REMOTE ACCESS TO THE VESPERS BEAMLINE USING SCIENCE STUDIO*

D. Maxwell#, D. Liu, E. Matias, D. Medrano, CLS, Saskatoon, Canada
M. Bauer, M. Fuller, S. McIntryre, J. Qin, UWO, London, Canada

Abstract

Science Studio is a web portal, and framework, that provides scientists with a platform to collaborate in distributed teams on research projects, and to remotely access the resources of research facilities located across Canada. The primary application for Science Studio is to provide scientists with remote access to the VESPERS beamline at the Canadian Light Source synchrotron in Saskatoon Saskatchewan, and to readily process data from this beamline at the SHARCNET high performance computing facility in London Ontario. The VESPERS beamline is a complex instrument that is composed of many devices, such as valves, motors and detectors, which are all controlled through the low-level EPICS control system. Science Studio implements a simple, intuitive and functional web-based interface to the beamline for device control and data acquisition. The Science Studio experiment management system allows the acquired data to be easily organized and shared with the research team. This paper will provide an overview of the design, implementation and capabilities of the Science Studio system, with a focus on remote control of the VESPERS beamline.

SCIENCE STUDIO OVERVIEW

The Science Studio web portal is mostly implemented in Java, and uses server-side web technology common to enterprise applications such as Java Servlets, Java Messaging Service (JMS), Java Database Connectivity (JDBC) and Java Server Pages (JSPs). In addition, many high quality open-source frameworks and libraries have been leveraged to build a highly functional web portal. The Spring [1] framework is used extensively throughout to build very robust and highly configurable servlets using the Model-View-Controller (MVC) architectural pattern. The iBATIS [2] Object-Relational Mapper (ORM) library is used to easily persist objects to a MySQL [3] relational database. The XStream [4] library provides fast object marshalling capabilities in both XML and JSON formats. Security functionality is provided by the JSecurity [5] framework using some custom extensions. Other Java libraries and tools include Apache Log4J [6], Apache Commons [7], Apache Tomcat [8], Apache ActiveMQ [9] and Jetty [10].

Data Model

Science Studio defines and implements a data model to capture the metadata associated with scientific research. Figure 1 is a data object relation diagram for this data model. The objects belonging to the experiment model have been indicated. A primary objective of Science Studio is to allow scientists, and other people, to collaborate; therefore an important part this data model is the *person* object. A *person* represents a user of the system and contains information such as their name, affiliation, email address and mailing address.

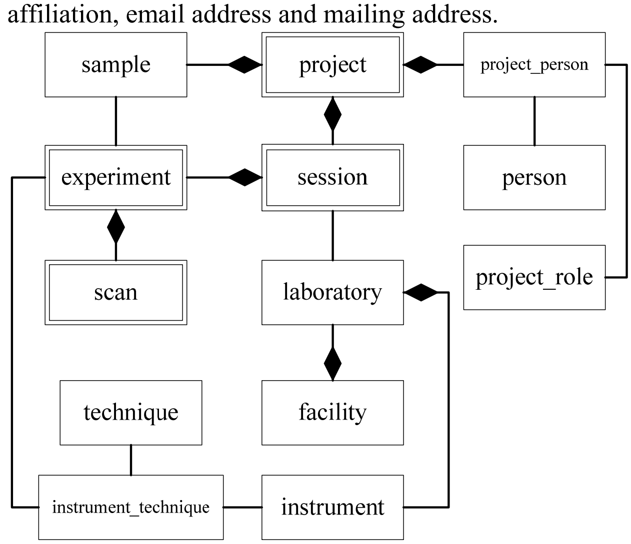

Figure 1: Data object relation diagram for the Science Studio data model, with the experiment model indicated.

Research projects are the foundation of experiment management in Science Studio. For that reason, the *project* object is the top-level organizational element for the hierarchical experiment model. A *project* is composed of *person*, *sample* and *session* objects. The collection of *persons* represents the people collaborating on a project, or simply a project team. A *sample* represents the physical specimen that is the subject of investigation for a project team. A *session* is composed of *experiment* objects and represents the reservation or allocation of resources to the project team for a specified time period. An *experiment* is composed of *scan* objects and references a *sample*, *instrument* and *technique* object. A *laboratory* is composed of *instruments* that are associated by location or function. An *instrument* references *technique* objects and represents a device or resource used to conduct an experiment. A *technique* represents the method or process used by an instrument to produce data. A *scan* represents the actual experimental data produced by an instrument, and contains information about its storage location and file format.

Standard Data Format

The manipulation of experimental data is a requirement for most scientific applications. Science Studio specifies a standard format for experimental data files to facilitate

*This work was funded by CANAIRE under R&D project NEP-01; and performed at the Canadian Light Source which is supported by NSERC, NRC, CIHR and the University of Saskatchewan.
#dylan.maxwell@lightsource.ca

the sharing of data between applications. This format is an extension of the more general purpose Common Data Format (CDF) [11]. The CDF is a self-describing format, with both binary and XML versions, used for efficient storage of scalar and multidimensional data. The standard format defines an overlying structure for the CDF that provides more information about the type of experimental data contained within a data file. Science Studio implements utilities for reading and writing files in the standard format, as well as, a framework for building custom data format converters.

Security

Science Studio provides security features such as authentication, authorization and session management. A web application implements shared services for user authentication such as the login and logout pages. This is indicated in Figure 2 by the Login Servlet. Security session management is handled mostly by JSecurity using a customized servlet filter. A servlet can be easily configured to use this filter, which will only allow access to authenticated users.

Authorization is provided using a project-oriented permission system. The members of a project team are associated with a project role. The project role determines the permissions that each team member has within the project. Currently only two project roles are used: Experimenter and Observer. Experimenters have full access to the project. They are permitted to create, read, edit and delete data objects belonging to the project. Observers are only permitted to read data objects belonging to the project. These permissions also apply to remote access. Experimenters are permitted to control the remote instrument, and Observers are only permitted to view the remote instrument.

Web Portal

Science Studio implements an extensible web portal that gives users a single, consistent entry-point for access to other services. This rich web interface is built using the Ext [12] JavaScript framework. In Figure 2, the server-side of this web application is indicated by the Core Servlet. A primary feature of the web portal is the ability for users to browse the data model. The data model is represented as data trees with *projects* as the roots, and *scans* as the leaves. Users can navigate to data objects using the tree, which will then provide different options based on the data object type. For example, selecting a *scan* allows users to view the experimental data, or selecting a *session* allows them begin remote access. Users can also create, edit and delete data objects, provided they have the required permissions.

VESPERS REMOTE ACCESS

The VESPERS beamline is located at the Canadian Light Source (CLS) synchrotron [13] in Saskatoon Saskatchewan. VESPERS is a microprobe beamline that operates in the energy range of 6 to 30keV using bending magnet radiation. The experimental station is equipped with both a CCD area detector and a four element Silicon Drift Detector (SDD). Together they are capable of multiple complimentary techniques such as X-Ray Diffraction (XRD) and X-Ray Fluorescence (XRF) spectroscopy.

XRD is a common technique used for determining the microcrystalline structure of geological samples. This technique uses the CCD detector to record the diffraction pattern produced by a sample when exposed to a focused x-ray beam. The CCD detector image size is 2084 x 2084 pixels or approximately 8MB. For an area of interest that

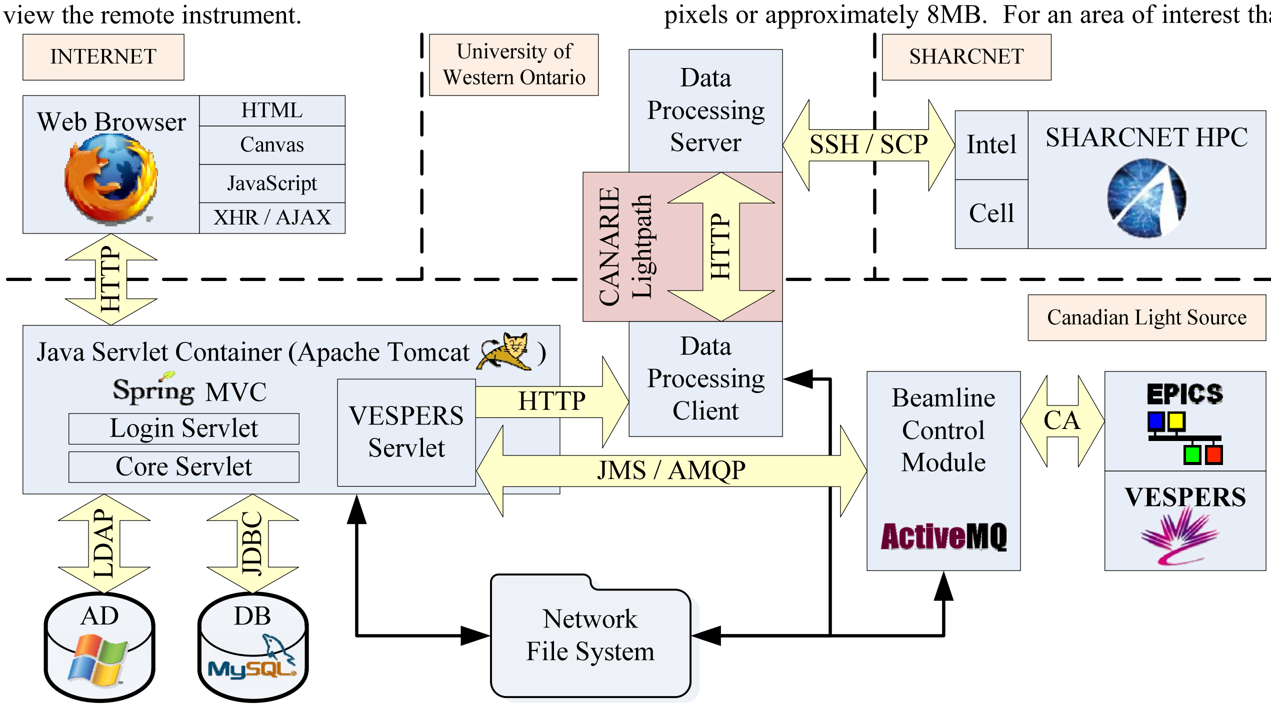

Figure 2: Science Studio architecture for VESPERS beamline remote access and data processing.

is 100 x 100 data points, the total size of the data set is approximately 80GB. A data set of this size requires many hours to process using conventional computers with standard software. However, using the computers at the SHARCHET [14] High Performance Computing (HPC) center, in conjunction with special software, this large data set can be processed in minutes. The SHARCNET HPC center is located at the University of Western Ontario (UWO) in London Ontario.

Science Studio allows users to remotely access the experimental capabilities of the VESPERS beamline, and then to readily utilize the computational capabilities of SHARCNET. Shown in Figure 2 is an architectural diagram of the main components, and their interaction, for the remote access and data processing systems.

EPICS Control System

EPICS [15] is the standard control system at the CLS and is used for control and data acquisition of nearly every device at the facility. The Channel Access (CA) protocol is used to communicate with EPICS over the network.

Beamline Control Module

The Beamline Control Module (BCM) is a Java application which provides a high-level interface to the EPICS control system. The BCM communicates with EPICS using a Java implementation of CA to monitor and change the state of devices on the VESPERS beamline. The BCM provides a device abstraction so that alternate low-level control systems can be used. This is important for use of the BCM outside of the CLS.

Web Application

The VESPERS beamline web application provides a user interface for device control and data acquisition. The Ext JavaScript framework is again used to build a rich interface that uses asynchronous requests to provide frequent (normally once per second) updates to the device information. This web application allows the user to interactively explore the sample, and then define a scan area by simply drawing a rectangle on the sample image. When the user starts a scan they are prompted to enter a name for the scan. The progress of a scan is displayed numerically, as the percentage complete, and graphically, as an animated dot that moves across the scan area. The user can also configure and test both the SDD and CCD detector. The web application also gives access to three video cameras, with pan, tilt and zoom capability, that show various views of the experimental station. Although all members of the project team, who have the Experimenter role, are permitted to control the beamline, only one user at a time is allowed to be in control of the beamline. In Figure 2, the server-side of this web application is represented by the VESPERS Servlet.

Data Processing Service

The raw data collected on the VESPERS beamline must be transferred from CLS to UWO for processing, and the processing results must to be transferred back to CLS for presentation to the users. In order to fully take advantage of the CANARIE Lightpath high-speed connection between the CLS and UWO, the File Transfer Server (FTS) and File Transfer Client (FTC) provide the following features:

- Simultaneous TCP connections.
- Start transferring multiple files with one request.
- Asynchronous client using non-blocking I/O.
- File content compression using *gzip* [16].
- File range transfer over multiple connections.

The File Monitor Service (FMS) provides file system event notification through HTTP. It is difficult to get I/O event notification from the CLS data acquisition system, by which experimental data is collected on the VESPERS beamline. Providing pseudo-realtime processing of experimental data requires that each piece of data be transferred to UWO once it is available on the CLS file system. This service can be deployed on any system with native support for *inotify* [17], and that has the *inotify-java* [18] library installed.

The basic sequence of events for the Data Processing Service is shown in Figure 3. The VESPERS Servlet first sends a request to the FMS to initiate monitoring of a specified directory. The notifications for these events are sent directly to the FTC, which then initiates the transfer of the experimental data files from the FTS.

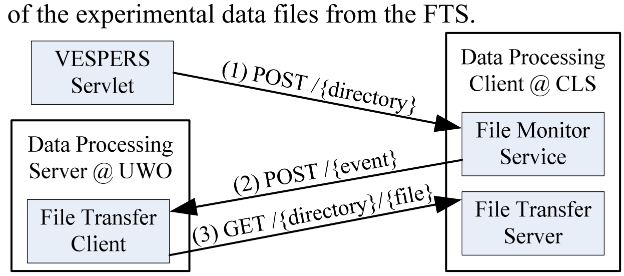

Figure 3: Event sequence for the Data Processing Service.

REFERENCES

[1] Spring <http://www.springsource.org/about>
[2] iBATIS <http://ibatis.apache.org>
[3] MySQL <http://www.mysql.com>
[4] XStream <http://xstream.codehaus.org>
[5] JSecurity (renamed to Apache Shiro) <http://incubator.apache.org/shiro>
[6] Apache Log4J <http://logging.apache.org/log4j>
[7] Apache Commons <http://commons.apache.org>
[8] Apache Tomcat <http://tomcat.apache.org>
[9] Apache ActiveMQ <http://activemq.apache.org>
[10] Jetty <http://jetty.codehaus.org/jetty>
[11] Common Data Format <http://cdf.gsfc.nasa.gov>
[12] Ext JS <http://www.sencha.com/products/js>
[13] Canadian Light Source <http://www.lightsource.ca>
[14] SHARCNET <https://www.sharcnet.ca/my/front>
[15] EPICS <http://www.aps.anl.gov/epics>
[16] gzip <http://www.gzip.org>
[17] inotify <http://inotify.aiken.cz>
[18] inotify-java <https://launchpad.net/inotify-java>