Automated Data Management for Climate Simulations

Daljeet Singh Sran[1], Christian Weihrauch[1], Kerstin Kleese van Dam[2], Mark Williams[2].

[1]University of Reading, Department of Computer Science
[2]CCLRC e-Science Centre

Abstract: This joint MSc project analysed an existing climate simulation process, based on the UK Meteorological Office's Unified Model, and developed a software system which automated the collection and storage of metadata for the simulations. The aim was to allow researchers to reuse past simulation data in order to save valuable compute time and other resources.

The project started with discussing the general scientific metadata problem and assessed the suitability of the CERA metadata model before focusing on implementing a model for the simulation model metadata. The CERA model was deemed suitable, with extensions, and a relational version of the model was implemented in a MySQL database.

The prototype system was designed as a web service coupled with a web service client. It currently supports ASCII and XML output formats and direct insertion into the backend MySQL database. This design provides a flexible, platform-neutral interface, which coupled with the web service WSDL allows easy development of client software.

Although a rudimentary web front end was implemented in PHP to search the metadata for this project, the design included integration with the CCLRC Data Portal as the ultimate search and retrieval tool.

1. Database Design & Metadata

The first half of this joint project was concerned with capturing the metadata requirements and implementing a database schema based on these requirements.

1.1 Project Aims

The aims of this half of the project were to:

- Analyse the UGAMP experiment process
- Identify required metadata
- Model the climate simulation process
- Assess CERA model suitability
- Database schema design
- Implement the database
- Implement a prototype system

1.2 UGAMP Experiment Process

The project domain was explored in conjunction with the UK Universities Global Atmospheric Modelling Programme (UGAMP) user group.

It was found that the UGAMP group had never really conceptualised how the experiment process worked in detail. They knew the process and had accepted certain elements as being black box scenarios; they knew what the process produced, and they deemed this sufficient for their purposes. The research was regarded as being important, not the technology used to conduct it. From the analysis it was discovered that the simulation process was more complex than had originally appeared. This was because as the simulation process was dissected more and more questions arose. The answers were not readily available and required further research on behalf of the UGAMP group and project members. While an overview was easy to develop there was much consultation needed.

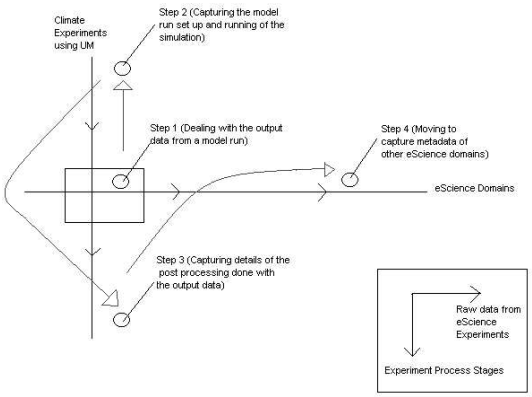

Figure 1: Climate experiment process

The main stages of the UGAMP Experiment processes were found to be:

Stage 1: Set Up

An Experiment can have many People working on it. A User configures the Unified Model User Interface (UMUI) in order to set up their Job. The

UMUI produces a set of Scripts which set up and run the Job on the UM. It also produces a Jobsheet which is a human readable version of the used configuration. An alternative to starting a Job simulation from the UMUI is to alter the Scripts from a previous use of the UMUI. Therefore there must be scope for capturing the details of the two distinct starting points.

Stage 2: Simulation

The Job simulation is run on the UM which is itself configurable by external Scripts. Any changes in the UM configuration must be recorded. Details are available of running the Job simulation known as the Job Log. Also a description of the Output needs to be detailed.

Stage 3: Post Processing

Processing of the Output data produced needs to be stored. This can involve further alteration of the actual output and/or the use of Visualisation Tools to produce Images. The Output data and Images can be used in subsequent Publications and References.

1.3 Metadata Requirements

From the preliminary analysis a decision had to be made regarding the required metadata. Five distinct levels were recognised and provided the naming convention in bold below.

For the researcher

e-Filing to catalogue their work, help them find items, remember details of experiments conducted in the past and general organisation of their work.

For the researcher's immediate colleagues (who use the same model)

e-Share to define the entry in the institute-specific centralised database containing various UMUI configurations. These entries can be changed or reused. To describe what an individual has added or changed in the original configuration of the job contained in the database. What UMUI set-up produced the images? Also what extra script files were used, and how these files altered the set up?

For broader collaborations (colleagues who use different but similar models)

e-Comparison to catalogue the experiment description (not details) so that different model results can be compared and contrasted.

For the community (who use other models but are involved in climate research)

e-Information to provide a broad description of the experiment so that others can derive information from it or re-use the data to perhaps drive other models.

For everyone else (who use data or are affected by it, interested parties and for educational purposes)

e-Education to provide a detailed description for a non specialist.

1.4 Climate Simulation

Climate simulations are the controlled running of a General Circulation Model, in this case the Unified Model. This is accomplished by setting the variables within the model set-up and specifying what you wish to measure, what variable you wish to alter in order to affect the outcome and specify some input data. The simulation then runs and the output produced is a set of values over a grid area, which can be as simple as temperature measurements.

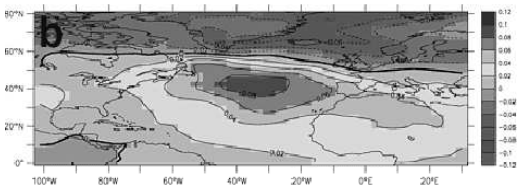

Figure 2: Product of a Climate Simulation

Simulations can also span up to thousands of years in simulated time. They are defined by a bounded area (Longitude and Latitude). Within these boundaries different parameters alter the simulated conditions. Measurements such as Radiation, Cloud Formation, Precipitation, Snow Cover etc. can then be made.

These simulations can be very costly in computation time and resources. The output produced can be very large in size. After initial analysis this data can be discarded, stored for future use, have a subset extracted or processed into an image such as shown in Figure 2.

1.5 CERA Model Assessment

Analysis of the Climate and Environmental data Retrieval and Archive (CERA) metadata model reveals that it is primarily designed to store raw climate data. It provides the mechanism to define the data's structure in order to provide an accurate description. This is done in the extension module Data_Organisation, combined with the core

blocks Coverage, Parameter and Spatial_Reference. A reasonable knowledge of the domain of climate science is needed to understand the different types of structure and nuances of climate data. CERA was designed specifically for this purpose. It is suitable to store the UGAMP group's data, and therefore could be used for this purpose.

It also provides high level details of the individual (Contact), Institute, Project (Metadata_Entry and Campaign), access restrictions (Distribution) and partially of post processing (Reference and the module Process_Step). This answered part of the UGAMP group's requirements. Extensions needed to be defined to capture the other remaining requirements, which are UM, UMUI, Job, Job Log, Scripts and Visualisation Tool. Therefore CERA could be used and extensions developed for the UGAMP group's metaUM database.

Benefits of using CERA included:

- An already implemented and tested solution that works
- Developed with domain knowledge of climate data's structure and fields
- Satisfies a core of the UGAMP problem
- Extendible framework

Shortcomings were:

- The purpose of the CERA architecture was first and foremost only to store the data sets produced from a simulation run
- It only answered a subset of the UGAMP storage requirements
- Still in development
- There exists a fair amount of complexity in integrating the fluid UM experiment process with the static nature of CERA

1.6 Metadata Schema

The complete metadata schema (metaUM, as it was called) comprises of the following components:

CERA Core v2.5
CERA Module DATA_ORGANIZATION v1.3
CERA Sub-Module PATCHED_DIMENSION v1.0
CERA Module DATA_ACCESS v1.0
UGAMP extension Module v1.0.
User Block
Job_Set_Up Block
Job_Run Block
Simulator Block
Post_Processing Block
Redefined CERA Module PROCESS_STEP v1.0

1.7 Schema Implementation

The metaUM schema was implemented in a MySQL database. The first task was to set up the database. The CERA tables were then converted and implemented into the metaUM database. Once this was achieved the relational schema derived in the ER analysis section was implemented. Together the metaUM database contains ninety-four relations, defined by eleven scripts named in the summary. Forty-four relations from the CERA core, thirty-two new relations from the UGAMP extension module and the remainder from the CERA defined extension modules were required for the project. Concurrently with this process the Web Client was developed to allow insertion, updating and retrieval of the data.

1.8 Prototype Query Interface

For the purposes of this project a prototype query interface was developed in PHP whilst the automatic data capture tool was also being developed.

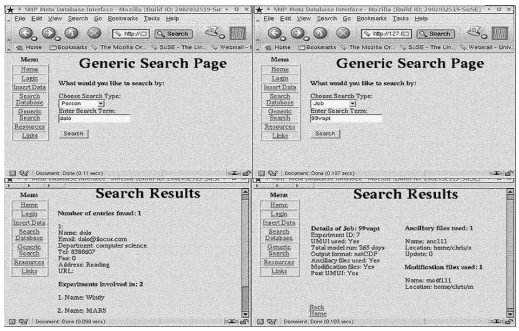

Figure 3: Prototype Screen Shots

The prototype web client implementation was tested successfully with set-up, query and closure functions.

2. Capture & Insertion Tool

The second half of this joint project focused on developing an automated metadata capture tool and a database insertion tool.

2.1 Project Aims

The aims were to:

- Develop a metadata capturing tool
  - Gather available metadata
  - Capture missing metadata

- Pre-process metadata
- Store metadata in XML and ASCII formatted files

- Develop a Database insertion tool
  - Store metadata in database

2.2 Design

The fundamental idea of the design is the use of a client-server approach. Therefore the software design is divided into two parts.

The first part requires a client program which works as a collector for the data on the user side, the second part is a server program which collects all the data from the clients and stores that data into an ASCII file, an XML file or sends the data via the network to the given metaUM database. The metaUM database was described in sections 1.6 and 1.7 earlier.

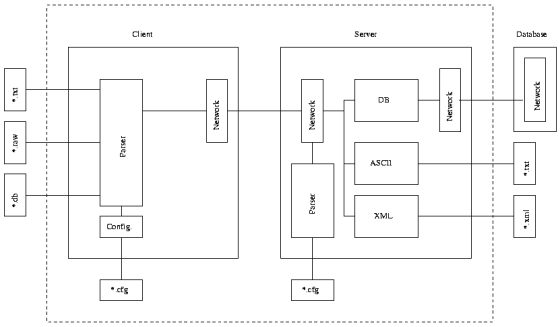

Figure 4: Abstract Client-Server Design

In Figure 4 we can see the abstract design of the client-server application. On the left side we can see the source files for the to be gathered metadata which also includes databases. In the middle left is the client with its configuration files which sends the gathered data to the server. In the middle right is the server program with its configuration which receives the data from the client and stores the data into a database, an ASCII file and an XML file.

2.3 Metadata Capture Tool

The client program, called metac, is a web services client and was implemented using the gSOAP toolkit. It should be executed by the Unified Model User Interface (UMUI) software that creates a basis file and a Jobsheet for each simulation. metac can also be executed at the console, requiring the name of the job as an argument. Parameters for server name, port-number, proxy settings, additional configuration files and so on can be specified in metac.cfg.

metac parses the basis file and, optionally, the Jobsheet and creates a SOAP message containing the gathered data which is then sent to the server (the insertion tool).

2.4 Database Insertion Tool

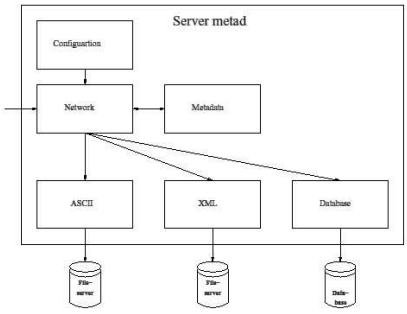

Figure 5: Data Flow of the Server

The server, called metad, is a standalone web service and was also implemented using the gSOAP toolkit. It is configured by a configuration file, .metad, stored in the UMUI user's home directory. It is a multi-threaded server application that allows up to 10 simultaneous client connections. The server receives a SOAP message from the client which contains a MetaData object. The data is then written to the database using an ODBC link and is also written to disk as both XML and ASCII files. A response is sent back to the client which allows the client to evaluate the success of a request.

3. Future Work

To allow easy use of the software the existing UMUI program needs changing to call the metac program after a simulation has been setup. To simplify the parsing and to minimise the number of configuration files used the UMUI configuration file should contain all parameters.

It is also recommend that XML be used as the output format for the basis file so that complex parsing can be avoided.

Further work will be required to integrate the metaUM metadata catalogue into the CCLRC Data Portal architecture which will also be able to search the metaUM database.

The browsing and retrieval interface of the web client will need to be completed.