STRICT: STRING 2 INSTRUMENTATION AND CONFIGURATION TOOL

T.Ladzinski, R.Saban, R.Schmidt, CERN, Geneva

Abstract

The validation of the LHC systems at CERN has started with Test String 1. During its operation a lack of central repository, where both the configuration parameters for data acquisition system (DAQ) and instrumentation information could be stored, has been observed. With the commissioning of String 2 [1], with double the number of instruments, the need for such a tool has become even more evident. This paper presents a software framework built to store instrumentation data and to facilitate configuration of test runs. The back-end of the system is an *Oracle* database; while the front-end is an application developed using the *PHP* scripting language extension of the *Apache* web server. Synoptic diagrams of the installation have been incorporated into the system to help retrieve instrumentation data using a graphical interface. The user of the system may set the data acquisition parameters for a given experimental run using the same web interface. The run configuration data can be retrieved from the database as a text file, which is used to arm the DAQ.

1 THE RATIONALE

String 2 has been built to individually validate the LHC subsystems and to investigate their collective behaviour. At the same time, the experience gained with the assembly, tooling, testing and commissioning procedures has been vital to the teams preparing for construction of LHC. Similarly, the experience gained with the former system: String 1 and the observations made during its operation have helped specify requirements for new software tools.

String 1 lacked a centrally maintained repository of signal data. Such a repository is indispensable at commissioning phase, when project engineers concerned with the technical aspects of their subsystems desperately try to set-up the data acquisition and archiving systems. Unfortunately, very often they neither have the time nor patience to study, which acquisition channel corresponds to the signal they have named using their own terminology. Introducing a central database eliminates this and similar problems, enforces naming conventions and reduces possible erroneous recording of data. Furthermore, modern data acquisition systems, capable of recording hundreds of channels simultaneously, require both expertise and time to set them up. These settings vary with the tests performed and should ideally be prepared well in advance of an experimental run, after a thorough consultation with experts in various domains. Whereas this was not the case for String 1, for its successor, with double the number of recorded signals, the need for a system facilitating the DAQ configuration became evident and ad-hoc setting of data acquisition was no longer permitted.

2 SYSTEM DESCRIPTION

The core of STRICT application is a relational database, with more than 20 tables. Its thorough description is beyond the scope of this paper; instead, the most important concepts and implementation issues will be explained in the following paragraphs.

2.1 Instruments and Signals

Instruments are normally defined in engineering specifications. These documents have traditionally been maintained in the form of *MS Excel* spreadsheets. Voltage taps, temperature sensors, electrical heaters are all examples of instruments. String 2 is divided into several sections called string components. A dipole or a quadrupole magnet, a cryogenics line are examples of string components. Each record in the instruments table is uniquely identified by a combination of instrument name and string component, on which the given instrument is physically located. An instrument record contains additionally data about the user and specialist responsible for given equipment, its type, flange, connector and wiring.

Signals use one or more instruments to record one value e.g. a quench detector signal is a voltage difference in a magnet measured with two (or more) voltage taps. A signal record is similar to an instrument record, as the unique key is also formed with the name and string component. The name of the responsible person is also provided. Additionally, all information for the correspondence between the signal and the hardware channel of the DAQ system is stored in the signals table.

Thanks to the relational database organization, additional tables could be added to store data relevant to only some signals. Most groups decided to store conversion parameters within STRICT. One request went even further and demanded storing data about signal conditioners (linear amplifiers) in the database.

2.2 Signal Classes

The DAQ system is composed of two independent subsystems, one for high frequency recording, another for low frequency recording, the threshold value being 1kHz. Most of the signals within String 2 have been recorded using the later subsystem. Thus, for some 600 signals, six parameters describing the triggering of recording, two sampling frequencies and three intervals have to be stored (see fig.1). In order to ease this task, the concept of signal classes has been introduced. Signal class is a logical expression of common recording characteristics - same frequency, recording intervals and default trigger. Several classes may contain the same signals, e.g. all the quench heaters can be grouped into a class named *Quench Heaters*, additionally some of them may also belong to class *Dipole Quench Heaters*.

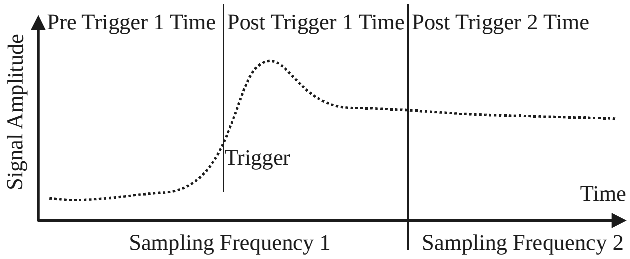

Fig. 1 Signal recording parameters.

2.3 DAQ Settings

Contrary to String 1, where the creation of DAQ configuration files has been an ad hoc process, the String 2 users can prepare the DAQ settings for a particular run well in advance of the actual experiment.

New settings creation is done in a wizard-like manner; i.e. the user introduces data in subsequent steps, aided by the system. First, the user chooses the signal classes that are to be recorded in a given run. If there are conflicts, i.e. there are signals, which belong simultaneously to two or more signal classes, the wizard proposes a solution. Two choices are possible: either keep the signal within only one of the classes or treat the signal individually. In the first case, the signal will be "removed" from other signal classes for this run (hence, we now talk of run specific signal classes), in the second case, it will be removed from all the classes and will become a "class" of its own. In the next step, assignment of DAQ parameters to each run specific class takes place.

Default trigger of a class is initially assigned to all the signals grouped in that class. However, a specific trigger can be assigned for a subset of signals of a given class located within one string component. The possibility of changing triggers in function of string component may help set up special configuration for such experiments as quench propagation tests.

Preparing the DAQ settings cycle includes initial DAQ settings introduction by the creator, elaboration phase, where all the experts concerned may modify the settings and production phase, where the settings are locked, configuration files generated and transferred to the DAQ system.

2.4 Software Implementation

The architecture of the system is schematically depicted in fig.2. The back-end of the system is a relational database (*Oracle ver.8i*), with appropriate table structure to store instrument and signal data, data acquisition settings and user data as well as trigger configuration information. A number of domain tables have been introduced storing lists of allowed values that the user can choose from (e.g. instrument type). Figure 3 shows the relational table structure, with domain tables coloured grey.

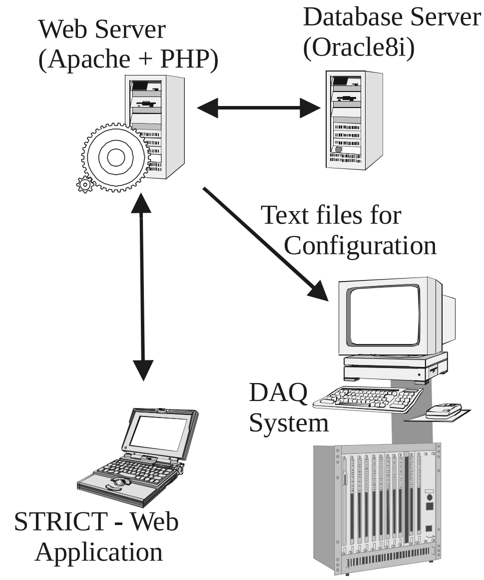

Fig. 2 System architecture.

As one of the goals was to enable consultation of data and preparation of experimental runs, easy-to-use remote access to the system was of primary importance. The database is therefore accessible via a web interface. A dedicated *Apache* [2] web server has been set-up in the String Control Room, extended with the *PHP* [3] module. *PHP* is a scripting language, similar to *perl* in syntax, suitable for generating web pages. Moreover, it has an extensive set of libraries to link with most of today's database systems.

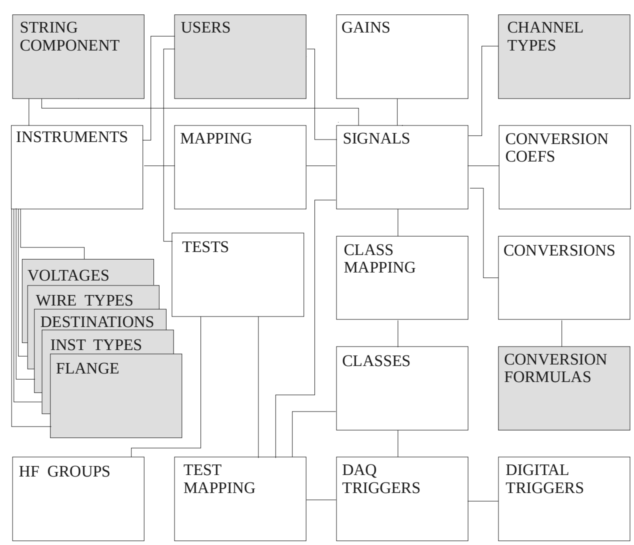

Fig. 3 Schematic diagram of table structure.

In general, the database is queried using the web browser and new run settings are prepared using the same interface. The resulting configuration files can then be transferred to the DAQ machine as well as signal conditioner gateway, using the file transfer protocol

(FTP). The choice of simple text files as means of configuration facilitated error detection. Thus, the configuration can be prepared remotely and in advance, and then stored on the target machine. The loading of configuration as well as arming of the system is done by the operator of the DAQ machine. This simple mechanism eliminates risk of remote intervention of non-authorized person and provides network independence at run-time.

2.5 Synoptic Diagrams

A number of synoptic drawings have been added to STRICT. Although these drawings have existed in various locations, both in traditional and electronic formats, having them grouped in one application is very useful when preparing acquisition runs or analyzing data. Moreover, by adding hyperlinks, it is possible to find information regarding an instrument or a signal stored in the database and shown in the synoptic with a simple mouse click (fig.4).

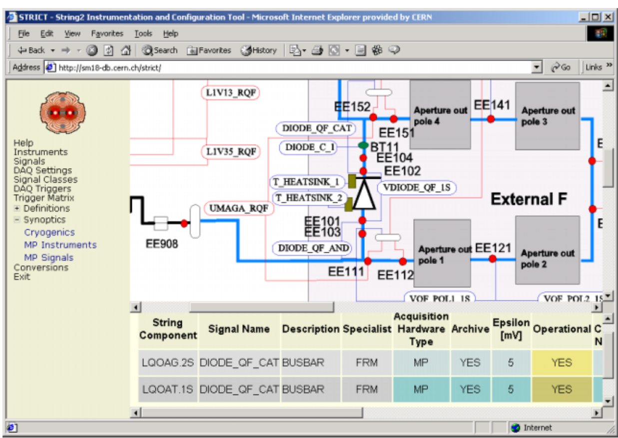

Fig. 4 STRICT window with synoptic diagram.

At the moment, two packages are widely used to prepare drawings at CERN: *AutoCAD* and *Micrografx Designer*. Inclusion of *AutoCAD* created designs in STRICT has been made possible by using the *AutoDESK WHIP!* plug-in [4] for *MS Windows* clients and *ZoomON Cadviewer* [5] for other platforms. Designer files are stored as simple GIF pictures, while custom written software has been developed to produce click-able image maps from postscript versions of the drawings.

3 CONCLUSIONS

STRICT has been greatly appreciated by users already in the commissioning phase [6], [7]. The interface proved intuitive, a central instrumentation and signal database resolved conflicts and standard naming conventions have been easier to impose. Similar technology could perhaps be applied not only to phase two of String 2, but also to the LHC, as the accelerator will be much less instrumented. However, one has to stress that STRICT limited itself only to the DAQ system and did not involve a vast number of SCADA signals.

The concept of signal classes proved particularly useful, enabling fast configuration preparation. The process of introducing a central repository did, however, encounter some problems. A database cannot replace a human coordinator and sometimes ‘fuzzy’ responsibility for various signals has been observed. Moreover, complex conversion functions proved hard to edit using the supplied web interface. Last but not least, the authors have underestimated the habits of users, accustomed to working with *MS Excel* spreadsheets. In the future, better import mechanisms should be implemented, to improve repository uploading from *Excel* files.

4 ACKNOWLEDGEMENTS

We would like to thank Davide Bozzini and Cédric Charrondière for maintaining, respectively, the registers of instrumentation and signals, which were used to initially populate the database. We are greatly obliged to Maciej Peryt for many discussions on system architecture as well as implementation choices [8]. Special thanks go to Adriaan Rijllart, who explained to us the details of the DAQ system. Last but not least, we would like to thank all the users from the teams participating in the commissioning of String 2 for providing their feedback.

5 REFERENCES

[1] F.Bordry et al., “The LHC Prototype Full-Cell: Design Study”, LHC Project report 170, 12 march 1998
[2] http://www.apache.org
[3] http://www.php.net
[4] http://www.autodesk.com/prods/whip/dwgembed
[5] http://www.cadviewer.com
[6] R.Saban et al., ”First Powering of the LHC Test String 2”, 17 Int. Conference on Magnet Technology, CICG, Geneva, Switzerland, 24 - 28 Sep. 2001
[7] A.Ballarino et al., ”First Results and Status of the LHC Test String 2”, this conference
[8] M.Peryt et al., “Generic Repository and Search Engine for LHC Equipment Test Data”, 7th International Conference on Accelerator and Large Experimental Physics Control Systems, Trieste, Italy, 4 - 8 Oct 1999