Electronic Collaboration Logbook

Suzanne Gysin, Igor Mandrichenko, Vladimir Podstavkov, Margherita Vittone
FNAL, P.O. Box 500, Batavia, IL 60510, USA

Abstract. In HEP, scientific research is performed by large collaborations of organizations and individuals. The logbook of a scientific collaboration is an important part of the collaboration record. Often it contains experimental data. At Fermi National Accelerator Laboratory (FNAL), we developed an Electronic Collaboration Logbook (ECL) application, which is used by about 20 different collaborations, experiments and groups at FNAL. The ECL is the latest iteration of the project formerly known as the Control Room Logbook (CRL). We have been working on mobile (IOS and Android) clients for the ECL. We will present the history, current status and future plans of the project, as well as design, implementation and support solutions made by the project.

1. Project History

This project began at FNAL as Control Room Logbook (CRL) for D0. It was a Java-based, non-web based application, which kept logbook entries on the local disk of the computer where the application was deployed. After a while, a web GUI was added on top of the data storage layer. In this architecture, the product gained about 20 different users at FNAL, including collaborations such as MINOS, Minerva, as well as smaller groups of scientists.

In 2010 the program was completely rewritten as a web-based database application. The main objective of this effort was to store all the data into the database instead of a file system. Also, the project got a new name – Electronic Collaboration Logbook (ECL), to reflect the fact that majority of its users do not actually use it in a control room, and it is not oriented to support control room operations.

All the data from old CRL instances were ported into ECL databases. Currently we have 24 instances of the ECL. Most of them are running on servers, maintained by the Computing Sector, but there is an instance run by the DES collaboration outside of FNAL.

2. Concepts

The logbook is a collection of logbook entries. Each entry has an author and a timestamp. An entry can be a simple text or a filled form. An entry can have any number of images or documents attached to it. Each entry belongs to one and only one category.

Forms are created by the ECL administrator using either the ECL form builder module, or by uploading an HTML form into the system. Forms are used to define entries with a standard format, e.g. begin or end shift entries.

Categories are defined by the ECL administrator and organized into a tree structure.

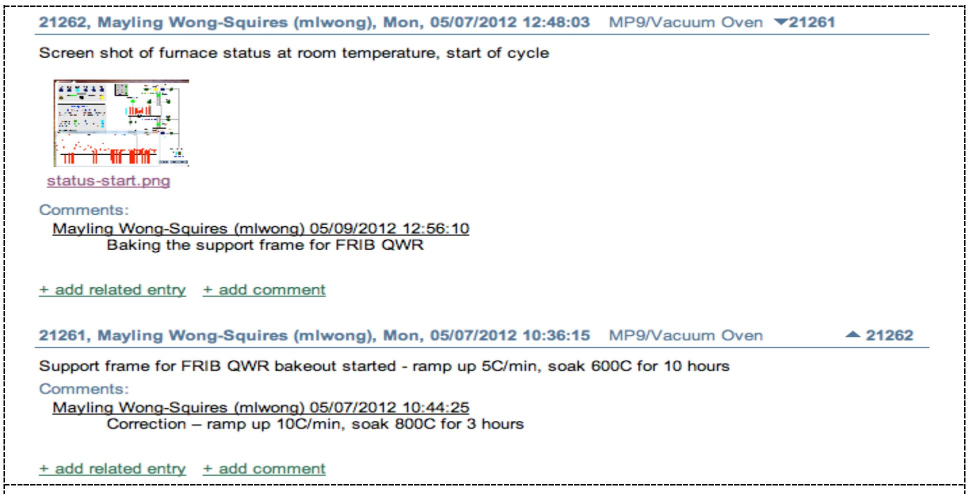
An entry can have zero or more tags attached to it. A tag is an arbitrary text string defined by the ECL administrator.

Once an entry is committed to the database, the entry content can not be modified. There are 2 ways to amend the content of an entry.

- Comments – a user can add comments to an existing entry. A comment is a piece of text and it has its own author and timestamp.
- Related entries – full featured entries with a reference to an older entry. The user can follow the links to the related entries, or show all inter-related entries as a thread.

Optionally, TexTile formatting can be used to format text in the entries.

Figure 1 shows a thread of 2 related entries. First entry has an image attached to it and you can see th eimage's thumbnail. Both entries have comments.



21262, Mayling Wong-Squires (mlwong), Mon, 05/07/2012 12:48:03 MP9/Vacuum Oven ▼21261

Screen shot of furnace status at room temperature, start of cycle

status-start.png

Comments:
Mayling Wong-Squires (mlwong) 05/09/2012 12:56:10
Baking the support frame for FRIB QWR

+ add related entry + add comment

21261, Mayling Wong-Squires (mlwong), Mon, 05/07/2012 10:36:15 MP9/Vacuum Oven ▲ 21262

Support frame for FRIB QWR bakeout started - ramp up 5C/min, soak 600C for 10 hours
Comments:
Mayling Wong-Squires (mlwong) 05/07/2012 10:44:25
Correction – ramp up 10C/min, soak 800C for 3 hours

+ add related entry + add comment

Figure 1. Example of a thread of related logbook entries

3. Features

3.1. Search

One of many important advantages of storing all logbook information in a database is the ability to search data by many different criteria. The ECL allows searching data by:

- Category
- Form
- Author
- Date range
- Tags
- Presence of attachments

- There are 3 types of text search
    - Exact substring search
    - Text expression search, e.g. expression "(rat | crab) & !cat" will find all entries with either word "rat" or word "crab" but not word "cat" in them.
    - Ranked text, e.g. "apple orange" will rank all entries by presence and proximity of words "apple" and "orange" and list entries according to the rank.

Users can create "saved searches" by saving a set of search criteria under an arbitrary name. Regular users create saved search for themselves only, but an administrator can create a "global" search, available to all users. Named searches have bookmarkable URLs so one can add such a URL to their bookmark list and rerun the search from one's browser.

3.2. User Authentication and Authorization.
The ECL uses 2 user authentication mechanisms – a "local" username/password pair and LDAP [1]. Passwords are sent over a secure HTTPS connection. Once the user is authenticated via one of the 2 methods, the authorization step determines permissions to access ECL data. There are 3 types of ECL users:

- Active user - can post new entries or add comments
- Inactive user - can only read existing entries
- Disabled user - can not log in to the sysrem.

An ECL instance can be either private or public. If the instance is public, unauthenticated user can read entries. A private instance is protected from unauthenticated access.

Table 1. ECL User Authorization.

ECL	**Unauthenticated**	**Disabled**	**Inactive**	**Active**
Public	read only	read only	read only	read,post,comment
Private	no access	no access	read only	read,post,comment

3.3. Subscriptions.
An ECL user can receive notifications about new entries in two ways.

- **A user can subscribe to e-mail notifications about entries posted into certain category or with a certain form. The user will receive one e-mail message per new entry.**

- **There is a set of RSS feeds, which can be used to receive notifications about new entries. The set includes feeds for every category, tag or form.**

3.4. XML/HTTP API.
There is a XML API which can be used to post or retrieve entries over HTTP/HTTPS connection. Currently, the API has 3 functions:

- Search entries by given set of criteria – returns list of entry ids
- Get an entry by entry id
- Post a new entry

XML API has 2 alternative authentication methods:

- Username and password sent over HTTPS connection
- Digital signature calculated over a combination of HTTP request body, random "salt" string, and user password, sent along with the request

Architecture
The ECL is a web-based application. It is written in Python 2.6 [2]. It is built using a Django [3] framework and a Jinja2 [4] templates package. It uses the PIL [5] module for image handling and Textile [6] for entry formatting. It uses PostgeSQL [7] version 9 as the database engine. The only reason why it uses this specific version is because starting in version 9, Postgres includes a text indexing [8] module, used by the ECL, as a standard feature. The ECL runs under Apache httpd [9] through mod_wsgi [10].

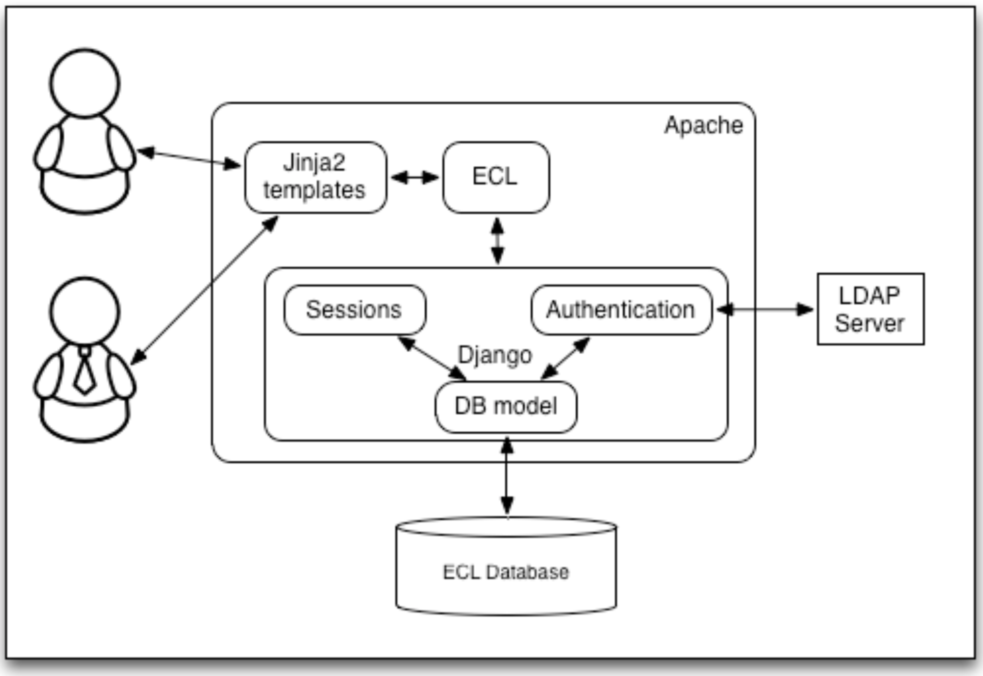


Figure 2. ECL Architecture

All entry data, including attached images and documents, are stored in the database. Storing everything in the database allows us to use single backup/recovery mechanism for ECL data and run multiple redundant instances of web interface to increase service availability (Fig. 3). Currently, we run 2 instances of the ECL web interface, but that number is completely arbitrary. No configuration changes are necessary to add, temporarily shut down or permanently remove a web server instance.

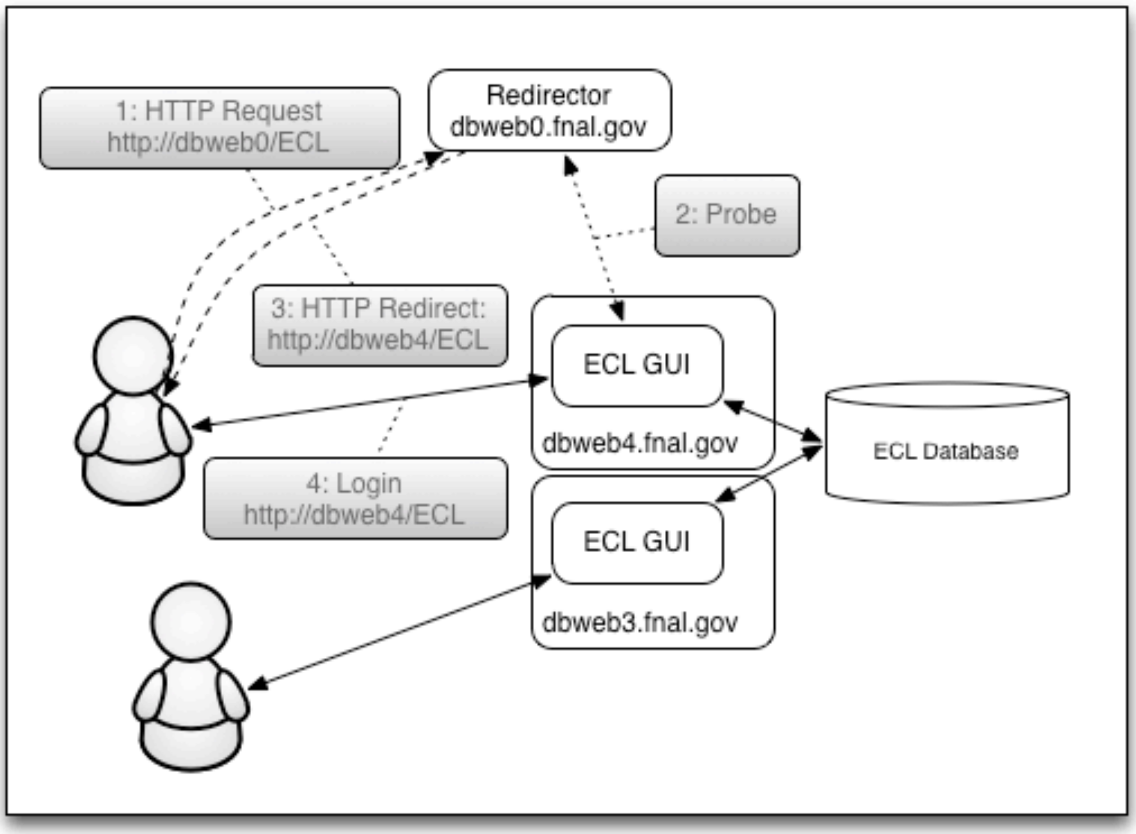


Figure 3. Figure with short caption (caption centred).

5. Mobile Application

We have developed an iOS ECL client aplicatiuon, which is capable of posting an entry to the logbook and attach one or more pictures taken by the mobile device. The application uses the XML/HTTP API to connect to the web server. It uses the same authentication method as the web client – password sent over HTTPS. We are working on Android application.

6. Shift Scheduler

The latest addition to the project is the Shift Scheduler. The Shift Scheduler allows the collaboration to define and keep track of its shift schedule. The schedule is defined in terms of schedule intervals. Each interval has a name and beginning and end date.

A shift is defined as a time interval repeating over a set of consecutive weekdays. A shift repeats itself every week for the duration of the scheduling interval. For example, a shift can be defined as 9am to 5 pm every Monday, Tuesday and Wednesday and repeat itself for the interval from January 1st to March 31st.

A shift has one or more roles, e.g.: Shift Captain, DAQ expert. When a person signs up for a shift, they sign up for specific role in the shift for the given week. Each role has certain number of points associated with it. When an individual signs up for a shift role, he earns associated points for himself and for his institution. The system provides reports, summarizing point collection for institutions and individuals.

The Shift Scheduler allows users to swap their shift assignments. Also, if the assignment is sufficiently far in the future, the individual can sign off a shift he signed up for previously. If the shift is too near in the future, only the administrator can cancel the shift assignment.

The Shift Scheduler shares the collaboration members database with the Logbook.

7. Project Status and Future Plans

Currently we have 24 instances of ECL running in production, used by collaborations and groups of various sizes. Our users include experiments Minerva, NOvA, DES, MINOS. The majority of instances run on common hardware – web and database servers. However, some instances use their own resources. NOvA and Minerva use their own database server, but common web server. One of DES instances runs ECL on their own servers.

Currently, we are primarily in a stable support mode without any major development planned, except in the area of mobile clients. In this area, we are planning to continue working in several directions:

- Expand mobile client functionality to include logbook searching/browsing
- Continue working on an Android client
- Develop the client for a larger tablet format
- Add entry geo-tagging functionality
- Export Shift Scheduler information as a calendar for mobile devices

Another area of project development is further integration of the Logbook and the Shift Scheduler. We can introduce an association between shifts and logbook entries and have cross-links between logbook entries and shift information, such as list of people who were on shift at the time the entry was made.

We are also working on packaging the application in a more suitable way for deployment outside of FNAL.

8. References

[1] http://tools.ietf.org/html/rfc4510
[2] http://www.python.org
[3] http://www.djangoproject.com
[4] http://jinja.pocoo.org
[5] http://www.pythonware.com/products/pil
[6] http://www.textism.com/tools/textile/
[7] http://www.postgresql.org
[8] http://www.postgresql.org/docs/current/static/textsearch.html
[9] http://httpd.apache.org/docs/2.0/programs/httpd.html
[10] http://code.google.com/p/modwsgi/

Acknowledgments

Authors wish to acknowledge assistance and encouragement from colleagues and valuable input from logbook and shift scheduler users.