FINAL REPORT DMQ2 WORK PACKAGE

Ensure effective and reliable connection of selected instruments and sensors to storage repositories (SRB) via CIMA middleware, and efficient use via changed work practices.

Version 1.0, MAY 2007

Lead Investigator: Mathew Wyatt

Prepared by:

Mathew Wyatt mathew.wyatt@jcu.edu.au

Franz Eilert frank.eilert@jcu.edu.au

School of Maths, Physics and Information Technology Faculty of Science, Engineering and Information Technology
James Cook University

Executive Summary

This report presents the findings of research undertaken as a part of the DART (Dataset Acquisition, Accessibility and Annotation e-Research Technologies) project, specifically for Work Package DMQ2, ensuring effective and reliable connection of selected instruments and sensors to storage repositories (SRB) via CIMA middleware, and efficient use via changed work practices. This work package is extremely closely coupled to the DMQ5 Work Package, 'Improve the intelligence of the storage framework' which provided the intelligent storage solution utilising the Kepler workflow tool.

The milestones for this work package have been achieved whilst also creating the solid foundation for development of the Distributed Integrated Multi-Sensor and Instrument Middleware (DIMSIM), planned for the ARCHER project. At the core is the Java Instrument framework developed through the significant effort invested in DMQ2 and through the partnership with University of Sydney and Indiana University.

The achievements with respect to each of the specified milestones are:

- *Scoping of instruments and work case assessment.* - A significant investment was made in scoping the instruments and assessing work cases. The implementation for the Rigaku monitoring, IMB and Monash, was named JCU And Indiana Instrument Service (JAINIS) to try to prevent confusion with other implementations of CIMA. Integral to any deployment was the work conducted by the DMQ5 work package with the Kepler implementation of the Data Manager, see DMQ5 project report.
- *An initial selection of instruments (HF surface ocean radar) and sensors (ocean temperature and weather) CIMA enabled and connected to the data storage via SRB* - The High Frequency (HF) surface ocean radar project is still being developed and subsequently there has been no progress from technical discussions. Proof of concept architecture for an instrument middleware layer within ReefGrid has been developed and the architecture and particularly the Sensor Abstraction Layer (SAL) will be utilised in the development of the next generation of instrument Middleware, DIMSIM. Deployments of CIMA at JCU, IMB, Monash, University of Sydney, and University of Adelaide, and
- *All intended instruments and sensors are connected to the data repositories* - The original intention was to enable three Crystallography instruments, JCU, University of Sydney and University of Adelaide and this has occurred and in fact this milestone has been exceeded with the addition of Monash and IMB. Unfortunately the problems with microwave connection between Davies Reef and Townsville has prevented a live *connection of ReefGrid and associated data.*

The project achieved its milestones and exceeded these except in case where there were dependencies on other work packages or organisations. It is recommended that development of Instrument Middleware, DIMSIM, be continued by the Archer project to bring value to the scientific community.

Table of Contents

1 INTRODUCTION 4

2 PROJECT MILESTONES 5

2.1 AGILE SOFTWARE DEVELOPMENT 5

3 PROJECT OUTCOMES 6

3.1 MILESTONE ACHIEVEMENTS 6

3.1.1 Scoping of instruments and work case assessment. 6

3.1.2 An initial selection of instruments (HF surface ocean radar) and sensors (ocean temperature and weather) CIMA enabled and connected to the data storage via SRB. 6

3.1.3 All intended instruments and sensors are connected to the data repositories. 7

3.2 SITE INSTALLATIONS 7

3.3 HARDWARE CONSTRUCTION 7

3.4 JAVA FRAMEWORK DEVELOPMENT 8

3.5 PLUGIN DEVELOPMENT 8

3.6 PORTAL DEVELOPMENT 9

3.7 ERROR FIXING 9

4 ARCHIVAL STORAGE OF PROJECT DELIVERABLES 11

5 RECOMMENDATIONS 12

6 PUBLICATIONS 13

7 TERMS OF REFERENCE 13

7.1 GLOSSARY 13

8 REPORT SIGNOFF 14

1 Introduction

Instruments and sensors and their accompanying actuators are essential to the conduct of scientific research. In many cases they provide observations in electronic format and can be connected to computer networks with varying degrees of remote interactivity. These devices vary in their architectures and type of data they capture and may generate data at various rates. The Common Instrument Middleware Architecture (CIMA) is a framework for making instruments and sensors network accessible in a standards-based, uniform way, and for interacting remotely with instruments and the data they produce. Some of the issues CIMA address include:

- flexibility in network transport,
- efficient and high throughput data transport,
- the availability (or lack of) computational, storage and networking resources at the instrument or sensor platform,
- evolution of instrument design, and
- reuse of data acquisition and processing codes.

This work package leveraged off CIMA's popularity with the X-Ray Crystallography community in the United States and introduced the framework into Australia. Although the work package was to provide a proof of concept implementation utilising crystallography infrastructure implementing CIMA, demand from the Crystallography community created a requirement to implement working software for a number of organisations.

2 Project Milestones

The following core milestones were defined in the original project specification:

- Determine details of how users expect to interact with selected facilities as regards remote interaction, data handling and overall experiment design and execution.
- Scoping of instruments and work case assessment.
- An initial selection of instruments (HF surface ocean radar) and sensors (ocean temperature and weather) CIMA enabled and connected to the data storage via SRB.
- All intended instruments and sensors are connected to the data repositories.

2.1 Agile Software Development

The initial milestones provided a clear vision of what was required and this was translated into a set of concrete software components for the work package.

- Installation of the Common Instrument Middleware Architecture (C++ implementation provided by Indiana University) at the James Cook University Advanced Analytical Centre.
- Develop a new an improved interface to the Common Instrument Middleware Architecture using JSR168 portlets.
- Create an SRB storage system for the Common Instrument Middleware Architecture.

It was decided that Kepler workflow (DMQ5) would be the appropriate product to provide storage management services, Data Manager, and it was incorporated into the CIMA framework. This fast tracked the storage services to SRB, which drew the attention of the Institute of Molecular Biology, University of Sydney, as well as Monash University.

After an initial period of development and attempted installation at the Institute for Molecular Biology (IMB) on a Rigaku Diffractometer, it became clear that the existing version of CIMA provided by Indiana University was unsatisfactory for deployment, as it contained various bugs, software incompatibilities (with SRB services developed as part of the work package) and hardware incompatibilities (X-Ray Diffractometer types).

A new Java implementation of the CIMA framework was then developed, by reverse engineering the existing C++ code base and forward engineering it to Java. Development was done primarily by the University of Sydney, with assistance from JCU. Other issues occurred with the Kepler workflow which hampered progress.

3 Project Outcomes

The vision of this project was to further develop an open CIMA framework which supports the distribution of instruments and sensors separated in different geographical locations.

3.1 Milestone Achievements

The milestones for this work package have been achieved whilst also creating the solid foundation for development of the Distributed Integrated Multi-Sensor and Instrument Middleware (DIMSIM), planned for the ARCHER project. At the core is the Java Instrument framework developed through the significant effort invested in DMQ2 and through the partnership with University of Sydney and Indiana University.

The current framework has implementations in the X-Ray Crystallography community, specifically relating to Bruker and Rigaku X-Ray Diffractometers. The developed framework also boasts deployment at 5 sites Australia wide.

Along with the framework is an associated Portal developed in JSR168 portlets, which can be deployed independent of the portlet container. The portal supports temp/hum sensors, and a modified kitchen scales. It also supports live video streaming as well as latest diffraction images from X-Ray Diffractometers.

3.1.1 Scoping of instruments and work case assessment.

A significant investment was made in scoping the instruments and assessing work cases. This work is ongoing and was supported by staff from DART at Monash, Institute for Molecular Biology (IMB) staff, JCU staff involved in the ReefGrid project (DMQ1) and University of Sydney. The implementation for the Rigaku monitoring, IMB and Monash, was named JCU And Indiana Instrument Service (JAINIS) to try to prevent confusion with other implementations of CIMA. Integral to any deployment was the work conducted by the DMQ5 work package with the Kepler implementation of the Data Manager, see DMQ5 project report.

3.1.2 An initial selection of instruments (HF surface ocean radar) and sensors (ocean temperature and weather) CIMA enabled and connected to the data storage via SRB.

The High Frequency (HF) surface ocean radar project is still being developed and subsequently there has been no progress from technical discussions. However, there have been the following major outcomes:

- Proof of concept architecture for an instrument middleware layer within ReefGrid. The architecture and particularly the Sensor Abstraction Layer (SAL) will be utilised in the development of the next generation of instrument Middleware, DIMSIM. See DMQ1 project report for more details.
- Deployments of CIMA at JCU, IMB, Monash, University of Sydney, and University of Adelaide, and
- Significant contributions and development of the CIMA codebase.

3.1.3 All intended instruments and sensors are connected to the data repositories.

The original intention was to enable three Crystallography instruments, JCU, University of Sydney and University of Adelaide and this has occurred and in fact this milestone has been exceeded with the addition of Monash and IMB. Unfortunately the problems with microwave connection between Davies Reef and Townsville has prevented a live connection of ReefGrid and associated data.

3.2 Site Installations

The popularity of CIMA concept caused site installations to be implemented at trial sites without a proper deployment process. Also, deployments of the initial system took extended amounts of time. Future deployments will be pre-packaged, based on instrument type, for simple installation. However, the complexity of network VLAN configurations and location of Storage Resource Broker (SRB) instances means each installation will require tailoring.

There are two components for a deployment:

Hardware: The hardware installation involves installing a PC, with a capture card, a Labjack – with temp/humid sensors, scales, and Video camera.

Software: The software installation involves installing, a web server (apache tomcat), the CIMA framework (with site specific configurations), a mounted samba share, the portlet portal, a Linux operating system (with Labjack compiled kernel), the Storage Resource Broker (SRB) and the Kepler workflow.

Overall there are CIMA installations at 5 sites:

- James Cook University,
- University of Sydney,
- Institute of Molecular Biology(IMB),
- Monash University, and
- University of Adelaide.

3.3 Hardware Construction

One use case defined by the X-Ray Crystallography community was the ability to monitor the liquid nitrogen levels left in nitrogen tank. Liquid nitrogen is a core component of an experiment as it is responsible for keeping the crystal sample of an experiment cool, so it does not loose its shape and cause the experiment to fail.

In order to monitor the liquid nitrogen levels in the tank, a circuit was created which was responsible for converting the output of common electrical household scales into a reading which could be interpreted by a Labjack U12. An associated plugin for the CIMA framework was also developed.

3.4 Java Framework Development

The Java framework developed to replace the Indiana University based C++ framework offers the base (core) components of the system. The base components are:

- abstract service implementations,
- abstract handlers for end to end communications (parcel handlers), and
- embedded web server (can run without the embedded webserver).

The original implementation of the framework supported one sequence of data transport and communication, based on a registration concept, i.e. sink registers with source to receive data, the source then pushes data to the sink. There are two problems with the implemented strategy, firstly there are times when a client wishes to do a get on a source, and secondly sinks may not be able to keep up with the rate at which data is being pushed to them. To resolve these two issues, extra communication/transport functionality was added to the new version. Initially the ability to do 'a get' to a CIMA service was added to the framework and then a push/pull hybrid system was developed. The hybrid is where a source notifies a sink when new data is ready to collect, and can do 'a get' on the new data when it is able.

In addition to the CIMA framework, an open Image conversion API was developed collaboratively with the University of Sydney specifically for the use on Diffraction images produced by X-Ray Diffractometers. As there are many different types of Diffractometers, there needed an easy way to convert diffraction patterns in proprietary formats into a common image format such as JPG, or PNG for display over the web. Currently the API supports Rigaku (.osc) and Bruker (.sfrm) image formats.

3.5 Plugin Development

Using the Java framework, we were able then to create instrument specific plugins knowing that a standard communication channel was defined. Plugins were developed collaboratively with University of Sydney and are listed below.

Bruker Frame Source: The Bruker Frame Source plugin is responsible for determining when a Bruker X-Ray Diffractometer has initiated an experiment. When an experiment has been initiated and new data is available, the plugin notifies the system that new data is available and it can be transferred via the standard communication channel.

Rigaku Frame Source: The Rigaku Frame Source plugin is responsible for determining when a Rigaku X-Ray Diffractometer has initiated an experiment. When an experiment has been initiated and new data is available, the plugin notifies the system that new data is available and it can be transferred via the standard communication channel.

Bruker Converted Frame Source: The Bruker Converted Frame Source plugin is an extension of the Bruker plugin. When new images are created by the diffractometer, the plugin converts the proprietary image into a standard image format then transferred via the standard communication channel.

Rigaku Converted Frame Source: The Rigaku Converted Frame Source plugin is an extension of the Rigaku plugin. When new images are created by the diffractometer, the plugin converts the proprietary image into a standard image format then transferred via the standard communication channel.

Temp/Humid Plugin; This plugin is responsible for retrieving the latest readings from a temperature/humidity sensor connected to a Labjack U12.

Scales Plugin: This plugin is responsible for interpreting the readings coming from the converted household scales connected to the labjack (used for monitoring liquid nitrogen levels). The plugin has the ability to set the lower (empty tank) and higher (full tank) limits for the scales.

Simulation Plugins: Various simulation plugins were developed which simulate how the actual plugins are to behave. These simulation plugins were used primarily for testing purposes, to simulate a live system before deployments.

3.6 Portal Development

The original portal developed at Indiana University was written in GS portlets specifically for the Gridsphere portlet container. It was considered that the portal did not adhere to the standard portlet concept, so the portal was re-developed to adhere to portlet concepts by creating JSR168 portlets which are not container specific. The standard base portal has 3 individual portlets defined (multiple instances of these portlets can be used to compose a portal).

Sensors Portlet: This portlet is responsible for displaying the double data readings from the labjack. The portlet can currently displays temperature, humidity, and the weight on the scales (for liquid nitrogen levels).

Converted Image Portlet: Responsible for displaying converted diffraction images (for both Rigaku and Bruker).

Live Video: This portlet projects live video feeds from the cameras in the laboratories.

3.7 Error Fixing

A very major consumption of time for this project was taken up by testing and debugging errors which occurred at deployment time, and could not be reproduced in simulation. Many other major hurdles were faced which hampered development, or produced inconsistent results during testing stages. These are listed below:

Parcel Inconsistencies: Indiana University published a set of standard communication parcels specifically for CIMA. These communication parcels define how data is packaged and interpreted by various nodes in the system. During development of the SRB storage services, many inconsistent results appeared. After a period of time it was determined that the C++ framework was inconsistent with respect to the documented standards. These inconsistencies were initially fixed, before moving on to the Java framework.

Axis, gSoap Incompatibilities: During the initial development of the SRB storage services, errors appeared running gSoap (C++ web service container) and Apache Axis (Java web service container) together. These errors were unexpected as a key component of Web Services is that endpoints can communicate with each other independent of the container, or development language and platform. Once the error was identified, it was immediately corrected, before later moving over to the Java framework.

Axis vs Xfire: During simulation and testing memory issues became a big problem because of the heavy use of XML, Web Services, and transfer of large data (20Mb images). After careful analysis of the Web Service containers, it was noticed that Xfire (Open source Codehaus Web Service container) was more robust and reliable than the Apache Axis Web Service Container.

DOM vs SAX: The CIMA framework implementation makes heavy use of XML and Web Services. Initially data parcels (defined standard of XML for packaging the data) were parsed using the Document Object Model (DOM), where the XML parcel is completely loaded into memory and can be easily interpreted in code. A problem with this system occurred very quickly as DOM used 10:1 memory compared to the original document. For every 20Mb image that was parsed or packaged, the system would leave a 200Mb footprint. This type of memory usage eventually caused the system to fail, as the data would accumulate too fast for it to be sent.

A solution to the problem came from using SAX (Simple API for XML), which allowed us to parse the XML in a serialised manner. Thought the code was a little more complex, it reduced the memory footprint dramatically, providing a more stable system.

4 Archival Storage of Project Deliverables

The software produced by this work package belongs to a group and does not specifically belong to this work package. Although a copy can be zipped up and shipped to Monash University for archival storage with the other software artefacts from the DART Project, the ownership of this software, JAINIS or other improvements to the CIMA codebase are not exclusively owned the DART.

5 Recommendations

The project achieved its milestones and exceeded these except in case where there were dependencies on other work packages or organisations. Over the course of development many lessons were learned and because of the short time frame for development, a large section of the framework is yet to be implemented. Details listed below:

Data Transfer: Currently data movement between nodes in the CIMA framework occurs by wrapping Base64 encoded binary data in XML. Problems with this are that data needs to be encoded/decoded, data in XML causes large memory use. A solution to this problem is to use:

- MTOM (Message Transmission Optimization Mechanism) - http://en.wikipedia.org/wiki/MTOM
- XOP (XML-binary Optimized Packaging) - http://en.wikipedia.org/wiki/XOP

These two technologies allow the developer to attach files to the web service envelopes rather than wrapping the data in XML, which is far more efficient.

Deployment: Currently deployment if the system is a multipart process which is very time consuming. Especially for those who deploy the system and are not closely involved with the development team. Improvement of the deployment process is essential to further uptake of the framework outside of Crystallography.

Security: Currently there are no security mechanisms built into the framework. In scientific domains where many groups utilise the same machinery and work with highly important datasets, it is a requirement to have a security model for the framework.

Registries: Currently, accessing a service for a specific instrument, or a particular storage device is a semi manual process. The process of discovery of services can be delivered to the client through the use of registries, making the system a more dynamic and attractive initiative.

User Interface: Current interface design has been completely restricted to portlets. Moving away from this restriction will allow for development of more dynamic, intuitive and artistic interface designs, with benefit being seen in user satisfaction.

6 Publications

I.M. Atkinson, et al. Common Instrument Middleware Architecture: Extensions for the Australian e-Research Environment, 2nd IEEE International Conference on e-Science and Grid Computing

D.F. McMullen, et al. Toward Standards for Integration of Instruments into Grid Computing Environments, 2nd IEEE International Conference on e-Science and Grid Computing

I.M. Atkinson, et al. Developing CIMA based Remote Access for Collaborative e-Research, 4th Australasian Symposium on Grid Computing and e-Research

7 Terms of Reference

7.1 Glossary

Acronym	Definition
CIMA	Common Instrument Middleware Architecture
DOM	Document Object Model
JAINIS	JCU And Indiana Instrument Service
SRB	Storage Resource Broker

8 Report Signoff

It is agreed between

Franz Eilert and Mathew Wyatt

and

Associate Professor Ian Atkinson

and

Dr Andrew Treloar

That the **Final Report Document** for the DART DMQ2 – 'Ensure effective and reliable connection of selected instruments and sensors to storage repositories (SRB) via CIMA middleware, and efficient use via changed work practices', gives a full account of the work undertaken for the DART Project.

Franz Eilert	0401 710 272
Mathew Wyatt	0401 470 365

- has been read and reviewed by all parties,
- shows that the DART DMQ2 – 'Ensure effective and reliable connection of selected instruments and sensors to storage repositories (SRB) via CIMA middleware, and efficient use via changed work practices' has been completed satisfactorily,
- clearly outlines the deliverables stated in the DMQ2 requirements documentation have been met.

Dated this 25th day of May 2007

Signed	Signed
Chief Investigator	For and on behalf of DART
Associate Professor	Project Director
Ian Atkinson	Andrew Treloar