SCIENCE AND TECHNOLOGY FACILITIES COUNCIL

Advanced Climate Research Infrastructure for Data (ACRID)

Report III: Work Package 3.1 & 3.2 – Tools for Managing Climate Research Software and Data

Arif Shaon, (STFC) and Colin Harpham (CRU, UEA)

1 June 2011

Revision History:

Version	Date	Authors	Sections Affected / Comments
0.1	01/06/2011	AS	Outline and Main Body

Contents

1 Introduction and Rationale

ACRID aims to develop effective means for publishing detailed workflows associated with the CRU's climate datasets. To this end, in previous phase of the project, we developed an information model to describe various aspects of the CRU workflows and designed a data management architecture to capture and manage the information defined by the model (Report II). This report details the development of the CRU data management architecture which involves the implementation and deployment of a number of tools to ensure adequate management of the software artefacts involved in processing climate data – including software versioning, configuration, publishing, and run-time metadata capture. In addition, these tools are also intended to facilitate the management of updates and versioning of datasets, capturing the workflow associated with their generation as well as enabling metadata export.

2 Requirements Specification

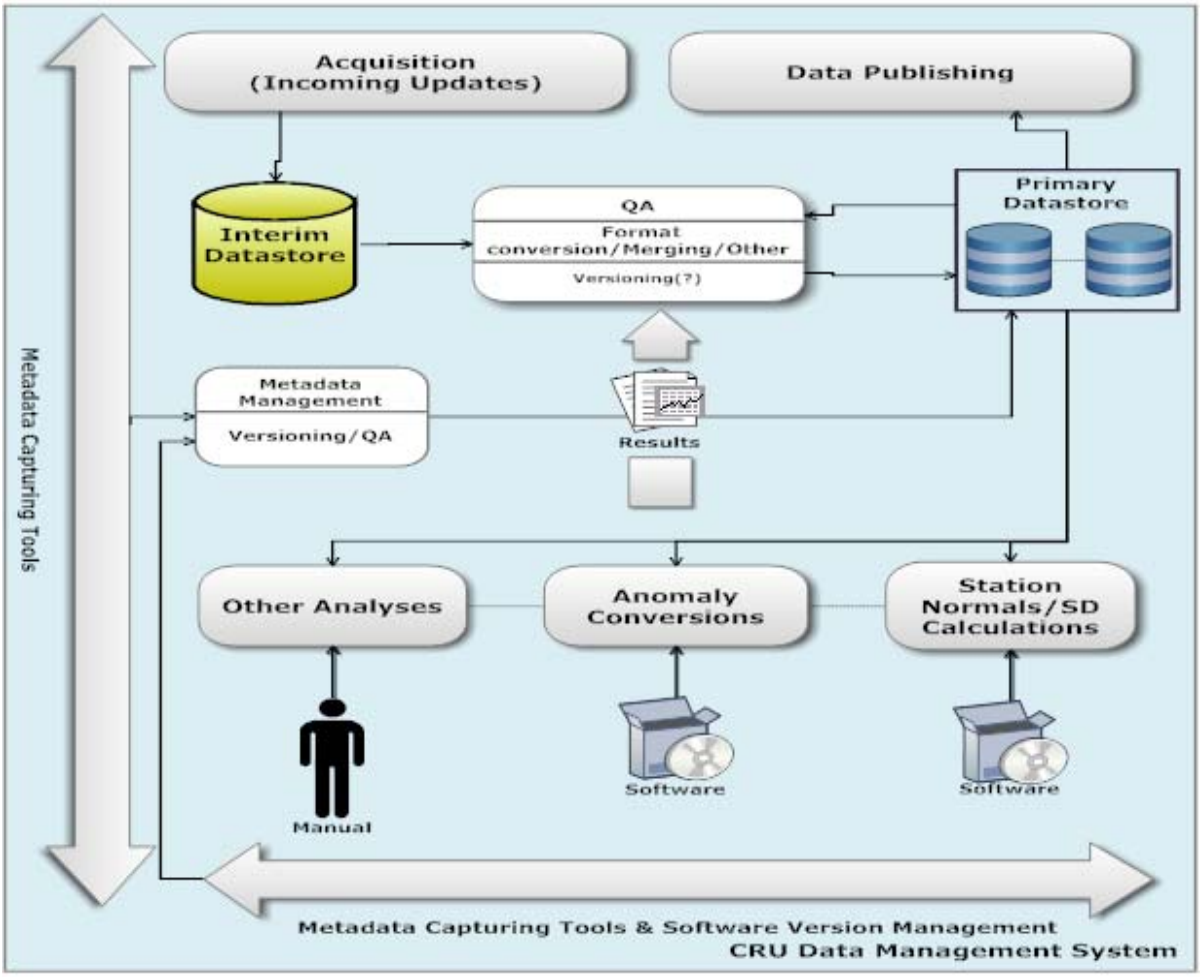

Figure 1: A Data management infrastructure for CRU

As illustrated in Figure 1, the tools were to meet the following functional requirements:

- **Capturing Source Observation Information** – this is to involve collecting relevant information (as defined in the ACRID information model) from the source measurements or observation datasets used to generate a CRU derived dataset, such as CRUTEM3 and Tree-ring chronologies.
- **Capturing Derived Observation information** – this is to involve collecting various metadata about the CRU observations defined in the ACRID information model
- **Capturing Process and Software Information** – this is to involve collecting detailed information (such as the version of the software used and software runtime parameters) about the process and software used to generate a CRU's derived observation in accordance with the ACRID information model.
- **Versioning and Storing the information captured** – this is to involve storing the information captured into a suitable storage media, such as a relational database to enable querying and access later. This should also involve efficiently handling updates to existing information using a suitable versioning mechanism.

3 Technologies Employed

We assessed the technical expertise available at CRU and chose the following technologies for developing the tools and ensuring their efficient managements, updates and administrations in future.

3.1 Programming Language

We have used Tool Command Language (TCL)[1] to develop the tools as CRU have substantial expertise of software development using TCL.

3.2 Software Version Management Tool

We have deployed Subversion[2] system (an open source version management system) within the CRU data management infrastructure to manage the software artefacts used to produce the CRU observational datasets. Our preference for Subversion over other similar systems, such as CVS[3] is mainly based on Subversion's capability to handle a wider range of file formats as well as attaching mote detailed metadata to files, and also it's increasing popularity within academic and research sector.

[1] TCL - http://www.tcl.tk/ [Last accessed 1 June 2011]
[2] Subversion - http://subversion.tigris.org/ [Last Accessed 1 June 2011]
[3] Concurrent Versioning System – http://en.wikipedia.org/wiki/Concurrent_Versions_System [Last accessed 1 June 2011]

3.3 Database

We have used PostgreSQL[4] with spatial database extension enabled through the PostGIS[5] plug-in to store various metadata about the CRU workflows. PostgreSQL (through the PostGIS extension) handles geometric attributes (e.g. point, polygon etc.) of a spatial dataset better than other open source databases, such as MySQL, by providing automatic serializations of such information in various spatial formats including Geography Markup Language (GML)[6].

4 Implementation and Methodology

We have developed a set of tools that implement the following functions:

4.1 Capturing and storing a new Source Observation record

This involves collecting the relevant information from a source file and making the following updates to the database:

- Adding a new record containing information about the feature of interest (e.g. station at which the observation was made) associated with the source observation to the database
- Adding (a) new record (s) to the database to represent the authorities associated with the source as appropriate.
- Add a new record representing the source observation itself and link it to the corresponding "feature of interest" and Process records (pre-populated via a web interface). This task also involves parsing the source observation file to generate CSML encoding of the data and storing it in the database. A TCL-based templating engine has been written to handle this task.
- Add new records representing the names and values of the parameters associated with the process, software etc. used to generate the observation.

4.2 Updating an existing Source Observation record

This involves collecting the relevant information from a source file and making the following updates to the database:

- Update feature of interest associated with observation in the Database using appropriate SQL update statements as needed.

[4] PostgreSQL - http://www.postgresql.org/ [Last accessed 1 June 2011]
[5] PostGIS - http://postgis.refractions.net/ [Last accessed 1 June 2011]
[6] GML - http://www.opengeospatial.org/standards/gml [Last accessed 1 June 2011]

- Add a new record representing the updated observation and link it to the corresponding feature of interest and Process records (pre-populated via web interface). Also mark the new record as the latest version of this observation using a combination of version and origin identifiers.
- Add new records representing the names and values of the parameters associated with the process, software etc. used to generate the updated observation.

4.3 Adding a new CRU Observation record

This involves extracting relevant information from a CRU observation data file (e.g. CRUTEM3 file) and the scripts/software used produce the file, and making the following updates to the database:

- Add a new record representing the observation data into and link it to the corresponding Process records (pre-populated via web interface).
- As with the Source observations, this task will also require parsing the observation data file to generate CSML encoding of the data and storing the CSML in the database. A TCL-based templating engine has been written to handle this task
- For some observations (e.g. Tree-ring), it may be needed to capture information about the sampled feature. This should be encoded in XML in accordance with the GML schema and stored in the database as XML. As with the Source observations, a TCL-based templating mechanism has been implemented to handle this task .
- Add new records representing the names and values of the parameters associated with the process, software etc. used to generate the observation.

4.4 Updating an existing CRU Observation record

This involves:

- Adding a new record representing the updated CRU observation data to the database and link it to the corresponding Process records (pre-populated via web interface) in the Process tables. Also mark the new record as the latest version of this observation using a combination of version and origin identifiers.
- Add new records representing the names and values of the parameters associated with the process, software etc. used to generate the updated observation.

5 Testing

We have tested the tools with some sample data files. But the tools will be subject to further more rigorous tests in the next phase of the project, which will involve configuring the GeoTOD linked-data server to expose the information captured by the tools as linked-data.

5 Conclusions

In the current phase of the project, we have developed and deployed a number of tools and software components that were proposed by the data management infrastructure presented earlier in Report II. In general, these tools and software systems are indented to enable capturing, versioning and storing various information about the CRU workflows, including process and software information in an efficient manner.

References

[1] ISO 19156:2010 - Geographic information — Observations and measurements