

VIRA: A CIMA Based System for the Operation of Remote Scientific Instruments

Ian Atkinson², Douglas J. du Boulay¹, Sandor Brockhauser⁴, Donald F. McMullen³
Romain Quilici¹, Andrew Sharpe² and Peter Turner¹

¹ School of Chemistry, University of Sydney
Sydney, NSW 2006, Australia
e-mail: p.turner@chem.usyd.edu.au

² School of Information Technology & High Performance Computing, James Cook University
Townsville, Qld 4814, Australia
ian.atkinson@jcu.edu.au

³ The Pervasive Technology Labs at Indiana University
Bloomington, IN 47404 USA
mcmullen@indiana.edu

⁴ Instrumentation Group, European Molecular Biology Laboratory
Grenoble, France
brockhauser@embl.fr

(Received: 27 November 2008; published online: 25 March 2009)

Abstract: the common Instrument Middleware Architecture (CIMA) has been adopted and is being further developed as part of a project that includes the use of virtual instruments in a Web browser driven system for remote access to scientific instruments. Enhancements include distinct separation of concern for the modular components that make up the system, and a flexible message parcel schema.

Key words: remote access, CIMA, virtual instrument, Web services, XML parcels

I. INTRODUCTION

The financial, functional and educational returns from investing in the development of remote access services are well known and widely recognised. 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. Remote access services would maximise returns on the high construction and operating costs of landmark national research facilities, such as synchrotrons and neutron sources, as well as more conventional laboratory facilities.

A significant disincentive to developing ‘custom-built’ remote access systems, is that there is a high coding overhead that may well reproduce functionality already provided by an instrument manufacturer. Recognition of this disincentive is reflected, at least in part, in the cur-

rent preference for the use of the remote desktop [1-4] approach to providing remote access to instruments at major facilities [5].

A significant advantage of the custom built interface approach however, is that the actions of the remote instrument user may be tightly controlled, while at the same time services outside the desktop environment can be provided to offer a richer operating environment. Communication between the user and the facility may be adjusted to suit prevailing bandwidth constraints.

Labour invested in constructing a remote access system may be subsequently leveraged if a framework approach is adopted, and used to support modules and plug-ins. The Common Instrument Middleware Architecture [6-9] (CIMA) project is a pioneering NSF Middleware Initiative project to research and define a consistent and re-usable middleware framework to enable and embed instruments as addressable Web and Grid resources. The framework provides a capability for reuse, and plugins provide service specificity.

We have accordingly adopted the CIMA model as a basis for developing a flexible remote access system enabling an instrument as a Web resource. In doing so we are further developing the architecture, and providing additional remote access services [10-12]. Here we report our latest work on developing a CIMA based remote access system. The project makes significant use of virtual models of the target remote instrument and is accordingly named VIRA: Virtual Instruments and Remote Access.

II. ARCHITECTURE DEVELOPMENT

In adopting and developing the CIMA model, we have more clearly provided for *separation of concern* for the system components, and revised the schema used for construction of the XML message parcels that underpin the system.

Fundamentally CIMA is a two container portal-portlet system, with one container representing the remote instrument and the other representing the client researcher. Communication between the containers, including instrument state and data transfer, is provided through the exchange of XML parcels.

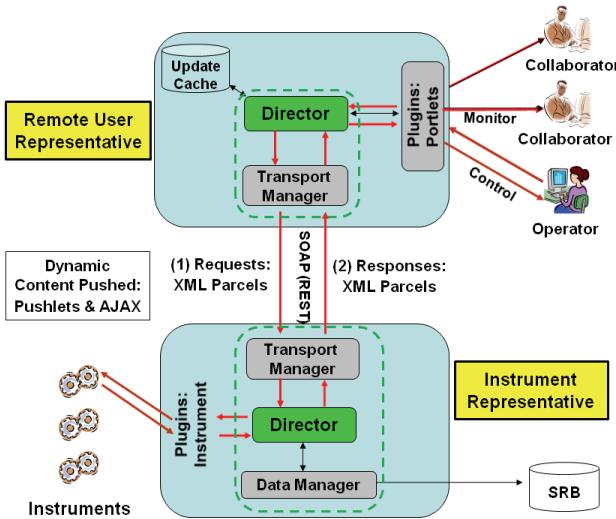


Fig. 1. Depiction of CIMA based remote access system

Processes in both the instrument and client representatives are mediated and coordinated by a central component known as a Director. All parcels entering or leaving a container pass through the Director, which in turn redirects the parcel to the appropriate container component, or directly to a plugin depending on the nature of the parcel. Figure 1 shows two manger components in each container; the dotted line boundary is to suggest the possible presence

of other components involved in the remote access service. Such components are referred to as framework components. Reflecting the embodiment of the principle of separation of concern, the nature of the components called into service, depends on the nature of the remote access task. Components are classed as either controllers or managers, with controllers effectively being interfaces allowing the separation of logic required for handling parcels from the associated process or action. Managers are then process containers. Currently components include a plugin controller/manager pair, session manager, subscription controller, transport manager and description controller. For instance, the plugin controller deals with Start/Stop plugin-typed parcels, and the subscription controller deals with subscribe/unsubscribe parcels. The transport manager (Fig. 2) is to provide support the use of alternative transport mechanisms. Currently Web services via SOAP is implemented, with support for REST communication under development.

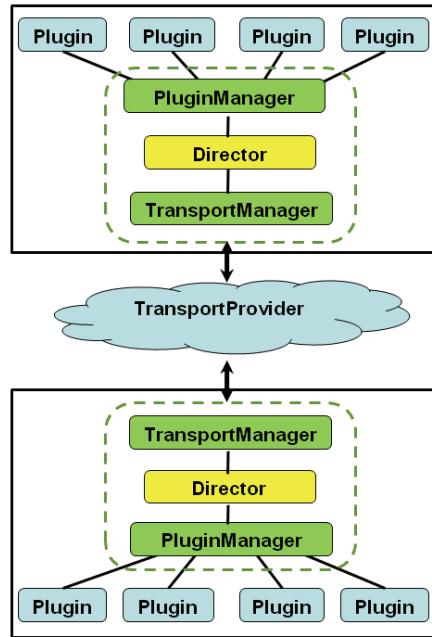


Fig. 2. The CIMA based framework facilitates plugin to plugin communication via schema based XML parcels

The XML message parcels are constructed with respect to a flexible schema. In essence there are two parts to a parcel; a base component and a body component. The base component contains information common to all parcel types, such as timestamp, sequence number, sender/recipient or session identifier. In contrast the parcel body is defined as abstract and empty by default. A parcel instance must then provide an extension of the body appropriate to its intended task. Base level elements are handled by the

appropriate framework components, with the separated body element then forwarded to the target plugin. The provision for abstraction of the parcel core then allows parcel body schema to be defined according to instrument and function type. Plugins must be developed accordingly. The structure of the plugins is such that a ‘handler’ interface can be used to accommodate new parcel content (e.g. PluginOperationExtension). The handler automatically extracts the relevant parameters and calls a plugin method to execute the command. The following is a section of a parcel used to drive motors on an X-ray diffractometer:

```
<body xsi:type="commandOperationType"> --> body
type extension for the command
<CommandOperation>
  <CommandName>Drive</CommandName>
  <parameter>
    <name>2Theta</name>
    <value>0.0</value>
  </parameter>
  <parameter>
    <name>Omega</name>
    <value>10.0</value>
  </parameter>
  same with other required params
</CommandOperation>
</body>
```

The target plugin is responsible for ‘translating’ these generic diffractometer instructions into instrument specific command messages relayed by the protocol required for that instrument.

The following describes noteworthy aspects of framework mediated message exchange between two plugins; a client and a server side plugin (the proxy for the remote instrument):

1. On behalf of the instrument user, the client sends a registration parcel that includes the name of the target plugin.
2. Following subscription validation, a new and unique session identifier is generated. A response is returned to the client acknowledging the subscription and containing the generated session identification. The identification will be included in all parcel messages between the client and server plugin.
3. The server-side plugin might for instance be a data producer, that will start to produce data and send it to all valid subscribers. A producer is defined as an entity that can produce data without on-going human prompting.
4. The server plugin might instead provide instrument control capabilities, or indeed might provide both control and data production.

5. Each time a parcel is sent, the session identifier is used to identify the correct recipient. If the parcel cannot be treated by the framework, it is forwarded directly to the target plugin.
6. At any time (possibly before subscription), a client can send a describe parcel to a plugin, in order to obtain the plugin description.

XMLbeans are used to generate the framework component classes and Spring is used for instantiation and injection. Maven is used for management.

III. SERVICE FEATURES

A Web service may of course be accessed by either a purpose built ‘stand-alone’ application, or via a browser. The immediate appeal of a browser is that is effectively ubiquitous and in principle its use does not require any software download and installation by the client user. The relatively recent emergence of Web 2.0 technology is enabling the Web as a distributed applications platform, and is changing the role of the ‘ubiquitous’ Web browser. For instance AJAX (Asynchronous Javascript And XML) [13] introduces a capability for a browser to function in a similar manner to a stand-alone GUI. In particular AJAX provides portlet specific dynamic ‘refresh’ of browser content, rather than the traditional full page refresh required when content changes.

A particular focus of the project has been to extend the CIMA based framework to provide support for instrument control; CIMA had previously been developed for instrument monitoring. Remote operation has intrinsic risks that may result in instrument component collision and costly damage. Ideally communication should be as close as possible to being effected in ‘real-time’. Pushlets [14] allow information or data to be continuously pushed to a browser – in effect over-turning the original browser get ‘paradigm’. Data push has obvious potential for the delivery of remote instrument operation services. Currently we are exploring the use of the AJAX based cometD [15] technology as a more scalable alternative to the older Pushlet technology.

Available network and internet bandwidth and traffic is of course highly variable and largely unpredictable. A remote access system should be able to accommodate variable latency. One approach we are developing is to use virtual instruments to provide a means of testing the viability and safety of a data collection undertaken with an X-ray diffractometer. The goniometer axes of the diffractometer are motor driven and their operation may result in a collision. Using a virtual model to ‘rehearse’ the move-

ments associated with a data collection strategy provides a means of checking for the possibility of a collision. The portlet shown in Fig. 3 enables a user determined strategy to be input and ‘run-through’ at a user selected animation speed.

Indeed the model may be used to determine a collision map, by systematically simulating the movement of all drivable components and checking for contact between colliding and collidable components.

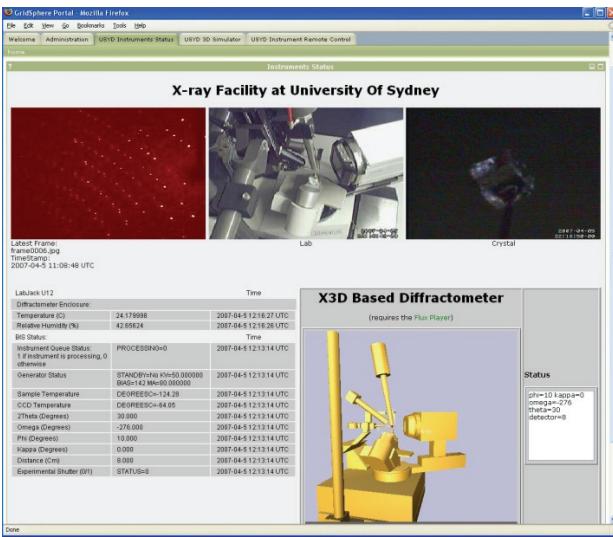


Fig. 3. Browser embedded virtual model used to rehearse a data collection strategy ahead of submitting instructions to the remote instrument

A further significant benefit of the use of a virtual model in a remote access system is that it provides a low-bandwidth, interactive and readily interpreted view (rendered client side) of the current state of the instrument. The model can be rotated arbitrarily and the view can be zoomed in and out as desired, and so provides greater view flexibility than that of a Webcam. The view can be constantly updated with just a small number of status parameters.

The inclusion of a virtual model representation of the instrument state then goes some way to solving the ‘dark-lab’ problem that arises when laboratory lighting is switched off or fails, or the Webcam fails.

The model also provides a means of safely training users without risking harm to trainee or instrument. Indeed new remote operation services and procedures may be safely tested. In short the provision of a virtual model provides for safe remote operation, improved throughput and new science facilitated by modelling and simulation.

In developing the use of browser based virtual models for remote instruments, we’ve selected X3D [16] as the

modelling ‘language’ or format. X3D has the attraction of being an ISO standard and XML schema based format, well suited to use in a Web context. X3D scenes can be externally scripted and updated by Javascript, via a scene access interface (SAI). Models and scenes are readily extensible, and it’s possible to build up libraries of virtual components.

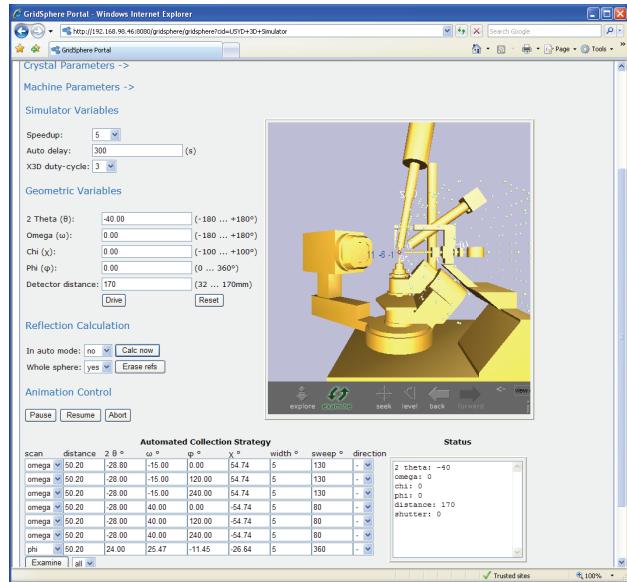


Fig. 4. The inclusion of a Pushlet driven X3D based virtual model to enhance remote instrument monitoring services

The following X3D segment provides the model shown in Figs. 3 and 4:

```
<Transform rotation="0 0 1 0" scale="1.0 1.0 1.0"
      translation="0.0 0.0 0.0">
<ProtoInstance name="Machine_axes"/>
<ProtoInstance name="BL19b-baseplate"/>
<ProtoInstance name="ColdStream"/>
<ProtoInstance name="Telescope"/>
<Group DEF="Omega">
  <Transform DEF="omega_angle" rotation="0 0 1 0.00">
    <ProtoInstance name="Omega_Block"/>
  <Group DEF="Kappa">
    <Transform DEF="alpha_plus" rotation="0 1 0 -0.872665">
      <Transform DEF="kappa_angle" rotation="0 0 1 0.0">
        <Transform DEF="alpha_minus" rotation="0 1 0 0.872665">
          <Group DEF="Kappa-circle">
            <ProtoInstance name="Kappa_Base"/>
            <Transform DEF="phi_angle">
              <ProtoInstance name="Phi_Block"/>
              <ProtoInstance name="Crystal"/>
            </Transform>
          </Group>
        </Transform>
      </Transform>
    </Group>
  </Transform>
</Group>
<Group DEF="ColdStream">
  <Transform DEF="ColdStream" rotation="0 0 1 0.00">
    <ProtoInstance name="ColdStream"/>
  </Transform>
</Group>
<Group DEF="Telescope">
  <Transform DEF="Telescope" rotation="0 0 1 0.00">
    <ProtoInstance name="Telescope"/>
  </Transform>
</Group>
<Group DEF="BL19b-baseplate">
  <Transform DEF="BL19b-baseplate" rotation="0 0 1 0.00">
    <ProtoInstance name="BL19b-baseplate"/>
  </Transform>
</Group>
<Group DEF="Machine_axes">
  <Transform DEF="Machine_axes" rotation="0 0 1 0.00">
    <ProtoInstance name="Machine_axes"/>
  </Transform>
</Group>
<Group DEF="Omega">
  <Transform DEF="omega" rotation="0 0 1 0.00">
    <ProtoInstance name="Omega"/>
  </Transform>
</Group>
<Group DEF="Kappa">
  <Transform DEF="kappa" rotation="0 0 1 0.00">
    <ProtoInstance name="Kappa"/>
  </Transform>
</Group>
<Group DEF="ColdStream">
  <Transform DEF="ColdStream" rotation="0 0 1 0.00">
    <ProtoInstance name="ColdStream"/>
  </Transform>
</Group>
<Group DEF="Telescope">
  <Transform DEF="Telescope" rotation="0 0 1 0.00">
    <ProtoInstance name="Telescope"/>
  </Transform>
</Group>
<Group DEF="BL19b-baseplate">
  <Transform DEF="BL19b-baseplate" rotation="0 0 1 0.00">
    <ProtoInstance name="BL19b-baseplate"/>
  </Transform>
</Group>
<Group DEF="Machine_axes">
  <Transform DEF="Machine_axes" rotation="0 0 1 0.00">
    <ProtoInstance name="Machine_axes"/>
  </Transform>
</Group>
```

```
<ProtoInstance name="Collimator"/>
<Group DEF="detector_block">
<Transform DEF="twotheta_angle" rotation="0 0 1 0.0">
<Group>
<Transform DEF="detector_distance" translation=".170 0 0">
<ProtoInstance name="Detector"/>
</Transform>
<ProtoInstance name="KBtn-TThetaCircle"/>
</Group>
</Transform>
</Group>
</Transform>
```

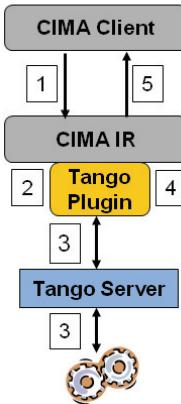
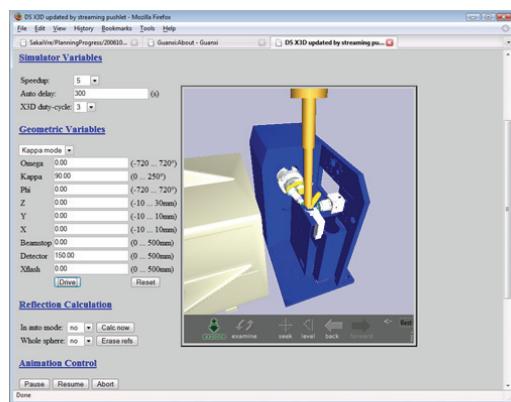


Fig. 5. Tango instrument control system as a CIMA plugin enabling control of a virtual model of an ESRF beamline instrument

Currently we use the Vivaty [17] (formerly FluxPlayer) browser plugin to render the X3D model. This requires manual download and installation, and it has some other drawbacks. Accordingly we are now exploring use of a Java Web-Start (JNLP) applet approach based on the open Source Xj3D [18].

The plugin based nature of CIMA architecture facilitates the use of the system in different instrument settings. Currently for instance, we are developing a plugin module for TANGO device servers [19] (Fig. 5). TANGO is an object oriented and distributed control system using

CORBA [20], and is being developed as an open source collaboration between the Alba, Elettra, ESRF and Soleil synchrotron facilities. The development of this plugin makes it possible to remotely drive an instrument simulator and virtual model for a beamline at the European Synchrotron Radiation Laboratory (ESRF) operated by the European Molecular Biology Laboratory (EMBL). The virtual model has an interface to the TANGO control system, such that driving the virtual model provides a viable basis for driving the beamline instrument.

Remote access services are generally provided for high value instruments – instruments for which efficient use is an expectation. This expectation of course applies to both local and remote instrument users. Diffraction experiments are best described in reciprocal space [21], which in essence is a Fourier transform of ‘real’ space. The effectiveness and efficiency of a diffraction data collection strategy is best assessed in reciprocal space.

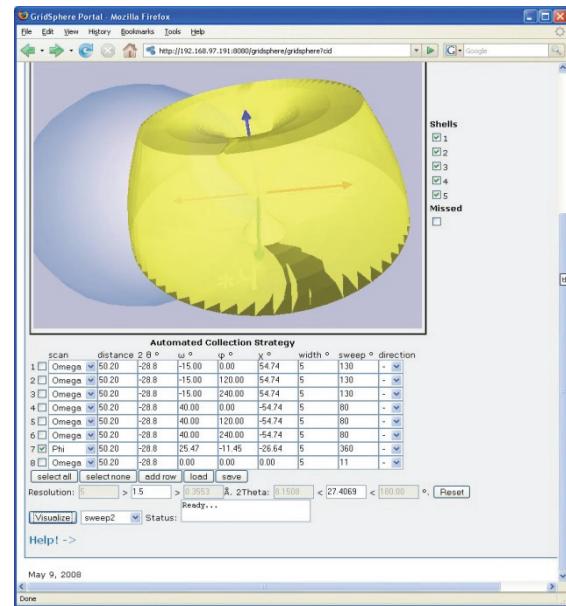


Fig. 6. Browser based X3D depiction of the coverage of diffraction space by a user defined data collection strategy

Figure 6 shows a portlet service that provides a number of display options to graphically represent the extent of coverage of diffraction (reciprocal) space provided by a user defined data collection strategy. The service includes an option to show regions of reciprocal space that are ‘missed’ by the given data collection strategy. Work is currently underway to provide for the representation of crystallography symmetry, such that symmetry equivalence and hence redundancy can be represented. Further work is to provide for automatic strategy determination.

IV. CONCLUSION

A CIMA based remote access system is being developed that utilises schema based XML parcels for communication between plugins representing are remote instrument and plugins representing the client. Safe operation of the remote instrument is facilitated through the use of a virtual model of the instrument. Future work will extend simulation capabilities to further optimise use of the remote instrument.

References

- [1] VNC: Virtual Network Computing; <http://www.realvnc.com/>. Accessed 27 November 2008. Variants include TightVNC, RealVNC, UltraVNC, and TridiaVNC.
- [2] CITRIX: <http://www.citrix.com/>. Accessed 27 November 2008.
- [3] SSGD: Sun Secure Global Desktop; <http://www.sun.com/software/products/sgd>. Accessed 27 November 2008. Formerly Tarantella.
- [4] NX NoMachine; <http://www.nomachine.com/>. Accessed 27 November 2008.
- [5] S. M. Soltis, A. E. Cohen, A. Deacon, T. Eriksson, A. González, S. McPhillips, H. Chui, P. Dunten, M. Hollenbeck, I. Mathews, M. Miller, P. Moorhead, R. P. Phizackerley, C. Smith, J. Song, H. van dem Bedem, P. Ellis, P. Kuhn, T. McPhillips, N. Sauter, K. Sharp, I. Tsypba and G. Wolf, *New paradigm for macromolecular crystallography experiments at SSRL: automated crystal screening and remote data collection*. Acta Cryst. **D64**, 1210-1221 (2008).
- [6] R. Bramley, K. Chiu, J. C. Huffman, K. L. Huffman and D. F. McMullen, *Instruments and Sensors as Network Services: Making Instruments First Class Members of the Grid*. Indiana University Computer Science Department Technical Report 588, December 2003.
- [7] T. Devadithya, K. Chiu, K. L. Huffman and D. F. McMullen, *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.
- [8] R. Bramley, K. Chiu, T. Devadithya, N. Gupta, C. Hart, J. C. Huffman, K. L. Huffman, Y. Ma and D. F. McMullen, *Instrument Monitoring, Data Sharing and Archiving Using Common Instrument Middleware Architecture (CIMA)*. Journal of Chemical Information and Modeling **46 (3)**, 1017-1025 (2006).
- [9] D. F. McMullen, T. Devadithya and K. Chiu, *Integrating Instruments and Sensors into the Grid with CIMA Web Services*. Proceedings of the Third APAC Conference on Advanced Computing, Grid Applications and e-Research (APAC05). Gold Coast, Australia, September 25-30, 2005. <http://grid.cs.binghamton.edu/projects/publications/integrate-APAC05/>
- [10] I. M. Atkinson, D. J. du Boulay, C. Chee, K. Chiu, T. King, D. F. McMullen, R. Quilici, N. G. D. Sim, P. Turner and M. Wyatt, *CIMA Based Remote Instrument and Data Access: An Extension into the Australian e-Science Environment*. Proceedings of IEEE International Conference on e-Science and Grid Computing (e-Science 2006). December 2006. Amsterdam, The Netherlands.
- [11] D. du Boulay, C. Chee, K. Chiu, R. Leow, D. F. McMullen, R. Quilici and P. Turner, *Portal Services for Collaborative Remote Instrument Control, Monitoring and Data Access*. 3rd IEEE International Conference on e-Science and Grid Computing, Bangalore, India, December 10-13, 2007.
- [12] D. J. du Boulay, S. Brockhauser, C. Chee, K. Chiu, T. Devadithya, R. Leow, D. F. McMullen, R. Quilici and P. Turner, *Remote Instrument Control with CIMA Web Services and Web 2.0 Technology*. International Journal of Online Engineering (iJOE) **4 (1)** (2008).
- [13] AJAX: Asynchronous JavaScript Technology and XML <http://java.sun.com/developer/technicalArticles/J2EE/AJAX>. Accessed 27 November 2008.
- [14] Pushlets: <http://www.pushlets.com/>. Accessed 27 November 2008.
- [15] CometD: <http://cometdproject dojo toolkit.org/>. Accessed 27 November 2008.
- [16] X3D: <http://www.web3d.org/about/overview/>. Accessed 27 November 2008.
- [17] Vivaty: <http://www.vivaty.com/downloads/player/>. Accessed 27 November 2008.
- [18] XJ3D: <http://www.xj3d.org/>. Accessed 27 November 2008.
- [19] TANGO: <http://www.tango-controls.org/>. Accessed 27 November 2008.
- [20] CORBA: <http://www.omg.org/>. Accessed 27 November 2008.
- [21] C. Giacovazzo, H. L. Monaco, G. Artioli, D. Viterbo, G. Ferraris, G. Gilli, G. Zanotti and M. Catti, *Fundamentals of Crystallography*. Oxford University Press, USA; 2nd edition (2002).



PROF. IAN ATKINSON has a background in computational chemistry that led him to the world of Supercomputing. More recently he has been engaged in e-Research and Grid computing service development focused on serving the Australian research community. He has a long-standing interest in eResearch methods, tools, scientific data management and user interfaces for HPC tools. Current interests are Grid mediated access to data sources. He is a CI on the Australian Archer eResearch project focussing on data grids and data collection from instruments and sensors. He holds joint appointments as the Manager of High Performance Computing within Information Technology and Resources at JCU and also a facility member of the School of Maths, Physics and IT at JCU.



DR. DOUGLAS DU BOULAY graduated from the University of West Australia with a PhD in Physics specializing in high precision, charge density crystallography. That was followed by software development work enhancing the Xtal software system for analyzing and modelling diffraction data. Doug also contributed to extending the power of the Crystallographic Information Framework (CIF) dictionaries, through parsing and executing embedded dictionary methods. Subsequently he was awarded Centre of Excellence and JSPS Fellowships for inorganic crystallography studies at the Tokyo Institute of Technology. More recently he's been exploring virtual reality as an aid to understanding and for remote control and monitoring of crystallography experiments.



DR. SANDOR BROCKHAUSER received his M.Sc. from Technical University of Budapest, Hungary, with a double major in Electrical Engineering and Computer Science and earned a Ph.D. from University of Leoben, Austria in petroinformatics with a thesis in hydrocarbon fluid flow simulation in anisotropic reservoir. He has worked in crystallography since 2004 when he joined the European Molecular Biology Laboratory in Grenoble, France. He is a co-inventor of the MiniKappa Goniometer Head and the founder of the International Kappa Workgroup that aims advancing data collection strategies by effectively introducing crystal reorientation features of newly re-appearing Kappa Goniometers in European synchrotrons and around the world. Since 2007 he has been the Head of the Computational Crystallography Association, EMBL, as well as responsible for the Macromolecular Crystallography Multi-Wavelength Anomalous Dispersion Beamline ID14-4 at European Synchrotron Radiation Facility, France.



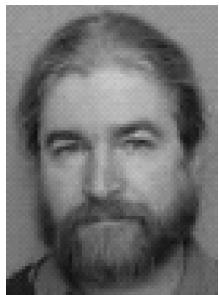
DR. RICK McMULLEN has until very recently been Director and Principal Scientist of the Knowledge Acquisition and Projection Lab in the Pervasive Technology Laboratories at Indiana University. He received a Ph.D. in Chemistry in 1982 and served in a variety of engineering and management roles in the chemical and electronics industries before joining the Supercomputer Computations Research Institute as a Research Scientist in visualization and high performance computing. Dr. McMullen came to Indiana University in 1990. Dr. McMullen's primary research interests are in ubiquitous computing, knowledge-based support systems, high-speed international networking, visualization, and virtual reality. Recent projects include work with the U.S. Navy to develop knowledge-based remote maintenance and repair (tele-maintenance) systems, development of distributed data acquisition with the Department of Energy Next Generation Internet (NGI) program, and management to develop distributed data acquisition, management and reduction systems for the Advanced Photon Source at Argonne National Laboratory and the Advanced Light Source at Berkeley. Dr. McMullen has just moved to be director of research computing and senior scientist at Kansas University.



ROMAIN QUILICI earned a MSc in mathematics and computer science from University of Aix-Marseille III, St Jerome, followed by a Engineer Degree from ENST Telecommunications. Following development work with HP, Romain joined the INRIA Oasis team led by Prof. Denis Caromel developing the ProActive Grid middleware system. Romain's contribution led to him becoming chief architect and technical leader for the ProActive Grid plug tests. Following this, Romain moved to the University of Sydney to make a leading contribution to the development of CIMA for remote instrument control.



ANDREW SHARPE graduated with a B.Sc. from James Cook University in 2004, with majors in computer science and mathematics. He then joined the Australian Partnership for Advanced Computing (APAC), and contributed to the development of computing and collaboration services for the Australian scientific community. Following that Andrew joined the ARCHER project to develop eResearch services, and this has included a significant contribution to the evolution of CIMA.



DR PETER TURNER is a crystallographer in the School of Chemistry at the University of Sydney, and is co-convenor for an Australian Research Council funded research network known as the Molecular and Materials Structure Network (MMSN). The MMSN has a particular focus on the development of eResearch services for the molecular and materials structure determination community. Peter's interests include the development of next generation remote access services and the development and utilization of virtual models and simulations. Interests also include the development of automation services for crystallography, and the development of data repository and publication services.