FLOGBOOK: FROM CONCEPT TO REALIZATION

B.S.K. Srivastava*, R.K. Agrawal, K.G. Barpande, P. Fatnani, C.P. Navathe
RRCAT, Indore, India

Abstract

Indus-1 and Indus-2, the Synchrotron Radiation Source (SRS) facilities at RRCAT Indore are national facilities and being operated on round the clock basis to provide synchrotron radiations to users as well as carrying out machine studies. Both of these accelerators are widely distributed systems and employ many sub systems for their operation. These sub systems are also made up of heterogeneous type of hardware and software modules. Indus-2 Control System is presently controlling approximately 10,000 input/output parameters for its operation. To keep the whole system up and running the faults & failures encountered during machine operations are attended at site and all observations and rectifications information are to be recorded electronically by the crewmembers. FLogbook has been conceived & developed to meet such needs. This web based software operates in the intranet environment over three tier software architecture. It mainly uses JavaServer Pages (JSP), JavaBeans and SQL databases for designing its building blocks. Using relational database features have been provided in the package for logging, e-mailing, searching & commenting the faults of various sub systems. Recently we have deployed the FLogbook in the field and machine operation crewmembers have been asked to use it as & when required.

INTRODUCTION

The main motivation of this software is to constantly monitor and improve the performance of Indus-1 & Indus-2 accelerators.

Indus-1 & Indus-2 are big & complex machines and involves various types of hardware and software for running it smoothly at round the clock basis. Seeing the complexities of various parts of both accelerators, it was felt to have a system that would systematically record the fault & failure information of accelerators encountered during its operations. This information would be recorded by accelerators crewmembers.

REQUIREMENTS

Based upon the previous experiences and feedback received from various system experts following basic requirements were considered before developing the FLogbook:

(1) It should be possible to use the FLogbook by multiple users simultaneously connected over campus network.
(2) The FLogbook should be easy to use for especially non computer experts.
(3) Only authenticated users should be able to use the FLogbook. It should be possible to log the textual as well as non textual data e.g. image, documents etc into FLogbook
(4) It should be possible to transmit the logged information with attachment to concerned sub-system experts electronically.
(5) It should be possible to query & retrieve the stored historical data with different search arguments.
(6) It should be possible to add comments on logged faults by concerned sub-system persons.

SOFTWARE DESCRIPTION

Access to FLogbook has been protected with password so that only authenticated users could use the system. System checks user's credentials using institute's central e-mail server. Hence users may use the software using his/her official e-mail login and password.

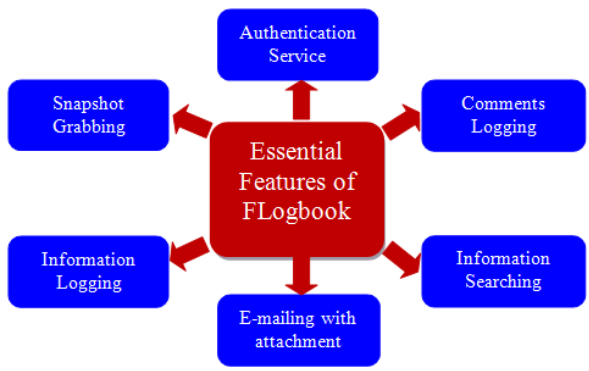

Figure 1: FLogbook Features.

Essential features of FLogbook have been shown in Figure 1. FLogbook is equipped with a software module for saving the computer's desktop in popular image files. This module has been developed in java and packaged in an executable jar file. On clicking the link provided in application this software module downloads & installs on user's machine & sits silently in system tray. This module is very useful for instant grabbing of machine control console desktop if any abnormality or unusual is noticed by the crewmembers.

*bsks@rrcat.gov.in

ISBN 978-3-95450-124-3

Data Input

Using HTML & JavaServer Pages (JSP) Flogbook provides the form for logging the fault related information. Some fields in this form are automatically filled by the system and file attachment (e.g. doc, excel, jpg, png, etc) along with textual data may be logged by the crewmembers for future reference. Logged textual data with attached document (if any) may also be sent immediately to system concerned persons through e-mails. FLogbook gives unique fault-id to each logged fault so that historical fault could be diagnosed in future by its fault-id.

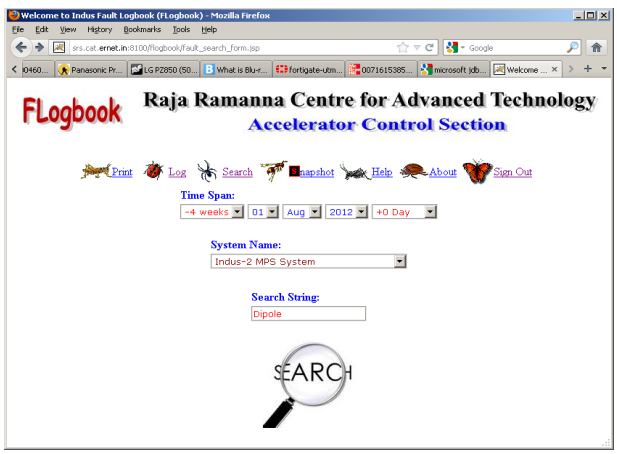

Figure 2: FLogbook Information Searching Form.

Data Output

Flogbook provides query form (Figure 2) for retrieving the logged historical faults with different search arguments e.g. timestamps, system name, text string. Feature has been provided by the software to log comments on retrieved specific historical faults and email it to selected concerned persons.

Software Design

Java Server Pages (JSP), JavaBeans and SQL databases have been used for designing & developing the various components of FLogbook. JSP with JavaBeans are used for developing the applications for web based environments. Java is main programming language used here for developing the complete FLogbook.

SOFTWARE ARCHITECTURE

The FLogbook follows the three-tier software architecture for designing & executing its building blocks. Here Web Browser resides on client machine and work as first or client tier. In our working environment we mainly use Microsoft Internet Explorer (IE) for accessing the web sites and web based applications hence we have developed and tested the Flogbook for IE users. FLogbook uses JavaServer Pages (JSP) & JavaBeans for developing the presentation/view and business/application logic. JavaServer Pages (JSP) technology enables Web developers and designers to rapidly develop and easily maintain, platform independent, information-rich, web applications that leverage existing business systems.

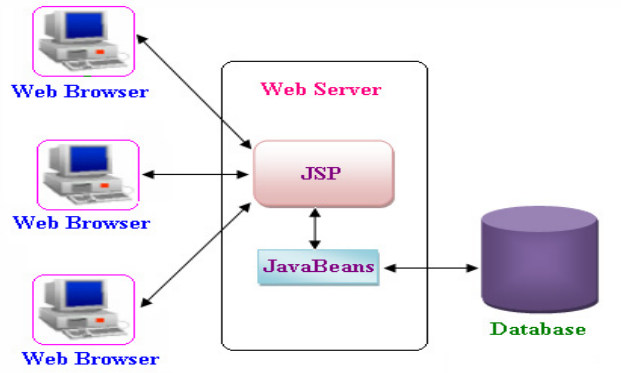

Figure 3: FLogbook Architecture.

JSP technology separates the user interface (content presentation) from content generation, enabling designers to change the overall page layout without altering the underlying dynamic content. Tags for content *access and presentation* reside in the webpage. Logic and programming code for content *generation* reside in reusable components. After receiving the client request, the JavaServer Page requests information from a JavaBean. The JavaBean can in turn request information from a database (Figure 3). Once the JavaBean generates content, the JavaServer Pages can query and display the Bean's content. JavaBeans components (beans) are reusable software programs that we can develop and assemble easily to create sophisticated applications.

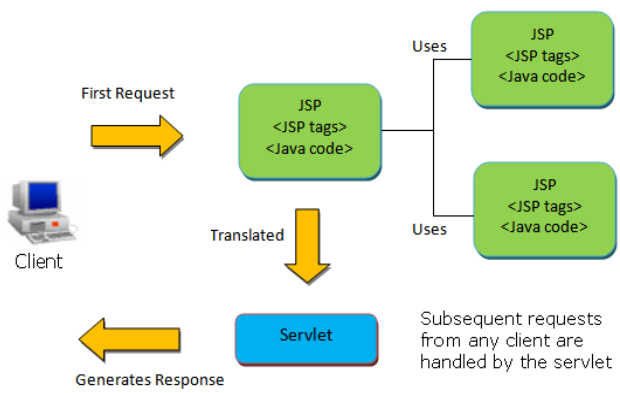

Figure 4: Execution of a JavaServer Page (JSP).

FLogbook uses JDBC (Java Database Connectivity) inside JavaBeans components for accessing the FLogbook database for inserting & retrieving the information.

ISBN 978-3-95450-124-3

The Type 4 Driver for JDBC has been used which provides JDBC access through any java-enabled applet, application, or application server. It delivers high-performance point-to-point and n-tier access to SQL database across the internet & intranets.

The JavaServer Page is identified to the server by a *.jsp* extension; this tells the server that special handling is required. As shown in Figure 4**,** the first time a request is made for such a file, the *.jsp* file is translated to a servlet & compiled into an object. (For that reason, there can be a slight delay on the first request for a *.jsp* page.) The output from the object is standard HTML which the browser interprets and displays as usual. After compilation, the compiled-page object is stored in memory on the server. On subsequent requests for that page, the server checks to see if the *.jsp* file has changed. If it has not changed, the server uses the compiled-page object stored in memory to generate the response to the client. (Because the object is stored in memory, the response is very fast.) If the *.jsp* file has changed, the server automatically recompiles the page and replaces the object in memory.

We have used Apache Tomcat as a web server for executing & serving web components of FLogbook. Apache Tomcat (or simply Tomcat) is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Oracle Corporation, and provides a "pure Java" HTTP web server environment for Java code to run. FLogbook uses JavaMail API for sending the e-mails composed of the information logged by the operation crewmembers into FLogbook database. The JavaMail API provides a platform-independent and protocol-independent framework to build mail and messaging applications.

Microsoft SQL Server based relational database was designed to implement the data tier of FLogbook, which stores the complete information in related tables. Subsystem names of both the accelerators (Indus-1 & Indus-2) and concerned persons details are stored in the database so that logged fault information could be mailed electronically.

SOFTWARE DEPLOYMENT

The FLogbook has been deployed on Apache Tomcat web server configured on Gateway machine. This computer as shown in Figure 5 is connected with accelerators' technical network (AccNet) as well as campus network (RRCATNet). The URL of the application has been mapped in DNS of both the networks so that it could be accessed uniformly from the machines of both networks. DNS Server of accelerators' technical network (AccNet) is running on Primary Domain Controller (PDC) Server machine.

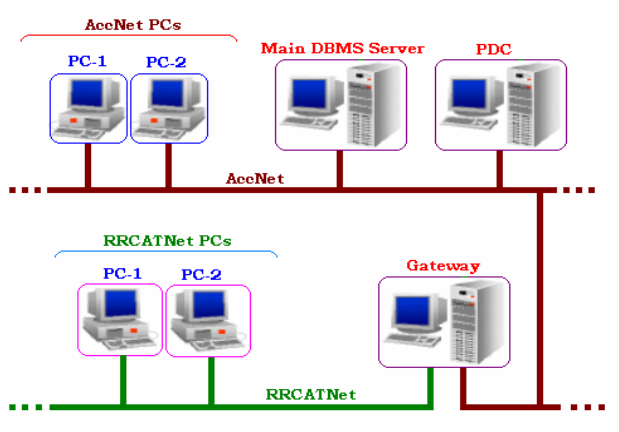

Figure 5: Deployment of FLogbook.

CONCLUSION

The first version of FLogbook has been deployed in the field and being used by the operation crewmembers of both accelerators (Indus-1 & Indus-2). The system is very useful for not only the accelerators operation crewmembers but also for machine/subsystem experts for tracking the faults and improving the overall machine performance.

Presently, we are using Microsoft Windows based infrastructure and if we switch to Linux based infrastructure then very little modifications are required in codes of the system during porting.

ISBN 978-3-95450-124-3