BLED: A TOP-DOWN APPROACH TO ACCELERATOR CONTROL SYSTEM DESIGN

J. Bobnar, K. Žagar, COBIK, Solkan, Slovenia

Abstract

In many existing controls projects the central database/inventory was introduced late in the project, usually to support installation or maintenance activities. Thus construction of this database was done in a bottom-up fashion by reverse engineering the installation. However, there are several benefits if the central database is introduced early in the machine design, such as the ability to simulate the system as a whole without having all the IOCs in place, it can be used as an input to the installation/commissioning plan, or act as an enforcer of certain conventions and quality processes. Based on our experience with the control systems, we have designed a central database BLED, which is used for storage of machine configuration and parameters as well as control system configuration, inventory, and cabling. The first implementation of BLED supports EPICS, meaning it is capable of storage and generation of EPICS templates and substitution files as well as archive, alarm and other configurations. With a goal in mind to provide functionality of several existing central databases (IRMIS [1], SNS db, DBSF etc.) a lot of effort has been made to design the database in a way to handle extremely large set-ups, consisting of millions of control system points. Furthermore, BLED also stores the lattice data, thus providing additional information (e.g. survey data) required by different engineering groups. The lattice import/export tools among others support MAD and TraceWin tools formats which are widely used in the machine design community.

INTRODUCTION

With the complexity of new machines growing, the complexity and size of the control systems also grows rapidly. Big machine control systems are normally distributed across several hundred I/O controllers, and each of them might run several different applications. In addition, the controllers are interconnected with numerous cables and to various kinds of equipment.

In such a large system it is recommended that an inventory of all the equipment and control points is maintained in a well organized structure, so that information is conveniently available. A good equipment and data management system can help during the design of the machine as well as developers and engineers during the development cycles. On the other hand the physicists and operators can benefit significantly from such a system during commissioning and operations.

BOTTOM-UP VS. TOP-DOWN APPROACH

Until today, several different systems have been set up in various laboratories around the world. Most of these systems are specific to each individual machine. They have usually been introduced late in the project, thus being created in a bottom-up approach: gathering all the information in the system and then the database was populated by reverse engineering the installation. Furthermore, the majority of the systems covers only one specific domain, such as for example inventory or lattice, depending on who initiated the use of the database and what stage the project was in at that time.

Having this in mind we wanted to create a general system, which would allow one to create the machine in a top-down approach, starting already in the design phase of the project and later being used during day-to-day operations.

This approach brings several benefits. Besides having a good overview of what has already been completed, it also allows simulating the system as a whole even before having all the parts in place. In addition, it can also serve as a data exchange system between developers and different types of development and design tools.

The database can serve as an input to the installation and commissioning plan. Well designed structure and tools can use the information about the machine and output it in a way required during each project phase. For example, at the very beginning it can be a powerful tool to enforce the naming convention for the whole facility, while later it can provide relevant data for engineers etc.

During the operational phase of the machine, the database serves as an image of the complete system and allows for an easy system-wide modification and change tracking of the installation.

MAIN OBJECTIVES

The first version of BLED was initiated by the European Spallation Source (ESS) [2]. ESS, currently in the design update phase, uses several different lattice design tools. Therefore, it was very inconvenient to share the data among the designers, since some use TraceWin [3] while others use MAD [4] as their lattice design tool. Having proper import/export tools for the database, the exchange of data can become much easier, because all users have access to all parts of the lattice design.

Beside the aforementioned benefits of top-down approach and data exchange, BLED shall also be used during construction to generate the survey data for engineers. Using the information that is provided by the lattice design tools, most of the required data is already available and only needs to be exported in the appropriate way.

During control system development and operation the database stores the data for all the control points in the system and it is possible to generate configuration files for control system, such as EPICS db files or BEAST and BEAUtY XML configurations [5]. In addition to pure control system configuration, other system description files could also be obtained from the database (e.g. XAL XML configuration [6]).

The database can also be a valuable tool to non-engineering departments (e.g. purchasing department), which could use it to track the list of components that remain to be purchased as well as to keep track of the purchases (i.e. delivery dates).

Another important aspect of the database is the naming convention, which is advised to be introduced early in the project. Selecting and enforcing consistent convention presents a high organizational risk, because all teams must conform to the convention. With introduction of BLED, it is the database and the tools that will take care of properly named items and verify every entry.

DATABASE STRUCTURE

Database has been designed as a Java object model equipped with Java Hibernate [7] annotations. Once the model is created, Hibernate takes care of the construction of the complete database (and later communication), thus making the model completely independent of the underlying database technology. Initially ESS will use MySQL, but if at any later time a decision is made to switch to a different database, only the database driver needs to be replaced and Hibernate will generate a new database.

The database is designed to provide functionality to store the history of the entities, which are expected to evolve during development and operation. Every change in any of these entities is recorded in order to be able to at any time revert to a previous version.

Since no database can satisfy the needs of every user, special custom tables have also been defined, which can be used to create virtual tables and store additional data, which may not belong to any of the existing tables.

Current version of the database consists of approximately 60 tables divided into 6 packages.

The Main Package

The main object of the model is the *Subsystem*, which represents every single part of the machine, either be it a logical system such as the *Target* or a real system such as an *RF Cavity* or a *Beam Position Monitor*. The subsystems are organized hierarchically, where each subsystem can have none or several child elements. Within each single subsystem all children are sequentially ordered, considering their physical position in the machine. This allows for organizing all the accelerator components according to their location and/or functionality (e.g. *spoke* subsystem can have several *spoke cavities* subsystems as its children).

The Devices Package

Subsystem represents the base class for all components of the machine. Several other classes (e.g. *Magnets*, *Power Supplies*, and *Cavities*) that describe physical components are derived from this base class. These components form the actual machine. In case of the accelerator the devices within each parent subsystem are ordered either into the lattice (if applicable) or bound to a specific device, which they are related to (e.g. *Power Supply* is related to a *Magnet*, but is not a part of the lattice). Beside the machine related parameters, the database also contains information about the precise location and orientation of the components, allowing for generation of survey data or for simulating the particle beam.

The hierarchy of the components is such that one can easily add another type of device and create appropriate tables for it.

The Control System Package

Control system package defines all controls software related items. On one hand it provides the list of all control points, associated with appropriate subsystems, and on the other hand it provides information about installed configurations for archiving and alarm systems.

The initial version of BLED is primarily focused on the EPICS framework. Therefore, it contains the collections of all process variables, EPICS applications and templates, substitution values etc. as well as provides links to the versioning system (Mercurial), where the source code of the related software (i.e. device support) is located.

The Infrastructure & Deployment Package

All *IOControllers*, *Racks*, *Servers,* and *Computers* are also parts of BLED. Together with the infrastructure parts (*Room* and *Building*) they present the deployment view of the machine. The deployment view describes on which location and on which equipment the particular control system software is deployed.

The Wiring Package

Wiring is another important part of the installation. The database provides facilities to record each cable that connects any two ports in the machine. It is possible to define the type of the cable (including physical characteristics such as diameter) as well as the type of connector on all free ends of the cable. Not only linear (2-end cables) are supported, but also storage of Y or even more complicated cables is allowed.

The Parameters Package

Every machine or individual parts of it have certain characteristics, which are not physical descriptions of particular equipment, but represent higher-level information such as energy or minimum spot size on the target. BLED provides necessary tables to store such parameters and associate them with subsystems, responsible personnel, references etc. [8].

DATABASE INTERFACE TOOLS

In order to effectively manage and make the most use of the database, certain graphical tools will be developed. These tools will verify that each new entry in the database corresponds with the rules set forth by the designers. Using only such designated tools it will be assured that the database integrity is not broken during updates.

Besides the management tools, the system will also provide a set of tools necessary to generate particular configuration used either in the control system or any other part of the machine operation. Such tools represent the main interface to top-down development approach.

All management and configuration tools will be based on the web technologies (*Google Web Toolkit*). This will eliminate problems with running the software on different platforms as well as make it available to all registered users. Additional tools or updates to existing ones will not require users to do individual updates, since all the deployment will reside on the server.

Naming Convention Tools

Naming convention is an important issue, which has to be introduced early in the design of the machine in order to enforce it on every possible step. The naming convention tool will help the developers construct names for their control points so they will be in coherence with the facility-wide convention. The tool will allow the user to specify which subsystem a particular control point belongs to and suggest the appropriate name for it, based on the hierarchy of the subsystem.

Lattice Generators

During the design development of the machine several lattice design tools might be used. The interoperability of those tools is limited, because each of them works on special set of configuration files that are valid only for each particular design tool. In order to ease the process several parsers and data exporters will be created, which will parse these configuration files and import data into the database. Other users may then export the data into format understood by their preferred design tool.

Control System Configuration Management

ESS is estimated to have an order of one million control points. Construction and maintenance of such a large system can introduce significant problems if it is not managed and overseen properly. BLED tools will minimize these problems by verifying the completed work and help generating the missing parts of the system. Once the control points are defined, BLED will take care of generating configuration files for services (archive, alarm etc.), eliminating the risk of mistyping names or skipping a particular part of the system. Also, the effort required to do the task will be minimized.

At some time certain control points may change, effecting numerous configurations across the whole system. By having a well organized database, such cases can be easily spotted and all the necessary configurations would be regenerated.

Equipment Inventory

As mentioned before, BLED will also serve as the equipment inventory database. Therefore, tools will be created to allow editing as well as browsing and searching through the list of available equipment. It will display the details of each selected piece of equipment including its connections to other pieces of equipment and also allow tracking the location of the component.

Machine Parameters Management

Parameters management tools will assure that the parameters list is alive and that people responsible for each section will have access to the lists and be able to inspect and modify them [8]. Several criteria for searching the parameters will be established, such as for example subsystem or team criteria. In addition, facilities to export the data in various formats (Excel, PDF) will be provided.

CONCLUSION

Maintaining accurate information on several ten thousands of devices and a million control points can be a challenging work and cannot be done without proper tools. Therefore, a good management system needs to be installed and supported by extensive set of graphical and utility tools, which can save a lot of time and reduce risks.

ACKNOWLEDGEMENTS

We would like to express our gratitude to all the people involved on the BLED project, especially controls group at ESS, which was the main initiator of the project and funded the first developments.

REFERENCES

[1] Integrated Relational Model of Installed Systems – IRMIS (http://irmis.sourceforge.net/)
[2] European Spallation Source – ESS (http://ess-scandinavia.eu/)
[3] Ions and electron beams trace calculation tool – TraceWin (http://irfu.cea.fr/Sacm/logiciels/index3.php)
[4] Accelerator design and simulation tool- MAD (http://cern.ch/mad/)
[5] Control System Studio – CSS (http://ics-web.sns.ornl.gov/css/devel.html)
[6] XML Accelerator Model – XAL (http://sourceforge.net/projects/xaldev/)
[7] Object-relational mapping library – Hibernate (http://www.hibernate.org/)
[8] K. Rathsman et al, ESS Parameter List Database and Web Interface Tools, IPAC 2011 Proceedings