REVIEW OF TWO YEARS EXPERIENCE WITH AN ELECTRONIC LOGBOOK

R. Kammering, O. Hensler, A. Petrosyan and K. Rehlich

Abstract

Now more than two years have past since the first introduction of the electronic logbook (e-LogBook) at DESY Hamburg. Due to its great acceptance, it is now no longer only used at the TESLA Test Facility (TTF) [1] at DESY, but also at SLAC (USA), INFN (Italy) and ESRF (France).

Triggered by the remote operation of TTF and the collaborative work within the TESLA collaboration, it showed that there is a strong need for an electronic version of the standard “shift logbook”. This lead to the development of a first version of the e-LogBook that has proven to be a software tool in the sense of a “Global accelerator network” (GAN) to support international collaborative activities.

The implementation follows the, in the area of web services widely used, client-server concept. On the client side every standard web browser can be used as user interface for input of text and retrieval of information. Graphical data can be inserted by use of low level postscript print services to offer a platform and program independent input interface. All data is stored in the XML format to allow interfacing with other web services and fast access for searching. On the server side a standard *Apache* and *Tomcat* web server [2] is generating dynamic content by use of JAVA server pages (JSP), XSL description of the layout and JAVA servlets [3].

The general concept of the e-LogBook, the used technologies, and the potential of this web service will be discussed.

THE TESLA TEST FACILITY (TTF)

The TESLA test facility is an advanced linear accelerator in the lower energy range (220 - 270MeV for phase 1, up to 1GeV for phase 2). Figure 1 shows a view of the linac tunnel in its assembly configuration of phase 1 (TTF1).

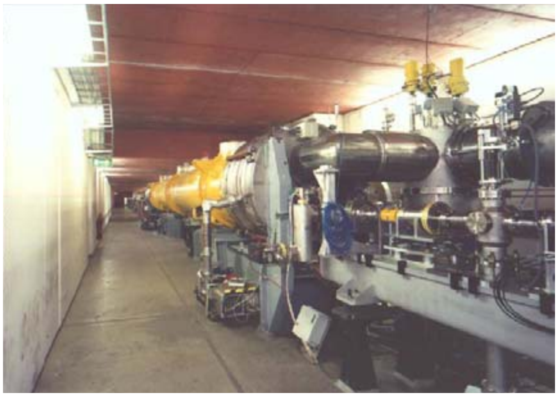

Figure 1: View of the TTF1 tunnel.

The main focus of the TTF (phase 1 and 2) is to:

- test the superconducting accelerator technology
- test the free-electron laser technology (FEL)
- be an reliable user facility for providing FEL light

In the second phase of TTF (TTF2) the facility is rebuild in a collaboration of 52 institutes from 12 different countries. To give an impression of the parameter range the TTF1 has been working in the following table shows an overview of the maximal performance values achieved in the year 2002 before TTF1 has been shut down to be extended to the TTF2 stage.

Table 1: TTF1 FEL performance data

Radiation pulse duration	50fs
Radiation peak power	~1GW
Spectrum width (FWHM)	1%
Bunch charge	0.2nC

The peak brilliance of the produced TTF1 FEL beam exceeds very other source at this wavelength about a factor of 1000 (see figure 2) [4].

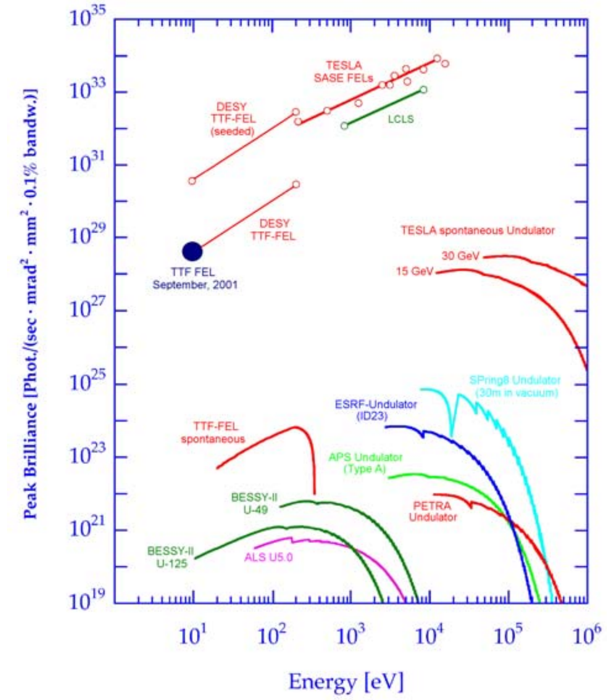

Figure 2: Spectral peak brilliance for the most brilliant FEL and synchrotron sources planed and already existing.

THE TTF E-LOGBOOK

In the year 2001 the controls group of the TTF1 decided to set up an electronic version of the usual shift logbook that is commonly used (or even required at some facilities). This decision has been made since a paper based version of such a logbook lacks a number of requirements to be fulfilled if one wants to work in a GAN like manner. Some of the advantages of an electronic logbook are:

- it makes remote shifts possible
- it can be made accessible for everyone
- be **the** common place for measurement results

After an initial testing phase of the e-LogBook in the real TTF operation, the operators quickly noticed that the electronic logbook version has many advantages and completely stopped using the "old" paper based version. Therefore we are now running the e-LogBook at the TTF1 for already two years. It has gained a very broad acceptance not only within the TTF operation team, but also for all people involved in the TTF operations.

Up till now about 17000 entries have been made and approximately 7.7GB of graphical and 12MB of textual data have been collected in the TTF e-LogBook.

THE E-LOGBOOK TECHNOLOGY

The general technology used for the e-LogBook follows the classical client-server architecture (see figure 3).

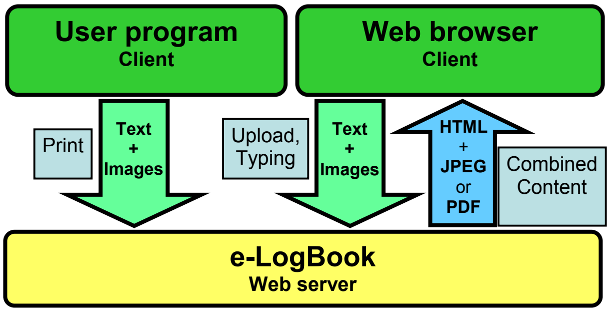

Figure 3: Basic architecture of the e-LogBook.

There are two ways to store data in the e-LogBook, which functions in this view as a kind of database. One way is mostly used for the 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*. Any user program that is capable of printing to a standard postscript printer is able to write graphical and textual data into the e-LogBook via this channel.

The second input channel is mainly meant for input of ASCII text data but we now also provide the common file upload functionality. A plain HTML "forms" page as front end in the browser is provided to directly type in comments to the graphical data and also to place text-only entries.

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 4 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 period of time (blue arrows) by a client.

Data input

In the case of input of graphical data a program (realized by a shell script) is "listening" at the already mentioned pipe which is connected to a printer port. If PS data is send to this printer port, the program saves the original PS data and creates a JPEG version (for display by the web browser) and a template XML file. This XML file provides the linkage information that uniquely connects the above mentioned files.

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.

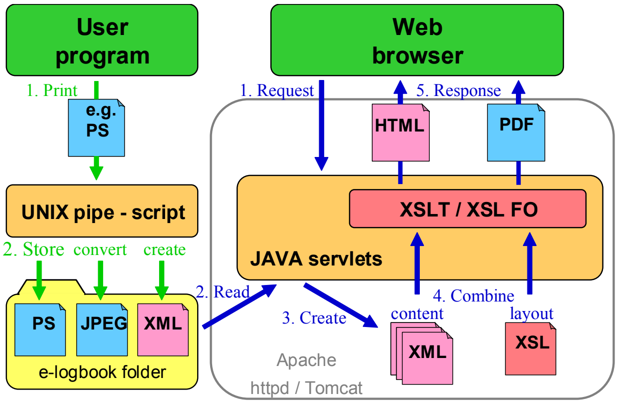

Figure 4: Data flow for graphic input and for a request of a certain period of time

WHAT HAS CHANGED?

We have ported all HTML code to JSP (JAVA server pages) to allow:

1. parameterization of e-LogBook specific values
2. one central (XML) conf. file per e-LogBook
3. to access this values via JSTL (JAVA standard tag library) from within every JSP files.

With this central configuration we can work in a modular design for the XSLs as sketched in figure 5. Due to the quite amazing number of e-LogBooks we are currently running (not only at DESY) the effort for keeping per e-LogBook XSLs up-to-date, would otherwise get to complex.

We have implemented JDBC (JAVA database connectivity) support to allow direct storing of main machine

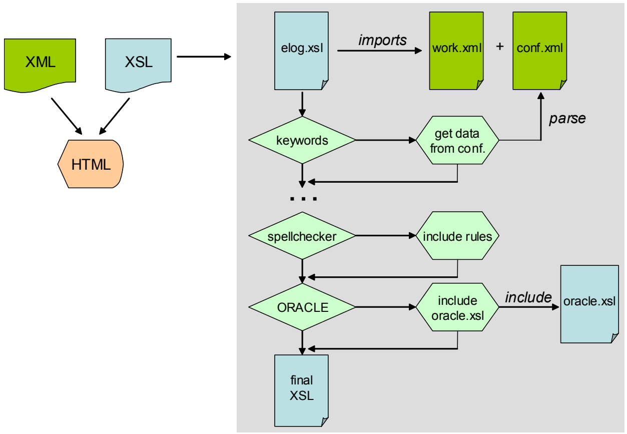

Figure 5: Sketch of the modular XSL architecture

parameters and standard shift settings (like names of the shift crew, goals, achievements etc.). This allows the operators to do most of the standard documentation from within the e-LogBook environment.

Further we now gathered first experience with an authentication and authorization schema. We are using the Apache httpd mod_ssl to provide a secure http (https) connection to authenticate the user via Apache Tomcat by use of JAAS (JAVA authentication authorization schema). This will not only allow to fine grade control the accessibility of certain resources on a per user basis but also allow personalizing the user's special desires.

Further is this personalization mandatory for working within collaborations.

THE FUTURE

One easy step towards a more comfortable deployment, upgrading and maintenance would be to build a web archive (*WAR* file) from all common e-LogBook parts, what can be fulfilled without great effort.

A further improvement to better reflect the user's desires would be to expand the parameterization of per e-LogBook specific values, to allow shifting these to the central configuration file. This would also ensure to work with a limited number of XSL files, which is a strong requirement since we already have been faced with many problems concerning the maintenance of these XSL files.

One surly necessary future development is to integrate a security, authentication and authorization schema into the e-LogBook core (the common WAR file). We have already made first tests and developed a general architecture to support such a security schema (but this will be described elsewhere – also see above).

Such a security schema would not only allow collaborating partners to access information from firewall protected partners (which is the standard), but would also allow a per user control of specific needs and desires like different sorting within the main view or addition of status information for example. This is commonly titled as personalization.

A feature towards a more comfortable layout and look-and-feel we are going to implement in the near future is to support some common text markup types like: bold, italic and underlined text. This markup can easily be realized by introduction of some special character sequences like: __bold__ to automatically recognize this sequence and print the string in **bold** letters within the main view of the e-LogBook.

There are a number of "standards" introduced be the more and more widely use of the so called *Wikis*. One has to be careful to not confuse the users by introducing furthermore new "homemade" markups, so that we will have to do some investigation of the today most widely used markups to select a schema that a user can easily adopt and furthermore might already know due to its use in the above mentioned Wikis.

As a fare goal we might think about expanding the e-LogBook to a full featured web-service [5] in the sense of supporting the SOAP protocol, building a web-service description (WSDL) and registering this web-service on a UDDI (Universal Description, Discovery and Integration) server. This would allow an intercommunication between this web-service and other web-services like an e.g. "central alarm and info server" [6]. Further would this enable the user to not only work within one e-LogBook but also to do e.g. search over a number of e-LogBook of interest or freely collect information from a number of different sites hosting e-LogBooks. This way the e-LogBook would really advance to a sophisticated information and knowledge pool which could provide an up to now not available resource for information within the particle accelerator community.

CONCLUSIONS

During the past two years we have seen that the acceptance for this more or less new technology is really overwhelming (today more than two dozens of the initial TTF e-LogBook version are running on approximately a dozen servers at different institutes). Therefore the ease for cloning of one e-LogBook has been a high priority and will probably remain so. This is one reason for the order and ranking of future developments, but this need might get a bit less important and thus will hopefully allow working on the more forward-looking features like the expansion to a full qualified web-service for example.

All in all the general use of web technologies in the area of particle accelerator control systems has been proven by this e-LogBook technology and one might expect a lot of new developments and improvements within this sector. This should strongly urge all people working in this areas to think about build collaborations.

REFERENCES

[1] http://tesla.desy.de
[2] http://www.apache.org
[3] http://java.sun.com
[4] *Eur. Phys. J. D20, 149-156*
[5] See e.g. www.sun.com
[6] Talk of O. Hensler given at the ICALEPCS 2003