PaNdata ODI 1\textsuperscript{st} Open Workshop

Dublin 24-25/28\textsuperscript{th} of March 2014

Co-located with the RDA 3\textsuperscript{rd} plenary at Croke Park

https://indico.desy.de//event/1stow

Booklet of collected presentations
ICAT workshop at trinity college in Dublin 24/25th of March

Brian Matthews: ICAT collaboration and its governance
ICAT site reports: ALBA, DESY, DLS, ELETTRA, ESRF, HZB, ISIS, SNS
Rolf Krahl: python-icat - a library for writing ICAT clients in Python
Tom Griffin: Domain specific ICAT GUI
Tom Griffin: ISIS authz rules
Rolf Krahl: Styles of rule and is the authz system adequate?
Jay Rainey: Mantid and ICAT
Brian Matthews: ISIS Data Journal and SCAPE
Steve Fisher: IJP for LSF
Jay Rainey: Experience of the ICAT API and Documentation
Steve Fisher: J2EE containers
Steve Fisher: ICAT, IDS & IJP
Wayne Chung: TopCAT
ICAT Steering Committee

Brian Matthews

Research Data Group
Scientific Computing Department
STFC Rutherford Appleton Laboratory

brian.matthews@stfc.ac.uk
Growing number of Partners in ICAT

- ICAT developed with a small number of partners
  - ISIS, DLS
  - SNS
- PaNData has included more
  - ILL, ELLETRA, ESRF etc
- Others also using or are interested
  - CLF, CCFE, ...
Managing an Open Source Collaboration

• To date
  – Developer group
  – Discussion with individual facilities
  – PanData

• But group getting larger
  – Need to make sure all have a say

• Sustainability
  – Current PanData project soon finishing
  – Need to ensure sustainability for ICAT
ICAT Steering Group

• Made up of representatives of ICAT User facilities
  – Production or near production users

• Purpose:
  – To advise on the likely usage and requirements for ICAT;
  – To provide senior management oversight for the work of the ICAT collaboration partners;
  – To provide strategic guidance on the development and deployment of ICAT.
Responsibilities of the ICAT Steering Committee

• to provide strategic advice to the ICAT collaboration;
• to protect the interests of the institutions;
• to advise on new initiatives and plans which may affect ICAT;
• to advise on major requirements which may affect ICAT;
• to oversee the ICAT development roadmap,
  – including the incorporation of additional components;
• to ensure that ICAT components continue to remain aligned to the users,
  – a common functionality and code base;
• to advise on ownership, licencing of components and commercialisation opportunities;
• to ensure that the institutions provide appropriate resources to the collaboration.
Membership

• The Committee has one representative from each of the stakeholders who are actively involved in developing, deploying and exploiting ICAT.
  – The Committee will regularly review membership

• The Project Manager of the ICAT Project should attend meetings in a non-voting capacity.

• A Chairperson elected from its members;

• A Secretary provided by STFC Scientific Computing.
Current Members?

- a representative of ISIS;
- a representative of DLS;
- a representative of CLF;
- a representative of SNS;
- a representative of Scientific Computing at STFC;
- a representative of ILL;
- a representative of ESRF;
- a representative of ELETTRA;
- a representative of the PanData consortium;
Meetings

• 1 – 2 times a year as needed

• Telco/VC
  – Though we will probably share a room at RAL

• 1 – 1.5 hrs

• Shared area on ICAT web-site
First Meeting

• ASAP
  – By end April?
  – Brian Matthews will temporarily chair until a new chair is elected.
• Representatives?
• Agenda?
  – Chair of Committee
  – Review ToR and Membership
  – 2014-15 Development Roadmap
  – Facilities Plans and Requirements
  – Future Directions
Agenda

- Project overview
- What’s next?
Project Overview

- Project Team:
- Current versions:
  - ICAT: 4.2.5
  - TopCAT: 1.9
  - Java 6
  - Glassfish 3

September 2013
Project Overview

User Office

PROPOSALS SYNC
Quartz Jobs

SCHEDULED TASKS FROM QUARTZ

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Start Time</th>
<th>End Time</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>09:00</td>
<td>10:00</td>
<td>1 hour</td>
</tr>
<tr>
<td>Task 2</td>
<td>10:00</td>
<td>11:00</td>
<td>1 hour</td>
</tr>
<tr>
<td>Task 3</td>
<td>11:00</td>
<td>12:00</td>
<td>1 hour</td>
</tr>
</tbody>
</table>

Copyright © 2015, ALBA. All rights reserved.

Support and Information: info@alba-project-site.org
OpenALP: Open Access Library Project
OpenALP License: GPL v3 or later
Project Overview

- **4 modules:**
  - **Core**: controller of the system interface to ICAT.ear
  - **Admin**: User/ Roles/ File Format web interface
  - **Storage**: Data ingestion done via file parsing processes.
  - **Webservice**: Interface to the Core.
- **CI**: Jenkins
Project Overview

- ICAT administration:
Project Overview

- TopCAT:
What’s next?

- Test environment (not production).
  - Need to test with real data.
  - Performance?
- IDS is still to be implemented.
  - Will not go to a production environment without.
- Interested in ingestion via python interface.
- Comments:
  - Upgrades:
    - Java 6 → Java 7
    - Glassfish 3 → Glassfish 4

Effort?
ICAT at ALBA

Thank you
ICAT status @ DESY

- ICAT installed and running (details: Jürgen)
  - Current plan: only for open access data
    - No concrete plan to host all petra3 data in icat
    - The oa ICAT will also serve as a platform to re-evaluate
  - Basic requirement: PI can alter acl’s at any time
    - Seems a major obstacle so far
  - Essentially no data ingested yet
    - Problem: open access = published data, but few meta-data recorded
    - Chicken-Egg: don’t invest effort for a scarcely filled data catalogue
  - Annotation and ingest of derived data via web-frontend
    - Access control through umbrella & basic (door)
    - WebDAV export
Diamond ICAT Status Report
Diamond migration from 3.3 to 4.2

- Diamond will only be using ICAT 4.2 from March 28.
Current status/volumes in Diamond
ICAT = 285,198,074 files

Diamond Total Data

Data Volume (Gb)

2007 2008 2009 2010 2011 2012 2013 2014

1,020,040

This image shows the diffraction pattern of an RNase crystal at 1.6Å, from Prof K. Ravi Acharya's group (at University of Bath) - the image that took DLS total data catalogued and archived > 1 Petabyte.
ICAT 4.2

• Successfully integrated and used as source of current visit by
  – Data Acquisition framework in Diamond (GDA)
  – Automated session/visit directory structure generation
  – Projects in DAWN
  – Workflows in Passerelle/DAWN
Example of workflow

Please remove the example ICAT connection and replace it with a new one in order for it to compute the new truststore path.
Look out for....

• Cache for data coming off tape ready for download to user via ICAT/IDS
  – DLS now have multi TB cache.
• Increased the filename size in DB.
• Problems with users and the ‘dataset’ concept
  – “I know the filename and directory....”
• Data rates from site ‘improving’
Plans to migrate to 4.3

• None yet.
More details in...

- More details in RDA presentations....
- **PaNSIG Meeting Friday 28th 13:30-15:00**
  - Specific topics for future collaboration
    - Data analysis issues and frameworks - Alun Ashton / DLS
      - ..... 
- **Structural Biology IG Session Thursday, March 27th, 13.30 -15.00**
  - Structural biology towards biomedicine and health
    - 13.30 -13.50 Data management and processing pipelines at DLS Alun Ashton
ICAT Deployment

Status @ Elettra

Dublin, 24 March 2014.

Milan Prica
IT – Scientific Computing Group
Elettra – Sincrotrone Trieste S.C.p.A.
Facility overview

• Two light sources

• **ELETTRA** Synchrotron Radiation Facility
  - Operational since 1993 – users since 1994
  - 20 beamlines, 4 more in construction/upgrade

• **FERMI@ELETTRA** Free Electron Laser
  - Operational since 2011 – users since 2012
  - 3 beamlines, 3 more planned
Virtual User Office (VUO)

- Operational since 1997
- Possible authentication with Umbrella (Nov. 2013);
  Facility LDAP > 8 years
- Contains plenty of useful metadata
- Uses DOIs
- Enforces Elettra's Data Policy
- Hosts data access service
  - Download / Upload
  - Proposal submitter(s) and selected BL scientist(s) may grant data access to other participants
- [https://vuo.elettra.eu](https://vuo.elettra.eu)
Data file formats

• Fermi@Elettra:
  • **Custom HDF5** files on all beamlines
    – Integrated with TANGO

• Elettra:
  • **Custom HDF5** files on some beamlines
    – Part of endstation DAQ TANGO
    – Converters from legacy formats
  • Variety of data formats on older beamlines
    – TIFF, RAW, ASCII...

• **NeXus**: Not adopted
ICAT metadata catalogue

- Three instances deployed (1 public)
  - Updated to the latest ICAT 4.3.2
  - Glassfish 3 and 4 containers
  - Topcat 1.11.0 and ICE
  - Authentication with local and remote db, simple
  - LDAP module not compatible with the Elettra LDAP
Data ingestion software

- **iGEST**
  - Python module
  - Simple to configure (template based mappings)
    - Works with any HDF5 file (NeXus included)
  - Supports parallelism during ingestion
  - Compatible with the FERMI acquisition pipeline
iGEST mapping example

```
[icat]
user=user@gmail.com
password=12345678901112
plugin=db

[template]
h5template=temp_vlabsample.h5

[icatmeta]
facility=entry/instrument/source/name
instrument=entry/instrument/name
investigationtype=entry/experiment_type
investigation=entry/experiment_identifier
datasettype=entry/dataset/type
dataset=entry/dataset/name
sample=entry/sample/name
sampletypename=entry/sample/type/name
sampleformula=entry/sample/chemical_formula
facilitycycle=entry/instrument/facilitycycle
```
Issues

• LDAP Authentication:
  • Custom module needs to be developed for ELETTRA
  • Umbrella – ICAT authentication could be used?

• Rules and Users:
  • Rules to be defined for ELETTRA
  • Users should be inserted automatically from VUO
Issues

• Front-end (TopCAT):
  • A custom front-end integrated with VUO would be a better solution for ELETTRA
  • Users often perceive the front-end as “the catalogue”
  • Configuration tools should be part of the front-end

• Schema:
  • Many cross dependencies
  • Quite complex to match to the facility needs
Thank you!

Questions?

milan.prica@elettra.eu
Status of metadata at ESRF

ICAT deployment
Metadata ingestion
Current status
Next steps
ICAT deployment

Current software stack

- Glassfish 4.0
- ICAT 4.3.1 + authn_db 1.1.1
- TopCAT 1.11 + IDS 1.0.0 (+ custom plugin)

Databases

- Oracle (test and production)
- Derby (development)

Machines

- Oracle VM
- Sun JDK 1.7.0

Security

- Full SSL with signed certificates (TERENA)
- Apache reverse proxy (mod proxy http, address translation)
Some Issues during deployment

IDS not fully suitable yet for ESRF setup

- We need more operations delegated to the plugin, including logic for available/not available, direct/archived, path reconstruction, …

Problem with TopCAT behind a reverse proxy (GWT issue)

Address translation ignored:

- All .gwt.rpc files need to be copied

or

- Archive needs to be modified to change the context root

Other issues with proxy or firewall

- Problems connecting to external ICAT on port 8181 (solved)
- Problems connecting to ILL using ill.eu URL (ill.fr works)
Pilot project (id 19)

First Pilot project for ID19

- Spec
- Tomo Xml
- Tomo to ICAT xml converter
- ICAT Xml
- ICAT xml ingest
- SMIS
- ICAT API
- RDBMS
- Web Service API

Data used for service verification imported using the Pilot
General case
Proposed implementation

1 stateful TANGO device on each beamline

1 stateless queuing and processing system to bind them all
Asynchronous processing

- Non-blocking TANGO devices (fire and forget)
- Queues serve as buffer in case of peak activity or slow/dead process

Data integrity

- Message persistence
- Transacted processing
- Error management, redelivery policy

Scalability

- Concurrent processing with configurable thread pools
- Several instances of queues and/or routes can be deployed

Input queue is a single point of failure

- Failover configuration / cluster deployment
- Local hdf5 file for offline processing
Advantages of Apache Camel

Most features already implemented
- Concurrent processing
- Transactions
- Automatic message translation (marshall/unmarshall, conversion)
- Logging / message tracing
- Error management

Technology agnostic
- Implements well-defined Enterprise Integration Patterns
- Independent of queuing system, transport technology, …

Very well integrated
- DSL for Java, Spring, Scala, Groovy, Annotations, Blueprint, …
- Options for deployment (standalone, EJB container, OSGi container)
- Pure Maven (archetypes, plugins), JUnit support, Spring integration
Prototype development
- Embedded Apache ActiveMQ
- Standalone Java application (using Spring)

Project metrics
- c.a. 1 month in the making
- c.a. 600 LoC (250 tests, 200 ICAT client, 150 data objects, beans)
- c.a. 800 lines of XML (incl. blank lines) (600 test, 100 configuration, 100 XSLT)
- Stored on ESRF gitlab

Not done yet
- SMIS integration
- Monitoring and error management interface

Authentication with Umbrella
- Reusing SMIS mechanism already in place
Next steps

Prototype almost ready for testing with TANGO device

Validation of the design

Testing and performance analysis

Stress testing, failure recovery

Dimensioning of the different components

Data volume for queues, number of concurrent processors for each steps

1 year = 1200 experimental session, 4800 visits, 1800 publications

on 30 public beamlines (+10 CRG)

Live testing on one volunteer beamline

We still need to define

- ESRF data policy
- Infrastructure needed for deployment
- Maintenance and support policy
Thank you for your attention
Issues with ICAT

Solved issues

- No migration script for Derby db (no python driver), no migration of rules
  - Python code (partially) ported to Java, rules recreated
- Some modification of the package required for TopCAT
  - To change the JPA log level
- Problem with TopCAT behind a reverse proxy (GWT issue)
  - All .gwt.rpc files need to be copied (address translation ignored)

Current issues

- ICE 1.0.0 not compatible with ICAT 4.3.x
  - Need to be updated at the same time than ICAT if considered as an admin module
- IDS not fully suitable for ESRF setup
  - We need more operations delegated to the plugin, including logic for available/not available, direct/archived, path reconstruction, …
  - The plugin currently needs a special packaging
Other issues

Issues with clients
- DAWN ICAT plugin
  - Only supports LDAP as authentication method
  - Cannot configure new ICAT repositories
- Most Python tools need at least Python 2.6
  - Version 2.4 stated in documentation
- XML ingestion tool does not work with ICAT 4.3.x
  - Would need a complete rebuild as it uses a modified 4.2 client

Issues with other ICATs
- Problems connecting to external ICAT on port 8181 (solved)
- Problems connecting to ILL using ill.eu URL (ill.fr works)
- Failed to connect to ISIS during SV 7 (investigation ongoing)
  - Likely linked to our proxy and firewall configuration too
HZB

Report on ICAT Deployment

Rolf Krahl

ICAT Meeting, Dublin, Mar 2014
ICAT deployment is in experimental stage at HZB.

- Virtual machine icat.helmholtz-berlin.de
- openSUSE 13.1 (x86_64)
- OpenJDK 1.7.0
- GlassFish Server Open Source Edition 4.0
- ICAT components:
  - authn_simple 1.0.0 (users: root, useroffice, idsreader)
  - authn_ldap 1.1.0
  - ICAT 4.3.2
  - ids.storage_file 1.0.0
  - ids.server 1.0.1
  - TopCAT 1.11.0
- Use central Oracle database as backend
- Access is proxied through Apache using mod_proxy_http
Users:

- ICAT users are authenticated by the HZB LDAP server.
- Users are imported from LDAP into ICAT by a Python script.

HZB user portal GATE:

- GATE is the HZB user portal and proposal submission system.
- Import instruments and proposals (as investigations) from GATE into ICAT using Python scripts.
- Problem: GATE has its own user base. Some GATE users do have an HZB account (LDAP), but not all.
- Plan: GATE users should get an HZB account routinely if their proposal is accepted, But workflow for this is not settled yet.
Systematic storage of experimental data is not yet operational for all instruments at HZB.

Purchase of storage hardware is planned.

E.g. the physical storage that IDS should be deployed on, does not exist yet.

A customized IDS storage plugin might be needed before going into production.

Question: How to keep data on storage and datafile objects in ICAT in sync?
ICAT at ISIS

Tom Griffin, STFC ISIS Facility
ICAT Workshop
Dublin
March 2014
tom.griffin@stfc.ac.uk
Status

- ICAT 3.3 : production for 5+ years
- ICAT 4.3 : production - parallel
- ICAT Data server: production
- Authentication : Full user office plugin
- Large Rule set - (mostly) implements our data policy
Review – Past Issues

- Rules system now supports most of what we need
- IDS reference implementation required only minor changes
- Investigation <-> Instrument change painful, but useful
- SampleType constraint changed
- TopCAT single sign on
Current Issues

• Network reliability at RAL
  – Due to network topology ISIS staff see this as an ICAT problem

• Periodic server hangs
  – Not always sure why (memory, load, ?)
  – Multi node setup for balancing and reliability
1 year plan – Lund

- Go live with 4.2 / 4.3 / 4.4 ✓
- Mantid integration with 4.x ✓
- Switch off 3.3 ✗
- Upload of processed data √
  - from Mantid ✓
  - From TopCAT ✗
<1 year Requirements - Lund

- Permissions control in TopCAT  ✗
  - ICAT auth rules to support this  ?
- GUI tweaks to TopCAT  ✓
- IDS reference implementation  ✓
- Data upload from Mantid  ✓
2014/15 plan

- Continue verification testing of 4.3
- Continue Mantid integration with 4.3
- Switch off 3.3

- New interfaces (see later presentation)
- ISIS ‘Scientific Computing Strategy’
Requirements

- See the roadmap (?)
- Further enhancements to rules
- GUI changes to TopCAT….maybe more
- Stability – **no breaking** changes
- Performance generally very good
  - Function of ingest server, database and network
  - Some rules can cause ‘slow’ queries
Questions...
ICAT4 Experience at ORNL/SNS

Shelly Ren
Scientific Information Systems
March 24, 2014
How is ICAT used at SNS today?

• Fully integrated with SNS post processing
  ➢ Auto catalog of raw experiment data
  ➢ Auto catalog of auto reduced data

• Web service to serve live experiment monitor
  ➢ Up to date information about a proposal
  ➢ Up to date information about runs of a proposal

• Web service to serve for search engines (findnexus, finddata, Filefinder):
  ➢ Summary of a proposal
  ➢ Metadata of an experiment/run
  ➢ Location of experiment data
Automated Data Catalog and Data Reduction

Active MQ

Queue Processor on fermi.ornl.gov

Post Process Admin for Parallel Processing

Remote Reduction Jobs via mpirun

Metadata ICAT Service

Queue Processor running on autoreducer1.sns.gov autoreducer2.sns.gov

Post Process Admin

Catalog Agent ~/ingestNexus.mq

Reduction Agent ~/reduce_XXX

Red-Catalog Agent ~/ingestReduce.mq

File System

Reduced Datafile

Reduced Metadata

Raw Metadata

Reduced Datafile
# Experiment Monitor

## Experiment Monitor

### SEQ IPTS-10633

**List of SEQ runs for IPTS-10633:**

<table>
<thead>
<tr>
<th>Run</th>
<th>Created on</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>48832</td>
<td>March 23, 2014, 6:07 p.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48831</td>
<td>March 23, 2014, 5:09 p.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48830</td>
<td>March 23, 2014, 4:09 p.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48829</td>
<td>March 23, 2014, 3:11 p.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48828</td>
<td>March 23, 2014, 2:07 p.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48827</td>
<td>March 23, 2014, 1:51 p.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48826</td>
<td>March 23, 2014, 12:38 p.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48825</td>
<td>March 23, 2014, 11:35 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48824</td>
<td>March 23, 2014, 10:07 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48823</td>
<td>March 23, 2014, 9:07 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48822</td>
<td>March 23, 2014, 7:46 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48821</td>
<td>March 23, 2014, 6:45 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48820</td>
<td>March 23, 2014, 5:46 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48819</td>
<td>March 23, 2014, 4:44 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48818</td>
<td>March 23, 2014, 3:46 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48817</td>
<td>March 23, 2014, 2:43 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48816</td>
<td>March 23, 2014, 1:44 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48815</td>
<td>March 23, 2014, 12:43 a.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48814</td>
<td>March 23, 2014, 11:41 p.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48813</td>
<td>March 23, 2014, 10:40 p.m.</td>
<td>complete</td>
</tr>
<tr>
<td>48812</td>
<td>March 23, 2014, 9:41 p.m.</td>
<td>complete</td>
</tr>
</tbody>
</table>

21 of 223 runs shown – show all runs

---

**Notes:**

- Experiment status
- List of experiments
Experiment Data Display

Run title: Run=42172,8a3NiSb2O9, powder,T=1.51 K, 7.5 meV, Fermi=180 Hz
Run start: March 23, 2014, 11:18 p.m.
Run end: March 24, 2014, 6:23 a.m.

Plot of auto reduced data

Experiment parameters

Experiment data location

Data files:
- /SNS/HYSS/PT5-10498/adara/HYS_42172.adara
- /SNS/HYSS/PT5-10498/nexus/HYS_42172.nxs.h5

Reduced files:
- /SNS/HYSS/PT5-10498/shared/autoreduce/4pixel/HYS_42172_4pixel.nxs
- /SNS/HYSS/PT5-10498/shared/autoreduce/msk_tube/HYS_42172_msk_tube_spe.nxs
- /SNS/HYSS/PT5-10498/shared/autoreduce/reduction_log/HYS_42172.nxs.h5.log
- /SNS/HYSS/PT5-10498/shared/autoreduce/4pixel/HYS_42172_4pixel_spe.nxs
- /SNS/HYSS/PT5-10498/shared/autoreduce/msk_tube/HYS_42172_msk_tube.nxs
- /SNS/HYSS/PT5-10498/shared/autoreduce/msk_tube_spe.png
More about ICAT tomorrow

- Web service to encapsulate database/objects from application users
  - Integrated user or guest portal
  - Experiment hall monitors
- ICAT driven fully automate data reduction
  - Auto reduce via the latest calibration, characterization runs
  - Auto reduce via strategies defined in ICAT
- Statistical distribution of experiment parameters, aka process variables
- ICAT assisted experiment design
ICAT Installation, Performance, Concerns

• Plan to roll ICAT 4.3 in the summer break
• Still need to work on and test out Rules
• ICAT 4.2 seems to run smoothly in production
• Potential performance concerns: load and search of large volume of dataset and datafile parameters
python-icat
A Library for Writing ICAT Clients in Python

Rolf Krahl

ICAT Meeting, Dublin, Mar 2014
SOAP is used as the access protocol for ICAT.
Clients exist for different programming languages, including Java and Python.
The most popular SOAP library for Python is Suds.
python-icat aims to make writing ICAT clients with Python simpler.
python-icat is built on top of Suds.

<table>
<thead>
<tr>
<th>Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep the general structure and flexibility of Suds.</td>
</tr>
<tr>
<td>Simplify things where possible.</td>
</tr>
<tr>
<td>Try to remove annoying details.</td>
</tr>
<tr>
<td>Make use of object-oriented design.</td>
</tr>
</tbody>
</table>

A typical python-icat program might be mistaken for a generic Suds program at first glance. It’s just somewhat simpler.
Example: Add a Datafile

Using plain Suds

dataset = client.service.search(sessionId, "Dataset[name='e201215']")[0]
format = client.service.search(sessionId, "DatafileFormat[name='NeXus']")[0]
datafile = client.factory.create("datafile")
datafile.dataset = dataset
datafile.datafileFormat = format
datafile.name = "e201215−7.nxs"
datafile.id = client.service.create(sessionId, datafile)
Example: Add a Datafile

**Using plain Suds**

```python
dataset = client.service.search(sessionId,
    "Dataset[name='e201215']")[0]
format = client.service.search(sessionId,
    "DatafileFormat[name='NeXus']")[0]
datafile = client.factory.create("datafile")
datafile.dataset = dataset
datafile.datafileFormat = format
datafile.name = "e201215-7.nxs"
datafile.id = client.service.create(sessionId, datafile)
```

**Using python-icat**

```python
dataset = client.search("Dataset[name='e201215']")[0]
format = client.search("DatafileFormat[name='NeXus']")[0]
datafile = client.new("datafile")
datafile.dataset = dataset
datafile.datafileFormat = format
datafile.name = "e201215-7.nxs"
datafile.create()
```
Example: Add a Datafile

Or even:

Using python-icat

```python
dataset = client.search("Dataset[name='e201215']")[0]
format = client.search("DatafileFormat[name='NeXus']")[0]
client.new("datafile",
    dataset=dataset,
    datafileFormat=format,
    name="e201215-7.nxs").create()
```
• `client.new(...)`: creates a new ICAT entity object. Use this in place of `client.factory.create(...)`.  
• `client.new(...)`: optionally accepts keyword/value arguments to set attributes.  
• ICAT API methods are defined as methods in the `python-icat` client. Replace `client.service.<method>(...)` by `client.<method>(...)`.  
• Don’t care about the session id, the `python-icat` client remembers it and adds it to the ICAT method calls as needed.  
• ICAT entity objects have their own methods: e.g. `datafile.create()`.
Example: Add Keywords to an Investigation

Using plain Suds

```python
investigation = client.service.search(sessionId, "Investigation [name='2010-E2-0489-1']")[0]
keywords = []
for k in ["Foo", "Bar", "Baz"]:
    keyword = client.factory.create("keyword")
    keyword.name = k
    keyword.investigation = investigation
    keywords.append(keyword)
client.service.createMany(sessionId, keywords)
```
Example: Add Keywords to an Investigation

Using plain Suds

```python
investigation = client.service.search(sessionId,
   "Investigation [name='2010-E2-0489-1']")[0]
keywords = []
for k in ["Foo", "Bar", "Baz"]:  
    keyword = client.factory.create("keyword")
    keyword.name = k
    keyword.investigation = investigation
    keywords.append(keyword)
client.service.createMany(sessionId, keywords)
```

Using python-icat

```python
investigation = client.search(
   "Investigation [name='2010-E2-0489-1']")[0]
investigation.addKeywords(["Foo", "Bar", "Baz"])
```
### Example: Create a Group

**Using plain Suds**

```python
users = [jbotu, jdoe, nbour]
group = client.factory.create("group")
group.name = "investigation_42_reader"
group.id = client.service.create(sessionId, group)
ugs = []
for u in users:
    ug = client.factory.create("userGroup")
    ug.user = u
    ug.group = group
    ugs.append(ug)
client.service.createMany(sessionId, ugs)
```
Example: Create a Group

Using plain Suds

```python
users = [ jbotu, jdoe, nbour ]
group = client.factory.create("group")
group.name = "investigation_42_reader"
group.id = client.service.create(sessionId, group)
ugs = []
for u in users:
    ug = client.factory.create("userGroup")
    ug.user = u
    ug.group = group
    ugs.append(ug)
client.service.createMany(sessionId, ugs)
```

Using python-icat

```python
users = [ jbotu, jdoe, nbour ]
group = client.createGroup("investigation_42_reader", users)
```
Using plain Suds

```python
client = suds.client.Client(url)
credentials = client.factory.create("credentials")
credentials.entry.append(
    [
        { 'key': 'username', 'value': username },
        { 'key': 'password', 'value': password }
    ]
)
sessionId = client.service.login(auth, credentials)

# ...
client.service.logout(sessionId)
```
Example: Login

Using plain Suds

```python
client = suds.client.Client(url)
credentials = client.factory.create("credentials")
credentials.entry.append(
    [ {'key': 'username', 'value': username },
      {'key': 'password', 'value': password } ])

sessionId = client.service.login(auth, credentials)

# ...

client.service.logout(sessionId)
```

Using python-icat

```python
client = icat.client.Client(url)
credentials = { 'username':username, 'password':password }
client.login(auth, credentials)
```
Drawback of python-icat: it depends on the ICAT version.

When the ICAT API changes, the library needs to get adapted to the new version.

Currently supported: 4.2.* and 4.3.*, the API version is checked automatically.

A module `icat.icatcheck` tests compatibility helps to adapt the library to new versions.

Advantage: some incompatibilities between ICAT versions are handled by python-icat and hidden from the application.
Example: Add Instrument to an Investigation

Using plain Suds

```python
investigation = client.service.search(sessionId,
   "Investigation INCLUDE 1 [name='2010-E2-0489-1']")[0]
instrument = client.service.search(sessionId,
   "Instrument [name='HIKE']")[0]
if client.service.getApiVersion() < '4.3.0':
    investigation.instrument = instrument
    client.service.update(sessionId, investigation)
else:
    ii = client.factory.create('investigationInstrument')
    ii.investigation = investigation
    ii.instrument = instrument
    client.service.create(sessionId, ii)
```

Using python-icat

```python
investigation = client.search("Investigation [name='2010-E2-0489-1']")[0]
instrument = client.search("Instrument [name='HIKE']")[0]
investigation.addInstrument(instrument)
```
Example: Add Instrument to an Investigation

**Using plain Suds**

```python
investigation = client.service.search(sessionId,
    "Investigation INCLUDE 1 [name='2010-E2-0489-1']")[0]
instrument = client.service.search(sessionId,
    "Instrument [name='HIKE']")[0]
if client.service.getApiVersion() < '4.3.0':
    investigation.instrument = instrument
    client.service.update(sessionId, investigation)
else:
    ii = client.factory.create('investigationInstrument')
    ii.investigation = investigation
    ii.instrument = instrument
    client.service.create(sessionId, ii)
```

**Using python-icat**

```python
investigation = client.search(
    "Investigation [name='2010-E2-0489-1']")[0]
instrument = client.search("Instrument [name='HIKE']")[0]
investigation.addInstrument(instrument)
```
A typical ICAT client always needs the same set of command line arguments: URL of the ICAT service, authentication plugin name, username, and password.

A module `icat.config` takes care of this: it defines the command line arguments.

Configuration options may be set via command line arguments, environment variables, configuration files, and default values (in this order, first match wins). The password may also be read from interactive keyboard input.

Of course, a program may define additional custom arguments.
Using plain Suds

```python
auth = sys.argv[3]
username = sys.argv[5]
password = sys.argv[7]
client = suds.client.Client(url)
credentials = client.factory.create("credentials")
credentials.entry.append(
    [ 
        { 'key': 'username', 'value': username },
        { 'key': 'password', 'value': password } ]
)
sessionId = client.service.login(auth, credentials)
```
Example: Config

Using plain Suds

```python
auth = sys.argv[3]
username = sys.argv[5]
password = sys.argv[7]
client = suds.client.Client(url)
credentials = client.factory.create("credentials")
credentials.entry.append(
    [  { 'key': 'username', 'value': username },
      { 'key': 'password', 'value': password } ]
)
sessionId = client.service.login(auth, credentials)
```

Using python-icat

```python
config = icat.config.Config()
conf = config.getConfig()
client = icat.Client(conf.url, **conf.client_kwargs)
client.login(conf.auth, conf.credentials)
```
usage: login-icat-config.py [options]

optional arguments:
  -h, --help show this help message and exit
  -c CONFIGFILE, --configfile CONFIGFILE config file
  -s SECTION, --configsection SECTION section in the config file
  -w URL, --url URL URL to the web service description
  --http-proxy HTTP_PROXY proxy to use for http requests
  --https-proxy HTTPS_PROXY proxy to use for https requests
  -a AUTH, --auth AUTH authentication plugin
  -u USERNAME, --user USERNAME username
  -p PASSWORD, --pass PASSWORD password
  -P, --prompt-pass prompt for the password
Example: Exception Handling

Using plain Suds

```python
try:
    sessionId = client.service.login(auth, credentials)
except suds.WebFault as e:
    if e.fault.detail.IcatException.type == 'SESSION':
        print "Login failed: %s" % e
    else:
        raise
```
Example: Exception Handling

Using plain Suds

```python
try:
    sessionId = client.service.login(auth, credentials)
except suds.WebFault as e:
    if e.fault.detail.lcatException.type == 'SESSION':
        print "Login failed: %s" % e
    else:
        raise
```

Using python-icat

```python
try:
    client.login(conf.auth, conf.credentials)
except ICATSessionError as e:
    print "Login failed: %s" % e
```
Example: Searching

Using plain Suds

```python
searchres = client.service.search(sessionId, "Facility")
if len(searchres) != 1:
    raise RuntimeError("Expected to find one facility")
else:
    facility = searchres[0]
```
Example: Searching

**Using plain Suds**

```python
searchres = client.service.search(sessionId, "Facility")
if len(searchres) != 1:
    raise RuntimeError("Expected to find one facility")
else:
    facility = searchres[0]
```

**Using python-icat**

```python
facility = client.assertedSearch("Facility")[0]
```

Rolf Krahl (HZB) python-icat
Example: Searching

Using plain Suds

```python
searchres = client.service.search(sessionId, "Facility")
if len(searchres) != 1:
    raise RuntimeError("Expected to find one facility")
else:
    facility = searchres[0]
```

Using python-icat

```python
facility = client.assertedSearch("Facility")[0]
```

Using python-icat (more)

```python
# Assert there is at least one Investigation
investigation = client.assertedSearch("Investigation",
assertmax=None)[0]

# Assert there is at most one Instrument
res = client.assertedSearch("Instrument", assertmin=0)
```
A module `icat.cgi` helps writing CGI scripts. It does session management: the ICAT session Id is set as a cookie in the user’s browser.

Methods `Entity.getUniqueKey()` and `Client.searchUniqueKey()` to create a unique object identifier and to search for the object corresponding to an identifier respectively.

Example scripts `icatdump.py` and `icatrestore.py` that dump the whole content of an ICAT to a file (YAML) and restore it from the dump file respectively.
icat.client.Client is a suds.client.Client. Everything you can do with a Suds client, you can do with a python-icat client.

The ICAT entity objects created by client.new(...) or returned by a search live in a hierarchy of classes based on icat.entity.Entity.

The ICAT entity objects mimic very closely the behavior of corresponding Suds objects. They are converted transparently from and to Suds objects as appropriate.
System Requirements

- Python 2.6 or newer (Python 2.6 requires a patch).
- Suds, either 0.4 or jurko fork, the latter is recommended.
- argparse (in system library in Python 2.7 or newer).
- The example scripts use PyYAML, but this is not needed to use the library itself.

Download

- python-icat 0.4.0 available at http://code.google.com/p/icatproject/wiki/PythonIcat
- BSD license.
### System Requirements

- Python 2.6 or newer (Python 2.6 requires a patch).
- Suds, either 0.4 or jurko fork, the latter is recommended.
- argparse (in system library in Python 2.7 or newer).
- The example scripts use PyYAML, but this is not needed to use the library itself.

### Download

- python-icat 0.4.0 available at http://code.google.com/p/icatproject/wiki/PythonIcat
- BSD license.

Thank you for your attention! Questions?
A customisable domain specific ICAT interface

Frazer Barnsley, Tom Griffin
ISIS Facility

Elettra, Trieste
November 2013
tom.griffin@stfc.ac.uk
Contents

• What?
• Why?
• Who?
• How?
• Where?
• When?
Why?

- TopCAT is very generic, so does not provide information to our users in a domain specific way
- This is fine for some things
- Some domains have specific workflows and standard ways of working
- Can we have ‘custom’ portals for them, without too much work?
Who?

- Customers
  - ISIS Disordered Materials Group
  - Inelastic spectra database

- Creators
  - ISIS Computing group
  - Me
  - Frazer Barnsley (graduate secondment)

- Will be added to contrib
How?

- Uses ICAT as the database and IDS to store and serve data
- Configuration file defines
  - Names
  - Numbers of samples
  - Parameters (Investigation, sample, dataset, datafile)
  - Datafiles (uploads)
  - Tooltips
  - Friendly names
- Creates ICAT entries (including ParameterTypes)
- Grails – Groovy on Rails
- Iterative development with science groups
- Not much code. Lots of nice things for free
- Tomcat, Glassfish, Jetty, etc
Where?

- Currently on a development server (RAL only)

Live Demo......

http://130.246.49.212:8080/SimpleICATBrowser/
When?

• Next release to customers – end of this week

• Must be finished by end of November
Questions...
Authorisation Rules at ISIS

Tom Griffin, STFC ISIS Facility
ICAT Workshop
Dublin
March 2014
tom.griffin@stfc.ac.uk
Introduction

- Implements a defined data policy
  - PaNdata policy based.
- Public Tables
- Public Steps
- Rules
- UserGroups
Background

- Rules – by default everything is closed.
- ISIS data policy requires >100 rules…..still not finished
- Are (now) sanity checked on creation
- Required at every level for direct access (see PublicSteps)
- Can be checked:

```java
isAccessAllowed(String sessionId, EntityBaseBean bean, AccessType accessType)
```
Public Tables

• For read only access to open tables
• Cached
• Instrument, Application, DatasetType, InvestigationType etc
• Can be used for link tables:
  DataCollectionDatafile, InvestigationUser
Public Tables

List<String> publicTables = new ArrayList<>();
publicTables.add("Application");
publicTables.add("DatafileFormat");
publicTables.add("DatasetType");
publicTables.add("Facility");
...<cut>...

publicTables.add("DataCollectionDatafile");
publicTables.add("DataCollectionDataset");
...<cut>...

List<EntityBaseBean> publicRules = new ArrayList<>();
for (String publicTableName : publicTables) {
    Rule publicRule = new Rule();
    publicRule.what = publicTableName;
    publicRule.crudFlags = "R";
    publicRules.add(publicRule);
}
port.createMany(sessionId, publicRules);
Public Steps

• Allow access to a related object (attribute)
• ONLY used in INCLUDE processing.
• Offer a significant speedup on INCLUDE
• Thing -> ThingParameter
• Investigation -> samples, publications, users, investigationInstrument etc
Public Steps

String[] parameterBasedPublicSteps = new String[]{"Investigation", "Dataset", "Datafile", "Sample"};

for (String table : parameterBasedPublicSteps) {
    PublicStep paramPublicStep = new PublicStep();
    paramPublicStep.origin = table;
    paramPublicStep.field = "parameters";
    publicSteps.add(paramPublicStep);
}
Public Steps

```java
String[] publicStepsFromInvestigation = new String[] {"samples", "publications", "shifts", "investigationUsers", "keywords", "investigationInstruments"};

for (String step : publicStepsFromInvestigation) {
    PublicStep invToSomething = new PublicStep();
    invToSomething.origin = "Investigation";
    invToSomething.field = step;
    publicSteps.add(invToSomething);
}
```
Rule Structure

- Administrators
- Safe Admin (read all)
- [Raw] Data Ingestor
- Instrument Scientists
- Investigators
- Disordered Materials Database
- DOI service
- Unembargoed data
Administrators

- Easy

List<String> allTables = port.getEntityNames();
for (String table : allTables)
{
    Rule rule = new Rule();
    rule.grouping = facilityAdmins;
    rule.crudFlags = "CRUD";
    rule.what = table;
    port.create(sessionId, rule);
}
Other super groups

- Safe admins – same rules, no ‘CUD’
- Data Ingestors: ‘CRU’, fewer tables
Instrument Scientists

• Defines access relative to instruments

SELECT i FROM Investigation i
JOIN i.investigationInstruments ii
JOIN ii.instrument inst
JOIN inst.instrumentScientists instSci
JOIN instSci.user u
WHERE u.name = :user

Investigation, Dataset, Datafile, Sample, SampleType + 4x Parameters
SELECT df FROM Datafile df
JOIN df.dataset d
JOIN d.investigation i
JOIN i.investigationInstruments ii
JOIN ii.instrument inst
JOIN inst.investigationScientists instSci
JOIN instSci.user u
WHERE d.name='Default'
AND u.name = :user;
Investigators

- Defines access relative to investigation role

DatafileParameter <-> Datafile <-> Dataset <-> Investigation <-> InvestigationUser <-> User <-> User [name = :user]

Investigation, Dataset, Datafile, Sample, SampleType + 4x Parameters
Disordered Materials Database

- Write (authenticated) and open read to domain specific ‘database’

```sql
SELECT df FROM Datafile df
JOIN df.dataset ds
JOIN ds.investigation i
JOIN i.type it
WHERE it.name = 'Disordered Materials'
GROUPING = 'Disordered Materials Publishers'
Access = CRUD
```
DOI Account

- Requires read access to generate landing pages

```sql
SELECT i FROM Investigation i WHERE i.doi IS NOT NULL

SELECT ds FROM Dataset ds WHERE ds.doi IS NOT NULL
```

- DOI creation runs as Data Ingestor
Unembargoed Data

• Read for all authenticated users

SELECT i FROM Investigation i WHERE i.releaseDate < CURRENT_TIMESTAMP

SELECT dfp FROM DatafileParameter dfp
JOIN dfp.datafile df
JOIN df.dataset d
JOIN d.investigation i
WHERE d.name = 'Default'
AND i.releaseDate < CURRENT_TIMESTAMP

Investigation, Dataset, Datafile, Sample, SampleType + 4x Parameters
Complications

- Lock updates when DOI <> null
- Granting permissions
  - Create a group per investigation
  - What about finer grain?
  - Adding new users
  - What is ICAT, what is User office?
- Allowing partial updates
  - Investigation.release_date only
- Performance
Questions...
Authorization Rules
Setup of Authorization Rules based on Groups of Users

Rolf Krahl

ICAT Meeting, Dublin, Mar 2014
Why using groups to setup the authorization rules?

- Authorization based on groups is very flexible.
- Easy to grant or to revoke permissions: simply add the user to or remove him from the corresponding group.
- Access policies may be individually defined for each investigation.
- Users may manage permissions themselves. They only need CRUD permission on UserGroup related to the corresponding group.
- InvestigationUser is also used for other purposes (e.g. TopCAT). One might wish to setup permissions independently from this.
For each investigation, create three access groups:

**investigation_<name>_writer**: Shall get CRUD permission on objects related to the investigation, such as Datafiles, Datasets, Samples, Keywords, Parameters and so on.

**investigation_<name>_reader**: Shall get R permission on objects related to the investigation.

**investigation_<name>_owner**: Shall get permission to manage access permissions on the investigation.

Here `<name>` is replaced by the investigation name.
Simple way to setup access rules: create a set of rules for each investigation.

**Rule**

- **crudFlags**: CRUD
- **what**: Datafile <-> Dataset <-> Investigation[name='<name>']
- **grouping**: investigation_<name>_writer

**Rule**

- **crudFlags**: CRUD
- **what**: UserGroup <-> Grouping[name='investigation_<name>_writer']
- **grouping**: investigation_<name>_owner
Simple.

Works fine in test installation.

Drawback: excessive number of rules, 28 rules per investigation, more than a half of all objects in the ICAT are rules.

Will this scale reasonably in a production size deployment?
Try to achieve the same result with a limited number of generic rules:

Rule

```
crudFlags: CRUD

what: 
SELECT df FROM Datafile df
JOIN df.dataset ds JOIN ds.investigation i
JOIN Grouping g JOIN g.userGroups ug
JOIN ug.user u
WHERE g.name =
CONCAT('investigation_',i.name,'_writer')
AND u.name = :user

grouping: NULL
```
### Rule

**crudFlags:** CRUD

**what:**
```sql
SELECT aug FROM UserGroup aug
JOIN aug.grouping ag
JOIN Investigation i
JOIN Grouping g JOIN g.userGroups ug
JOIN ug.user u
WHERE ag.name = 
CONCAT('investigation_',i.name,'_writer)
AND g.name = 
CONCAT('investigation_',i.name,'_owner'
AND u.name = :user
```

**grouping:** NULL
Works in principle in test installation.
Only fixed set of static rules.
Drawback: incredible slow! Seven minutes to answer a simple query on a test ICAT having about 700 investigations.
Why is it so slow?

<table>
<thead>
<tr>
<th>Query</th>
</tr>
</thead>
</table>
| SELECT df FROM Datafile df  
JOIN df.dataset ds JOIN ds.investigation i  
JOIN Grouping g JOIN g.userGroups ug  
JOIN ug.user u  
WHERE g.name = CONCAT('investigation_\',i.name,\'_writer\')  
AND u.name = :user |

Missing relation between Investigation and Grouping  
⇒ Need to evaluate string expression on full cartesian product.

Complexity: $O(n^2)$ in the number of investigations.
Possible solution: Add the missing relation. Add to ICAT schema:

**InvestigationGroup**

Many to many relationship between investigation and grouping
Uniqueness constraint: grouping, investigation

Relationships:

<table>
<thead>
<tr>
<th>Card</th>
<th>Class</th>
<th>Field</th>
<th>Cascaded</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,1</td>
<td>Investigation</td>
<td>investigation</td>
<td>No</td>
</tr>
<tr>
<td>1,1</td>
<td>Grouping</td>
<td>grouping</td>
<td>No</td>
</tr>
</tbody>
</table>

Other fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>role</td>
<td>String [255]</td>
</tr>
</tbody>
</table>
Add relations between Investigation and Groups:

**InvestigationGroup**

- **investigation**: Investigation `<name>`
  - **grouping**: Grouping `investigation_<name>_writer`
  - **role**: writer

and accordingly for reader and owner.

Add rules:

**Rule**

- **crudFlags**: CRUD
  - **what**: Datafile <-> Dataset <-> Investigation <-> InvestigationGroup [role='writer'] <-> Grouping <-> UserGroup <-> User [name=:user]

- **grouping**: NULL
**Rule**

**crudFlags:** CRUD

**what:**
```
SELECT tug FROM UserGroup tug
JOIN tug.grouping tg
JOIN tg.investigationGroups tig
JOIN tig.investigation i
JOIN i.investigationGroups uig
JOIN uig.grouping ug
JOIN ug.userGroups uug JOIN uug.user u
WHERE (tig.role = 'writer'
OR tig.role = 'reader')
AND uig.role = 'owner' AND u.name = :user
```

**grouping:** NULL
- Solves the issue: only one fixed set of static rules. Only three Grouping and three InvestigationGroup per Investigation.
- Provides all the flexibility.
- Should scale reasonably, at least no obvious reason why it should not.
- Requires a change in the ICAT schema.
- But: this change is limited to the addition of the new type. Already existing types are not altered $\Rightarrow$ no compatibility issues. Sites not using it should not be affected in any way.
Thank you for your attention!

Comments? Discussion?
Which Objects to Setup Rules for?

Writers get CRUD permission on:

- Sample <-> Investigation,
- Dataset <-> Investigation,
- Datafile <-> Dataset <-> Investigation,
- InvestigationParameter <-> Investigation,
- SampleParameter <-> Sample <-> Investigation,
- DatasetParameter <-> Dataset <-> Investigation,
- DatafileParameter <-> Datafile <-> Dataset <-> Investigation,
- Shift <-> Investigation,
- Keyword <-> Investigation,
- Publication <-> Investigation,
- InvestigationInstrument <-> Investigation,

they get RU permission on Investigation, and R permission on InvestigationUser <-> Investigation.
Mantid & ICAT
About me

{
    "name" : "Jay Rainey",
    "affiliation" : "ISIS, Mantid",
    "email" : "jay.rainey@stfc.ac.uk",
    "github" : "@jawrainey"
}
Agenda

1. Mantid
2. ICAT within Mantid
3. IDS within Mantid
4. Future plans
5. Questions
Mantid

- What is Mantid?
  - Framework that facilitates HPC & visualization and analysis of neutron and muon data
  - Collaboration between ISIS and SNS
  - Supports various techniques
    - Inelastic (Direct/Indirect)
    - Diffraction
    - Muon
    - Disordered Materials
Mantid

- Features include:
  - Flexible: Instrument/Facility/Technique independent
  - Cross platform (UNIX, Win, OS X)
  - Open source (@mantidproject)
  - Ease of use (Expert -> Visitor)
Mantid - visualisations

- MantidPlot
Mantid - visualisations

- Custom interfaces
ICAT within Mantid

• Why is it used?
  – Load data directly from archives
  – Easily access catalogued data

• How is it used?
  – **New** Catalog == **new** adaptor
ICAT within Mantid

- Search parameters
ICAT within Mantid

- Search results

<table>
<thead>
<tr>
<th>Investigation id</th>
<th>Title</th>
<th>Instrument</th>
<th>Run range</th>
<th>Start date</th>
<th>End date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1390019</td>
<td>The structure of the ionic conductance</td>
<td>GEM</td>
<td>62862</td>
<td>2013-05-30</td>
<td>2013-05-30</td>
</tr>
<tr>
<td>1390020</td>
<td>Refinement of the ND3 deficit in...</td>
<td>GEM</td>
<td>62855</td>
<td>2013-05-29</td>
<td>2013-05-29</td>
</tr>
<tr>
<td>1390046</td>
<td>Undeformed, R242 5x15 mm</td>
<td>GEM</td>
<td>62831, 6303...</td>
<td>2013-05-27</td>
<td>2013-06-11</td>
</tr>
<tr>
<td>1390028</td>
<td>Al cylinder with 10% deformation</td>
<td>GEM</td>
<td>62838-62839</td>
<td>2013-05-27</td>
<td>2013-05-28</td>
</tr>
<tr>
<td>1390031</td>
<td>Al cylinder with 40% deformation</td>
<td>GEM</td>
<td>62832-62833</td>
<td>2013-05-27</td>
<td>2013-05-27</td>
</tr>
<tr>
<td>1390030</td>
<td>Al cylinder with 30% deformation</td>
<td>GEM</td>
<td>62834-62835</td>
<td>2013-05-27</td>
<td>2013-05-27</td>
</tr>
<tr>
<td>1390029</td>
<td>Al cylinder with 20% deformation</td>
<td>GEM</td>
<td>62836-62837</td>
<td>2013-05-27</td>
<td>2013-05-27</td>
</tr>
<tr>
<td>CAL_GEM...</td>
<td>Y2O3 pos 5</td>
<td>GEM</td>
<td>62414</td>
<td>2013-05-15</td>
<td>2013-05-15</td>
</tr>
<tr>
<td>CAL_GEM...</td>
<td>Si SRM640c 8mm pos 6</td>
<td>GEM</td>
<td>62410</td>
<td>2013-05-14</td>
<td>2013-05-14</td>
</tr>
<tr>
<td>CAL_GEM...</td>
<td>Empty CCR 4K 15x40 mm</td>
<td>GEM</td>
<td>62402-62409</td>
<td>2013-03-29</td>
<td>2013-05-14</td>
</tr>
</tbody>
</table>
ICAT within Mantid

- Investigation datafiles
IDS within Mantid

• Downloading (GET)
  – Off-site access
  – Local archive access not required
  – Access latest datafiles
IDS within Mantid

• Publishing (PUT)
  – Share analysed data
  – Automatically upload workspace history
IDS within Mantid

- Publishing GUI

![Catalog Publish Window](image.png)

**Catalog Publish**

Allows the user to publish datafiles or workspaces to the information catalog.

- **Workspace**: GEM62862
- **Name in catalog**: eureka.nxs
- **Investigation number**: 1193002
- **Description**: A useful description of operations performed on data
Future plans

• Cross facility querying (optional)
  – One query to be sent to each logged in catalog.
  – Results collated on search GUI
Future plans

• DOIs generated for published *public* data
Questions?
IRO Viewer (Data Journal)

Preservation, Context and Linking

Brian Matthews and Antony Wilson
STFC

ICAT Meeting
Dublin, 25
Overview

- Facility data and research objects
- Components
- Demo?
- Preservation
- Summary

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Facility Data and Research Objects
DOI Data Access Process
Is this enough?

• What we have so far is good for:
  • us to manage data
  • users to access their own data
  • citation of raw data

• But
  • Publication and exchange of Investigations
    • Traceability and Validation?
    • Reuse of the data?
  • Preservation of Investigations

• Need to make context more explicit
  • Focussing on the dataset is the wrong subject of discourse
Facilities Data Lifecycle

Scientist submits application for beamtime

Facility committee approves application

Facility registers, trains, and schedules scientist’s visit

Facility registers, trains, and schedules scientist’s visit

Scientists visits facility, run’s experiment

Raw data filtered, and stored

Tools for processing made available

Record Publication

Subsequent publication registered with facility

Data analysis

Proposal

Approval

Scheduling

Experiment/Investigation

Data storage

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Publishing Investigations

• So what we want is a record of EXPERIMENTS not data.

• Thus want the record of the context
  • The experimental intention and actors
  • The instruments and configurations used
  • The sample
  • The environmental parameters and context
  • The Raw Data

• Thus we want to publish a record of the whole INVESTIGATION
  • Can get most of this from what we have

• The Investigation becomes a “first class” research object
  • Published
  • Identified and treated as a single entity
  • Cited and credited
  • Record of the output of the facility

• Analogous to a Journal Article
  • Investigation as the unit of discourse for scientific facilities.

• But also as an access point for validation and reuse
  • Because we have a record of what actually happened.
Our DataCite entries are in fact Investigations

(red is for “data” notion, and green is for “investigation”)

Data collected on the GEM instrument at the ISIS facility.

RB920025

Investigation title: Crystal and magnetic structures of EuW01+xN2-x.

Creator: Kusmartseva, A
Creator: Rodgers, J A
Creator: Attfield, J P

DOI: 10.5286/ISIS.E.24071239

Date of Experiment: Tue Aug 04 14:38:23 BST 2009

Publisher: STFC ISIS Facility

Data format: RAW/Nexus
Select the data format above to find out more about it.

Data Citation

The recommended format for citing this dataset in a research publication is as:
[author], [date], [title], [publisher], [doi]

For Example:

Abstract

Eu2+ d0- transition metal perovskites are of interest as potential multiferroics when undoped, or as CMR materials. EuWO1+xN2-x is a new magnetoresistive material and exists over a broad range of x = -0.2 to 0.5. It has a ferromagnetic ordering transition at TC = 12 K. Neutron diffraction is needed to determine the \(\overline{1}112/m\) monoclinic superstructure evidenced by TEM that arises from O/N ordering and octahedral tilting, and the magnetic order. This may include a coexistence of antiferromagnetic/ferromagnetic orders (as found in a previous GEM study of the analogue EuNbO2N). 2 days on GEM are needed to study 2 samples with different x values (one stoichiometric x =0, the other highly doped x = 0.5) because of high absorption by Eu.
• Represent the “investigation” as a Research Object
  • Research Objects (ROs) are semantically rich aggregations of resources that bring together data, methods and people in scientific investigations. Their goal is to create a class of artefacts that can encapsulate our digital knowledge and provide a mechanism for sharing and discovering assets of reusable research and scientific knowledge
    • [www.researchobject.org](http://www.researchobject.org) and elsewhere

• Use RDF, Semantic Web and Linked Data Technologies to support this
  • And we can experiment with these tools for our data
Building an Investigation Research Object

Investigation #n
DOI:STFC.xxx.n

:investigator

:instrument

:sample

Software Repository

Software Package 1

cito:cites

cito:cites

:inputDataset

:outputDataset

:dataset

:application

:relatedDataset

:publication

:publication
Annotations

- Annotations used to construct a Research Object
  - http://www.w3.org/community/openannotation/
  - More on this later
Components
Annotation – Linking Software

This diagram illustrates the linking of software with an annotation and investigation. The annotation is represented by "2014-03-01T11:30:23Z" and "j.s@example.org". It is linked to the person "John Smith" with the email "j.s@example.org". The annotation is motivated by "2014-03-01T11:30:23Z" and serialized by "DataJournal". The foaf:Person "John Smith" is the agent responsible for the annotation. The software is represented by "Software" and the investigation by "Investigation".
Demo
Investigations are grouped into cycles.
This is the main index for the ISIS Data Journal.

There is an entry in the journal for every investigation carried out at ISIS that has a DOI or is no longer in the embargo period. The investigations are grouped into cycles, where a cycle is a period of time during which ISIS was running. The references for the investigation are in the form [Author], [Date], [Title/RB Number], [Publisher], [DOI], where RB Number is a unique number allocated to the investigation. The investigations are sorted by start date. Long running investigations may appear in more than one cycle.

- cycle_13_4 (11/11/2013 - 31/01/2013)
- cycle_13_3 (01/09/2013 - 10/11/2013)

Dr Winfried Koekemoer et al; (2013): 1230027, STFC ISIS Facility.

Filters are used to restrict results.
Investigations are presented in the style of references.
The data from the triple store contains links to images on the ISIS web site. Details about an investigation are retrieved from the triple store.
Data are retrieved from the PANKOS ontology based on the instrument name.

Saving will create an annotation containing the URI of the ontology object.
Creating SIP

Now contains merged data from the ontology
Preservation
User looks at data journal entry

Is it ready to publish?

Yes

Does it meet min checks?

Yes

Run process to make it preservable

Finish

No

Inform user of the missing aspects

No

This is the question that the creator of the experiment will ask

No

Does it meet min checks?

Yes

Run process to make it preservable

Finish

No

Inform user of the missing aspects

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
Replacing Na ions in NaxCoO2 with divalent Ca ions results in the formation of ....
Automated Preservation

User looks at data journal entry

Is it ready to publish?

Yes

No

Does it meet min checks?

Yes

Run process to make it preservable

Inform user of the missing aspects

Finish

SCOUT looks at embargo date on the content

Embargo date passed?

Yes

Is there an Archival package?

Yes

Start AIP creation process

No

No

Advise user

May use error log

No, look at the next item

Does it meet min stds?

Yes

Add any extra rep info

Create AIP

No

Finish

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
• When is a Investigation Research Object Complete?
  • Does this change over time?
  • Does it changes depending on who you are?
• Preserving links – how much trust do you have in others?
  • What does this mean for the preserved object
• Is all data of the same value?
  • For ISIS data in SCAPE : should there be different processes for different investigation types, samples etc?
Summary

• **DONE**
  • Initial IRO built from ICAT data
  • Links to ISIS web site automatically added
  • User may add links to PaNData ontology
  • User may add links to software packages
  • User may initiate archive process
  • IRO archived via SCAPE connector to Fedora repo

• **What next**
  • Add more data to archived IRO
  • Use SCOUT to automate triggering of archive process
  • Link in data from institutional repository

---

This work was partially supported by the SCAPE Project.
The SCAPE project is co-funded by the European Union under FP7 ICT-2009.4.1 (Grant Agreement number 270137).
ICAT Job Portal

a generic job submission system built on a scientific data catalog

24-25 March 2014

Steve Fisher, Kevin Phipps and Dan Rolfe
Rutherford Appleton Laboratory - STFC
Use Case

- LSF operate OCTOPUS imaging cluster: lasers coupled to interconnected microscopy stations.
  - a large number of data files
  - applications to process and visualise them
  - interactive program with an easy to use GUI to offer lists of raw and processed datasets and offer the ability to process those datasets

- Some requirements
  - GUI and command line from on and off site.
  - Consult the metadata to locate the data.
  - Submit batch jobs to Linux nodes; listing, cancelling and retrieving output.
  - Interactive GUI based analysis/visualisation jobs able to access data.
  - Select and submit multiple datasets for processing through applications.
  - No facility dependence: configurable menus, datasets types, jobs and associated job parameters.
A solution

- Build a batch and interactive job portal on top of ICAT and IDS
- Implement GUI access via Google Web Toolkit
- Provide command line access via RESTful interface
- Use other tried, tested, scalable and preferably open source components
Architecture Overview

User's PC

- Web browser
- CLI
- Remote Desktop client

Head Node

- ICAT Job Portal webapp
- Torque batch server

Worker Node 1

- Prepare job
- Facility software
- Torque worker node

Worker Node n

Submit batch job
Assign interactive job
Assign batch job

https
rdp
Head Node Architecture

- Head Node
- Torque batch server
- JEE Application Server
  - ICAT
  - IDS
  - GWT
  - REST
- Metadata database
- File storage
- Jobs database
- XML Job Descriptions and Job Dataset Parameters

IJP

Torque batch server

Head Node
### Job Portal Main Panel (Datasets)

**Datasets**

<table>
<thead>
<tr>
<th>project</th>
<th>Any instrument</th>
<th>Any experiment type</th>
<th>Any number of channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unknown Instrument</td>
<td>Unknown experiment type</td>
<td>Any number of channels</td>
<td></td>
</tr>
<tr>
<td>OctopusSM2</td>
<td>Colocalisation</td>
<td>1 channel</td>
<td></td>
</tr>
<tr>
<td>OctopusSM3</td>
<td>Undefined</td>
<td>2 channels</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>startDate</th>
<th>BETWEEN</th>
<th>endDate</th>
<th>BETWEEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012 Jan 1 12:00:00</td>
<td>2013 Jan 1 12:00:00</td>
<td>2012-11-27T14:18:17Z</td>
<td>undefined</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>20120524_0002_0001_632c1e69-9f52-4a39-a648-855ed5592c27</td>
<td>Drosophila 638 nm laser</td>
<td>1234</td>
</tr>
<tr>
<td>20120525_0004_0001_0bb1b30e-d287-4b9a-8929-3b52f5ddc208</td>
<td>T47D 3 Affibody 639 nm laser</td>
<td>5678</td>
</tr>
<tr>
<td>20120524_0002_0001_5e2805b5-f99a-4b4a-93d7-61a962a38912</td>
<td>T47D 3 Affibody 639 nm laser</td>
<td>9101</td>
</tr>
<tr>
<td>20120524_0002_0001_c1b3c5-55-0f9d-4ca4-985291f81e2</td>
<td>T47D 3 Affibody 639 nm laser</td>
<td>1111</td>
</tr>
<tr>
<td>20120524_0002_0001_d29e970-7b0d-40f6-9e06-b32670896d1d</td>
<td>T47D 3 Affibody 639 nm laser</td>
<td>1212</td>
</tr>
<tr>
<td>20120524_0002_0001_7ee07c8e-dc7d-4b6c-a699-6e62abf92fe</td>
<td>T47D 3 Affibody 639 nm laser</td>
<td>1313</td>
</tr>
</tbody>
</table>

**Attributes**

- **endDate**: 2012-11-27T14:18:17Z
- **experiment_type**: Undefined
- **id**: 7201
- **instrument**: OctopusSM3
- **location**: Dummy investigation 1/20120524_0002_0001_7ee07c8e-dc7d-4b6c-a699-6e62abf92fe
- **name**: 20120524_0002_0001_7ee07c8e-dc7d-4b6c-a699-6e62abf92fe
- **nchannels**: 1
- **nframes**: 571
- **sample_description**: T47D 3 Affibody 639 nm laser
- **startDate**: 2012-11-27T14:16:21Z
Job Options

MSMM Viewer Project Options

View type  
- View  
- View beads  
- View whitelights  
- View reg residual frames  
- View reg model frames

Track method

Show variance image instead of image  

Do not load traces

Read features/tracks from hdf5 files (slow)

Set min,max for colour scale

Regular expression for images in directory

Do not clean levels/stats  (default=0) (min=0) (max=1.0)

Min number of detected features per frame range of a level/state  (default=2)

Threshold for the Chauvenet's outlier test  (default=2) (min=1) (max=5)

Set the (real) EM gain by hand

Quantum efficiency  (default=0.010000026) (min=-1.0) (max=1.0)

Set the (real) electron/ADU by hand

A unique identifier of the EMCCD  (default=Command Line)

Quit immediately after initialisation completes

Add a string to the view window title

Submit  
Close
## Job Status Panel

![Job Status Panel](image)

### Table: Job Status

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Worker Node</th>
<th>Batch Filename</th>
<th>Submitted</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>81.sig-10.esc.rl.ac.uk</td>
<td>sig-12.esc.rl.ac.uk</td>
<td>qmybdzrphr.sh</td>
<td>01-03-2013 14:41:54</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>78.sig-10.esc.rl.ac.uk</td>
<td>sig-12.esc.rl.ac.uk</td>
<td>icfhkvjhf.sh</td>
<td>12-02-2013 13:51:57</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>77.sig-10.esc.rl.ac.uk</td>
<td>sig-12.esc.rl.ac.uk</td>
<td>agefjhzfwf.sh</td>
<td>12-02-2013 13:51:51</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>76.sig-10.esc.rl.ac.uk</td>
<td>sig-12.esc.rl.ac.uk</td>
<td>xezhuyccms.sh</td>
<td>12-02-2013 13:40:39</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>75.sig-10.esc.rl.ac.uk</td>
<td>sig-12.esc.rl.ac.uk</td>
<td>fcebrhyxvp.sh</td>
<td>12-02-2013 13:40:29</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>74.sig-10.esc.rl.ac.uk</td>
<td>sig-12.esc.rl.ac.uk</td>
<td>iqlkkvabkk.sh</td>
<td>12-02-2013 10:50:48</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>73.sig-10.esc.rl.ac.uk</td>
<td>sig-12.esc.rl.ac.uk</td>
<td>dnfsmuakvy.sh</td>
<td>12-02-2013 10:48:21</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>64.sig-10.esc.rl.ac.uk</td>
<td>sig-12.esc.rl.ac.uk</td>
<td>ahtpltkhzc.sh</td>
<td>11-02-2013 15:27:14</td>
<td>COMPLETED</td>
</tr>
<tr>
<td>65.sig-10.esc.rl.ac.uk</td>
<td>sig-12.esc.rl.ac.uk</td>
<td>phqzrubckz.sh</td>
<td>11-02-2013 15:27:14</td>
<td>COMPLETED</td>
</tr>
</tbody>
</table>
Multiple Dataset Handling

- Jobs can accept a single or multiple datasets (specified in XML Job Description)
- Multiple datasets can be submitted to a job specified as accepting multiple datasets as input
- A separate batch job can be submitted for each dataset (with a single click)
- With multiple datasets selected, Job Options Form offers only options common to all datasets
Interactive jobs
Configuration

1. Create XML files for each dataset type picking out dataset features relevant to Job Options

2. Create XML Job Descriptions

3. Write applications (or wrappers around existing applications) – loading and saving datasets from IDS and recording provenance in ICAT
Job Options from XML

XML Job Description on Head Node

```
<jobOptions>
  <name>View</name>
  <groupName>View type</groupName>
  <type>boolean</type>
  <programParameter>--show-reg-beads</programParameter>
  <condition>numBeadFiles>0 && numChannels>1</condition>
</jobOptions>
```

```
<jobOptions>
  <name>Track method</name>
  <type>enumeration</type>
  <programParameter>--trackmethod</programParameter>
  <values>Simple</values>
  <values>SLH</values>
  <values>Biggles</values>
  <values>Simulation</values>
</jobOptions>
```

```
<jobOptions>
  <name>Do not clean levels/stats</name>
  <type>integer</type>
  <programParameter>--levels-no-clean</programParameter>
  <defaultValue>0</defaultValue>
</jobOptions>
```

Job Options Form in Web Browser

**MSMM Viewer Project Options**

- **View type**
  - View beads
  - View white lights
  - View reg residual frames
  - View reg model frames

- **Track method**
  - View type

- **Show variance image instead of image**

- **Do not load traces**

- **Read features/tracks from hdf5 files (slow)**

- **Set min, max for colour scale**

- **Regular expression for images in directory**

- **Do not clean levels/stats**
  - (default=0) (min=0) (max=10)

- **Min number of detected features per frame range of a level/state**
  - (default=2)

- **Threshold for the Chauvenet's outlier test**
  - (default=2) (min=1) (max=5)

- **Set the (real) EM gain by hand**

- **Quantum efficiency**
  - (default=0.910000026) (min=1.0) (max=1.0)

- **Set the (real) electron/ADU by hand**

- **A unique identifier of the EMCCD**
  - (default=Command Line)

- **Quit immediately after initialisation completes**

- **Add a string to the view window title**

**Submit**
Command Line Interface

- RESTful web service and Python client for job handling
- Alternative to using web browser
- May become preferred interface for some users
- Enables scripted interaction with IJP

```bash
$> ijp login db username fred password -
password: d3f58cf7-23fb-4e0a-89bc-292dcc142e20

$> ijp session
User ingest connected to ICAT 4.2.5 at https://rclsfserv010.rc-harwell.ac.uk:8181 with 1439 minutes left.

$> ijp jobtype
Available job types are:
view_ingested is interactive
ingest is batch
view_project is interactive
quincy is batch

$> ijp submit ingest gggg
2.rclsfserv010.rc-harwell.ac.uk

$> ijp status
2.rclsfserv010.rc-harwell.ac.uk, R
```
Status

- System has been implemented deployed and given to LSF for feedback
  - The system has the desired functionality and is responsive
  - Short informal weekly meetings between the developers and LSF have ensured the delivery of the desired product
- Other STFC facilities and groups are interested
Future Developments

- Improvements following user feedback
- Visualisation of Provenance
- Workflow Support
- Administration console
- Alternative remote desktop mechanism
- Alternative batch systems
- Portability
Experience of the ICAT API and Documentation
Agenda

1. ICAT API Experience
2. Documentation
3. Questions
ICAT API Experience

- C++ & gSOAP
- Very simple API
- Each call is (roughly) the same
  - One concrete example needed
- ICAT4 speed!
- Great return errors (IDS & ICAT)
Documentation

• Excellent IDS documentation
  – States what is required, and what is optional
Documentation

• Query Examples
  – No complex example of JQPL

    SELECT DISTINCT inves
    FROM Investigation inves
    JOIN inves.investigationInstruments invInst
    JOIN invInst.instrument inst
    JOIN inves.keywords keywords
    WHERE inst.fullName = 'GEM'
    AND keywords.name IN ('solid')
    ORDER BY inves.id DESC
Documentation

• ICAT Scheme documentation
  – Investigation name
  – Is the investigation ID (this is not documented)

• ICAT Client API documentation
  – Date format: Must be Y/M/D TIME (2011-12-15 00:00:00)
Questions?
Containers

24-25 March 2014

Steve Fisher
Rutherford Appleton Laboratory - STFC
Why not just Glassfish?

- Some sites are already running other containers and don’t want to know about another one.
- Application bugs might be revealed by testing in different containers.
- Glassfish support from Oracle changed in November last year - pointed out to me by Andy Goetz.

Oracle said that it would: "no longer release future major releases of Oracle GlassFish Server with commercial support — specifically Oracle GlassFish Server 4.x with commercial Java EE 7 support will not be released."

"Oracle recommends that existing commercial Oracle GlassFish Server customers begin planning to move to Oracle WebLogic Server."
ABANDON FISH!
MIGRATING FROM GLASSFISH TO JBOSS OR TOMEE

So long, and thanks for all the fish!
Findings

Report indicates that there is not much to choose.

Differences:

- Clustering
- JPA

WildFly 8

Commercial support via RedHat (JBoss)
Use Hibernate as JPA implementation
Java EE7 Certified

TomEE+

Derived from OpenEJB
Collection of Apache Components - including OpenJPA
Commercial support via Tomitribe
Choices

- Do nothing
- Try another container
- If successful could routinely test ICAT components on both or could switch
- Try a third container?
  - Having done two then three should be easy???
Proposal

- Try WildFly 8 and update the install scripts to be able to use either GlassFish or WildFly
- Add TOMEE+
- Decide on preferred container and do future testing only on that though bug reports on the other “ICAT supported” containers will be accepted and dealt with.
- Low on my priority list at the moment - but it doesn’t have to be done by me.
ICAT core, IDS and IJP Status and Roadmap

24-25 March 2014

Steve Fisher
Rutherford Appleton Laboratory - STFC
ICAT core - Status

Version 4.3 in October with subsequent bug fix releases
Emphasis now on stability and backwards compatibility
Main features of 4.3:

- The notification mechanism has been completely changed.
- Call logging has been added either to file or to a log table or both.
- Lucene free text search via the new searchText call.
- Various “small” schema changes
- Added new alternate query syntax (JQPL based)
  - If you are new to ICAT prefer the JPQL style
- Python setup script which works both on Linux and Windows.
Possible changes to the setup installation script

- **Objections to**
  - separate config step
  - reports of differences from example

- **Proposal**
  - avoid separate config step
  - `--vv` required to report differences from example
  - updated scripts will be packaged with new releases
ICAT Core - Next Release (May 2014)

Clustered deployment for better performance

• Chose to avoid container based clustering:
  ○ Tied to chosen container
  ○ Glassfish solution not suitable

• Issues
  ○ Lucene access must be synchronized
  ○ Gatekeeper has state - cached:
    ■ Set of public tables (derived from rules)
    ■ Set of public steps
  ○ Must disable JPA caching
Clustering Deployment

Bean Manager -> Lucene Singleton Bean

Lucene -> Gatekeeper

Bean Manager

JMS Destination

MDB

Gatekeeper

Using remote api - CORBA
ICAT Core - Next Release (May 2014)

- Improved API documentation – in particular queries and authorization
  - Necessary drudgery
ICAT Core - Next Release (May 2014) - but not in the road map

- A get function which takes an object with uniqueness constraint fields set
  - inv = get(sid, inv, "INCLUDE 1")

- Java client was generated for each server release - now being done separately.

- From WSDL which has no comments. If do it from server code can include:
  - Full Javadoc from the entity documentation held by the server
  - Useful constructor for each object taking the uniqueness constraint fields
    - inv = new Investigation(facility, name, visitId)

- getApiVersion() returns a number held by the server
  - could add new call getVersion and deprecate the old call

- authz addition suggested by Rolf
ICAT core - RoadMap (Sep 2014)

- The provision of a JSON based RESTful web service
  - this should be a rather easy add-on

- Ability to migrate metadata/data between ICAT/IDS instances.
  - allow a file to be prepared of data to be fed into ICAT
  - old XMLIngest functionality will be included
  - will be able to deal with any ICAT data
    - not just a hierarchical projection of that data
  - It will also allow a new ICAT to be configured very rapidly.
ICAT core - RoadMap (Jan 2015)

- ICAT - study the use of a non-relational database to give better performance at large scale.

- Some kind of hybrid between relational and non-relational might be optimal.
Questions

? Suggested changes to the roadmap ?
IDS - Status

First version recently released
Good feedback so far - but a few problems have shown up.

Structure of zip file:

"ids"/facility.name/investigation.name/investigation.visitId/dataset.name/datafile.name

- if any of the names have characters not suitable for chosen OS it does not work - the "/" is a special case
- in addition ESRF want to be able to control the zip file structure to give facility dependency which I had hoped to avoid.
- mapping the names can lead to duplicates
IDS - Proposed change

- Change the plugin interface to provide more information to the plugin
  - e.g. the Dataset.location field has been requested
- Add an extra call which the plugin must implement to compute the full file name of a zip file entry.
  - Could provide the existing algorithm as a default
  - Facility may choose to base the zip file entry on its own storage structure
    - trivially guarantees unique names
IDS - roadmap

Apr 2014
- The IDS computes checksums when data files are uploaded - use this information to provide background checking that all data can be correctly read to provide the expected checksum.

Aug 2014
- Add the ability to work well when deployed on a shared file system
  - upload and download without file movement
- Extend to provide FUSE file system authorized by ICAT
Questions

Suggested changes to the roadmap
IJP - Portability

Currently only works on a single platform
Our users want it to run on at least three very different platforms (different OS, different batch system etc.) - but to present a single GUI
This requires (at least)
1. abstracting the batch system
2. providing a very simple “meta-scheduler”
IJP - Other things

- Expect to make a series of small enhancements to meet user needs.
- Also want to make IJP available to others so we will provide a very simple demo configuration with jobs such as concatenating copying and deleting datafiles.
Questions

Suggested changes to the roadmap
TopCAT

March 2014

Wayne Chung
wayne.chung@stfc.ac.uk
TopCAT

- Current release TopCAT 1.11.0
- Added support for the ICAT 4.3
- Added support for IDS 1.1.0
- Updated GWT 2.5.1
- Updated GXT 2.31
- Added timed announcement message via TOPCATAdmin
- Added Free Text search (ICAT 4.3 only)
Next Release

April 2014

- Added External Redirect authentication type plugin
- Last used ICAT server and authentication type saved to cookie
- Provide data upload facilities via the IDS
- Remove support for ICAT below 4.2
- Bug fixes (https://code.google.com/p/topcat/issues/list)
Road Map

July 2014

- Add support for multiple facilities
- Rationalise and simplify the code to reduce dependencies
  - Result in more maintainable software
  - If plugin are desired, should not be compiled into distribution
- Desirable to separate off code for doing federation search, possibly into a small independent web service.
October 2014

- Add visualisation features (simple analysis of one and 2 dimensional data) by packaging such functionality already developed by the ICAT team at STFC for one of their facilities.

- Find replacement for GWT

- ICE (ICAT Editor) might be included in TopCAT
What Direction?

- What requirements do the community want for TopCAT?
- Is there really a use case for federation?
- Would it be better to provide a simple GUI for a single ICAT server?