

## MIT Open Access Articles

*The CMS electronic logbook*

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

**Citation:** Bukowiec, Sebastian et al. "The CMS Electronic Logbook." 2010 17th IEEE-NPSSIEEE Real Time Conference (RT), 2010. 1–3. © Copyright 2010 IEEE

**As Published:** <http://dx.doi.org/10.1109/RTC.2010.5750482>

**Publisher:** Institute of Electrical and Electronics Engineers (IEEE)

**Persistent URL:** <http://hdl.handle.net/1721.1/76609>

**Version:** Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

**Terms of Use:** Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



# The CMS Electronic Logbook

Sebastian Bukowiec, *Member IEEE*, Gerry Bauer, Barbara Beccati, Ulf Behrens, Kurt Biery, James Branson, Eric Cano, Harry Cheung, Marek Ciganek, Sergio Cittolin, Jose Antonio Coarasa Perez, Christian Deldicque, Samim Erhan, Dominique Gigi, Frank Glege, Robert Gomez-Reino, Derek Hatton, Yi Ling Hwong, *Member IEEE*, Constantinos Loizides, Frank Ma, Lorenzo Masetti, Frans Meijers, Emilio Meschi, Andreas Meyer, Remigius K. Mommsen, Roland Moser, Vivian O'Dell, Luciano Orsini, Christoph Paus, Andrea Petrucci, Marco Pieri, Attila Racz, Olivier Raginel, Hannes Sakulin, *Member IEEE*, Matteo Sani, Philipp Schieferdecker, Christoph Schwick, Dennis Shpakov, Michal Simon, *Member IEEE*, Konstanty Sumorok and Andre Sungho Yoon

**Abstract**– The CMS ELogbook (ELog) is a collaborative tool, which provides a platform to share and store information about various events or problems occurring in the Compact Muon Solenoid (CMS) experiment at CERN during operation. The ELog is based on a Model–View–Controller (MVC) software architectural pattern and uses an Oracle database to store messages and attachments. The ELog is developed as a pluggable web component in Oracle Portal in order to provide better management, monitoring and security.

## I. INTRODUCTION

A logbook was originally a book for keeping the data from the log, a navigation tool used by mariners to estimate the speed of a vessel through water. Currently the term logbook is widely used in science and industry for recording various data on complex machines such as nuclear plants or particle accelerators. The reason for using a logbook remains the same, however the technology which is used has changed – now a computer based electronic logbook is used instead of paper.

The CMS [1]-[2] high-energy physics experiment, part of the Large Hadron Collider (LHC) at CERN has its own Electronic Logbook.

The Elog fulfills an important role as a tool to exchange opinions and store information about the experiment, problems and solutions, which are taken to solve them

allowing for a better understanding of the detector and creating a platform for the future improvements.

The first version of CMS ELogbook was an open source electronic logbook developed by Stefan Ritt [3]. This stand-alone application was file based and due to its file based nature, the searching did not meet our requirements. As a result, it was decided to write a new, dedicated Elog version for the CMS experiment, while keeping the same look and feel as the old one. This new version of the tool has the same interface making it easily recognized and accepted by the users.

The new Elog uses Oracle technologies and is written in the Java, XHTML and JavaScript programming languages.

## II. ARCHITECTURE

The ELog is a portlet running on OracleAS Portal (Fig. 1) that provides implementation of the portlet container [4]-[5]. OracleAS Portal [6] is a component of Oracle Application Server that provides a solution for building, deploying, and maintaining enterprise portals. OracleAS Portal displays information in the form of portal pages. There are three elements that define a portal page: region, item and portlet. Region defines page layout and contains either portlets or items. Item is defined as a static content stored by the portal and a portlet is a dynamic, reusable, pluggable web component that can draw generated content from many different sources. As different portlets can be placed on a common page, the user receives a single-source experience. Portlets generate content, and the portal controls the appearance. There are four types of portlets in OracleAS Portal: prebuilt portlets, portal tools, partner portlets and programmatic portlets. The Elog is the programmatic portlet based on the PDK-Java API and the Struts framework. Along with the PDK-Java API, OracleAS Portal gives a possibility to develop portlets using either JSR 168 standard [7] or PL/SQL API.

The OracleAS Portal has been chosen because it is scalable, secure and gives the possibility to monitor and manage all data and users. It is also well integrated into the CERN infrastructure, since the authentication uses a single sign-on (SSO) solution. Building separated applications and placing them together, gives the user a powerful, consolidated interface for monitoring and reporting, which is available for the whole collaboration.

Manuscript received May 28, 2010. This research project has been supported by a Marie Curie Early Initial Training Network Fellowship of the European Community's Seventh Framework Programme under contract number (PITN-GA-2008-211801-ACEOLE) and in part by the DOE and NSF (USA) and the Marie Curie Program.

U. Behrens, D. Hatton and A. Meyer are with DESY, Hamburg, Germany.

B. Beccati, S. Bukowiec (corresponding author, phone: +41 22 767 8086, fax: +41 22 766 8124, email: Sebastian.Bukowiec@cern.ch), E. Cano, M. Ciganek, S. Cittolin, J. A. Coarasa Perez, C. Deldicque, D. Gigi, F. Glege, R. Gomez-Reino, Y. L. Hwong, L. Masetti, F. Meijers, E. Meschi, R. Moser, L. Orsini, A. Racz, H. Sakulin, C. Schwick, and M. Simon are with CERN, Geneva, Switzerland. P. Schieferdecker were with CERN, Geneva, Switzerland. P. Schieferdecker is now with Universität Karlsruhe, Karlsruhe, Germany.

J. Branson, M. Pieri and M. Sani are with University of California San Diego, La Jolla, California, USA. A. Petrucci was with University of California San Diego, La Jolla, California, USA. He is now with CERN, Geneva, Switzerland.

S. Erhan is with University of California, Los Angeles, California, USA and CERN, Geneva, Switzerland.

K. Biery, H. Cheung, R. K. Mommsen, V. O'Dell and D. Shpakov are with FNAL, Batavia, Illinois, USA.

G. Bauer, C. Loizides, F. Ma, C. Paus, O. Raginel, K. Sumorok and A. S. Yoon are with Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.

In summary, the key advantages of the use of portlets and OracleAS Portal are:

1. Portal container provided services;
2. Built in user customization;
3. Single sign-on mechanism;
4. Built in management system of users and data.

Considering user interface, jQuery and jQueryUI were used for handling more demanding requests on the client side and to provide more useful and modern user interface. jQuery is a fast and concise JavaScript Library that simplifies HTML document browsing, event handling, animation, and Ajax interactions for rapid web development [8]. jQuery UI provides abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets, which are built on top of the jQuery JavaScript Library and can be used to build highly interactive web applications [9].

As was said earlier, the new Elog is database oriented and therefore whole structure of the tool is defined in the database. The Elog uses Oracle database for storing the messages, and the attachments.

The integration of the application flow and business logic is not part of the portlet APIs. By using the Struts framework, it is possible to leverage the MVC architecture to create and publish applications within the portal. By applying the MVC to the ELog, core business model functionality was separated from the presentation and application logic. The Model represents the business or database code, the View represents the page design code and the Controller represents the navigational code. Such separation allows multiple views, clarifies the design and provides an efficient modularity that makes developing, testing and maintaining easier. Struts framework and MVC model is detailed in section III of this paper.

### III. STRUTS FRAMEWORK

Struts is one of the most popular frameworks for building Java-based Web applications based on the Model-View-Controller (MVC) design pattern which is responsible for organizing code [10]. It is a foundation for building an MVC application providing not only libraries but also utility tools, thereby making development faster and easier. As the name suggests MVC consists of three parts Model, View and Controller that creates a separation of concerns and provides an infrastructure for Web application [11]. In Fig. 2 the struts flow diagram is shown. The Model represents the business or database code and controls data access and persistence. The View represents the page design code, it controls how data is presented to the user and it collects user input. The Controller represents the navigational code, it validates input, and it handles data flow and transformation between Model and View. Beside separating the code, this makes simultaneous work of a group of programmers on the same project possible, enabling a better understanding of the application and making maintenance much easier. Struts also offers utilities that provides several useful features such as simple and complex validations in the Controller or custom tags in the View. Whole flow control of the application is declared in the struts-

config.xml file. Struts only places restrictions on the View and Controller. The developer is free to implement the Model portion in any way he wishes.

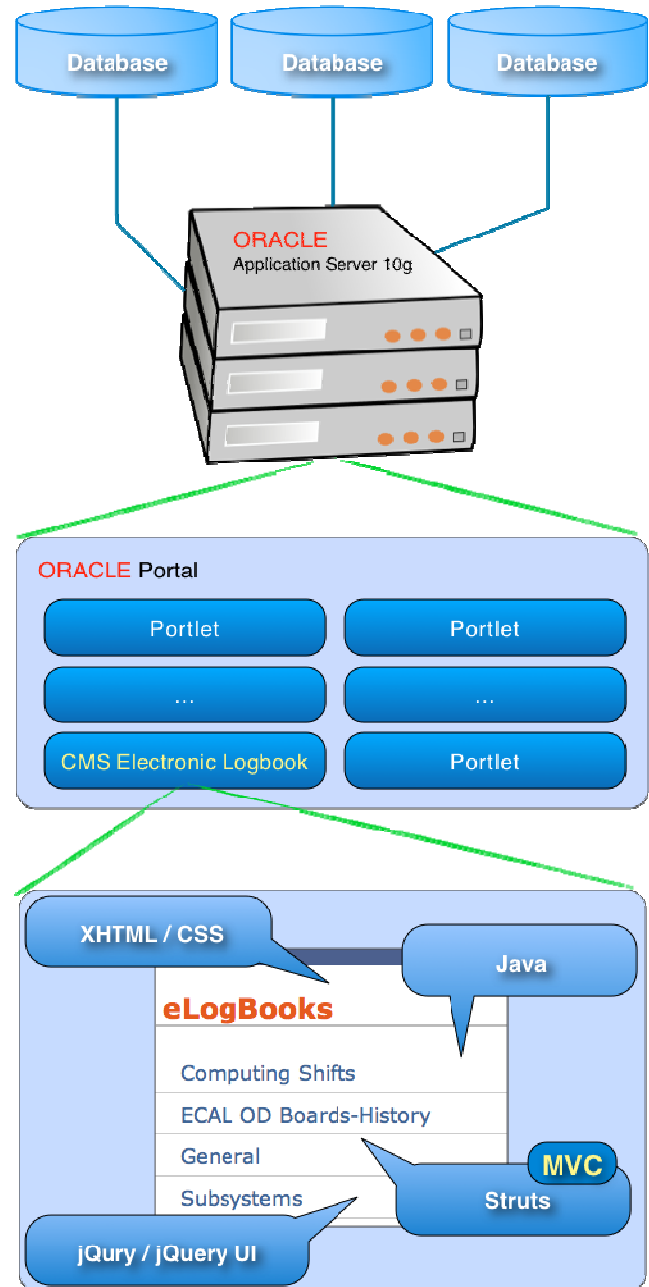


Fig. 1. Architecture overview.

### IV. FEATURES

The ELog organizes messages into categories, where each category may have unlimited levels of other categories and subcategories. Each subcategory has its own page where the entries are displayed as a list. Inside the list messages are organized in threads.

Four views are available for browsing: Full, Summary, Threaded and Expanded. Users can navigate page by page through the messages in the current subcategory or display all chosen subcategories together in a special view for a selected

time period (day, week, month). This view makes use of a special personalization mechanism provided by OracleAS Portal allowing the chosen subcategories to be saved in the profile of the user. Personal customization is done per user for each portlet separately. The mechanism is similar to the cookie mechanism in web browsers; the difference is that information is kept on the server side and is available for all browsers to which is logged in the user.

The data for personal customization can be kept as a file or in the database with the difference that data stored in the file is lost after each deployment of the portlet. OracleAS portal is fully responsible for accessing the data in the personalization, the programmer must only prepare the java bean with properties, setters and getters for them. File based personalization does not require any prerequisites while the database version requires a special table in the database.

The user can create a new message that can have attachments with an optional description. Two editors are available: rich and plain text editor.

The editing functionality allows only the author to edit the existing messages, while the reply functionality gives user the possibility to reply to any existing message inside a thread. The ELog also provides the feature to subscribe to chosen subcategories. Any update (new or edited message) in the subcategories is sent to the subscribers by e-mail. It also includes a search engine, where messages from different subcategories can be retrieved on the base of given filters.

Using message id it is possible to display the message directly without browsing the whole list. Additionally the ELog contains RSS Feeds.

## V. CONCLUSION

The ELog is extensively used by the entire CMS collaboration, more than 3000 scientists and engineers from 38 countries, as a tool for communicating and archiving information related to the commissioning and running of the experiment. Every week users create around 1500 messages.

The tool is open for the future extensions in order to meet the dynamically changing needs of users. The main goal of the developers is to prepare a reliable tool with a modern interface that allows users to easily find stored information based on a wide range of search options.

## ACKNOWLEDGMENT

The development team would like to give the credits to Stefan Ritt, the developer of the ELOG, for the Cascading Style Sheets (CSS) that we have been partially used in the CMS ELogbook.

We thank Omer Inam Alvi and Ad Aerts for developing the first version of the current CMS Elogbook.

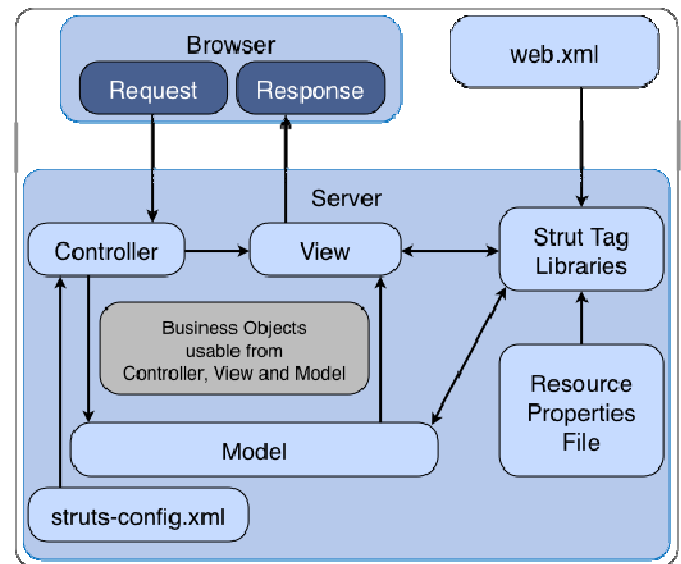


Fig. 2. Struts flow diagram.

## REFERENCES

- [1] The CMS Collaboration, CMS Technical Proposal, CERN LHCC 94-38, 1994.
- [2] The CMS Collaboration (R. Adolphi et al.), "The CMS Experiment at CERN LHC", JINST 3 S08004 p. 361, 2008.
- [3] ELOG Project, Stefan Ritt, <https://midas.psi.ch/elog>.
- [4] Oracle Database 10g, <http://www.oracle.com/technology/products/database/oracle10g/index.html>.
- [5] OracleAS Portal 10g, <http://www.oracle.com/technology/products/ias/portal/10g1014.html>.
- [6] OracleAS Portal 10g R2: Build Portlets with Java, Volume 1, Student Guide.
- [7] Java Portlet Specification (JSR 168), <http://www.jcp.org/en/jsr/detail?id=168>.
- [8] jQuery, <http://jquery.com>.
- [9] jQuery UI, <http://jqueryui.com>.
- [10] Struts, MVC architecture, <http://struts.apache.org>.
- [11] James Holmes, "Struts: The Complete Reference, 2nd Edition", pp. 3-10, McGraw-Hill Osborne Media, 2006.