

# AN XML BASED WEB SERVICE FOR AN ELECTRONIC LOGBOOK

R. Kammering\*, O. Hensler\*, A. Petrosyan† and K. Rehlich\*

## Abstract

Since October 2001 the documentation of all accelerator operation relevant information at the TESLA Test Facility [1] is accomplished by using the "electronic logbook". Due to the broad acceptance and usage of this service this e-LogBook is now also used at the HERA accelerator and for other applications.

Every standard web browser can function as a user interface for the input of text and retrieval of information. Graphical data is inserted by low level postscript print services to offer a platform independent input interface. The generation of PDF (Portable Desktop Format) is provided for high quality printouts.

The data is stored in the today widely used XML format to allow high performance searches and also to allow interfacing with other web based services. A standard web server like Apache is generating dynamic content by use of JAVA servlet technology.

The key technologies used for this e-LogBook will be discussed and a practical overview of its applications will be demonstrated.

## 1 MOTIVATION

In the praxis of operation of huge accelerator facilities one is used to document all machine relevant parameters in a so called *logbook* (see figure 1).



Figure 1: The "old" paper logbook.

Since this way of documentation can be very time consuming and sometimes also doesn't produce a traceable history of the machine status, the need for a improvement in this direction is obvious.

A further advantage of having this information stored in a electronically version is that this allows interested persons (other then the operators) to have instant access to the actual status of the machine.

Furthermore is such an electronic version an essential tool to perform operation from remote locations – since a paper version of this logbook can hardly be shipped to all interested collaboration partners.

## 2 REQUIREMENTS

To be a good substitute for the old paper logbook version the eLogBook has to fulfil a number of requirements.

### 2.1 Simplicity

First of all one must accommodate the needs of the operators – since they will be the ones who are heavily making use of the logbook.

Since the input to this eLogBook should be at least equally or even easier than it has been in the paper version, an easy to use front end is a must.

### 2.2 Availability

Due to the requirement for the use outside the hosting institute, the electronic version of the logbook must be available at a usual Windows, Linux, Solaris, etc. system.

There should not be any need for special "set-up" and "installation" procedures at the guest institutes to implement the front end.

### 2.3 Stability

Because most of the bigger accelerator facilities are running around the clock 24 hours 7 days the week such a electronic logbook has to be always available. This gives the eLogBook a status of a high availability software product with all the know requirements.

## 3 MEETING THE REQUIREMENTS

Due to the above mentioned requirements it is just logical to investigate the use of modern web technologies for this purpose.

A usual web browser – called client in the paper – (like Netscape or Internet Explorer) meets the requirements for simplicity – everybody is used to work with it – and also for availability since nearly every PC system includes a web browser in the standard factory configuration. Since it is out of our scope to provide an advantage concerning the stability of such web browsers, we have to live with these products as they are.

The concern of stability arises mainly at the side providing the data for the clients (this side will be called server in what follows). This server will be the core of the

\*DESY Hamburg.

†Yerevan Physics Institute

e-LogBook system since it has direct access to the data inputs from the operators, manipulates them in a certain way and delivers them to the clients.

This kind of problem is pretty well known on the usual sector of web technologies. If one relies on the common systems that are available on the market, one can take profit of many years of experience that accumulated in these products.

Among the most popular products there are: The web server of the Apache software foundation; IBM's HTTP server; Sun's IPlanet and many more.

#### 4 THE BASIC TECHNOLOGY

Figure 2 shows a sketch of the basic input and output mechanism for the e-LogBook.

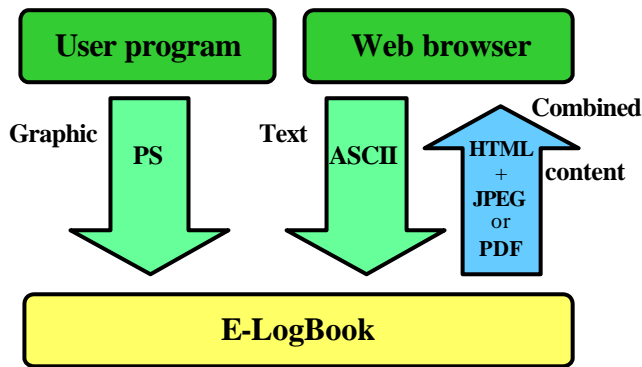


Figure 2: Data in- and output for the e-LogBook.

There are two ways to get data into the e-LogBook, which functions in this view as a kind of database. One way is mostly supposed for input of graphical data and is realized by a simple low level Postscript (PS) print service. This input channel can easily be realized on any standard UNIX system that provides the so called *pipe mechanism*. Every user program that is capable of printing to such a pipe will be able to write (graphical) data into the e-LogBook.

The second input channel is mainly meant for input of ASCII text data. We provide a plain HTML "forms" front end in the browser to directly type in comments to the graphical data and also to place text-only entries.

As already briefly mentioned can the graphical input channel also be used to get textual data into the e-LogBook. This is possible since one can "peek" at the pipe to gather what kind of data is waiting and take the necessary action to store this data in the adequate way.

All requests to the e-LogBook are realized by usual HTTP communication – meaning that every standard web browser can function as a front end for data retrieval from the e-LogBook. The e-LogBook delivers all stored data either as HTML page with embedded JPEG (joint picture experts group) graphics or as PDF for printing purposes.

#### 5 INTERNAL DATA HANDLING

The core functionality provided by the e-LogBook is based on the internal data handling and storing by using the *extensible mark-up language* (XML).

Figure 3 shows a more detailed view of the dataflow that will be invoked by an input of graphical data (green arrows) and the request for certain amount of data (blue arrows) by a client.

##### 5.1 Data input

In case of a input of graphical data a program (in our case realized by a shell script) is "listening" at the already mentioned pipe that is connected to a printer port. If now PS data is send to this printer port, the program saves the original PS data and creates a JPEG version (for faster access) and a template XML file.

This XML file provides a kind of linkage information that uniquely connects the above mentioned files.

##### 5.2 Data output

If a request is send to a JAVA servlet running on the web server, the servlet reads the appropriate data and creates a temporary XML file holding all requested entries.

The servlet now triggers an XML parser to read and pass the data to an XSLT (XML *stylesheet language transformation*) processor in case of HTML output or to an XSLFO (XML *stylesheet language formatting objects*) processor for creation of PDF output.

The servlet redirects this output stream to the requesting web browser and the request is fully processed.

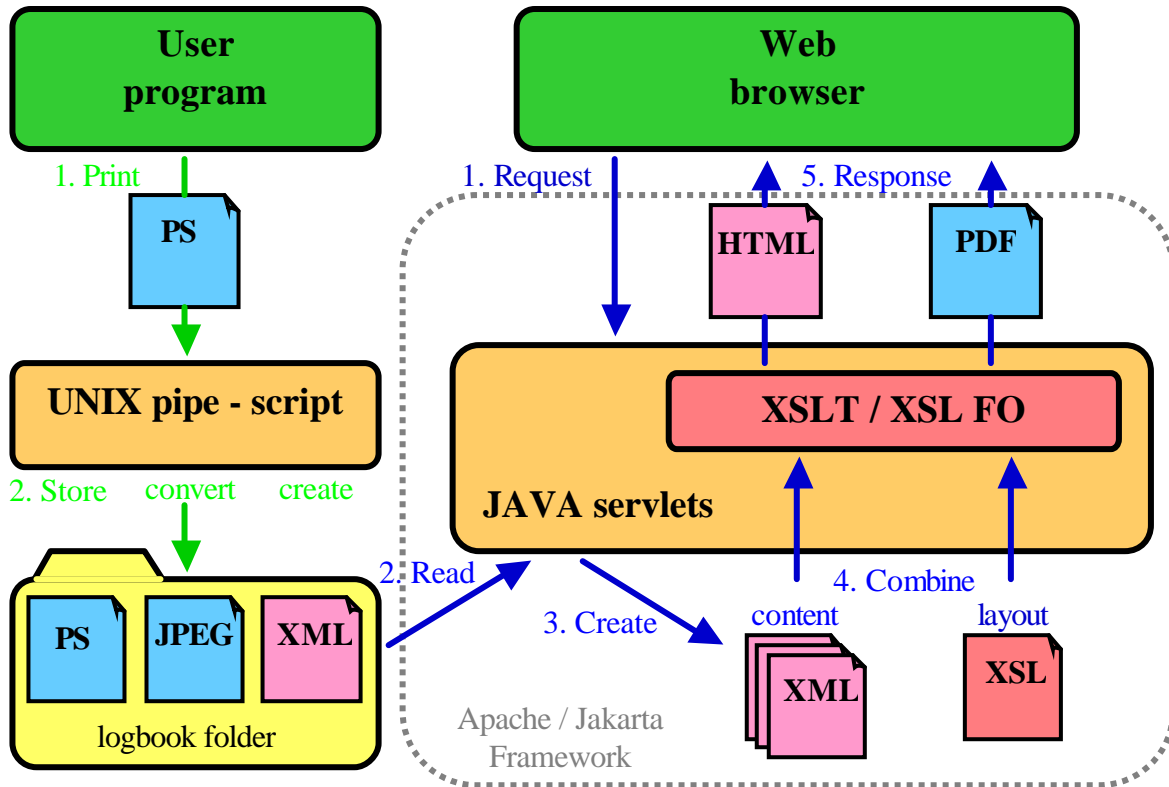


Figure 3: Internal data flow invoked by a write operation (green arrows) and due to a request by a client (blue arrows).

## 6 SOFTWARE COMPONENTS

The overall structure of the e-LogBook can be divided into the classic *client – server* architecture (see figure 2) and we will discuss the used software components corresponding to this classification.

### 6.1 The client side

On the client side every user program that is able to print to a standard postscript printer can play the role of a client. We are currently working MATLAB, Labview and ROOT, which produce very good results.

Every standard JAVA capable web browser is furthermore able to do a bi-directional communication via the usual HTTP communication with the server, meaning that it is able to write ASCII text into and also can request both text and graphical data, from the e-LogBook. Netscape 6.x upward and Internet Explorer 5 upward show a very good performance.

### 6.2 The server side

On the server side a JAVA servlet engine is the minimum requirement, but to build up a more convenient in-

terface one should also implement a standard web server. Further an XML parser and an XSLT processor is needed.

These software components build up the core of the e-LogBook server side (grey dotted area in figure 3).

We are currently running a Apache web server [2] in combination with the JAVA servlet engine from the Apache Jakarta project (alias “Tomcat” see [2] for details). The XML parser “Xerces” and the stylesheet processor “Xalan” are also part of the Apache XML framework.

The already mentioned pipe mechanism for the graphic input can easily be installed on any UNIX system without additional software components.

The remaining component is the storage pool (named “Logbook folder” in figure 3). We are currently working with a plain directory structure mapping the underlying hierarchical date ordering that is determined by the machine/shift cycle.

## 7 THE TTF E-LOGBOOK

Figure 4 shows a snapshot of the e-LogBook as it is used for the TESLA Test Facility (TTF).

The full screen is divided into three distinct areas:

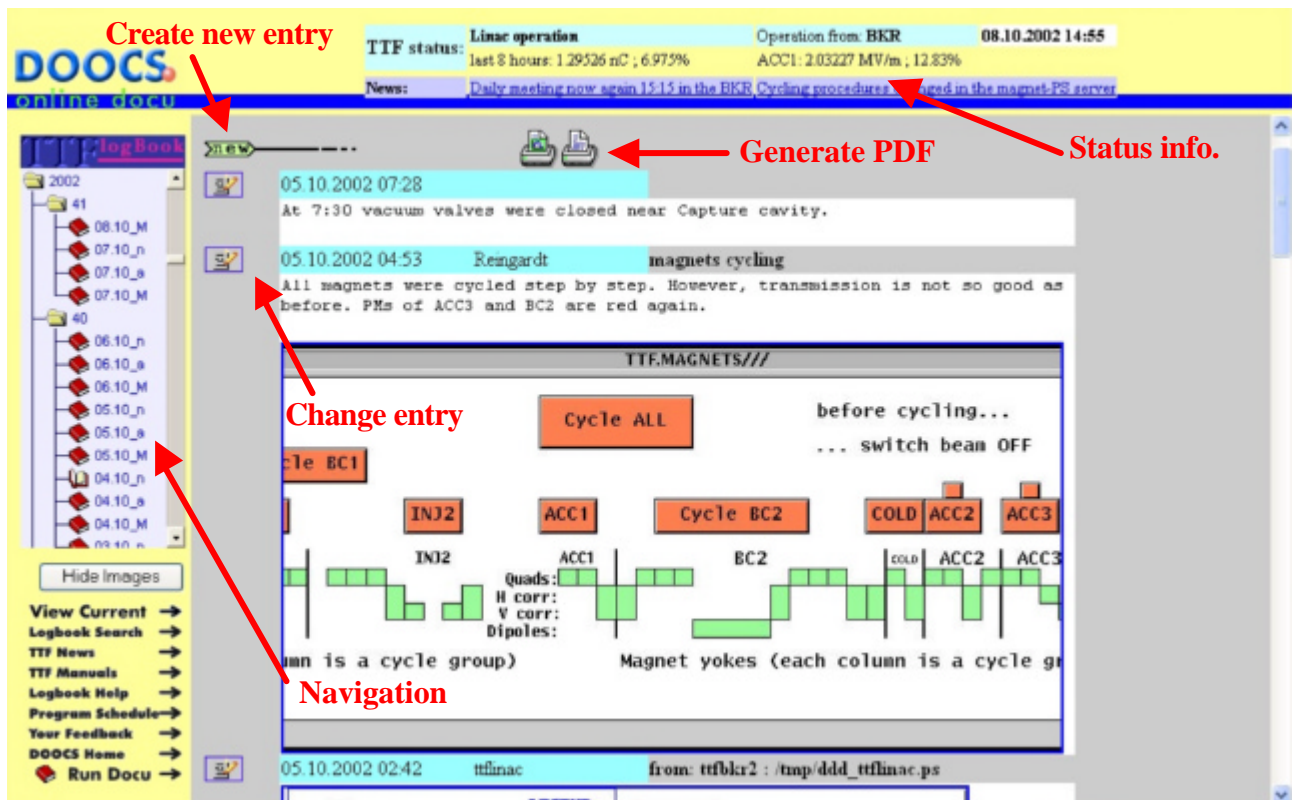


Figure 4: Snapshot of the e-LogBook used at TTF.

The *top* frame holding status and other important information. The *navigation* frame for access to the full database of the e-LogBook and some links to other related pages. And the so called *list* frame which receives the output to a request to the e-LogBook (see also figure 3) and holds the main control elements for access to the e-LogBook.

This layout and its functional bindings are just one possible way of interfacing the e-LogBooks core functionality. One could even imagine a complete JAVA Applet based or Visual Basic interface for the client side, but this would in our case violate our goal for availability.

## 8 CONCLUSION

Until now we gained the experience of one year of machine operation at the TTF using the e-LogBook.

From the response of the machine operators and also other people involved in the machine operation, it is obvious that no one wants to go back to the old paper log-

book. It is the common opinion that the electronic logbook is a success, since it offers new possibilities and also allows a more reliable and faster editing.

All this resonance and feedback shows that the requirements and assumptions we made to design and implement this e-LogBook were at least met if not surpassed.

The core technology and the software components used to build up this core, are all freely available and one could imagine to introduce such a tool in a variety of other institutes and for a plenty of other areas of interest.

All modules, except the web server and the servlet engine can be downloaded from [3].

## 9 REFERENCES

- [1] <http://tesla.desy.de>
- [2] <http://www.apache.org>
- [3] <http://ttfinfo.desy.de/doocs/elogbook/logbook.html>