

DEVELOPMENT OF WEB-BASED PARAMETER MANAGEMENT SYSTEM FOR SHINE

H. H. Lv, Shanghai Institute of Applied Physics, Shanghai, China
Y. B. Leng, Y. B. Yan, Shanghai Advanced Research Institute, Shanghai, China
C. P. Chu, Institute of High Energy Physics, Beijing, China

Abstract

A web-based parameter management system for Shanghai High repetition rate XFEL aNd Extreme light facility (SHINE) is developed for accelerator physicists and researchers to communicate with each other and track the modified history. The system is based on standard J2EE Glassfish platform with MySQL database utilized as backend data storage. The user interface is designed with JavaServer Faces which incorporates MVC architecture. It is of great convenience for researchers in the facility designing process.

INTRODUCTION

SHINE is a hard X-ray free electron laser user facility which is recently proposed. When completed, it will provide sophisticated research methods for physics, chemistry, life sciences, materials science, energy science and other disciplines.

Generally speaking, the modern accelerator is complex and composed of many subsystems. Each subsystem is composed of several units with independent functions, while each unit is composed of thousands of devices. During the designing process, accelerator physicists devise the basic parameters according to the design goals of the facility. After that, researchers working in kinds of subsystems devise the specific devices' parameters for their respective subsystem. Moreover, the parameters of one subsystem depend on the other. For example, the design of the power supply depends on the requirements of the magnet. In reality, the official version of parameters will be identified through several revisions. Traditionally, the parameters are saved in static text files, which is not only hard to guarantee consistent values of plenty of parameters for several subsystems and a great number of devices, but also violence against the management and analysis [1]. As a consequence, it's very essential to develop an advanced system to store and manage the design parameters for SHINE. A user interface on which users could conveniently retrieve or modify the parameters is also desired. Then a web-based parameter management system has been developed to use relational database for data storing. Details of the application architecture, database schema and system functions are described in the following sections.

APPLICATION ARCHITECTURE

The system is based on standard J2EE (Java 2 Platform, Enterprise Edition) Glassfish platform with MySQL database used as backend data storage. J2EE platform contains various components, service architectures and technical levels which have common standards and specifications.

It's compatible between different platforms following the J2EE architecture [2]. Therefore, the parameter system could be well integrated with other systems. This makes it feasible to integrate all the data involved in SHINE design, building, installation and commissioning process together.

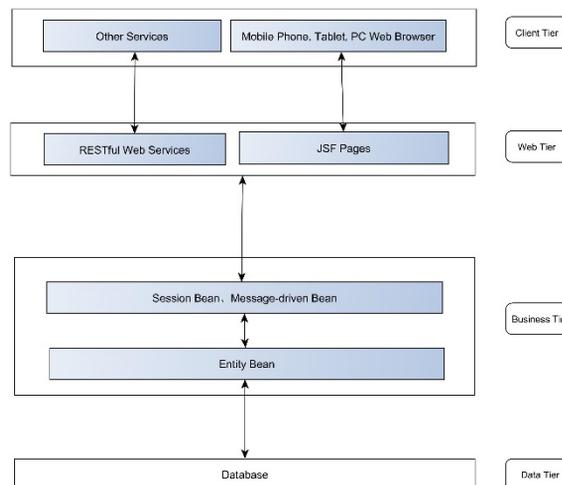


Figure 1: Application architecture.

As shown in Fig.1, the application architecture diagram is composed by four parts: database, business layer, web layer and client layer. The database is a general data storage container for parameter data. The business layer implements the business logic such as retrieving, searching, batch storing, updating and deleting records from database. They are handled by enterprise beans. Web layer includes web pages created by JavaServer Faces technology and services which could be invoked by other services. JavaServer Faces technology builds on servlets and JSP technology. It provides a user interface component framework for web applications. Client layer is a thin client that do not query databases, execute complex business rules, or connect to legacy applications. Such heavyweight operations are off-loaded to enterprise beans, where they can leverage the security, speed, services, and reliability of server-side technologies.

Database Schema

The database schema saves the properties as Name/Value pairs and has built a universal database schema. Details are illustrated in Fig. 2.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

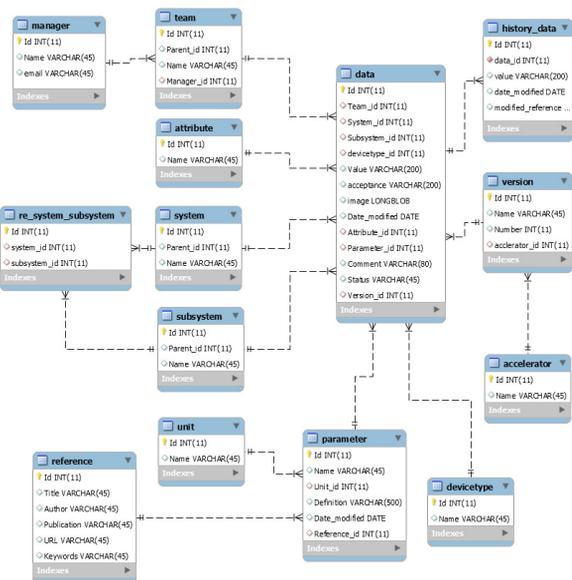


Figure 2: Database schema.

As shown in Fig. 2, the name/value pair is handled by parameter table and data table while the system, subsystem, team it belongs to are handled by respective tables. Usually the value is numerical or characters, but sometimes it is complicated and demonstrated as a curve. For example, beta function representing the amplitude of the transverse motion is an important parameter in the accelerator design period. For this reason, the field of saving an image value is included in the data table. In addition, the parameter value will be modified over and over again and it is important to track the modification history. The history data table meets the needs and it provides much convenience for physicists and engineers to check the value with its modification list. In general, more than one accelerator facilities do coexist in an institute. Therefore, the schema satisfies the requirements of storing several accelerators' parameters. Summarily there are 14 tables in total to satisfy the need of flexibility and popularity.

Business Layer

The business layer in the application architecture contains the business logic, interface between the database and JSF pages. To facilitate database access in programmatic way, a set of Java entity classes for object-relational mapping(ORM) are generated following Enterprise Java Bean(EJB) standards. These EJB classes provide base interfaces for database accessing. Database query through these JPA classes is much simpler than direct SQL with JDBC (Java Database Connectivity) API. Each database table has a corresponding entity class as direct mapping.

On top of entity class set is the Data API which provides convenient access to the database. JavaServer Pages or other services can call the API directly. These data APIs are predefined database queries to facilitate application programming.

Web Layer

The web layer is built with JavaServer Faces technology that is integrated with MVC (Model, View, Controller) architecture for separating logic from presentation. Therefore, the code for view layer can be changed and typically it does not need to change model layer, while controllers are used to process user actions [3]. This is essential for the application being used by other accelerator labs because the presentation contents will be different from each other with the universal database schema and uniform Java Beans.

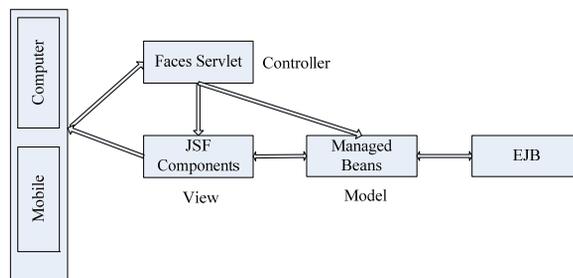


Figure 3: JSF architecture.

The details of JSF architecture is illustrated in Fig. 3. Managed Bean works as Model for UI component. It is a Java bean managed by JSF framework which contains getter and setter methods, business logic, or even event handling. Working as the interface between UI components and EJB, Managed Bean can be accessed from JSF pages.

JavaServer Page works as View showing the user interface. In the application, a lightweight library called PrimeFaces is utilized as the UI framework, which can be used not only to develop a standard desktop web interface but also provide a usable interface for mobile devices built upon an existing application. This allows engineers to check parameter information on the mobile phone or a tablet anywhere.

The controller handles processing of user actions. It includes FacesServlet Class, several configuration files and a series of action handlers.

APPLICATION FUNCTION

Except for basic functions of query, modification, saving and deleting a record (as shown in Fig. 4.), the application also provides several other functions meeting the specific demand for accelerator facilities. Subsequently the details will be described.

SubDivision	System	Parameter Name	Attribute	Unit	Value	Image	Change Date	Comment	Definition	Reference Title	Reference Author	Reference Publisher	Reference URL	Keyword	Attachment
0	0	Pulse length	nominal	%	20.50		2019/06/14								
0	0	Peak brightness	nominal	1	5*10 ¹²		2019/06/14								
0	0	Average brightness	nominal	1	5*10 ¹²		2019/06/14								
0	0	Total beam length	nominal	km	3.1		2019/06/14								
0	0	注入器长度	nominal	km	3.1		2019/06/14								

Figure 4: Display by web browser.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Basic Functions

As shown in Fig. 4, users can browse all the parameters' information including the parameter name/value and the system, subdivision it belongs to. The value will become editable if clicking the exact cell in Value Column. Additionally, a record can be deleted by the "delete" button in the last column. A new record can be added by the "Add a Record" button while an adding page will be shown later. Basically, the default items shown on the web page are comparatively comprehensive but sometimes an engineer is only concerned about a couple of them. In this circumstance, the "Choose Columns to View" button can be used to choose these particular items. It also provides filtering function for researchers to ignore uninterested records.

Image Uploading

Sometimes the value is complicated and demonstrated as a curve, then the uploading button in Image Column can be used to load it and the "eye" button is utilized to view the image existing in the database.

Batch Uploading

Normally all the parameters' information is saved in a spreadsheet. For convenience, batch uploading program is provided to load the parameter name/value pair, system, subdivision and other related data to the database at one time. In the uploading process, data consistency and integrity will be checked and the relations will also be transformed from the basic information contained in the spreadsheet.

Authorization

In practice, the system's parameters are designed by researchers working in that particular area. And the group leader should be responsible for the correctness of each value. For this reason, the permission to edit or delete a record is only allowed by the manager of the team to which that record belongs, while other workers can only browse or search the information. When a user has no corresponding authority, a warning box will pop up as shown in Fig. 5. This authorization method is of great importance to ensuring the parameter values' being consistent.



Figure 5: Deleting alert box.

Viewing History Data

In actual practice, the parameter value will be modified over and over again and it's helpful to check the modification history and why it was modified. As shown in Fig.6, this information is displayed and meanwhile the modified reference is saved as an attachment.

Figure 6: Modification history.

CONCLUSIONS

The application of Web-based Parameter Management System for SHINE has been developed. MySQL database is utilized to store all the information. A series of web-based service APIs and user interfaces are developed to manage the data. The system provides great convenience for physicists and engineers to regulate the parameters. This system has been deployed in a dedicated server, operating stably for several months.

REFERENCES

- [1] C. P. Chu, Y. S. Qiao, C. H. Wang, and H. H. Lv, "HEPS High-Level Software Architecture Plan", in *Proc. 9th Int. Particle Accelerator Conf. (IPAC'18)*, Vancouver, Canada, Apr.-May 2018, pp. 1884-1887.
doi:10.18429/JACoW-IPAC2018-WEPAF030
- [2] The Java EE 8 Tutorial,
<https://javaee.github.io/tutorial/>
- [3] JavaServer Faces Technology Tutorial,
<https://docs.oracle.com/javaee/6/tutorial/doc/bnaph.html>