Some of the Why's of ICAT

Shoaib Sufi
13th August 2008

Introduction

- This talk: Background on why things are they way they are in the icat schema and interfaces
- Another talk: If you want to know how to actually install the icat schema please contact databaseservices@stfc.ac.uk

Where Did it come from

- Analysis of Data Management requirements from SR lead to CCLRC Scientific Metadata Model version 1
- Work on BADC, MPIM, e-Minerals, e-Materials, ISIS back catalog lead to CCLRC Scientific Metadata Model version 2 (CSMD)
- ISIS Backcatalog – ICAT 1 & 2 (based on CSMD)
- Facilities programme lead to ICAT 3
- If there was an STFC Scientific Metadata Model version 3 it would include many of the ideas from ICAT 3

From ICAT2 to ICAT3

- Some Schema artefact's that existed in ICAT2 – they were helpful for modeling certain scenarios
- Removed many-to-many mappings when developing ICAT3 for efficiency reason
- Schema took ICAT2 removed as many tables and added simplifications
- Then added in only what was necessary and justified after discussions with all stakeholders and other facilities requirements
- So some of the column naming might be cranky but what is in there is there for good reason. Some notes in the data dictionary on columns which we first thought would be needed but were never used in practice so could be removed in future versions.

ICAT Schema

- Schema developed in ICAT done in a JDeveloper project
  - Visual & can suck in existing schema
  - Can produce schema DDL automatically
  - Free
  - Likely to become somewhat of a standard way of doing things with schema design in Oracle
  - Can do most of what you want to do
    - Partitioning missing
  - https://esc-cvs.dl.ac.uk/svn/dl/metadata/icat/trunk/jdeveloper/icat

ICAT Schema more

- Based on lots of discussion with ISIS and DLS
- e.g. One DLS specific thing:
    - Shifts – to model time at the beamline used by data acquisition system to check what the logged in user is associated with
    - ISIS did not like this but came round to it – one happy family scenario
- Raised the issue of how do you keep all of the facilities happy all of the time !

ICAT Schema generic aspects

- Parameter tables
  - Parameter – stores definition of permitted parameters
  - Dataset parameter – name-value pairs of valid parameters for datasets
  - Datafile parameter – name-value pairs of valid parameters for datafiles
  - Sample parameter – name-value pairs of valid parameters for samples

How are parameters loaded

- From the facilities spreadsheet
- Detour – Facilities spreadsheet
- Holds information about facility specific things which are loaded into an ICAT instance at install time
- Why .XLS – friendly – something people are familiar with and will to put data into
  - Not really – actually turned out useful for ICAT curators/instance maintainers

Facility Spreadsheet

- A type of facility proto ontology (more a controlled vocabulary of terms)
- Holds information about parameters, cycles, instrument/beamlines, facility scientists amongst other things
- After installation of the database schema used to initialise the database
- Customise the ICAT schema for each facility while not changing the metadata model that's why you have to be strict with e.g. parameters

Facility Spreadsheet

- Each excel tab converted to a csv file and then loaded into Oracle external tables from which the data is loaded into the actual ICAT Schema tables
- All routines written for Oracle so would have to be re-written for another DBMS
- Not done dynamically done as a one step process
- From the Mod_id's and Create_id's you can tell which processes put in the information, so if additional data is put in from the proposal/business system you can track this down and update the spreadsheet at the next iteration.

Oracle – why oh why

- Leading DBMS system
- CCLRC/STFC paid big money for licenses
- E-Science DB servers to get the most out need to use Oracle Specific features
- We have a database services team – who are specialists in Oracle – be a shame not to make use of them
- ££££ - A free version of Oracle now exists - Oracle Express

More Oracle reasons

- If you want it to perform
  - Making use of software resources
  - Making use of hardware resources
  - Making use of staff resources
  - Making use of potential expertise pool (contractors)
- For Example – the Oracle contractors helped the ICAT API how ...

Oracle Contractors to the Rescue

- Advised ICAT API author on how to structure EJBQL
- Wrote SQL to do the same thing
- Were able to do comparisons to show that the EJBQL was comparable to the SQL
- So we knew that the EJBQL that was being written was of good quality and not performance impaired.

Installation

- So we have the Schema
- Modification to the schema (hash partitioning of datafile_parameter)
- Installing common objects (logging tables for procedures, creating the external tables, loading from the spreadsheet process, creating sequences, investigation view, global parameters)
- Installing schema specific object (ISIS: LOQ commercial patch, DLS: batch migration system, setup of jobs)

Rationale

- Everything that can be done in JDeveloper should be – mainly on the DDL front
- Then Anything that can't be done in JDeveloper – e.g. partitioning in additional common scripts.
    - Have to now manual sync the mod script view of the object and what is in JDeveloper.
- Aside: Why partitioning –
- A large part of the generic support is provided by the parameter tables as discussed however this is also turned out to be the slowest aspect

More Aside

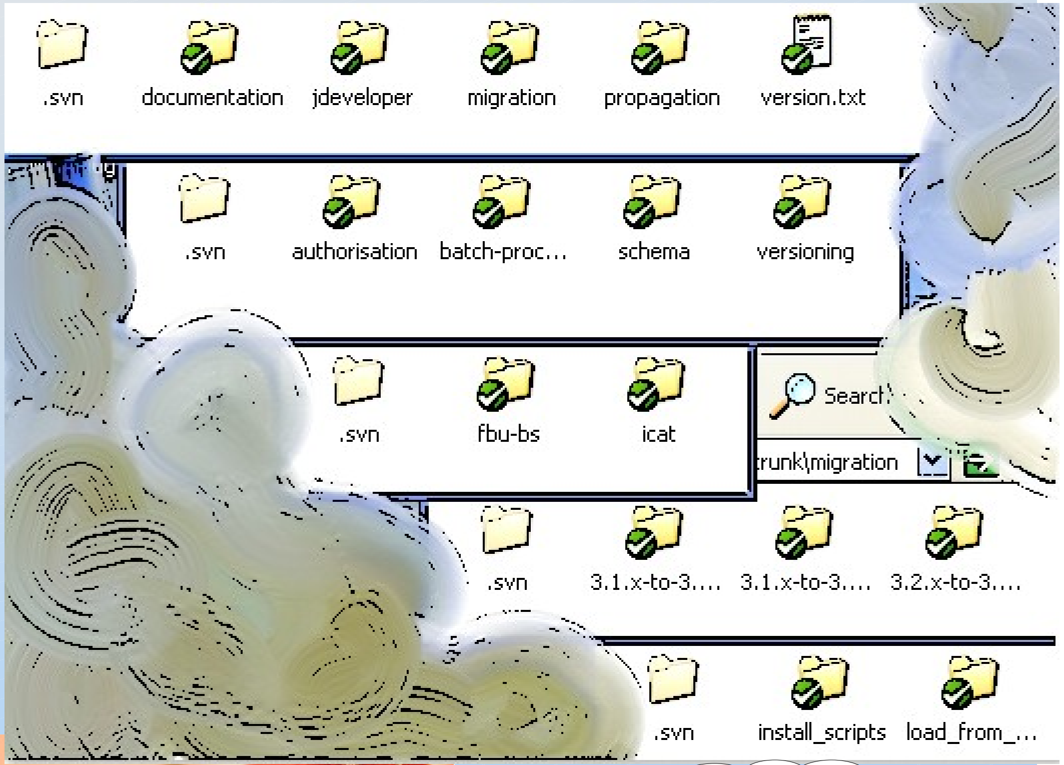
- Luckily we had the oracle contractors at the time and after trying various partitioning options came to the conclusion that hash partitioning worked really well – we are talking about a table with 25million+ records.
- Changing query times from minutes to less than a second.
- Database services subsequently have done lots of optimisation work to make the ICAT sing-along – I have to admit during the acceptance tests which all passed yesterday there was one of the common queries which managed 50 concurrent executions in less that 0.5 seconds !! - which is very impressive.

Back to Rationale

- So everything in JDeveloper thats common
- What is not but generally applicable to all ICATs should be in a separate set of files – i.e. the schema mods (e.g. partitions) & common objects
- Then schema specific things – i.e. LOQ patch for ISIS and Batch Migration system for DLS will be specific for those installations
- Has worked well – i.e. trying to KEEP THINGS AS COMMON AS POSSIBLE

Subversion Layout: ICAT trunk

- https://.../svn/dl/metadata/icat/trunk
- directories and what's inside them
- Some historical accidents got a chance to do a lot of clearing yesterday when version 3.3.0 became official.



Version.txt

- States the Major series e.g. 3.3 that is being worked on
- A quick and easy way to figure out the major version number
- When it changes it's a good time to make an announcement.
- For example, change from 3.3 to 3.4 is specific changes planned. A move from 3.3 to perhaps 4.0 may signify something more radical is planned.

scripts

- Contains a Python script
- Updates the ICAT 3.2 with file locations
- Was used as an interim method to publish Windows DFS locations
- We now use data.isis for various good reasons (tying into existing infrastructure, not attempting to oust ISIS computing or any other such subversive behavior)
- Perhaps this should be Deleted.

Propagation

- The meaty important one (suitable for vegetarians also !)
- Contains the ICAT installation files, schema mods, facility specific object and facility spreadsheet.
- Contains the specific directories and files for ISIS, DLS, CLF, Ikitten and test users.
- Contains a copy of the master ddl script – script1.sql so installation can be tested without affecting schema development.

Migration

- Details what is needed for migrating from one version to the next.
- Basic premise is create new schema, initialise and copy over from the old instance – different from just modding the existing one – has some benefits e.g. able to re-run until you get it right.
- Meant to be from one minor version to the next i.e. 3.1 to 3.2 and 3.2 to 3.3 but ideal was hard to keep.
    - e.g. the fact that a specific item was migration from 3.1 to 3.3 to DLS and yes the ideal seems far away
- Useful though at least for documentation of what is and what is not covered in the migration

JDeveloper

- This ones my favourite ... only kidding
- Contains the JDeveloper files for the ICAT Schema
- Also contains the nascent FBU-BS – stands for Facilities Business Unit Business System – a wonderful parallel to Council of the Central Laboratory of the Research Councils
  - Perhaps a punishment for underperforming staff – say the expansion of FBU-BS CCLRC – 1000 times in their lunch break or better still write it out !
- FBU-BS one needs further development based on when the STFC Useroffice system creates the correct views – please see Richard Browning – note he needs extreme harassment and pressurisation from his Boss and then he might do something for you maybe.

Reminder for Mike ...

£5
Bank of England
I PROMISE TO PAY THE BEARER ON DEMAND THE SUM OF
FIVE
Pounds
5
HH37 802549
Chief Cashier
For the Govr. and Compa. of the BANK of ENGLAND
E R
HH37 802549

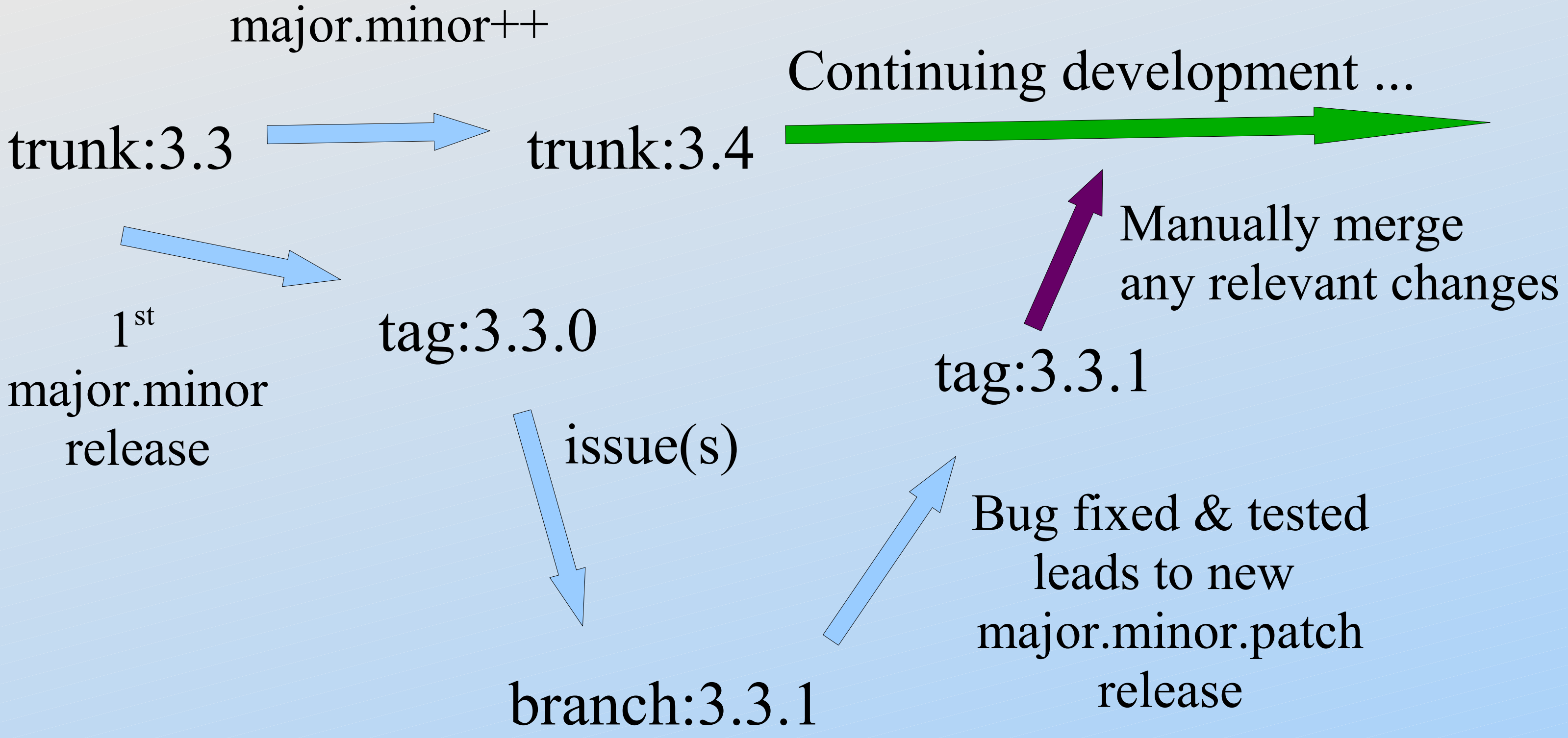
Documentation

- https://esc-cvs.dl.ac.uk/svn/dl/metadata/icat/trunk/documentation
- Data Dictionary (esc+Devigo)
- Batch Process Specification (Devigo)
- Authorisation (esc+devigo)
- Versioning (esc)

Versioning

- A way to identify version that are deployed
- Naming scheme x.y.z approach, major.minor.patch
- As complicated as it needed to be but no more
- A short document (2 pages)
- How to deal with changes – branching then tagging once tested
- Clarity over apparent utilisation of space
- Tag a whole new tree even if one file changed – just makes semantics easier and efficient storage on server sorted out by Subversion repository

Version Example



major.minor++

trunk:3.3

trunk:3.4

Continuing development ...

1st major.minor release

tag:3.3.0

issue(s)

branch:3.3.1

Bug fixed & tested leads to new major.minor.patch release

tag:3.3.1

Manually merge any relevant changes

Batch Process Spec

- Why is it called that – well as not to tie the implementation to a specific language – could be implemented in Java although actually done in PL/SQL
- Details procedures needed to update database periodically

Authorisation

- Details the Entity Hierarchy Approach
- Have you read the documentation – it's a great read
- Basically you have Roles which map onto a set of actions
- The actions are what's important
- In dependent software its the actions that need specific support – so adding new actions are non-trivial.

Data Dictionary

- Reference work
- Description of tables and columns
- Deserve an award if you can read through it without Caffeine or any other stimulants.
- Extracts the comments from the schema and formats them into a pdf via latex.
- WHY ? - to keep the comments in the schema '**Fresh and Useful**™'

applications

- Contains Oracle Application Express applications
- Not really part of 3.3 or 3.2 but were of use in 3.1
- Monitoring lookup data and generally browsing the metadata
- Could resurrected
- But really should be removed and newer applications should use the ICAT API
- This may create further requirements for the ICAT API which is good as it is meant to be THE API.

Other Interfaces

- DUO Desk to ICAT
  - Very different worlds
  - Fresh data as validated/verified/cleaned by the facility very useful
  - Other systems can then use subset (e.g. GDA and some of the MX software)
  - Complex could do an hour long presentation on this
  - Written in PL/SQL – please see again why Oracle

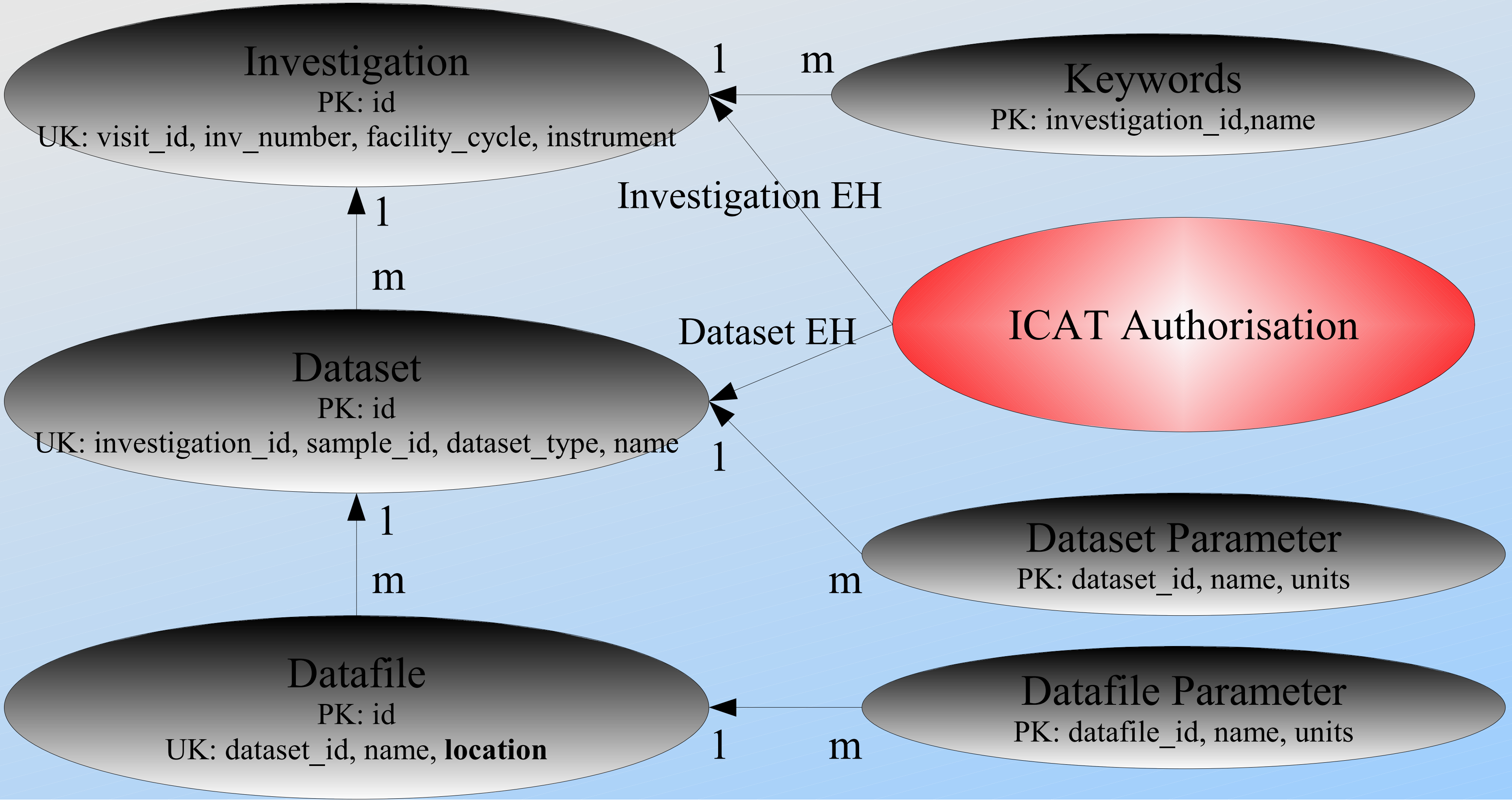
More Other Interfaces

- ICAT API
  - DDH
    - DLS file registration into ICAT
  - Trigger Mechanism
    - ISIS file registration into ICAT
- (Future if enough pressure applied) FBU-BS
  - Unless they decide to use Duo Desk (not sure if that is on the cards)

Questions

- Any more Whys ?

ICAT 3 overview



Investigation
PK: id
UK: visit_id, inv_number, facility_cycle, instrument

Keywords
PK: investigation_id,name

Dataset
PK: id
UK: investigation_id, sample_id, dataset_type, name

ICAT Authorisation

Dataset Parameter
PK: dataset_id, name, units

Datafile
PK: id
UK: dataset_id, name, **location**

Datafile Parameter
PK: datafile_id, name, units

ICAT 3

- Contains information about experiments and the data they produce.
- Based on the STFC Core Scientific Metadata Model v2 with many extensions
  - http://epubs.cclrc.ac.uk/work-details?w=30324
- Investigation->dataset->datafile orientated
- Indexing using keywords and parameters
- Constrains parameters building block of generic abilities, but also open to abuse
  - Garbage in , garbage out
- Involved but efficient Authorisation framework taking various authorisation user stories into account
  - https://esc-cvs.dl.ac.uk/svn/dl/metadata/icat/trunk/documentation/authorisation/icat3_authorisation_spec.doc