Developing CIMA-Based Cyberinfrastructure for Remote Access to Scientific Instruments and Collaborative e-Research

Ian M. Atkinson,[1*] Douglas du Boulay,[2] Clinton Chee,[2] Kenneth Chiu,[3] Paul Coddington,[4*] Andrea Gerson,[5] Tristan King,[1] Donald F. McMullen,[6] Romain Quilici,[2] Peter Turner,[2*] Andrew Wendelborn,[4*] Mathew Wyatt,[1] Donglai Zhang[4]

[1]School of Information Technology, James Cook University, Townsville, QLD 4811, Australia.
[2]Department of Chemistry, University of Sydney, Sydney, NSW 2006, Australia.
[3]Computer Science Department, SUNY Binghamton, Binghamton, NY, USA.
[4]School of Computer Science, University of Adelaide, Adelaide, SA 5005, Australia.
[5]ACeSSS, University of South Australia, Mawson Lakes, SA 5095, Australia.
[6]The Pervasive Technology Labs at Indiana University, Bloomington, IN 47404 USA.

Ian.Atkinson@jcu.edu.au, boulay_d@chem.usyd.edu.au,c.chee@chem.usyd.edu.au,
kchiu@cs.binghamton.edu paul.coddington@adelaide.edu.au, Andrea.Gerson@unisa.edu.au,
Tristan.King@jcu.edu.au, mcmullen@indiana.edu, r.quilici@chem.usyd.edu.au,
p.turner@chem.usyd.edu.au, andrew.wendelborn@adelaide.edu.au, Mathew.Wyatt@jcu.edu.au,
donglai.zhang@adelaide.edu.au

Abstract

An infrastructure for remote instrument access, data acquisition and data management is being developed for e-Research. The Common Instrument Middleware Architecture (CIMA) is being used to provide a scalable and extensible basis for the cyberinfrastructure, and X-ray diffraction is targeted as an ideal development domain. Australian research is enhancing the CIMA model to enable federated Grid storage via SRB, and the use of the Kepler workflow system. Kepler has been introduced to enable automated data management, and the facile extraction and generation of instrument and experimental metadata. The system permits real-time deposition of experimental data into an SRB data store using a schema that allows for searching on the basis of metadata or user supplied annotations, image file previewing and data management and download. The CCLRC scientific metadata model is being adopted for metadata definition. In addition to monitoring, CIMA is being further extended to support instrument control, and is being embedded as a component in a feature rich portal for remote instrument access. The architecture supports remote access to multiple instruments from a single portal. The use of Pushlet and AJAX technologies has been introduced for push based portlet refresh and updating. An X3D based 3D virtual representation of the instrument provides data collection simulation and (pseudo) real time instrument representation. A tool for multi-user collaborative image evaluation is being developed for the infrastructure system.

Keywords: Remote Access, Cyberinfrastructure, e-Research, CIMA, GridSphere, Grid services, Web services, Kepler, SRB, Crystallography, Virtual Instrument, Collaboration.

1 Introduction

The development of a reusable software model for the provision of remote access to instruments and other real-time data sources, is an outstanding e-Research challenge. State of the art high performance laboratory instrumentation, such as high flux X-ray diffraction systems and powerful electron microscopes, is increasingly expensive and too costly to replicate in multiple locations. Not only is there the high initial capital cost, there is the on-going burden of technical staffing and specialised maintenance costs. There are obvious financial, functional and educational returns in developing collaborative access services, including instrument control, for remote instrumentation. Likewise, remote access services would maximise returns on the high construction and operating costs of the landmark national research facilities currently under construction; the OPAL research reactor and the Australian Synchrotron.

The Common Instrument Middleware Architecture (CIMA: Bramley et al. 2003, Devadithya et al. 2005, Bramley et al. 2005) project is a NSF Middleware Initiative project seeking a consistent and re-usable middleware framework to enable and embed instruments and sensors as addressable cyberinfrastructure elements. Core goals of the CIMA project are:

1. The development and use of plugins to a Web services CIMA stack to enable code re-use and support applications in different instrument settings.
2. Integration of instruments, computing and storage as Web and Grid services.

---

3. Facilitate discovery and adaptation through the development of instrument descriptions based on the Web Ontology Language (OWL-DL).

A notable features of the CIMA model is that it facilitates access to multiple instruments and sensors from a single portal. The Indiana University project team initiated a collaboration with the Open Grid Computing Environments (OGCE) project to develop GridSphere (JSR 168) portlets providing user interfaces to CIMA instruments. The use of portal-portlet technology has a number of incentives, including the provision of rich functionality and global accessibility together with minimal or no need for any client software other than a suitable Web browser.

Crystallographic structure determination offers a particularly attractive test domain, with well defined work-flows and data structures, and has relatively common (if not standardised) instrument types. Accordingly, portlets specific to the crystallography domain have been developed to monitor laboratory instruments, and follow experiments in progress (CIMA Crystallography Portal).

Herein we describe Australian adaptations and extensions to the CIMA model, driven by Australian e-Research projects and undertaken in collaboration with the US CIMA project team.

2 Workflow, Data Management and Grid Storage Developments

CIMA project collaborators at James Cook University (JCU) have taken advantage of an inherent CIMA capability to provide a Web services interface to other services in a workflow environment. In particular the Kepler (Kepler 2006) workflow system has been introduced to facilitate the automation of data management, and the facile extraction and generation of instrument and experimental metadata (Atkinson et al 2006). Kepler has the attraction of a system that is relatively simple to develop and has been shown to very useful in rapid the development of new or revised workflows.

Kepler provides a visual programming user interface for designing and executing scientific workflows. These workflows define the flow of data between a set of analytical steps and can be used to perform a wide variety of different tasks. Each step in the workflow is represented in Kepler as an *Actor*. Each actor is a compiled piece of java code which defines what the actor does, and what data can be input to and output from the actor. Kepler provides a few simple base actor classes which are extensible and able to create new actors in order to perform non-standard tasks.

The data storage mechanisms, data and metadata associated with crystallography are very specific. For use cases outside of crystallography the data management requirements may well be very different. The problem of data management customisation would normally require an abundance of different data management codes to cater for the broad scientific community. However, by using Kepler workflows the development effort can be greatly reduced. A customised data manager workflow can be composed in days, compared to weeks or months if the code was to be written in Java or C. Kepler also makes tuning a workflow much easier than a full project recompile, or project branch. Another advantage Kepler offers is the ability to import and export workflows. Workflows in Kepler can be exported and can then be deployed to other instances of Kepler at different sites. Another possibility for Kepler data management is collaborative workflow creation (via VNC), and sharing of workflows amongst research communities.

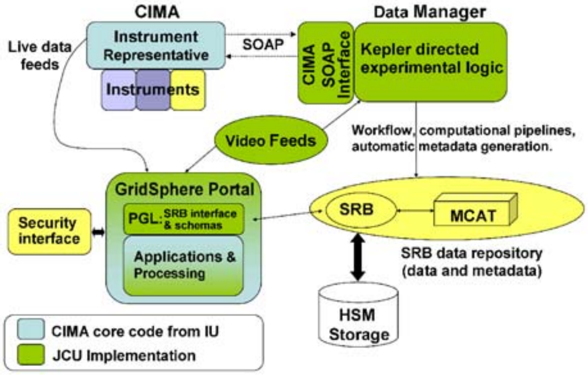


Figure 1. JCU Adaptations and extensions to the CIMA model.

For the implementation of CIMA in Australia, Kepler has been introduced as a service deployed with a specific workflow. To date, the Web service architecture is not built into Kepler, so an intermediate Web service container (Apache Axis) has been used. This intermediate contains a deployed Web service that acts as a sink, for data sent after registration by a CIMA Instrument Representative registry. As the CIMA parcels arrive at the sink they are packaged into a java bean and forwarded to the Kepler service. Ultimately Web service actors may be built into Kepler.

Kepler can now play a key composition and mediating role in work flow and data management for CIMA based instrumentation. In particular, Kepler can be used to quickly compose the integration and use of a Storage Resource Broker (SRB 2006) managed data store (see Fig 1). The integration between Kepler and SRB is now well developed and robust, and the JCU group have introduced the use of SRB and the MCAT as an alternative to the NFS and MySQL data manager originally provided in the CIMA software. The system permits experimental data to be deposited into the SRB data store in real time and provides data management, replication and download. SRB is robust, scalable and file system independent.

The introduction of SRB storage provides a foundation from which to build a network of Grid enabled instruments that will facilitate data fusion and analysis across instruments, disciplines, and facilities. To this end, a test-bed network is being built that encompasses instruments at James Cook University, Monash University, the University of Queensland and the University of Sydney (Fig. 2). It is envisaged that the network may ultimately be integrated into a National Grid resource such as that provided by the Australian Partnership for Advanced Computing (APAC).

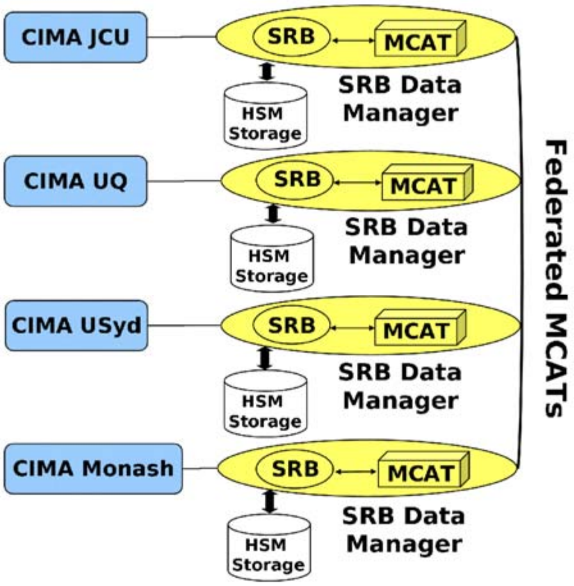


Figure 2. Proposed instrument network linked by federated SRB storage.

JCU portal enhancements and new portlets include the use of the JCU Personal Grid Library (PGL 2006) system for user friendly SRB data access; image file previewing; the ability to create metadata schema definitions for experimental metadata and to extract metadata in real-time from instrument output using Kepler; annotation, search and retrieval of data sets; searching on the basis of metadata or user supplied annotations. Described below in part 4 are portal and portlet developments at the University of Sydney that are further contributing to the Australian enhancement and extension of the CIMA model.

3 Real-Time Data Storage and Processing

Complementing the work at James Cook University and the University of Sydney, researchers at the University of South Australia and the University of Adelaide are investigating the provision of real-time metadata capture and remote data storage using SRB, and near-real-time processing of synchrotron data using high-performance computational resources available via a grid interface. The initial experiments being targeted are microdiffraction fluorescence probes, which will be installed at the Canadian Light Source (CLS) and the Australian Synchrotron. Our investigations are currently using UNI/XOR, a similar beamline at the Advanced Photon Source (APS) in the US. Related to this, CIMA software has been installed (CIMA 2006) at the ChemMatCARS beamline of the APS and this affords further opportunity for exploring the practicalities of real-time downloads. The Australian Synchrotron Research Program, together with the University of Sydney, currently operates the Australian Specialist Crystallography at the Advanced Photon Sources program (SCrAPS 2006) using the ChemMatCARS beamline at the APS.

Current practice for microdiffraction-fluorescence experiments is for researchers to analyse their experimental data after the experiment is completed, and this may take many days of processing time on a PC. The data and metadata are typically stored in an ad-hoc fashion, and not readily available to collaborators and other researchers for whom it may be useful. For the initial beamline experiments of interest, we are designing and developing a system to provide researchers with remote access to the experiment, real-time data download, storage and archiving of data and metadata. A capability for near-real-time data analysis and visualization would allow an experimental procedure to be changed, or tuned, based on a preliminary analysis of the data while measurements are in progress.

Of particular relevance to our research, is work being undertaken in the UK for the CCLRC Data Portal Project (CCLRC Data Portal 2006, Drinkwater et al. 2004), which is developing leading edge tools that are shaping emerging e-Research environments. The project is developing portal services for data access, utilisation and management across major UK research institutions such as the Diamond synchrotron and ISIS neutron facility. The CCLRC Scientific Metadata Model (CSMD: Sufi & Mathews 2004) is designed to provide a general model of metadata for scientific studies, and is being adopted for metadata catalogues by major UK research facilities (Dove & de Leeow 2005, Stevens et al. 2004). The model is not yet comprehensive, but seems to offer an excellent base upon which to build, and from which to interoperate with other resources.

The CSMD is hierarchical, following a pattern of investigation common at instrument facilities. At the top level is a study (or project), which comprises a number of investigations, such as experiments or simulations. These have raw data associated with them; the metadata catalogue will provide links to such data, for example, as SRB identifiers. This is achieved by modelling data storage as a collection of Atomic Data Objects, where an ADO contains a locator, in a format specific to an implementation, specifying the physical location of data.

CSMD holds information about a study, such as topic keywords, provenance information, material related to a study or investigation, experiment parameters, and so on. A Web services based data portal has been developed at CCLRC, operating with the ICAT, a relational database storing metadata in CSMD format. The ICAT is used at the ISIS facility in the UK and is to be used at the Diamond synchrotron, currently under construction in the UK, and the Spallation Neutron Source, which is nearing completion in the US. The ICAT is to be installed at the X-ray crystallography facilities of James Cook University and the University of Sydney. It may also be used at OPAL and the Australian Synchrotron.

The UNI/XOR beamline generates CCD image files in Princeton Instruments proprietary SPE format, which uses a basic 16-bit image representation with a header containing metadata. A mapping from the metadata in the SPE files to CSMD has been developed as part of the work being reported here, so that the metadata for each data file can be captured as the data is generated, and stored into an ICAT database in CSMD format. Much of the required metadata is not available in the SPE file, for example information about the UNI/XOR facility, the experimenter, the research project, and the sample. This information must be generated manually, although much of it can be re-used. The software that generates the metadata can merge this information with the automatically generated metadata information contained in the SPE file.

Experiments have been undertaken to check the feasibility of real-time data download and storage (in an SRB repository) from the APS in Chicago to the University of Adelaide. Depending on the type of experiment, the size of the image data files can range from around 1 megabyte to over 8 megabytes, and the time to generate each image can range from around 2 to 25 seconds. Typically the data generation rate is around 1 MByte/sec. Our experiments have shown that it is feasible to sustain this performance between APS and Adelaide (without any TCP tuning), but only if multiple TCP streams are used. We have used up to 10 concurrent file transfers using scp (the default maximum number of connections). The bandwidth obtained is highly variable, particularly for different times of the day, with values for a single scp stream ranging from 30 to 130 kbytes/sec being observed. The data download program developed for this work allows for lossy compression of the CCD images, by throwing away the required number of least significant bits of the image, if the downloads cannot keep up with the rate of experimental data generation. This allows for near-real-time data analysis to be done, as long as the number of discarded bits is not too large (for example, the two least significant bits appear to be indistinguishable from random noise). If lossy files are used, the complete data files can be downloaded and archived after the experiment is finished.

Some of the data analysis is very compute intensive, and it can take days or weeks on a PC to analyse the thousands of image files obtained from an experiment. Accordingly, work is being undertaken to modify the data analysis program so that the most computationally intensive parts can be run on grid compute resources. Some parts of the analysis are embarrassingly parallel, in that they can be done independently on each image file. Other aspects of the reconstruction are best done by parallelizing across rows of the images, so that each processor has the same set of rows for all images.

The analysis program also assumes that the input and output data files are stored on local disk. A further part of our work is to enable the experimental data files to be accessed from SRB, and the output of the data analysis to be stored back into SRB, with metadata (in CSMD) that associates the analysis data with the appropriate experimental data.

The software components being developed by the South Australian team will be integrated with CIMA-Kepler data manager workflow system being developed at JCU.

4 Remote Control and Collaboration

The remote desktop approach to remote instrument access, such as typified by the use of Virtual Network Computing (VNC 2006) and its many variants, CITRIX (CITRIX 2006), Sun Secure Global Desktop (SSGD 2006) and NX NoMachine (NX 2006), has the significant advantages of ease of set-up and familiarity. While convenient, these approaches are not ideal and can afford remote instrument users with excessive control over expensive and potentially dangerous instruments. A significant disincentive to building custom-built remote access systems, is that there is a high coding overhead that may well reproduce functionality already provided by an instrument manufacturer. A significant advantage of the custom built interface approach however, is that the actions of the remote instrument user can be tightly controlled, while at the same time services outside the desktop environment can be provided to offer a richer operating environment.

To this end, the University of Sydney group is developing a CIMA based GridSphere portal/portlet system for remote instrument control, monitoring and data processing (Atkinson et al. 2006). Thus far CIMA has been developed solely for remote instrument and sensor monitoring, and accordingly extensions to the architecture have been required to enable CIMA mediated instrument control. The need for real-time updating of instrument status and data displays for effective and safe instrument control has driven the introduction of Pushlet (Pushlets 2006) and AJAX (AJAX 2006) technologies to enable (pseudo) real-time data push from the instrument to the client. The utilisation of these technologies has in turn improved the functionality of the monitoring portlets.

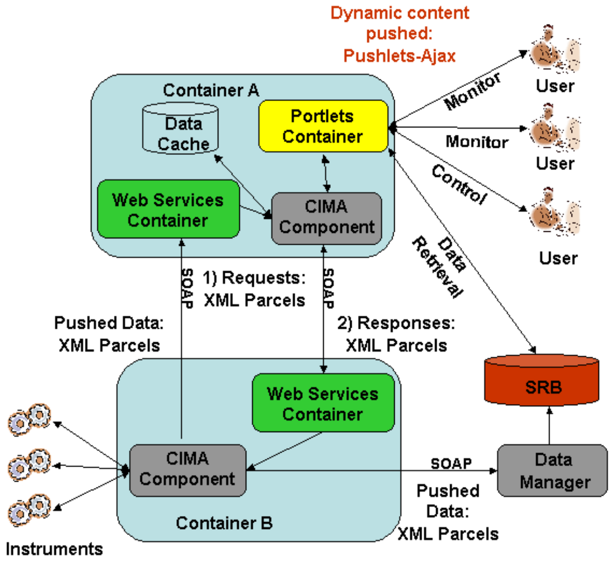


Figure 3. Remote instrument control portal structure. Only one person at a time can control any given instrument.

A modular service oriented architecture (SOA) model using Web services has been adopted to provide maximum flexibility, with Web services offering language and location independence. There are two primary containers; one located at the instrument site for instrument services and a second for user access interface components that need not be co-located with the instrument (see Fig 3). For instance, several institutions under a common project may want to provide remote access to a shared instrument or set of instruments. In that case a user interface portlet container may be beneficially located at each of the user institutions. Alternatively, container A could be shared between multiple institutions to provide remote access to multiple shared instruments (not necessarily located at one site). The second model may be desirable for large facilities such as a synchrotron, for which a single Container B could serve multiple instruments or beamlines. The architecture is flexible and other combinations are possible.

As Fig 3 suggests, CIMA service endpoints are integrated into Web services containers (currently Axis) located in servlet containers (currently Tomcat). Container A holds the user interface elements, including a portlets container, a Data Cache, a Web services container receiving SOAP calls from container B, and a CIMA component used to send XML parcels via SOAP to container B. Container B contains elements to interact with the instrument (or instruments). In particular, a Web services container to receive SOAP messages containing XML parcels from container A, and a CIMA component to interact with the instruments as well as to send XML parcels to container A. Container B delivers data and metadata to the data manager developed at James Cook University.

There are two primary modes of communication between the two containers. One mode involves a request/response pair in which a request is sent from container A to container B, and for which a response is then expected. The response may simply be an acknowledgement, or the result of a command. The CIMA component in Container B extracts data from XML parcels included in the request from A, builds instrument-specific commands and finally returns a response as an XML parcel. Requests are of two kinds: those that affect the instrument state (SET request; eg for instrument control) and those that retrieve instruments information (GET requests; eg for instrument status information).

The second type of interaction occurs when an instrument pushes information. The push may be the result of an earlier asynchronous request, state changes in the instrument, or because the instrument otherwise sends data on a regular basis. In this case the CIMA component on B (CIMA source) sends an XML parcel via SOAP to the CIMA component of container A (CIMA sink). The parcel content is then extracted and relevant data is transferred into a temporary store, or Data Cache, and the portlet content is updated.

The data is sent (pushed) from the instrument into Container B, where the ‘raw’ X-ray detector data (CCD frames) are parcelled for transfer out to a CIMA data manager and into a storage resource. Container B also provides a service to convert proprietary CCD data image files into smaller file size JPEG images for rapid transfer to Container A and portlet display.

The University of Sydney group has introduced the use of Pushlets, used in conjunction with AJAX, for portlet refresh and dynamic updating of the portlet content. The Pushlet based server push mechanism provides (pseudo) real-time status and data display on the client browser. It is then possible for instance to view CCD data frame images along with metadata essentially as they are generated by the instrument.

The Data Cache contains status information about the instrument as well as temporary files (such as data frame images for portlet display and review), and is populated when data pushed from the instrument arrives, or when a response containing instrument information is received. The Cache is used to minimise SOAP calls to Container B, when a GET request is issued and the desired data is already available in the cache. The cache has the same lifecycle as Container A, and might not be applicable for all instruments.

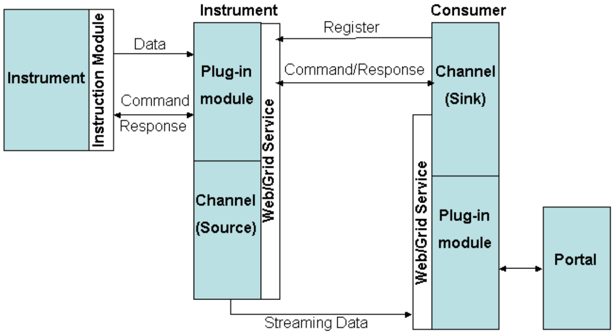


Figure 4. CIMA channels and plug-in modules including the new Instrument Instruction Module.

The development of Instrument control services has been undertaken in accord with the CIMA channels and plug-in model[1] (Fig. 4). As part of this Australian initiative, new parcel types were required to support the command/response mode. Moreover an Instrument Instruction Module has been introduced and serves as an instruction interface to proprietary instrument software (or device drivers). The instructions may be to get instrument status information or to change the state of the instrument (control the instrument). The module translates CIMA parcels into instrument specific instructions to be sent to the instrument interface. Work is currently underway to generalise this initiative.

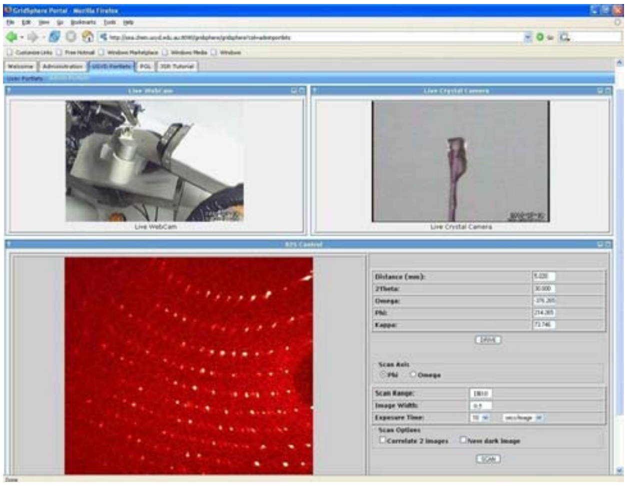
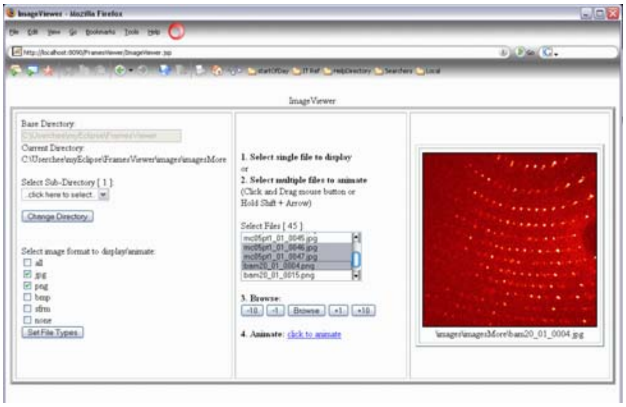


Figure 5. GridSphere portlet for X-ray diffraction instrument control and data view.

The current form of the GridSphere instrument access interface provides an instrument control pane (Fig. 5) allowing the user to define and initiate simple data collections. Options for the provision of more complex data collections are currently being evaluated. The pane provides for data collection parameter input, web cam monitoring of the instrument and crystal sample, and a display of the current data image. As mentioned, the images are dynamically updated though the use of Pushlet technology.

A second tabbed pane augments the browser interface with instrument status information and, for example, displays the X-ray generator voltage and current settings, the cooling water temperature, CCD camera temperature and laboratory temperature and humidity.


ImageViewer - Mozilla Firefox

http://localhost:8090/FramesViewer/ImageViewer.jsp

ImageViewer

Base Directory

Current Directory
C:\Users\hee\myEclipse\FramesViewer\images\imagesMore

Select Sub-Directory [1]

Select image format to display/animate:
☐ all
☑ jpg
☑ png
☐ bmp
☐ sfrm
☐ none

1. Select single file to display
or
2. Select multiple files to animate
(Click and Drag mouse button or Hold Shift + Arrow)

Select Files [45]
mc05prt1_01_0045.jpg
mc05prt1_01_0046.jpg
mc05prt1_01_0047.jpg
bam20_01_0004.png
bam20_01_0015.png

3. Browse:

4. Animate: click to animate

\images\imagesMore\bam20_01_0004.jpg

Figure 6. Portlet for diffraction image browsing and scanning. A selected set of mages may be scanned at a user determined animation rate.

Work is in progress to integrate a tabbed pane for diffraction image inspection (see Fig 6). An individual image may be selected for display in the portlet, or a range of images may be selected and viewed at a user selected display rate (image set animation). It is intended that further image analysis capability will be added to this portlet. The images may be those held in the Data Cache or images located in the storage resource.

While multiple users may join a remote access session through a browser, only one user can control the instrument, though control may be transferred to another user. The system allows collaborators to jointly monitor an experiment and collectively determine the best approach to data collection and processing.

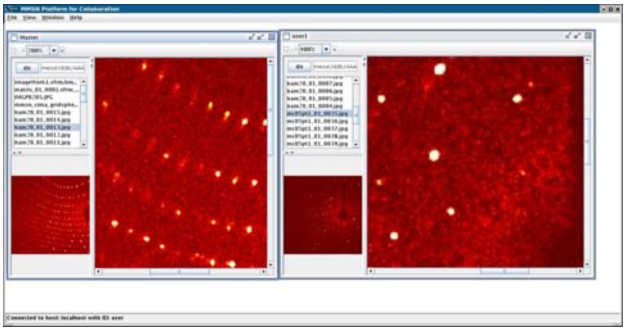

Connected to host: localhost with ID: user

Figure 7. PfC, a Platform for Collaboration allowing multiple users to independently select and collaboratively interact with diffraction images.

To further extend the collaborative infrastructure, a multi-client Java RMI application is being developed for distributed assessment of X-ray diffraction images stored on a remote host. Nominally one remote participant acts as a "master" user, guiding or supervising the remote file system access of other participants, but all of whom are able to independently browse images, subject to their access permissions. The status of each client is then broadcast to all participants who are free to eavesdrop or interact with their colleagues (see Fig. 7).

In this manner images may be selected independently by the collaborators and viewed ‘side by side’ for feature comparison, and independently manipulated by all participants. The viewer has an image zoom function and future aims include annotation capabilities and image content analysis. Currently the embryonic tool operates as a stand alone application for use over a VPN, and we are exploring the possibility of incorporating its functionality into a portlet.

A strong disincentive to providing remote client control of a physical device is that unskilled operators, or simple data entry errors, may lead to costly damage to the instrument. Ideally the instrument control software installed at the remote site would include a collision map to prevent accidental damage. In practice however collision map software is not always provided, and when such software is available it may have weakness or bugs. As suggested above, the risk of collision damage can be reduced by limiting the functionality of the remote instrument access interface. The risk of damage can also be mitigated through the use of an instrument simulator or virtual instrument.

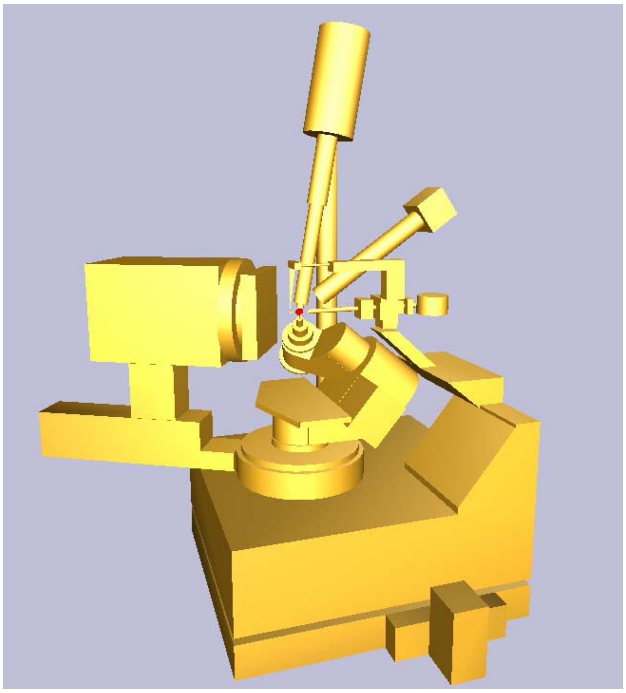
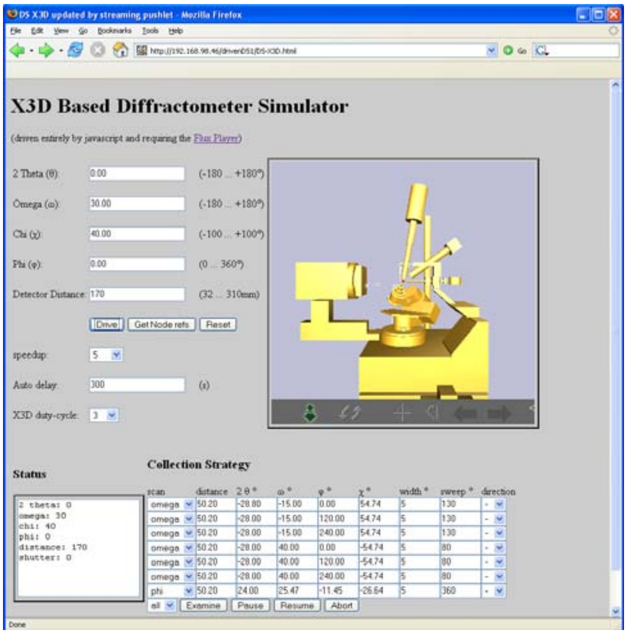


Figure 8. A DS and X3D based virtual X-ray diffraction instrument. Components displayed include the goniometer, detector, video microscope (partial), cryosystem (partial) and X-ray collimator. The X-ray source is not shown.

Our approach has been to adapt the DS diffractometer simulator package (Zheng, Yao & Tanaka 1995) for portlet incorporation of a virtual instrument, and for data collection strategy evaluation. As part of this endeavour, the DS package has been ported from its C based OpenGL implementation under HP Unix, to Java using the JOGL (JOGL 2006) native interface to OpenGL. A disadvantage in the direct use of OpenGL in an applet or application, is the need to hard code different machine specifications into the source, and thence its compilation and distribution. Accordingly we're now using the open standard X3D (X3D 2006) format for the virtual instrument representation (Fig. 8), and as a means for developing a library of virtual device specifications that can be easily customised and distributed on a per-instrument basis.

The X3D XML files are externalised from the source, easily editable and can be examined by an increasing number of renderers and web-browser plugins. In addition, X3D can be externally scripted via JavaScript, allowing for inclusion directly within XHTML pages and thereby providing a virtual representation within a portlet. Currently we use FluxPlayer (FluxPlayer 2006) as browser plug-in to display the virtual instrument (see Fig. 9). Accordingly the user can 'zoom in' on the virtual instrument and adopt any viewing position around the instrument, including preset positions.



X3D Based Diffractometer Simulator

(driven entirely by javascript and requiring the Flux Player)

2 Theta (θ): 0.00 (-180 ... +180°)
Omega (ω): 30.00 (-180 ... +180°)
Chi (χ): 40.00 (-100 ... +100°)
Phi (φ): 0.00 (0 ... 360°)
Detector Distance: 170 (32 ... 310mm)
speedup: 5
Auto delay: 300 (s)
X3D duty-cycle: 3

Status

```
2 theta: 0
omega: 30
chi: 40
phi: 0
distance: 170
shutter: 0
```

Collection Strategy

scan	distance	2 θ °	ω °	φ °	χ °	width °	sweep °	direction
omega	50.20	-28.80	-15.00	0.00	54.74	5	130	-
omega	50.20	-28.00	-15.00	120.00	54.74	5	130	-
omega	50.20	-28.00	-15.00	240.00	54.74	5	130	-
omega	50.20	-28.00	40.00	0.00	-54.74	5	80	-
omega	50.20	-28.00	40.00	120.00	-54.74	5	80	-
omega	50.20	-28.00	40.00	240.00	-54.74	5	80	-
phi	50.20	24.00	25.47	-11.45	-26.64	5	360	-

all

Figure 9. Portlet to operate a X3D and DS based diffraction instrument simulator, tailored for an instrument in use at the University of Sydney.

As a solution to the 'dark lab' problem, our portlet deployment of the DS and X3D based virtual instrument utilises Pushlets to provide (pseudo) real-time monitoring of the remote instrument. Bandwidth requirements are minimal, as only a small number of status parameters are needed from the remote instrument for the portlet representation. The instrument state can continue to be represented and assessed should the video feed fail, or should the remote laboratory lights be turned off.

The X3D standard is evolving beyond graphical representation, texturing and world navigation into full featured 3D object simulation that includes rigid body physics properties, collaborative interaction and, in this context, the particularly attractive benefit of collision detection. The use of X3D means that a diffractometer specification for a portlet simulation can be easily re-used to simulate and evaluate various data collection strategies within a separate, stand alone, server side, or client side application.

A further utilisation of the virtual instrument is as a test-bed for developing instrument control plug-ins for CIMA. The simulator may also serve as an indestructible training tool.

5 Conclusion

In collaboration with the CIMA project team, the capability of the CIMA model is being significantly extended. The core CIMA software is being developed and integrated into a comprehensive remote instrument control and monitoring system that is Grid enabled and made efficient through the use of workflow tools. The system provides for collaborative remote control and monitoring of instruments, and support for collaborative interaction is being enhanced. The architecture supports remote access to multiple instruments from a single portal. Pushlet and AJAX technologies have been utilised for responsive and safe instrument control. These technologies further enhance the utility of the monitoring portlets. Remote instrument control is facilitated with an X3D based virtual instrument representation. The CCLRC Scientific Metadata Model is being used for metadata definition and management. Data flow is directed into user friendly Grid storage infrastructure (SRB) via the Kepler workflow tool. As there may well be circumstances where immediate data transfer from the remote site to the local is a desired (or essential) alternative to remote processing and management, the feasibility of (pseudo) real-time data transfer has been established.

Future work will continue to investigate the use of Grid resources for local and remote data processing, and will explore opportunities for workflow automation. The capabilities of the simulator are to enhanced to provide sample specific simulations of the diffraction process, and the collaborations tool is to be incorporated into the portal system. Also currently being considered are options for voice communication for collaboration.

A significant challenge yet to be addressed is that of security. The JCU group has funding support from the MAMS (MAMS 2006) project to investigate the application of Shibboleth (Shibboleth 2006) as a basis for the provision of single-sign-on secure access to remote instrumentation. A robust yet easy to use security system is a critcial requirement for a remote resource access service.

Acknowledgements.

The support of the Australian Department of Education, Science and Training (DEST), for the DART project (dart.edu.au), the Australian Research Council (ARC), GrangeNet and the ARC Molecular and Materials Structure Network (mmsn.net.au) is gratefully acknowledged. CIMA is supported by National Science Foundation cooperative agreements and grants SCI

0330568 and MRI CDA-0116050. The assistance of Kia L Huffman and Tharaka Devadithya is gratefully acknowledged. Dr Min Yao is thanked for generously providing the DS source code, and the assistance of Dr Darren Spruce is also gratefully acknowledged.

References.

AJAX: Asynchronous JavaScript Technology and XML : http://java.sun.com/developer/technicalArticles/J2EE/AJAX/. Accessed 10 Oct 2006.

Atkinson I.A., du Boulay D., Chee C., Chiu K., Huffman K.L., McMullen D.F., Quilici R., Turner P., Wyatt M. (2006): Common Instrument Middleware Architecture: Extensions for the Australian e-Research Environment. *Proceedings of the UK e-Science All Hands Meeting September 18-21 2006*; **http://www.allhands.org.uk/2006/proceedings/**. Accessed 10 Oct 2006.

Bramley R., Chiu K., Huffman J.C., Huffman K.L., and McMullen, D. (2003): Instruments and Sensors as Network Services: Making Instruments First Class Members of the Grid. Indiana University Computer Science Department Technical Report 588, December 2003.

Bramley R., Chiu K., Devadithya T., Gupta N., Hart C., Huffman J.C., Huffman K.L., Ma Y., McMullen D.F. (2006): Instrument Monitoring, Data Sharing and Archiving Using Common Instrument Middleware Architecture (CIMA). *Journal of Chemical Information and Modeling*, **46**(3):1017-25, 2006.

CCLRC Data Portal: http://www.e-science.clrc.ac.uk/web/projects/dataportal. Accessed 10 Oct 2006.

CIMA Crystallography Portal: iri164.informatics.indiana.edu:8080/gridsphere/gridsphere. Accessed 10 Oct 2006.

CITRIX: www.citrix.com. Accessed 10 Oct 2006.

Devadithya I., Chiu K., Huffman K.L., McMullen D.F. (2005): The Common Instrument Middleware Architecture: Overview of Goals and Implementation. *Proceedings of the First IEEE International Conference on e-Science and Grid Computing (e-Science 2005)*, Melbourne, Australia, Dec. 5-8, 2005: 578-585, IEEE Computer Society.

Dove M.T., and de Leeuw N.H. (2005): Grid Computing and Molecular Simulations: The Vision of the eMinerals project. *Molecular Simulation*, **31**(5):297-301.

Drinkwater G., Sufi S., Manandhar A., Blanshard L., Kleese van Dam K. (2004): Data Management with the CCLRC Data Portal. *Proc. Int. Conf. on Parallel and Distributed Processing Techniques and Applications*. Las Vegas, June 21-24 2004. http://epubs.cclrc.ac.uk/work-details?w=30294. Accessed 10 Oct 2006.

Flux Player: www.mediamachines.com. Accessed 10 Oct 2006.

JOGL: https://jogl.dev.java.net/. Accessed 10 Oct 2006.

Kepler: www.keplerproject.org. Accessed 10 Oct 2006.

MAMS: www.melcoe.mq.edu.au/projects/MAMS. Accessed 10 Oct 2006.

NX NoMachine; www.nomachine.com. Accessed 10 Oct 2006.

OGCE: Open Grid Computing Environments: www.collab-ogce.org/ogce2/. Accessed 10 Oct 2006.

PGL: Personal Grid Library Project, VeRG Lab, JCU: https://plone.jcu.edu.au/hpc/staff/projects/hpc-software/personal-grid-library. Accessed 10 Oct 2006.

Pushlets: www.pushlets.com. Accessed 10 Oct 2006.

SRB: Storage Resource Broker: www.sdsc.edu/srb. Accessed 10 Oct 2006.

SCrAPS: Specialist Crystallography at the Advanced Photon Sources, www.ansto.gov.au/natfac/scraps.html. Accessed 10 Oct 2006.

Shibboleth: http://shibboleth.internet2.edu. Accessed 10 Oct 2006.

SSGD: Sun Secure Global Desktop: http://www.sun.com/software/products/sgd. Accessed 10 Oct 2006. Formerly Taeantella.

Stevens R.D., Tipney H.J., Wroe C.J., Oinn T. M., Senger M., Lord P.W., Goble C.A., Brass A. and Tassabehji M. (2004): Exploring Williams-Beuren Syndrome Using myGrid. *Bioinformatics*, **20** (1):i303-i310.

Sufi S., Mathews B.: (2004): CCLRC Scientific Metadata Model: Version 2, *CCLRC Technical Report DL-TR-2004-001*, e-Science Centre, CCLRC, Daresbury Laboratory, 2004. http://epubs.cclrc.ac.uk/work-details?w=30324. Accessed 10 Oct 2006.

VNC: Virtual Network Computing ; http://www.realvnc.com/. Accessed 10 Oct 2006. Variants include TightVNC, RealVNC, UltraVNC, and TridiaVNC.

X3D: http://www.web3d.org/about/overview/. Accessed 10 Oct 2006.

Zheng, C., Yao M., and I. Tanaka. (1995): DS - a 3D graphics simulation program for single-crystal diffractometer. *.J. Appl. Cryst.*, **28**:225-227.