Providing Portlet-Based Client Access to CIMA-Enabled Crystallographic Instruments, Sensors, and Data

Hao Yin[1], Donald F. McMullen[2], Mehmet A. Nacar[2,4], Marlon Pierce[2], Kianosh Huffman[3], Geoffrey Fox[2,4], Yu Ma[4]

[1]School of Computer Science, Sichuan University, No. 24, South Section 1, Yihuan Road, Chengdu, 610065, China

[2]The Pervasive Technology Labs at Indiana University, 501 N. Morton St. Bloomington, Indiana 47404 USA

[3]School of Informatics, Indiana University, 901 E. 10th St., Bloomington, Indiana 47408

[4]Department of Computer Science, Indiana University, 215 Lindley Hall, 150 S. Woodlawn, Bloomington, Indiana 47405
{hayin, mcmullen, mnacar, marpierc, kihuffma, gcf, yuma}@indiana.edu

Abstract— **The Common Instrument Middleware Architecture (CIMA) project, supported by the NSF Middleware Initiative, aims at making scientific instruments and sensors remotely accessible by providing a general solution for services and user interfaces to remotely access data from instruments and to remotely monitor experiments. X-ray crystallography is one of several motivating applications for the development of CIMA. Data such as CCD frames and sensor readings may be accessed by portals through middleware services as they are being acquired or through persistent archives. CIMA software may be used to federate online instruments in multiple labs, so this project must also address problems in data management and data sharing. This paper describes a collaboration between the CIMA and the Open Grid Computing Environments (OGCE) project to enable remote users to monitor instruments and interact with data gathered from CIMA-enabled crystallography laboratories through various Web portal components (*portlets*) running within a standards-compliant portal container. We also discuss an approach taken to develop portlets that use Web Services for data management and solutions for managing distributed identity and access control.**

I. Introduction

Remote access to shared instrument resources is a major outcome of e-Science development projects in many disciplines. Shared remote access improves instrument utilization, collaboration between users and instrument experts and provides “hooks” for automating the processing of data coming from instruments by pre-configured workflows [1].

One of the key issues in developing shared instrument systems is how to create an open and flexible approach to user interfaces for access to instruments and the data streams coming from them. In related work we have shown how portals can be used to organize access to instruments through the Common Instrument Middleware Architecture (CIMA) [2] and how individual portlets can provide specialized, role and task specific functionality as users, technicians and system administrators interact in the generation, analysis and management of data from shared instrument resources. In this paper we will focus on the approach taken to develop portlets for managing crystallographic data in a group of cooperating laboratories.

II. Problem Statement

X-ray crystallography is an analytic technique to help scientists understand and determine the precise molecular structure of a crystalline substance. However the instruments used to perform these types of studies, X-ray diffractometers, are quite expensive and require a highly trained operator. In a previous paper [3] we discuss the integration of a group of laboratories that employ X-ray diffraction techniques using CIMA as a common instrument interface methodology shared by all labs participating in the federation. This federation is based on sharing of instruments, information and expertise, and poses a number of functional requirements for user interfaces to the underlying CIMA-based data acquisition systems and other services provided by the labs. To facilitate interactions between people, instruments and data a design decision was made to organize access to instruments and data through a GridSphere [4] portal. Individual functions in the portal are to be implemented as portlets accessing back-end services with Web Services interfaces. The primary design and implementation questions at this point center on developing a straightforward and reusable technique for constructing portlets that integrate well with CIMA and other services used to access instruments and data in the lab federation, and with the authentication and authorization services provided by GridSphere.

For the current work, a subset of requirements relating to user and administrative interaction with data was chosen to evaluate approaches to portlet development. The major requirements include the following: users are classified as lab administrators, lab users, and general public; groups of users may be defined as well; remote users and in-lab crystallographers must be able to monitor an experiment in progress and view current and previously collected CCD frames and related environmental and technical parameters; data sets are initially owned by the lab that performed the experiment and collected the data; individual users can view, but not modify or delete their samples and related sensor readings; lab administrators should be able to set the end time of an experiment; lab administrators must be able to add and remove users to an access control list for a sample; some sample data may be provided to the general public for educational or public science awareness purposes; individual functions that are of general utility should be implemented as software components in a reusable, standards-based manner within portlets that can be added or removed by administrators or end-users as appropriate; the portlets must interact with a lab's data manager software via Web Services calls; users and groups will be managed through the portal and access to all functions of the portal will be provided by a single sign-on; and nothing more than a web browser should be needed to interact with all functionality provided by the portal.

Although the JSR 168 portlet specification [5] provides some capabilities needed to meet the above requirements, these are somewhat limited, and the specification categorically does not support the composition of reusable component widgets. Portlets may be developed from a number of technologies, including Struts, Velocity, JavaServer Pages, and JavaServer Faces (see below) through "portlet bridges". Three key requirements are that portlets will contain various GUI widgets possibly beyond those in HTML forms, that the presentation widgets would be decoupled from the underlying data model they use, and that widgets be easy to test, preferably independently from portlets that contain them. JavaServer Faces (JSF) [6], a specification for building user interfaces for server-side applications, has proven to be a good solution for meeting these requirements. One advantage of JSF is that it provides rich tag libraries to build components which run on the server, handle events generated by a client, and which can be rendered back to the client. The other advantage is that JSF is based on the Model-View-Controller (MVC) model, offering a clean separation between presentation and logic. In addition to improving the flexibility with respect to back end data sources and reusability of portlets in general, GUI component beans developed for JSF applications can easily be unit tested and can reused in other types of applications, including non-web applications.

The requirements outlined above led us to develop the following portlets as test cases for a portlet design strategy based on JSF: *lab overview*, providing the current status of a facility and its instruments; *administrative*, to support management of sample ownership and other parameters related to individual experiments; *public sample*, providing sample data to all portal users and the general public; and *user sample*, to show logged-in users their samples and other samples on whose access control lists they appear. These JSF-based portlets directly access Web Services data sources and render the output.

III. Conclusions

The work described here provides a model for organizing and accessing data from CIMA instruments and identifies a problem and provisional solution for managing identity relationships more complex than those provided by GridSphere. The GridSphere framework of the CIMA crystallography portal uses extensively vetted standards, to which JavaServer Faces adds a flexible approach to GUI component development and interoperability with Web Services back-ends.

Difficulties encountered in this project with authorization indicate that more work is needed to provide a flexible, general solution for authentication and authorization within and between portal instances, and that authentication and authorization services should be easily accessible to portlets. We have taken a pragmatic approach to solving the problem of merging portal and data service identity. However, our system is a good candidate for investigating Web Service security standards [7,8] for securely handling authentication tokens and access privileges. Another interesting unresolved problem is the integration and federation of multiple instances of the CIMA crystallography portal.

Acknowledgment

CIMA is supported by National Science Foundation cooperative agreement SCI 0330568 and grant MRI CDA-0116050. OGCE is supported by the NSF Middleware Initiative grant SCI 0330613. We thank Professor Jiliu Zhou, School of Computer Science, Sichuan University, China for supporting H.Y.; Prof. R. Bramley and T. Devadithya, IU CS Dept. for contributions to CIMA design and implementation; Dr. J. C. Huffman, IU Chemistry Dept., and crystallographers at collaborating labs worldwide for their help and feedback; and Ji Young Chong of the IU Telecommunications Dept. for graphical design of the CIMA crystallography portal.

References

[1] R. Bramley, K. Chiu, T. Devadithya, N. Gupta, C. Hart, J. C. Huffman, K. Huffman, Y. Ma, D. F. McMullen, "Instrument Monitoring, Data Sharing, and Archiving Using Common Instrument Middleware Architecture (CIMA)", J. Chem. Inf. Model., 2006. DOI: http://dx.doi.org/10.1021/ci050368l

[2] T. Devadithya, K. Chiu, K. Huffman, D.F. McMullen, "The Common Instrument Middleware Architecture: Overview of Goals and Implementation", In Proceedings of the First IEEE International Conference on e-Science and Grid Computing (e-Science 2005), Melbourne, Australia, Dec. 5-8, 2005. http://doi.ieeecomputersociety.org/10.1109/E-SCIENCE.2005.77

[3] D. F. McMullen, K. Huffman, "Connecting Users to Instruments and Sensors: Portals as Multi-user GUIs for Instrument and Sensor Facilities", In Proceedings of the Workshop on Grid Computing Portals held in conjunction with SC05 (GCE05), Seattle, WA, Nov. 18, 2005.

[4] J. Novotny, M. Russell, O. Wehrens, "GridSphere: a portal framework for building collaborations", Concurrency - Practice and Experience, 2004, 16(5), pp. 503-513.

[5] A. Abdelnur, S. Hepper, "Java Portlet Specification version 1.0", 2003, http://www.jcp.org/aboutJava/communityprocess/final/jsr168

[6] C. McClanahan, E. Burns, and R. Kitain, Eds. "Java Server Faces Specification version 1.1", 2004, Available http://java.sun.com/javaee/javaserverfaces

[7] OASIS Web Services Security TC, "WS-Security", OASIS, Available http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

[8] OASIS Security Services TC, "Security Assertion Markup Language (SAML)", OASIS, Available http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security