Fermilab FERMILAB-Conf-01/305 November 2001

An Electronic Logbook for the HEP Control Room

G. Roediger[1], *P. Pomatto*[1], *J. Kyriakopulos*[1], *S. Panacek*[1], *P. Canal*[1],*A. Kubarovsky*[1]

[1] Fermi National Laboratory, USA

Abstract

The Control Room Logbook (CRL) is designed to improve and replace the paper logbooks traditionally used in the HEP accelerator control room. Its features benefit the on-line coordinator, the shift operators, and the remote observers. This paper explains some of the most attractive features for each of these roles.

The features include the ability to configure the logbook for the specific needs of a collaboration, a large variety of entry types an operator can add by simply clicking and dragging, and a flexible web interface for the remote observer to keep up with control room activities.

The entries are saved as UTF-8 based XML files, which allowed us to give the data structure and meaning such that it can easily be parsed in the present and far into the future. The XML tag data is also indexed in a relational database, making queries on dates, keywords, entry type and other criteria feasible and fast.

The CRL is used in the D0 control room. This presentation also discusses our experience with deployment, platform independence and other interesting issues that arose with the installation and use of the logbook.

Keywords: Logbook, Control Room, JAVA, XML

Introduction

Since the Control Room Logbook was deployed in the D0 control room in December, 2000, the shift operators have entered 9000 entries, with a current daily average of 150. The entries contain graphs, screen shots, output (logs) from external processes, check lists, forms, and formatted text. The CRL has become an indispensable part of the daily life in the D0 control room.

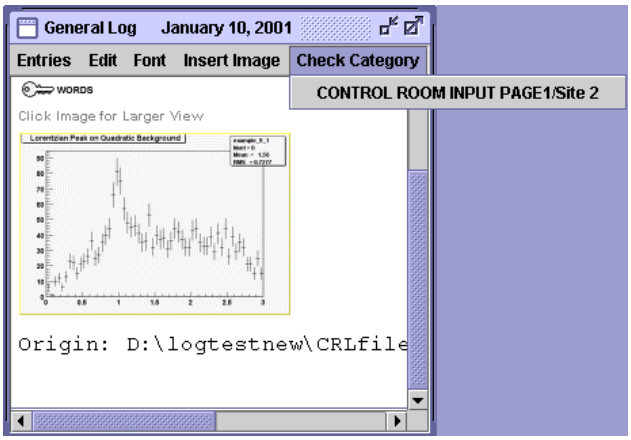


Figure 1: a ROOT histogram in a CRL entry

Outside the control room, D0 collaborators are keeping up by reading the entries on the web using keywords and date/time queries to display the relevant entries.

Furthermore, Minos is installing the CRL and plans to use it during the far detector installation. CMS, MiniBooNE, and BTeV have made positive evaluations and plan to use the CRL.

The CRL is very attractive for these users thanks to its flexibility, long list of features, and ease of use.

The Control Room Logbook (CRL) is designed to improve and replace the paper logbooks traditionally used in the HEP accelerator control room. Its features and configurability benefit online coordinators, shift operators, and remote collaborators. As an electronic logbook, it facilitates the entry of control room information so that more of the information needed by an experiment can be logged than what is normally put into a paper logbook. It also provides easier retrieval mechanisms based on the keywords and categories set up by the experiment's logbook administrator. The logbook is based on platform-independent technologies, such as XML, Java, and the web, that will endure throughout the life of an experiment and provide easily accessible, permanent storage of critical control room information. Gary Roediger and Pat Pomatto developed the Control Room Logbook as an HEPNRC project. As a development team, they are currently joined by members of the Fermilab Computing Division's Computational Physics Department.

Features

The CRL benefits the online coordinator by its highly configurable features. Since each experiment has different needs, the CRL desktop is almost entirely configurable by the logbook administrator. This desktop includes tabbed panes, each with their own set of menus at the top and toolbars on the right-hand side. In addition, each station in the control room can have its own unique set of desktops defined. For example, the Data Acquisition station may have separate pages defined for DAQ, store, and run while the Detector station has its own pages, menus, and toolbars. Each log entry created on a page will have its own pre-defined keywords to facilitate easy retrieval by keyword or category at a later time. For specific situations and needs the operator can add additional keywords to the log entry. A log-in/log-out facility provides for signing of entries when they are archived by an operator. The online coordinator can choose to run the logbook on Windows, Linux, or Unix (Sun Solaris). The database used for faster retrieval of entries can be installed on another system, if desired, and use either MySQL or Oracle. Microsoft Access database support will be added soon.

The CRL benefits the control room operator by providing a Graphical User Interface that facilitates his/her choice of the kinds of entries, such as text, pictures and binary files. For example, the jpeg image of a histogram can be added to and displayed by the logbook. The CRL also supports the Cross IPEN, which allows the operator to draw on the IPEN to enter free-hand notes or figures into the window of the CRL. In addition, the CRL can directly log information originating from a running program thanks to a simple, XML based, socket interface.

The CRL provides read access to logged entries via any web browser for remote observers. This kind of access is virtually impossible for a paper logbook to provide. The remote observer has the capability of retrieving entries based on the same criteria as the operators: keywords, categories, date/time, operator, entry type, and word search.

Configurability

The CRL is designed to conform to each experiment's needs as opposed to providing a fixed configuration requiring the collaboration to conform to a logbook. In the CRL this is accomplished through an XML based configuration file. This file defines the experiment's "view" of the information being logged. The XML file allows an administrator the ability to define the desktops available to the operator. For each configured desktop, he can also define the containers used for cataloging the entries as they are inserted into the logbook. The containers are selected from a cascading menu bar on each desktop. It is this cascading menu bar that can be defined in the XML configuration file. In addition to the containers the XML file also allows the administrator to specify the multimedia entry types available to the operator on each desktop. These entry types are the essence of the information being logged. The configured entry types appear as a button on a vertical toolbar along the right edge of the desktop. The operator can simply drag one of these entry types into a container to start the logging of information. Keywords may be associated with desktops, containers, entry types, or the overall logbook. As an entry is being made the operator can chose any

of the valid configured keywords to be associated with the entry. These keywords can later be used in a relational query to retrieve entries.

The CRL is designed to be extended to meet the growing needs of collaborations. There are currently 4 types of containers used to add and retrieve entries. The container type associated with a menu item on the desktop is specified in the XML configuration. A standard container is used for the simple case of adding entries through the drag and drop method. A scheduled container type is used for adding entries on a periodic basis much like Unix cron jobs. The only difference between adding entries one at a time or periodically is the type of container the entry is dropped into. An example of a periodic entry would be a live picture from a web cam. The 2 remaining types of containers currently provided are for reporting of existing entries. The report container holds the results of relational queries and/or specifically selected entries. The thread container holds the results of a thread of interest. Threads can be compared to a newsgroup. The named thread can hold any number of entries and any entry can belong to multiple threads.

There are 6 types of entries currently provided. The entry types are not built into the logbook but are rather configured to a desktop via the XML configuration. An abstract base class is provided from which all new entries are inherited. The CRL entry types can be extended by creating new multimedia entries class, which extend the base class and configuring them to appear on a desktop toolbar through the XML configuration.

The existing types include the following collaboration independent types: standard text, images (from files or URLs), arbitrary binary files, forms, and execute which runs any arbitrary command and logs its standard output. An example of a collaboration specific entry is the entry type used to retrieve images from the Fermilab Accelerator Division image database and log that image.

The process of creating an entry involves selecting a toolbar button and dragging it into a desktop container. After editing the entry the operator can associate his/her name with the entry and archive it. The archival process creates 3 separate items, An XML file representing all information within the entry, an SQL database update containing only partial entry information required for relational queries and an HTML files for rapid display of entry information from the Web

Java reflection is used to interrogate the entry to determine what instance variables are required to save the entry. This relies on the entry being coded as a JavaBean with a specified property descriptor table. Simple instance variables such as int, char, and String are converted to an XML form: *<name>value</name>*

More complex items such as a JTextPane require extending the object type in a package called xmldatatypes. These extended objects e.g. XMLJTextPane, have methods included in their property descriptor that can create a XML Document Object Model (DOM) or read in a DOM and instantiate the object. This process is carried out recursively for an entry thus producing an XML representation for an object made up of complex and simple object types. The use of a property descriptor table prevents pure reflection from breaking an entry into its atomic parts for objects that are not controlled by future modifications to the entry. An example of this is an entry, which may contain a JPanel. Reflection applied recursively would produce hundreds of component parts which may even change in a future release of JPanel. Since the JPanel would not be needed to reconstitute an entry and could always be derived, it should not be saved as XML. Providing a property descriptor in a BeanInfo file for complex objects clearly defines how far reflection and introspection will go in determining what must be saved as XML. The resulting XML has all the entry information embedded with no external references.

Web interface

The web interface to the CRL is implemented using Java Server Pages. This technology allows the embedding of Java code inside an HTML web page, which lets us re-use a lot of the code of the main CRL. For example, queries that have been created and saved using the main graphical interface can be re-used to create web searches.

Since web browsers can recognize a large variety of files using their MIME type, the web interface gives convenient access to the binary files that have been saved. Pictures are directly displayed. Direct links to the binary files that need an external viewer are provided so that the user can either directly start the viewer or save the file on his local machine.

In a control room logbook, it is important to insure the integrity of the information. For this reason, the addition of new log entries is often restricted to the persons that are physically inside the experiment's control room. However, remote observers sometimes wish to be able to comment on a particular log entry. The CRL provides the ability for the remote user to annotate the entries. These annotations can be created either locally or remotely and are carried by the entry as addenda that will be seen each time the entry is retrieved.

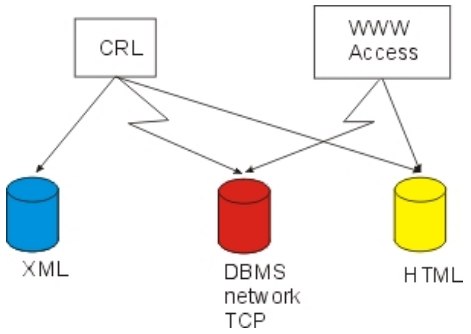
Data Storage

Often one of the legal requirements of an experiment's logbook is for the information to be available for 10 years after the end of the experiment. In order to support this requirement, the CRL is using an XML based file format for its main data storage.

XML is an international standard defining a method for storing structured data in a text file. XML is license-free, platform-independent and well supported. These qualities of the XML format insure that the main data of the CRL will still be retrievable even if the CRL software itself or any of the products it uses become unavailable. XML is supported by a wealth of software tools that makes it easy for a third party to develop software taking advantage of the CRL's XML entry files.

All the information needed for retrieval, searching, indexing and displaying is available inside the XML entry file. However, this is not the most efficient format with which to do searches. In order to improve the search speed, the CRL uses an auxiliary SQL database to index the most used data fields. The CRL has been tested using Oracle, MySQL and postgreSQL and should work with any SQL compliant database.

Also, to simplify and optimize entry retrieval via a web server, the HTML version of the entry file is produced in parallel with the creation of the XML file. In addition this gives the user more flexibility and potential security on the location of the files. For example, the XML might be saved on a secure and restricted-access disk while the HTML version might be saved on world readable (via a web server) disk.



Deployment.

The CRL was written in JAVA and is very easy to port to new platforms. During each port, most of our time so far has been spent in properly installing the products the CRL depends on. These include the MySQL database, the RS232 device driver for the electronic pen and software such as Apache and Tomcat, which enable Java Server Pages and Servlets to be run in a web server.

To simplify the installation task, we now provide an InstallShield™ executable. When run, this executable installs all the components of the CRL including the device driver for the Ipen and the SQL database that are needed for control room use on the target platform.

The platforms currently supported are: Windows, Linux and Solaris, which can be downloaded from: *http://www-cpd.fnal.gov/CRL*. Using InstallShield Multi-Platform Edition allowed us to quickly provide the installation executable for all 3 platforms in 2 possible modes: Complete installation or Update of an existing installation.

Before installing the CRL, it is necessary to install Java 1.3. Running the CRL also requires an SQL database, such as MySQL, Oracle or PostgreSQL to be installed and configured. Last but not least, to enable web access to the CRL content, you will need to install a web server and a JSP implementation such as Tomcat.