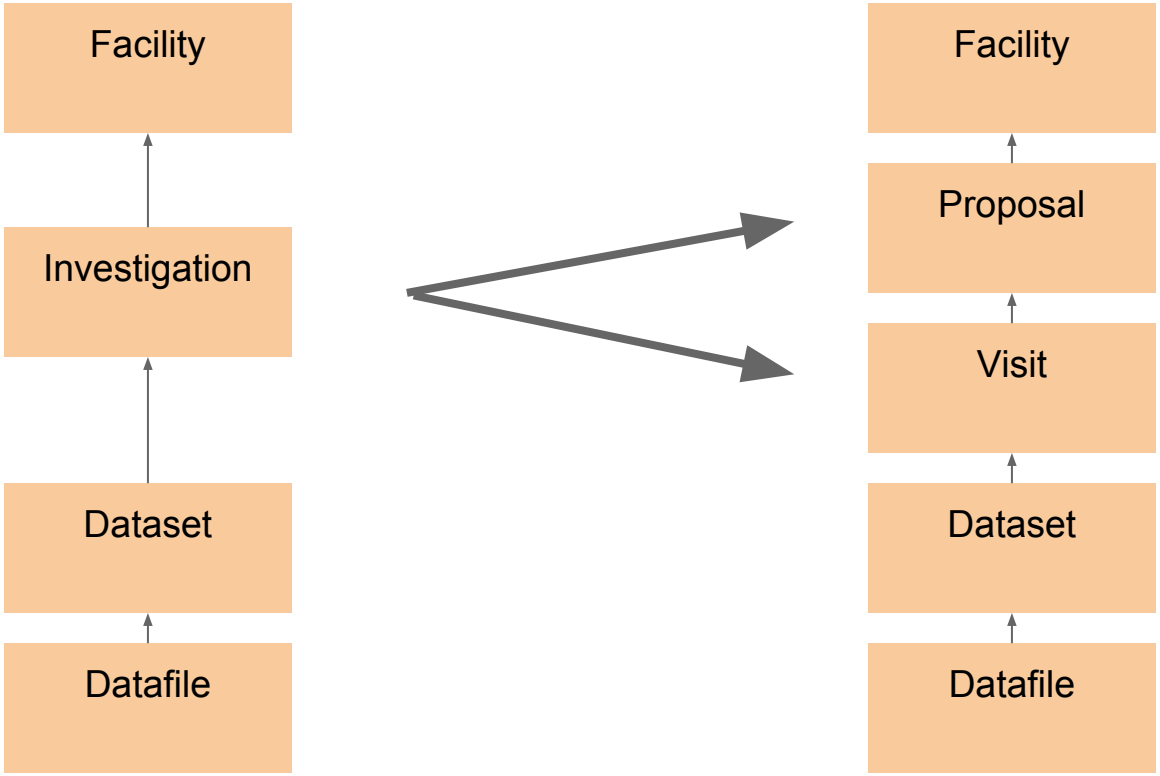
Eliminate all unused entities and attributes

- Candidates include: Study, Publication, Keyword
- It would require a survey of facilities to identify what to remove

- Simpler schema

- ?

Rename Investigation to Visit and introduce Proposal/Investigation



- Hierarchical structure becomes more regular and redundancy is avoided.
- Each entity identified by a name relative to its parent

- Affects almost all applications

Add count of datafiles and total size of datafiles to datasets and all the way up the hierarchy to the facility.

- Speeds up some queries

- Slows down writing as adding, updating or deleting a datafile would also update the dataset and everything above it.
- It breaks normalisation and allows the database to become inconsistent.
- File systems don’t do it - and I don’t believe that we should. Unix uses *du.*

Get rid of the Dataset and replace it with a Directory which may contain other Directories or Datafiles.

- Investigations (visits) would now have a one to many relationship to a directory.
- A directory would be in exactly one investigation or directory
- A directory may contain soft/symbolic links to a directory or datafile or hard links to a datafile (with normal file system semantics).

- Directly addresses the needs of those who want a file system view
- Much more flexible.
- Would no longer need DataCollections and RelatedDatafiles leading to a much simpler and more flexible model.
- Fits reasonably with Authz model - though may need some extension to say “everything below a node”

- Applications need to be changed
- It might require calls analogous to unix *find* and *du* commands
- There is a lot to think through!

Remove Datafile and Dataset/Directory from RDBMS.

- Why? - much influenced by DLS and CLF
  - Facilities especially those already having data - want a file system view and putting datafiles into ICAT Datasets is a nuisance. They only care about the Datafile.location.
  - The Datafile table is huge
    - Leads to possibility of very slow queries
  - DLS currently have an auxiliary table outside ICAT even bigger than the datafile table to offer a file system view of ICAT data (FUSE).
  - Most facilities actually have very little metadata.
  - Large facilities generally have a two tier IDS
    - The archive layer offers a file system view - possibly implemented with a huge table to locate a Datafile on tape.
  - ICAT currently provides little else except the ability to store metadata as DatasetParameters and DatafileParameters etc.
- How?
  - See next slide ...

Solution 1

- Use ICAT RDBMS for everything down to the visit/investigation. The visit would then have a location field which would identify a top level directory for use by that visit. This would make the RDBMS of ICAT very small. With this model a datafile would exist if in either main or archive IDS storage
- Authz could be represented by ACLs in the file system
- Webdav interface will be trivial
- Metadata:
  a. could use extended file attributes (indexed by lucene). However note the OS dependent behaviour of such attributes and size restrictions.
  b. could use RDBMS for metadata (indexed by lucene).
  c. could use lucene. Today lucene does not store the data it indexes but only the id of the object. If we chose to store the data that was indexed would have little reason to store metadata elsewhere. Would also index on location to find metadata associated with a Datafile.
- TopCAT could look almost like now except that drill down is by directory structure (old location)
- IDS can look similar on the outside
- ICAT very different

Solution 2

- Eliminate the RDBMS and use a directory structure with ACLs to represent facility/proposal/visit.
- Use small set of directory structures to allow TopCAT to work.

- I like solution 2 on top of 1c:
  - No RDBMS.
  - Use file system
  - Use lucene
  - No redundancy
- Conceptually very simple
- Avoids scaling problems

- Huge change - but worth thinking about

?

?

?

?

Science & Technology Facilities Council