UPGRADING THE SPS OPERATIONAL SOFTWARE FOR THE LHC ERA THE SPS-2001 SOFTWARE PROJECT

Pierre Charrue, Bernard Denis,
Michel Jonker, Marc Vanden Eynden,
CERN, Geneva, Switzerland

Abstract

In December 1996, it was decided to set up a project to re-engineer the CERN SPS Accelerator application software to improve the operational efficiency and to cover the specific needs of the SPS as LHC injector. The mandate of this project is to provide a first working version of the software in June 2001 and a final version in July 2003. It will make use of the emerging CERN accelerator control infrastructure amongst which the new Java Application Programming Interface (this conference). This presentation reviews the reasons why the project was launched, its scope and objectives, the project planning and the methods used to run the analysis. The first results of the analysis phase and their implications for the application software, for the control infrastructure and for the accelerator equipment software services will be presented.

1 MANDATE OF THE SPS-2001 PROJECT

The application software used today to control the SPS is the result of 20 years of evolution of the accelerator and computer technology. Many applications have been designed for and by equipment specialists. Although parts of the application software are much appreciated by the operators, today software does not allow the most efficient exploitation of the SPS machine. Since the SPS will probably run for another 20 years, it has been decided to set up a project to do the re-engineering of the SPS application software to allow more efficient operation and to cover the additional needs for the LHC era, that is the usage of the SPS as LHC injector [1].

The purpose of the project is to provide an application software suite to operate the SPS and its transfer lines in the LHC era that is allowing LHC filling, fixed target operation and machine developments (MD's). An important benefit of this project is to provide a homogenous software suite to the control room.

The mandate of the SPS-2001 project consists of the definition of the strategy to operate SPS in 2001 onwards, the definition of the strategy to implement the new software and the production of all the software that will be used by the operation crew to operate SPS from 2001 onwards.

In order to provide a homogenous solution to the control room, the SPS-2001 project will specify the interface to all equipment that must be accessed for the operation of SPS, define the standard services that this equipment should provide, define a standard Man Machine Interface and specify general purpose services and their interfaces needed to build the SPS-2001 software; e.g. Data Management Service, Alarm Service, etc.

The SPS-2001 project will establish a communication path with the equipment groups and other entities that might be affected by the project. In addition, it will communicate SPS-2001 specific needs and requirements to other relevant projects such as the PS/SL Controls Convergence Project and its sub-projects (i.e. Java API, Middleware, Timing Convergence, etc.)

The SPS-2001 project will provide the strategy to operate SPS from 2001 onwards, the strategy to implement the new software, the SPS-2001 certification document, guidelines to produce equipment services as well as application software, a first working version of the application suite in June 2001 and a complete final version in July 2003.

2 THE LHC REQUIREMENTS

The LHC machine has specific requirements on the SPS operations that are driving the SPS-2001 project. These are fast and flexible cycle change, high machine availability and new operational concepts.

The most obvious one is what is called the Fast and Flexible cycle change. Changing from one cycle or supercycle to another is possible today in the SPS but takes a non negligible amount of time. The future SPS software should provide some facilities that will allow the change of cycle in a much more flexible and rapid way. This will allow high machine availability and switching from LHC filling to fixed target physics will be done smoothly and rapidly. But this implies that equipment should not include assumptions about the super cycle structure and equipment must manage resident cycles and that new functionality has to be implemented in cycle dependant equipment (such as load, unload, activate…).

Switching from one mode of operation to another requires new equipment surveillance to ensure that equipment is in the desired state for future mode changes. Subscription mechanisms would give the future software efficiency and flexibility.

3 RELATION WITH OTHER PROJECTS

The SLI project [2] provides all LHC requirements that should be taken into account by the SPS-2001 project.

One member of the SPS-2001 project is also member of the SLI project and the SLI project leader is considered as one of the sponsors of the SPS-2001 project.

The SPS-2001 is responsible for the SPS operational software but it relies heavily on other controls projects that are defining the Controls Infrastructure of the coming years. These projects are organized in the so-called PS/SL convergence project [3]. The JAVA API project [4][5] is responsible for defining a standard accelerator device model and then for building a Java Application Interface (API) deploying this model. The middleware project is responsible for providing a Software Communication Architecture and services allowing inter-objects communication and supporting, as first priority, the Java API Technical Specification, including the Standard Accelerator Device Model [6]. Finally, the timing project is responsible for building a reciprocal understanding of the two timing systems, of maximizing the sharing of the analysis, design and implementation of timing hardware, firmware and software components and of providing the new central timing facility required to operate the CERN accelerator complex in the LHC era [7].

4 WHAT WILL CHANGE - NEW CONCEPTS

4.1 New Sequence Concept

Several concepts used for the control of the accelerator have been re-visited and defined during the first analysis phase. The beam-process, cycle and sequence concepts are described here.

4.1.1 Beam-process

A beam-process is a specific manipulation of the beam within the context of an accelerator cycle. Examples of beam-processes are transfer, injection, capture/bunch, etc. Beam-processes are usually limited in space (transfer zone, ring) and/or time (injection, acceleration, extraction). A beam-process is defined by its characteristics. The major ones are the location (i.e. the parts of the machine involved), the time-extent and the set of associated beam control parameters. Furthermore, one can usually define explicit and independent performance indicators to beam-processes. (e.g. transmission, emittance blow-up...).

4.1.2 Cycle

A Cycle is a predefined set of beam-processes to transport (and transform) beam from source to destination. These beam-processes can have a predefined fixed length or they can have an undetermined length (accumulation, storage). The number of beam-processes in an accelerator cycle does not have to be predefined either; an example is multiple injection for beam accumulation.

4.1.3 Sequence

A sequence is a predefined group of cycles. An accelerator can be multi-cycling with several cycles executed in a sequence. The sequence of cycles can be a fixed repetitive pattern (super cycle) or a more flexible sequence of cycles adapting to the status of the users and equipment. To speed up sequence switching, several sequences may be resident in the equipment, and executable under control of the timing system. The beam scheduling system will respond appropriately to changing interlock and request conditions in order to select the sequence that makes best use of the available resources. One should further note that a given cycle could be present in more than one sequence.

4.2 Simple view of the new operating strategy

A. Before each period (TBD) physicists and OP crew define the physics program of the next period. This physics program will allow Users to issue Beam Requests.

B. All the corresponding sequences that should be used to handle this physics program are selected from the sequence database and the sequence settings are made resident in the SPS equipment.

C. Then, depending on the user and operators requests and the possible machine state, the active sequence is chosen within the resident sequences depending on the priorities of the client requests and operators veto. NOTE: This choice of the active sequence can be made via human intervention or automatically.

D. The central timing system is directs the scene by "playing" the active sequence in a loop until another sequence is chosen.

E. NOTE: The equipment is not aware of which sequence is running. It has been loaded with the settings corresponding to the possible cycles that could be run. The local timing system will warn them that the cycle x will be played soon and then send all the timing events/telegrams for the running cycle.

F. During operation, conditions may change either in the user or operators requests or in the current machine state (for example due to interlocks). Therefore, the active sequence becomes invalid and a new active sequence will be selected as explained in C above.

4.3 SPS-2001 Equipment Interface

The SPS-2001 project proposes to define a standard equipment interface that will be deployed on top of the proposed architecture given by the middleware project. This standard interface will be defined in conjunction with the equipment groups and the operation crew and will be the base of the SPS-2001 application software. It will define services that equipment can provide and also services that operation would like to see provided by the equipment. These services will be classified into groups of services, such as identification, global status information, state management, cycle management, cycle parameters, cycle functions or measurements.

4.4 Standard Man Machine Interface

Another piece of standardization that the SPS-2001 project would like to perform is the definition of a standard Man Machine Interface. This MMI will offer to the operation crew a standard environment and therefore a homogeneous view of the SPS operation. The definition and the implementation of this standard MMI will be carried out in the coming months in close collaboration with the operational teams.

5 THE APPLICATION SOFTWARE

The application software represents all that is between the MMI and the equipment interface and represents the biggest part of the SPS-2001 project. An Object Oriented Analysis and Design phase (applying Unified Software Development Process [8]) is currently under way [9] to define and then implement these applications.

The purpose of the Analysis phase is to provide a complete, consistent, readable, and revisable description of the software needed to operate the SPS accelerator in the LHC era.

The objectives of the Analysis exercise are to describe the functions the future SPS software will perform and to state what kind of performance and resources are required.

The scope of the analysis will cover all behavioral aspects of the future software, not its form. Issues related to software component design and implementation will not be pursued during this phase.

The actions required to reach the objectives of the SPS 2001 analysis phase will be organized around two main ideas, the identification of the functions the future system has to provide and the analysis of the behavior of the system for each function.

5.1 Identifying System Functions

This activity will focus on the identification - from the requirement point of view - of the functions the future system has to provide. We consider functions as the expression of the need rather than the expression of solutions. The aim of expressing the need through its functions is to find a better way to breakdown the concept and to clarify the perception of the problem. Once identified, the functions are then clustered according to hierarchies.

5.2 Behavior Analysis

This key activity of the analysis phase will concentrate on running practical scenarios for each function of the system. These scenarios will help to find the entities the future software will be composed of. Next, the relations between these entities and the services they provide will be studied in greater detail. Initiators of these scenarios (i.e. operators, external triggers) and contributors (i.e. external service providers) will also be identified during this behavioral analysis phase. This identification will help, amongst other things, to specify interfaces with external hardware or software system required to run the SPS as LHC injector.

Analysis will be successfully complete when scenarios for all fundamental (central to the application's purpose) system behaviors will have been developed and validated by the experts in accelerator operations and people responsible for the external systems that will interface with the future SPS software.

6. CONCLUSIONS

The SPS-2001 software project is an ambitious project to handle the new LHC requirements for the SPS as LHC injector and also to provide a coherent application software suite to operate the SPS. This task would not be possible without comprehensive analysis of the needs. Neither would it be possible without a good separation of responsibilities between all the parties involved. The definition of standard equipment access as well as standard Man Machine Interface will ease the main task of writing the application software.

REFERENCES

[1] The SPS-2001 project - Web site http://venice.cern.ch/~slaps/hot/sps_lhc/intro.html

[2] The SLI project - Web site http://nicewww.cern.ch/sl/sli/Sli.html

[3] The PS/SL convergence project - Web site http://hpslweb.cern.ch/pssl

[4] Java Controls API Technical Specification - PS/CO/Note 98-16 - P. Charrue, J. Cuperus, I. Deloose, F. Di Maio, K. Kostro, M. Vanden Eynden, W. Watson - 11 August 1998

[5] The Java API - Web site http://hpslweb/pssl/projects/javapi/javapi.html

[6] The middleware project - Web site http://hpslweb/pssl/projects/middleware/middleware.html

[7] The timing project - Web site - http://hpslweb/pssl/projects/timing/timing.html

[8] The Unified Software Development Process - I.Jacobson, G.Booch, J.Rumbaugh - Adison Wesley

[9] The Analysis Phase Plan of the SPS-2001 project -http://venice/~slaps/hot/sps_lhc/docs/docs.hml