The Electronic Logbook for the Information Storage of ATLAS Experiment at LHC (ELisA)

Alina Corso-Radu¹, Giovanna Lehmann-Miotto², Luca Magnoni²

¹University of California, Irvine, USA
²CERN, Geneva, Switzerland

Contact: alina.radu@cern.ch

## Introduction

The ATLAS collaboration is composed of about 3000 scientific users from 36 countries who, together, run one of the largest high energy physics experiments in the world.

An electronic logbook is used to record and share messages about ATLAS data taking activities by system operators, experts and automated services. Since the log messages are of interest for the whole ATLAS collaboration, the logbook tool has to meet strict requirements:

- High-availability and scalability
- Fast and effective message browsing, editing and visualization
- Standard API for message retrieval

## MVC and Spring

The Model-View-Controller approach introduces a well defined structure in the ELisA architecture. The Spring framework provides the means and the technology to implement it. Model objects, mapped to messages, are retrieved from the Oracle database via Spring Data and Hibernate tools. Client requests, coming both from the web front-end or the REST API, are handled by Controllers and resolved in dedicated Views. Controllers are standard Spring beans and they are responsible for handling the mapping between user requests and model objects.

- The ELisA back-end server, hosted by the Apache Tomcat container, is deployed on two dedicated server machines, one used for production and the other reserved for fail-over.
- User authentication is based on CERN Shibboleth Single-Sign-On and on plain LDAP for usage inside the restricted experiment network.

## REST API

An HTTP-based REST API is provided for client applications to access and modify logbook messages. Following the REST architectural principles, a client makes a HTTP request using the standard verbs (GET/PUT/POST/DELETE) to a structured URL. Operations such as creating a new message, editing existing ones or getting a list of entries with specified criteria are supported in this way. The ELisA web front-end is designed to provide an effective and responsive user interface to browse and edit log messages. Based on JSP technology, the front-end heavily uses JavaScript for client-side processing, data formatting and visualization. Users are provided with functionalities to browse messages, submit new logs, reply or update existing ones, perform advanced research based on the multiple message properties and configure the layout of the page.

An asynchronous communication with the ELisA back-end is implemented using Ajax principles:

- Basic message properties are pre-loaded on client side
- Detailed message content is retrieved on user request

This allows for a fast and effective user experience, which is one of the main successes of the ELisA project.

## Towards a new implementation

Already in its first version, the logbook was designed with a web front-end for message visualization and an Oracle back-end for archiving messages. Nevertheless, the original implementation showed limitations in scalability, performance and functionality due to the adopted web technologies.

ELisA, the new logbook implementation, is a modern web application:

- it implements the Model-View-Controller approach,
- is based on the Spring Web MVC framework,
- it privileges client side processing for message visualization and
- uses the AJAX technique to asynchronously retrieve data on client request.

This approach adds responsiveness to the user experience, reduces the load on the server and decouples the implementation from the back-end database. Moreover, using standard technologies and frameworks reduces the in-house developed code-base and thus facilitates software maintenance and evolution.

## Performance

The plot in this box presents the comparison of the front-end loading time for the old logbook implementation (ATLOG) and for ELisA.

The improvement of ELisA compared with the old ATLOG implementation is clearly visible in the plot. The new implementation scales well with the number of messages retrieved, and it is adding a minor time overhead on top of the Oracle query-response time. The database schema has been preserved for backward compatibility, but it will be subject to revision and optimization in the future to further improve the data retrieval performance.

## Conclusion

ELisA is used by ATLAS in production since the beginning of data taking in 2012 and has been positively received by many users. Client-side processing allows for a faster and more effective user experience.

The adoption of a MVC-driven architecture has allowed to focus code development on specific features of the logbook, while profiting from the reliability of established third-party technologies, such as the Spring framework.

The implementation of a REST API provides access to the logbook features to clients and services beyond the provided web front-end.

## References

2. Spring Source Tool Suite [http://www.springsource.org/sts](http://www.springsource.org/sts)
3. DataTables [http://datatables.net](http://datatables.net)