Portal Services for Collaborative Remote Instrument Control, Monitoring and Data Access

Douglas du Boulay,[1] Clinton Chee,[1] Kenneth Chiu,[2] Richard Leow, [1] Donald F. McMullen,[3] Romain Quilici,[1] Peter Turner[1*]

[1]Department of Chemistry, University of Sydney, Sydney, NSW 2006, Australia.
[2]Computer Science Department, SUNY Binghamton, Binghamton, NY, USA.
[3]The Pervasive Technology Labs at Indiana University, Bloomington, IN 47404 USA.
boulay_d@chem.usyd.edu.au, c.chee@chem.usyd.edu.au, kchiu@cs.binghamton.edu, mcmullen@indiana.edu, r.quilici@chem.usyd.edu.au, p.turner@chem.usyd.edu.au

Abstract

A two component portal system is being developed for collaborative remote instrument and data control and monitoring. The system builds on and enhances the Common Instrument Middleware Architecture (CIMA) model for Web services based monitoring of remote scientific instruments and sensors. The architecture supports remote access to multiple instruments from a single portal.. Plugin modules are used to provide flexibility and re-use, and the notion of plugin control is being developed. The use of Web 2.0 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. An important component of the system is a Webs services driven portlet for collaborative image viewing

1. Introduction

There are obvious financial, functional and educational returns in developing collaborative access services for remote scientific instrumentation. State of the art high performance laboratory instrumentation, such as high flux X-ray diffraction systems and powerful electron microscopes, is increasingly expensive and often 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.

The popular remote desktop approach to providing remote instrument access, such as typified by the use of Virtual Network Computing[1] and its many variants, CITRIX[2], Sun Secure Global Desktop[3] and NX NoMachine[4], has the significant advantages of ease of set-up and familiarity. While convenient, these approaches are nonetheless less than ideal. Remote desktop access can afford remote instrument users with excessive control over expensive and potentially dangerous instruments. Any additional service and functionality is limited to that provided by the remote desktop and the policies of the remote institution. Significantly, while readily accessible through the internet, the instrument effectively remains an isolated resource.

The use of portal-portlet technology offers a number incentives, including the provision of rich functionality and global accessibility, without the need for any client software other than a suitable Web browser. Importantly, the performance distinction between stand-alone interfaces and browser based interfaces is being eroded by Web 2.0 technologies, such as AJAX[5] and DOJO[6]. Portal services are intrinsically collaborative, with multiple authorised users able to access a given service. Robust security can be provided through certificate based single-sign-on authentication and authorisation, and the level of access provided to the remote user can be tightly controlled. Portal services can be further enhanced through the utilisation of Web services.

The use of Web services offers several compelling attractions:

- Location, platform and language independent Web resource access.
- Integration of the instrument into the Web (and Grid).
- Supports Service Orientated Architectures (SOAs). Can integrate distributed and heterogeneous services.

0-7695-3064-8/07 $25.00 © 2007 Crown Copyright
DOI 10.1109/e-Science.2007.25

- Robust underlying security model (WS-Security supports security tokens and certificates).
- Use of HTTP as the underlying Web transport protocol facilitates firewall passage.
- Facilitates the linkage of multiple users and resources for collaborative interactions across the Web.

The use and performance of Web services for the remote control of relatively simple laboratory devices, such as a waveform generator, has been described by Yan et al.[7]. At the other end of the spectrum, the GridCC project to Grid enable large scale facilities, such particle accelerators, is underpinned by Web services[8].

The pioneering Common Instrument Middleware Architecture[9,10,11] project is a NSF Middleware Initiative project researching a consistent and re-usable middleware framework to enable and embed instruments as addressable Web and Grid resources, through the use of Web services. Thus far CIMA has been developed solely for remote instrument and sensor monitoring, although aspects of CIMA mediated instrument control have been canvassed[12].

We describe herein extensions and enhancements to the CIMA model, being undertaken as part a collaborative endeavour to develop a comprehensive and feature rich remote access portal system[13].

2. Design and Functionality

The principle components of the portal system are shown in Fig. 1. A modular service oriented architecture (SOA) model using Web services has been adopted to provide maximum applicability, flexibility and capability.

The architecture is comprised of two primary containers; one located at the instrument site (Source) and providing instrument services, and a second that provides user access interface components (Sink) and need not be co-located with the instrument or the instruments host institution. 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 by 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, or each instrument could have its own instance of Container B. A third model would have both containers located at the instrument site or facility, but with Container A serving in a DMZ.

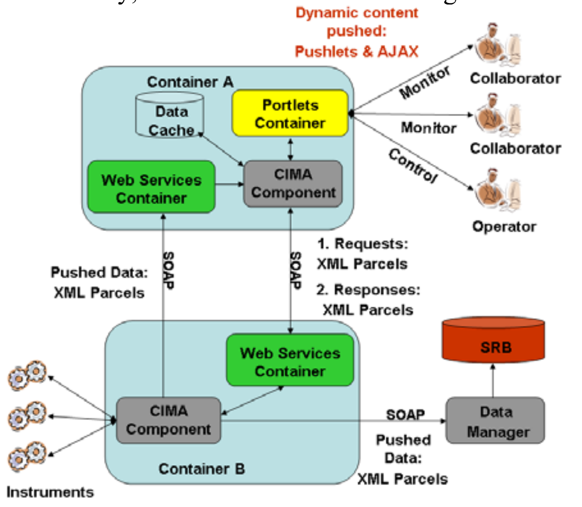


Figure 1. Remote instrument control elements. Only one person at a time can control any given instrument.

Both Source and Sink containers have Web services components that receive SOAP calls from the corresponding component in the partner container. Complementary Web services stubs (not shown in Fig 1) are responsible for assembling and sending the SOAP messages. The CIMA components assemble and provide the XML parcels delivered in the SOAP messages. Communication between the complementary Source and Sink CIMA components is formally described in terms of *channels* that are, in effect, defined by the XML parcel exchange endpoints (Fig. 2).

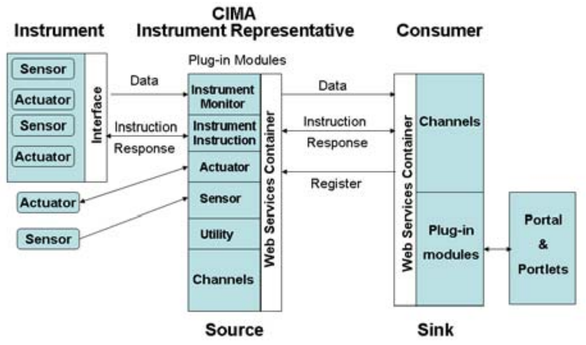


Figure 2. CIMA based components: Source components reside in Container B and the Sink components are located in Container A (see also Fig 1).

Flexibility and a capability for re-use is provided through the use of plugins to the CIMA components of each container. The relationship of the plugins to the CIMA framework is schematically illustrated in Fig. 2.

A prospective client registers for plugins and services, with access depending on assigned client rights.

The introduction and development of instrument control services has been undertaken in accord with the CIMA channels and plug-in model[12]. Plugins may serve individual sensors and/or actuators, or may serve the aggregation of sensors and actuators that defines an instrument. Instrument control has been implemented through the introduction of the Instrument Instruction Module and its partner plugin, the Instrument Monitor Module. Multiple instruments may be served trough the loading of corresponding modules for each instrument. Alternatively, the architecture and its implementation supports the use of dedicated instances of container B for each instrument.

The Instrument Instruction Module serves as an instruction interface to specific instrument software (or device drivers). That is, the module translates CIMA parcels (see below) into instrument specific instructions to be sent to the instrument interface. The instructions may be to get instrument status information, change the state of the instrument or operate the instrument (e.g. collect data).

As Fig. 2 suggests, other utility plugins may be used and we have, for instance, introduced a plugin for the conversion of an instrument data image into an image format suitable for display in the client browser interface. Data management may also be provided through an additional plugin or through a data management service located elsewhere and accessed by Web services[13].

Plugin selection and configuration is currently static, in the sense that the specification is provided in an XML configuration file loaded on container start-up. The same process provides for service registration and hence channel definition. Registration details are persistent in that they are stored in a file that can be read if Container B needs to be re-started.

There are two primary modes of communication between the two portal containers. One mode is synchronous and 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 from the instrument. The CIMA component in Container B extracts data from the XML parcel included in the request from A, builds instrument-specific instructions and finally returns the response from the instrument as an XML parcel. The response may simply be an acknowledgement, or data/meta-data associated with the instruction. New parcel types have been introduced to support the request/response mode (see below).

Requests are of two kinds; those that affect the instrument state (e.g. SET and OPERATE requests for instrument control) and those that retrieve instrument information (e.g. GET for instrument status data).

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 of Container B (Source) sends an XML parcel via SOAP to the CIMA component of Container A (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 following example illustrates a GET request to retrieve a goniometer parameter:

```
<Parcel>
  <Type>http://www.cima.usyd.edu.au/2006/Get</
Type>
    <Body>
        <Source>BIS</Source>
        <Variable>OMEGA</Variable>
    </Body>
  </Parcel>
```

A second parcel example illustrates the simple nature of an OPERATE instruction parcel

```
<Parcel>
  <Type>http://www.cima.usyd.edu.au/2006/Operat
e</Type>
    <Body>
        <Source>BISControl</Source>
        <Command>DRIVE</Command>
        <Paramters> … </Parameters>
    </Body>
  </Parcel>
```

Although the skeletal parcel structure is simple, quite complex parcels can nonetheless be assembled, and this inherent parcel flexibility facilitates the introduction of new control system modules. Currently we are developing a plugin module for TANGO device servers[14]. TANGO is an object oriented and distributed control system using CORBA[15], and is being developed as an open source collaboration between the Alba, Elettra, ESRF and Soleil synchrotron facilities.

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.

Recently we have introduced a capability for plugin control by the on-site administrator and, to a slightly lesser extent, by the remote user. The currently rudimentary capability (Fig. 3) provides a basis for exploring the potential utility and benefit of such control.

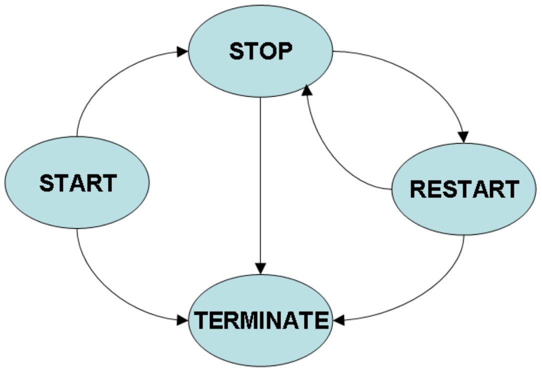


Figure 3. Schematic depiction of current capability for the plugin 'state machine'.

The START instruction starts the plugin, and cannot be called remotely. STOP stops the plugin, but keeps the plugin registered and remotely accessible. RESTART stops the plugin and then starts it again, and it remains remotely callable. TERMINATE removes or cleans the plugin from the application. After a TERMINATE instruction, the plugin cannot be accessed remotely. Further plugin handling instructions planned but not yet implemented include SUSPEND and RESUME for thread control.

3. Implementation and Services

Crystallographic molecular structure determination instrumentation has been selected as a particularly attractive system development domain, with well defined work-flows and data structures, and relatively common (if not standardised) instrument types. The remote access portal system is being developed to be applicable both conventional and major facility instrumentation.

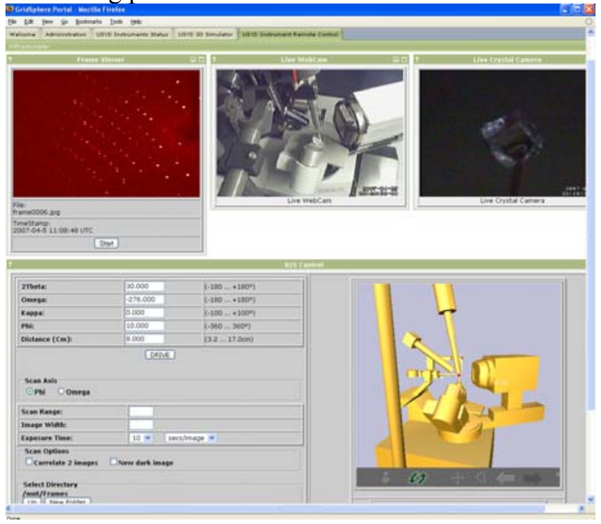
The two primary system components (Source and Sink) are deployed in separate Servlet containers, and currently we use either Tomcat[16] or Jetty[17]. Tomcat may used for both containers, however there is also the option of using Jetty for Container B. The use of Jetty allows multiple instruments to be served through multiple 'stand-alone' instances of the Instrument Representative (IR) of Container B (see Fig 2). Perhaps a drawback in some contexts, if each instance resides on the same machine then each must be assigned a distinct port number.

Multiple Tomcat instances could be used, however we find Jetty has the advantage of being relatively lightweight. While suitable for development, we have yet to determine if Jetty would be suitable in a production setting.

Alternatively, and perhaps more attractive, the introduction of plugin control facilitates multiple instrument access through the loading of multiple instrument modules in the Instrument Representative within a single Tomcat instance of Container B.

We have used both Axis[18] and XFire[19] for the provision of Web services and currently use embedded XFire as this allows programmatic initiation. Any JSR 168 compliant portlet container may be used for Container A, and we currently use GridSphere[20].

The goal of real-time updating of instrument status and data displays for effective and safe instrument control has driven the introduction of Pushlet[21] and AJAX Web 2.0 technologies to enable (pseudo) real-time data push from the instrument to the client. It is then possible for instance to view CCD based X-ray detector images, along with metadata, without polling. The utilization of these technologies has in turn improved the functionality of our instrument monitoring portlets.



File:
frame0006.jpg
TimeStamp:
2007-04-5 11:08:48 UTC

2Theta:	30.000	(-180 ... +180°)
Omega:	-276.000	(-180 ... +180°)
Kappa:	0.000	(-100 ... +100°)
Phi:	10.000	(-360 ... 360°)
Distance (Cm):	8.000	(3.2 ... 17.0cm)

Scan Axis
[✓ Phi] Omega

Scan Range:	
Image Width:	
Exposure Time:	10 secs/image

Scan Options
Correlate 2 images New dark image

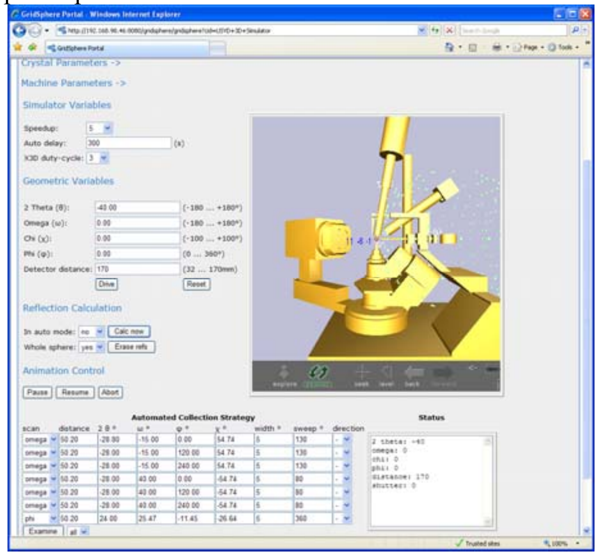
Select Directory
/mnt/frames

Figure 4. GridSphere portlet for X-ray diffraction instrument control.

The current form of the GridSphere instrument access portal provides an instrument control portlet (Fig. 4) allowing the user to define and initiate simple data collections. Options for the provision of more complex data collections are currently being developed (see below). The pane provides for data collection parameter input, webcam monitoring of the instrument and crystal sample, and a display of the current CCD detector data image. As mentioned, images and data are dynamically updated through the use of AJAX and Pushlet technologies.

An important element of the control portlet is an X3D[22] based virtual representation of the instrument. This provides an easy to assess representation of the current state of the instrument and has low bandwidth requirements in requiring only a small number of (pushed) instrument parameters. The virtual representation at least partially solves the 'dark laboratory' problem that arises when lab lights are turned off or when webcams fail.

Currently we use FluxPlayer[23] as a browser plug-in to display the virtual instrument. Accordingly the user can 'zoom in' on the virtual instrument and adopt any viewing position around the instrument, including preset positions.



Speedup: 5
Auto delay: 300 (s)
X3D duty-cycle: 3

2 Theta (θ): -40.00 (-180 .... +180°)
Omega (ω): 0.00 (-180 .... +180°)
Chi (χ): 0.00 (-100 .... +100°)
Phi (φ): 0.00 (0 .... 360°)
Detector distance: 170 (32 .... 170mm)

In auto mode: no
Whole sphere: yes

Automated 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

Status

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

Figure 5. Portlet to operate a X3D based diffraction instrument simulator. Available diffraction data may also be simulated.

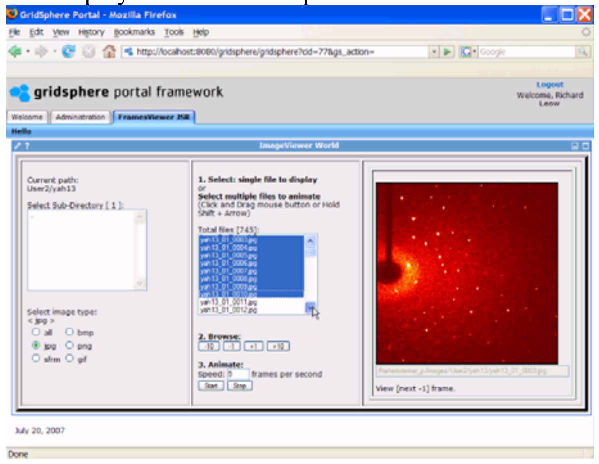
A strong disincentive to providing remote client control of a physical device is that unskilled operators, or simple data entry errors, may lead to costly damage to the instrument. Ideally the instrument control software installed at the remote site would include a collision map to prevent accidental damage. In practice however collision map software is not always provided, and when such software is available it may have weakness or bugs. 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 the virtual instrument as a simulator to test the viability of a data collection (Fig. 5).

Given sufficient preliminary scanning information (sufficient to provide the crystallographic orientation matrix), the simulator now has a capability to display the expected location of all of the diffraction data for the sample. The effectiveness and efficiency of the data collection strategy may then be assessed.

The simulator is being developed as an integral part of the instrument control system. Once a satisfactory data collection strategy has been determined, it will ten be possible to issue that strategy to the remote instrument.

The virtual instrument offers a visual test-bed for developing further instrument control plug-ins. The simulator is currently being linked to a TANGO device server for instance, and the same could be done for the EPICS[24] control system. The simulator may also serve as an indestructible training tool.

Another tabbed portlet 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. Like the control portlet, the current data frame, webcam view and virtual instrument state are also displayed in this status portlet.


GridSphere Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/gridsphere/gridsphere?cid=77&gs_action=
gridsphere portal framework
Logout
Welcome, Richard Leow
Welcome
Administration
FramesViewer JSR
Hello
ImageViewer World
Start
Stop
Done

Current path:
User2/yah13
Select Sub-Directory [1]:

Select image type:
< jpg >
○ all ○ bmp
● jpg ○ png
○ sfrm ○ gif

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

Total files [745]:
yah13_01_0003.jpg
yah13_01_0004.jpg
yah13_01_0005.jpg
yah13_01_0006.jpg
yah13_01_0007.jpg
yah13_01_0008.jpg
yah13_01_0009.jpg
yah13_01_0010.jpg
yah13_01_0011.jpg
yah13_01_0012.jpg

2. Browse:
-10 -1 +1 +10

3. Animate:
Speed: 5 frames per second

framesviewer_p/images/User2/yah13/yah13_01_0003.jpg
View [next -1] frame.

July 20, 2007

Figure 6. Data image inspection portlet. A set of images may be 'animated' at a user selected rate.

A further portlet service provides diffraction data image inspection (Fig. 6). The directory tree is easily traversed though the portlet 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).

The service allows a user to decide if the sample is suitable for a full data collection, or to retrospectively inspect features in a completed data collection.

4. Collaboration Service

As suggested in Figure 1, the use of a portal system is inherently collaborative with multiple users able to monitor a data collection via a browser. In order to provide further collaborative capability, we have begun to develop a Web services and Pushlet driven portlet for collaborative image evaluation. Images that have been selected independently by geographically distributed collaborators may be viewed 'side by side' and manipulated by all participants. The image viewing panes maybe independently resized, and the portlet includes a 'chat' text box (Fig. 7).

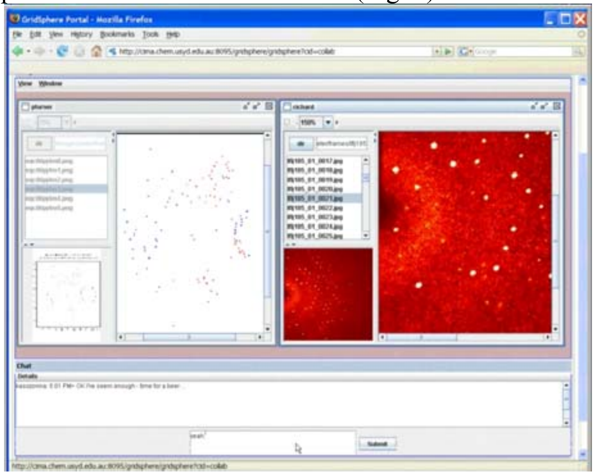

GridSphere Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://cima.chem.usyd.edu.au:8095/gridsphere/gridsphere?cid=collab
View Window
pturner
richard
150%
Chat
Details
Submit

Figure 7. Web services and Pushlet driven portlet for collaborative image analysis. Multiple users may participate, each having an independent viewing pane and access to a text chat pane (only two users shown here).

The multiple viewing windows are synchronised using notification packets pushed via Pushlet connections tat are maintained between every client and the portal server. Pushlet clients can subscribe as listeners to events from particular resources and the server provides selective notifications of when relevant events occur for each. A client notifies the server that its state has changed, providing it with new status information and then the client publishes a notification event to the server for broadcast to every subscribed client.

The resources themselves can be either server pre-configured, or agreed upon between the clients themselves. Communication between clients and server is via Web services (SOAP) calls. FlexDoc[25] is used to provide the image and text windows and GUI functionality. Multiple widows can be shown in the same display area, or alternatively a tabbed view option can be selected. Such that each users display appears in a single tabbed pane.

5. Conclusion

The Source and Sink components of the CIMA model have been augmented with Control components providing instrument control and plugin control. These components form the core of a Web services, AJAX and Pushlets driven portal system for collaborative remote instrument control and monitoring. The portal system provides portlets for instrument and experiment simulation, and for data inspection. Data collection can be simulated with an X3D based virtual instrument representation that also provides at least a partial solution for the dark laboratory problem. A TANGO device server client plugin module is being developed to provide a capability for remote monitoring and operation of synchrotron beamline instrumentation. Collected data may viewed and animated and a collaboration tools allows multiple users to independently select and collaboratively inspect data and other images.

An outstanding deficiency in the system as it stands is the lack of comprehensive security infrastructure. Currently security depends on the login role assignment provided by GridSphere. Exploratory work is underway to address this issue, through the use of certificates and tokens.

6. Acknowledgements

The support of the ARC e-Research SRI, 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..

7. References

[1] VNC: Virtual Network Computing; http://www.realvnc.com/. Accessed 10 July 2007. Variants include TightVNC, RealVNC, UltraVNC, and TridiaVNC.

[2] CITRIX: www.citrix.com. Accessed 10 July 2007.

[3] SSGD: Sun Secure Global Desktop: http://www.sun.com/software/products/sgd. Accessed 10 July 2007. Formerly Tarantella.

[4] NX NoMachine; www.nomachine.com. Accessed 10 July 2007.

[5] AJAX: Asynchronous JavaScript Technology and XML :http://java.sun.com/developer/technicalArticles/J2EE/AJAX/. Accessed 10 July 2007.

[6] DOJO: dojotoolkit.org. Accessed 10 July 2007.

[7] Yan, Y. , Liang, Y., Du, X. (2005), "Controlling remote instruments using Web services for online experiment systems. ", *Proceedings. 2005 IEEE International Conference on Web Services, 2005 (ICWS 2005).*

[8] GridCC: www.gridcc.org. Accessed 10 July 2007.

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

[10] Devadithya I., Chiu K., Huffman K.L., McMullen D.F., "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.

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

[12] McMullen D.F., Devadithya I., Chiu, K., "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/

[13] Atkinson, I.M., du Boulay, D. J., Chee, C., Chiu, K., King, T., McMullen, D.F., Quilici, R., Sim, N.G.D, Turner, P., Wyatt. M., "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.

[14] TANGO: www.tango-controls.org. Accessed 10 July 2007.

[15] CORBA: www.omg.org. Accessed 10 July 2007.

[16] Tomcat: tomcat.apache.org.Accessed 10 July 2007.

[17] Jetty: www.mortbay.org. Accessed 10 July 2007.

[18] Axis: ws.apache.org/axis. Accessed 10 July 2007.

[19] XFire: xfire.codehaus.org. Accessed 10 July 2007.

[20] GridSphere: www.gridsphere.org/gridsphere/gridsphere. Accessed 10 July 2007.

[21] Pushlets: http://www.pushlets.com. Accessed 10 July 2007.

[22] X3D: www.web3d.org/about/overview. Accessed 10 July 2007.

[23] Flux Player: www.mediamachines.com. Accessed 10 July 2007.

[24] EPICS:http://www.aps.anl.gov/epics/about.php. Accessed 10 July 2007.

[25] FlexDock: flexdock.dev.java.net. Accessed 10 July 2007.