ELisA: the ATLAS logbook facility extensions

Alina Corso-Radu[1], Giuseppe Avolio[2]

[1] University of California, Irvine, Department of Physics & Astronomy, 4129 Frederick Reines Hall, Irvine, CA 92697-4575, US

[2] CERN, Esplanade des Particules 1, 1211 Geneva 23, Switzerland

E-mail: alina.radu@cern.ch

Abstract. Information concerning the operation, configuration and behavior of the ATLAS experiment needs to be reported, gathered and shared reliably with the whole ATLAS community which comprises over three thousand scientists geographically distributed all over the world. To provide such functionality, a logbook facility, Electronic Logbook for the information storage of ATLAS (ELisA), has been developed and actively used since the beginning of the LHC Run 2 period. The facility includes a user-friendly web interface to browse activity logs and to report on system operations with a configurable email notification system; a RESTful API used programmatically by other tools and services of the data acquisition infrastructure, and a set of client API libraries and utilities to help user's interaction with the REST API. Given its generic configuration capabilities, the ELisA facility has been recently deployed as a stand-alone logbook for other projects such as the commissioning of different sub-detectors and the offline assessment of data-quality. To ease this operation and to potentially extend ELisA usage to other projects, an extension of the database backend support is being implemented to reduce the dependency on the ORACLE database for the logbook deployment. Also, the deployment process of the logbook is being improved. This contribution will present the status of the logbook facility as well as the extensions and improvements implemented to ease the logbook portability to other projects.

1. Introduction

ELisA (Electronic Logbook for the information storage of ATLAS) is a web tool used by the ATLAS experiment [1] at CERN LHC to keep track of the daily activities of the experiment's operations, commissioning and deployment work. The logbook is used by the system operators, experts, and automated services to record and share information. The logbook is highly configurable, and it comprises a web user interface, a REST API server, and a set of client libraries.

The logbook is in production since 2012, and it was exclusively used during LHC Run 2 operations after the previous logbook utility has been dismissed early 2014. Developed primarily as the experiment's operations logbook, users adopted it quite quickly as the logging facility for different standalone activities such as detector development and commissioning work. Thus, the need for more straightforward tool setup and deployment outside of the ATLAS working environment appeared. The final goal of the current improvements described in these proceedings is to offer the users an entirely out-of-the-box logbook utility.

2. The logbook constituents

ELisA is a Spring framework-based web application, following the Model-View-Controller (MVC) design pattern [2]. It offers a user-friendly web interface (see figure 1) to browse and search activity logs (called entries), to download attachments, to add, update or reply to logbook entries [3]. ELisA privileges client-side processing for message visualization. We use the DataTables plugin [4] which is a powerful Javascript and jQuery library to add interactive features to HTML tables. ELisA uses AJAX techniques to retrieve data on client request asynchronously. The logbook configuration and the data entries are stored in an Oracle database (the choice of database technology was limited to ensure the portability of the old data).

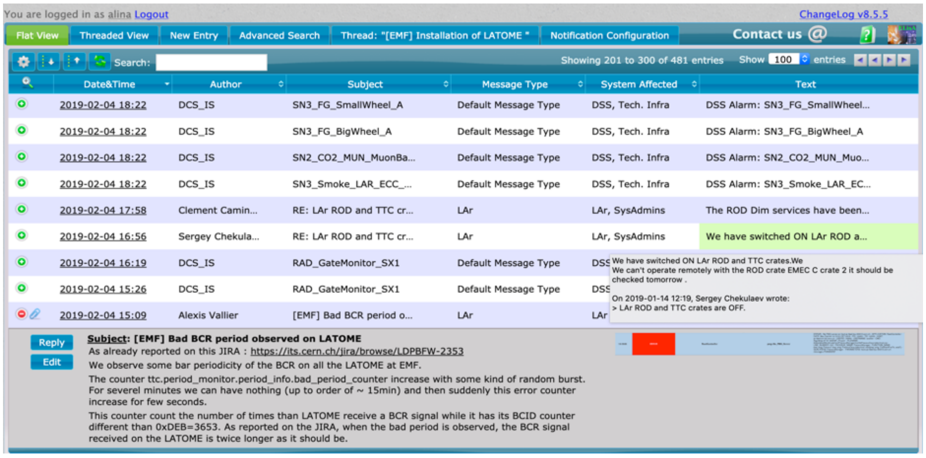

Figure 1. Screen capture showing the logbook web interface

ELisA also provides a REST API to its functionality [5], i.e., an HTTP-based interface to create, access and modify entries. For each client HTTP request, the REST API server responds with a structured XML or JSON representation of a resource. The model objects are mapped to the structure of the Oracle database used for configuration and data storage. The Spring framework is used for implementation, and the model classes are using JSR 303 annotations [6] for data validation and JAXB annotations [7] for the marshaled XML structure. The REST API server clients are the tools and services of the data acquisition software infrastructure as well as user-developed web pages and dedicated scripts (see figure 2) that should insert log entries automatically.

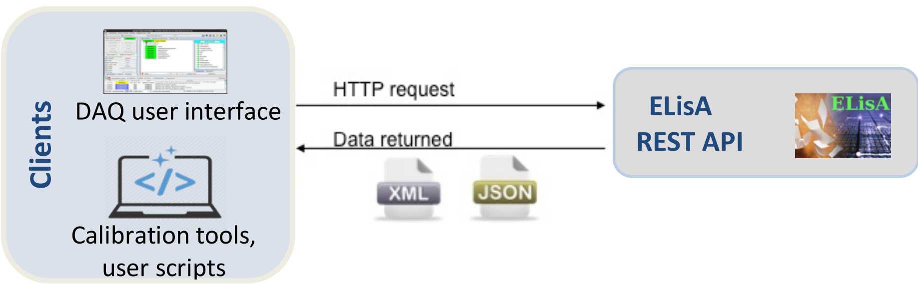

Figure 2. Schematic view of the interaction between the REST API and its clients

ELisA REST API is exposed to Python, Java, and C++ via dedicated client libraries. Also, a set of command line utilities are available on top of the Python client API library to provide developers with a programmatic-free facility to perform the most common logbook operations.

The logbook provides a handy email notification mechanism fully configurable depending on different entry properties, e.g., message type, systems affected list or based on a specific option. A specific tab on the web interface can be used to query the email notification configuration. Also, a reply-to utility is available to insert into the logbook a reply to a given entry directly from the user's preferred mail client, without accessing the web interface.

The logbook implements restricted access to authenticated users only. Users can connect to the logbook from the CERN General Public Network (GPN) or the experiment ATLAS Technical and Control Network (ATCN). To cover both cases, a double authentication mechanism is implemented: first one is based on CERN SSO and is used on GPN only; the latter implements a failover mechanism based on the LDAP servers and is used to authenticate inside the ATCN.

3. The logbook improvements

3.1. Editor templates

ELisA allows for pre-formatted, fixed templates to be automatically loaded into the entry editor in case of recurrent problems for which entries need to be posted. The format is thus standardized, and all the desired information is requested explicitly. An example of an entry template can be seen in figure 3. This approach makes the logbook information more thorough, more consistent, and, ideally, quicker to post. The templates are configurable depending on specific entries properties, and one can define as many templates as needed. The user editing a log entry can choose to use a specific template or to freely edit the text by choosing 'No template' option. The configuration information is stored in the same database as the logbook configuration.

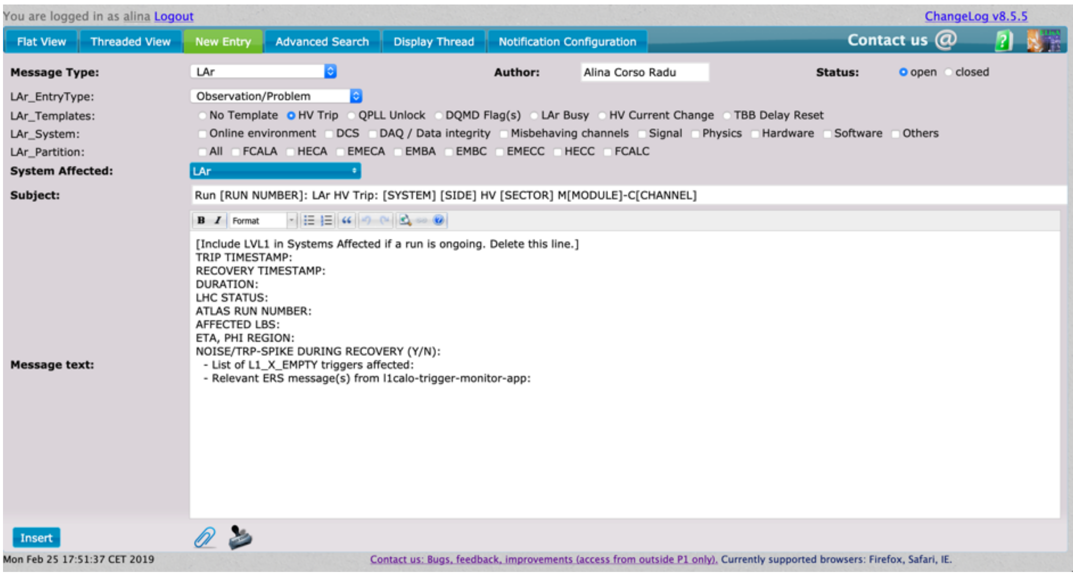

Figure 3. Screen capture showing the editor template options for a given message type

3.2. Support for multiple logbooks

Though initially the logbook was developed for ATLAS experiment's operations, requests from different sub-systems for private tool usage started to be gathered. Therefore, development was done to support configurations for multiple logbooks stored in the same Oracle database which is managed by the CERN IT department. A switch-startup page was implemented to help the user to choose the right logbook when accessing the web interface. The REST API server was improved as well to support logbook_name specification in the request URLs. For example, in the following HTTP request: /logbook_name/messages/12345, if the logbook_name is not specified, the ATLAS logbook is used by default, to retrieve the message with ID 12345. Else, the request is addressed to the specific logbook database.

3.3. Support for multiple databases technologies
One of the problems of the private logbook setups is the usage of the centralized Oracle database which is quite a complex operation to set up. Therefore, one way to improve the setup was to implement support for another SQL database. We choose MySQL which is easy to install and configure for a user. The MySQL configuration data uses the same structure names as those used by the Oracle configuration. Support for PostgreSQL will be added as well.

3.4. Migration to Spring Boot framework
As we are looking for ways to improve the deployment process, the migration to Spring Boot framework [8] seems to be the right approach as it helps to automate the configuration and the deployment while making it easier to implement new features. Additionally, Spring Boot reduces the XML configuration almost entirely, making the development more straightforward. Spring Boot also provides Tomcat server embedded into the framework, thus reducing the maintenance work.

4. Logbook deployment
The product is stable and mature enough, being in production for a few years already. No performance issues have been observed during the whole LHC Run 2 operations period. One server instance is enough to ensure logbook high-availability and good scalability, and user-friendly experience when accessing the logbook functionality.

Currently, five different logbook setups are deployed within the ATLAS experiment, and two setups are used by the ProtoDUNE [9] experiment at CERN, and few more are expected to be set up and deployed in the near future.

5. Conclusion
We developed a web facility for the electronic logbook used by the ATLAS experiment to keep track of the daily activities of the experiment's operations, commissioning and deployment work. The logbook is used by the system operators, experts and automated services to record and share information. We are implementing solutions to address the portability of the logbook as it is used as well in private setups outside its initial scope. Support for other database technologies is added, to remove the dependency from the Oracle database. We are reducing the deployment dependencies and maintenance effort by using Spring Boot utilities. These developments will improve the setup and deployment of the logbook.
Contact the developers: alina.radu@cern.ch or atlas-tdaq-cc-wg@cern.ch

References

[1] ATLAS Collaboration 2008 JINST 3 S08003
[2] Pivotal, "Spring framework" [software], version 5.1. Available from http://www.springsource.org/
[3] Corso-Radu A, Magnoni L and Lehmann-Miotto L Electronic Logbook for the Information Storage of ATLAS Experiment at LHC 2012 *J. Phys. Conf. Ser.* **396** 012014
[4] SpryMedia Ltd. "DataTables" [software], version 1.10. Available from http://www.datatables.net/
[5] Corso-Radu A, Magnoni L and Murillo Garcia R The ELisA Facility – RESTful API and Client Libraries 2013 *NSS/MIC* 10.1109/NSSMIC.2013.6829714
[6] JSR 303: Bean validation [software], version 1.0. Available from https://beanvalidation.org/1.0/spec/
[7] Java Architecture for XML Binding (JAXB) home page. https://www.oracle.com/technetwork/articles/javase/index-140168.html
[8] Pivotal, "Spring Boot framework" [software], version 2.1.4. Available from http://spring.io/projects/spring-boot
[9] http://cenf.web.cern.ch/