ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S06SA03

The LEP Alarm System

M. W. Tyrrell
CERN - European Organisation for Nuclear Research
SL Division, Controls Group
CH-1211 GENEVA 23, Switzerland

Abstract

Unlike alarm systems for previous accelerators, the LEP alarm system caters not only for the operation of the accelerator but also for technical services and provides the direct channel for personnel safety. It was commissioned during 1989 and has seen a continued development up to the present day. The system, comprising over 50 computers including 5 different platforms and 4 different operating systems, is described. The hierarchical structure of the software is outlined from the interface to the equipment groups, through the front end computers to the central server, and finally to the operator consoles. Reasons are given for choosing a conventional, as opposed to a 'knowledge based' approach. Finally, references are made to a prototype real time expert system for surveying the power converters of LEP, which was conducted during 1990 as part of the alarm development program.

I. INTRODUCTION

The Large Electron Positron Collider (LEP) was constructed during the years 1983 to 1989 and is situated in a 27km tunnel of diameter 3.8 metres at a depth varying between 50 and 175 metres. It contains 4 experimental halls situated symmetrically around the ring of size approximately 80 metres long and 23 metres in diameter. From the very outset it was decided to survey the whole complex both for personnel safety and equipment status by 1 alarm system due to the sheer size of LEP and cost of system installation. This task was given to an 'Alarm Team'. A major milestone for the project was the use of a complete prototype for the 'LEP Injection Tests' in July 1988. By 1989, in time for 'LEP Switch On' the system had become stable and operational, but with an incomplete coverage of the complex and only a rudimentary display for information presentation. A continued development concentrated on improving the Man Machine Interface (MMI), and extending the scrutiny of the surveillance system while investigating alternative techniques to improve the overall system.

II. THE ALARM SYSTEM PROJECT

A. Definition

The alarm system can be thought of as a window through which operators can view the status of any part of the process. Here the process concerns the whole accelerator, equipment associated with personnel safety and the control system itself. By definition, if there is nothing wrong with the process, it is assumed that the overall state is good, and therefore no alarm information is presented. On the other hand, whenever a piece of equipment does not work, or an abnormal state is detected, a description of this situation should be passed to the system. This description is termed a Fault State (FS) and covers both warning and alarm situations. The alarm system concerns the acceptance, treatment and display of these FS's.

B. Organisation

The project began in earnest in 1987 and has been continually staffed by 1 permanent and, on average, 3 temporary personnel. During the 5 years of system design, implementation and development, 16 temporary personnel, each working on average 1 year have contributed 18 man years to the project. These people were all trained in computer science, apart from 1 who was an experimental physicist. The permanent member managed and coordinated the project which was divided into 4 areas: display and presentation of information to operators, a Central Alarm Server (CAS), an interface to the users responsible for the equipment, and a database.

As part of the organisation of the LEP project, it was stated that each equipment group should be responsible for its own equipment surveillance. It was envisaged that the main part of this task would be done in the equipment groups' local control environment called Equipment Control Assemblies (ECA's). Unfortunately, in reality little surveillance of equipment was implemented at that level. This required that the interface between the alarm system and the equipment groups, in many cases passed beyond the ECA right down to the level of the equipment. For this reason it was even more important to define exactly where each line of responsibility was drawn. In practice this was always done at the level of a database definition, either at the FS description level or a combination of a command/response definition to acquire an equipment state and its corresponding FS definition.

C. Influencing Factors

The main influencing factors were the following:

1. At the very beginning of the project various groups at CERN were evaluating the possible uses of Expert Systems (ES). At one time an 'Expert System Interest Group' was set up which encouraged a free exchange of ideas and helped keep abreast of developments. Although nothing of practical value resulted for LEP, one area, a project in the Controls Group of

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S06SA03

the Proton Synchrotron Division at CERN [1], did develop, and continues today.

2. Throughout the design and implementation of the alarm system, close collaboration was maintained with the group responsible for the safety system in the experimental zones, namely the General Surveillance System (GSS) [2]. Both system designs were exposed to the conventional approach, but early on in the GSS project, an ES shell developed at the Electricite de France, called Genesia [3], was evaluated and finally used. The alarm team decided against its use for the following reasons: original lack of portability, lack of internal control of inferencing, limited interface to the outside world, constraints on the use of variables, rule set generation time very long and no temporal reasoning facilities. One conclusion of this work was the realisation that such an approach encouraged the use of 'rules' to define logical relationships but that the maintenance of a large interrelated rule set is by no means trivial.

3. The environment in which the alarm system had to run was also a strong influencing factor. At the equipment level, equipment groups built their own ECA's containing different processors and operating systems. This included the 4 GSS systems, which for connection purposes were considered as ECA's. UNIX was chosen as the operating system for the control system with 'C' the programming language. To-date various elements of the alarm system run on 5 different platforms covering 2 operating systems including 4 flavours of UNIX.

4. The LEP machine falls under the jurisdiction of the French authorities which consider accelerators as nuclear installations. They must conform to strict safety regulations with respect to safety of personnel and particularly to radiation exposure. This meant that the design of the alarm system had to provide the necessary features required by these regulations, including formal FS definitions for safety, redundant FS transmission paths and the use of a fail-safe power supply network.

III. Control System Architecture

The control system [4] is based on 3 levels of processing. At the lowest level are the ECA's of the equipment groups. They are connected to the next level using the MIL-1553-B multidrop bus. The controls group is responsible for the interface between the ECA's and this level, including: the hardware and software interface; a command/response protocol for control; a local name server; and an alarm channel for passing alarm information. The intermediate level consists of Personnel Computers (PC's), 386 machines, known as Process Control Assemblies (PCA's). These act as concentrators of processing power both for equipment control and alarm handling. There may be up to 10 such machines at one LEP site. Each machine is either connected to a local Ethernet segment and then a Token Ring (TR) (IEEE 802.5) or directly to a TR which in turn connects to a Time Division Multiplexer (TDM) (CCITT-G700) system for long distance transmission. The TDM connects the various PCA's to local TR networks in the different control centres. It is on these local networks that the third level of processing exists in the form of consoles for control and the display of FS information.

IV. Fault State Characteristics

A FS has been defined as something wrong with the process. It is described as a triplet: Fault Family (FF); Fault Member (FM); and Fault Code (FC). The FF is a collection of similar parts of the process, exhibiting similar FS's. A FM of this FF is an instance of one of these parts of the process. The FC describes the problem. For a perfect FF, all FC's will apply to all FM's of that FF. As an example, in the services environment consider the fire detection system. All fire detectors are grouped under the FF 'FIRE_ZONE'. The FM defines uniquely each fire detector or circuit. The FC describes the problem: smoke-detected alarm level; smoke-detected warning level; detector under maintenance; detector out of service; and detector fault. For the machine, consider the power converters. They are all grouped under the FF 'POWER_CONVERTERS'. The FM is the name of each individual power converter and the FC describes the problem: faulty; timing error; local mains variation; tune loop control error; ECA system reset; spike detected; etc..

Although this triplet definition defines the FS uniquely, it is not sufficient for alarm management or operators receiving these FS's. For this reason a 14 field, character string was formally defined. The first 7 fields are obligatory and consist of the following: a string version number; the triplet; a flag indicating whether the FS has just become active, or that it has now terminated; description of the problem; and a priority indicating severity. The remaining optional fields enable a user to define more precisely the FS. A formal name was given to this string namely: User Ascii Version 1 (UAV1), which is used to describe FS's in ECA's , PCA's and any other computer performing surveillance.

An important aspect of a FS is time. A FS is considered to be 'active' during the time it has a status 'true'. Such a state has 2 times associated with it: the time at which it became active; and the time it terminated. During the interval between it is referred to as an Active Fault (AF) and is added to a list containing all AF's, this list being called the Active List (AL). An AL is maintained at each level of the alarm software. It represents the current state of the process as surveyed by that software level.

Not all temporal aspects of problems are covered by the concept of an AF, since there are very important 'events' which take place at an instance of time, and thereafter have no further meaning. Examples of these are: an ECA microprocessor 'resetting'; a spark in an electrostatic separator; and a software task 'timing out'. These situations are treated differently by the system. They are described as 'Instant Faults' and have a corresponding identification flag in the UAV1 string. Since they only have 1 time stamp associated with them, they cannot be part of an AL. Instead they are passed through the system and finally offered for display.

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S06SA03

V. Hierarchical Structure of the Alarm System

LEP has 8 access points distributed equally around its circumference. These areas are used for personnel access, concentration of general services, and points of localisation for the control system; in addition the even points house the 4 underground experiments. Control centres for LEP are distributed widely over the CERN site, with the machine control room situated on the Prevessin Site, and the Technical and Safety Control Room (SCR) situated on the Meyrin Site, some 3km away.

This geographical layout lent itself to a hierarchical alarm system approach both in terms of hardware and software. At the lowest level, ECA's provide either FS or status information to the PCA's; the 4 GSS systems of the experimental zones mirrors this behaviour. Included at the PCA level are computers responsible for the surveillance of systems, not connected to ECA's. This layer then passes the FS's to the CAS, where they are centralised and grouped according to operators' areas of interest, such as machine operation, safety etc.. Finally, if a console has been initialised to receive one or more of these FS areas of interest, the CAS will send all corresponding states to that console, where alarm software will display them.

VI. Alarm Software Within the PCA's

A. General Structure

There are 3 layers of software within a PCA [5]. At the lowest level there is the Low Level Alarm Server (LLAS). The middle layer concerns Surveillance Programs (SP's) which can be divided into 2 sub-layers: that dealing with Standard Surveillance Programs (SSP's) which always exist, and that dealing with User Surveillance Programs (USP's). Finally at the top there is the Local Alarm Server (LAS). All these layers communicate in a standard way using messages and all processes are either active, or waiting for a message and or a timer: they never die.

B. Low Level Alarm Server

The LLAS receives FS's from 'intelligent' type ECA's via the alarm channel of the multidrop bus. They may be the complete description of a problem detected and transmitted by the most intelligent class of ECA's, or a pseudo FS from a less intelligent ECA simply indicating that a change in state has taken place. In the latter case, the pseudo FS may contain the current state as data, which must then be interrupted at the SP level. The role of the LLAS is first to transform the received FS into a standard UAV1 string, then to extend it into a Standard Ascii (SAV1) string through the addition of the computer name, SP name and arrival time. Finally, using the FF and an internal correspondence table, generated from the database, the LLAS directs the FS to its corresponding SSP at the SP level.

C. Standard Surveillance Programs

A SSP deals with all instances of one type of equipment connected to a PCA. This allows logical analysis of FS's concerning a particular equipment area, which is usually completely different to that required for another equipment area. This leads in some cases to dedicated software within certain SSP's which is well separated from the standard software.

To improve the maintainability, the ability to accept changes, and to provide a 'template' for a SSP, each SSP is data driven using 2 standard flat files generated from the database. These files are used to build internal tables at initialisation. One contains all possible FS's considered valid for that SSP, together with a 'LEP Mode Mask' for each FS which indicates the applicability of each FS under each machine mode. This table is used for: the reduction of FS's within or across FF's; the treatment of oscillating FS's; and the possibility to inhibit, on request from a control centre console, the transmission of particular FS's. In all cases it is possible for operators to view remotely the underlying FS details. The second table contains command/response information necessary to contact the attached ECA's, and in some cases the correspondence between bits in an equipment status word and their FS description.

SSP's attached to intelligent ECA's use these tables to request periodically the state of each relevant ECA to verify the consistency of their AL's, rectifying any discrepancies found. If an ECA does not respond, it is considered a problem, and a corresponding FS is generated. For those SSP's receiving pseudo FS's, indicating only a change in state, the ECA must be accessed to retrieve the true equipment status for analysis. ECA's which have no alarm 'intelligence' are only capable of providing the status of their equipment and have no notion of what defines a correct as opposed to an incorrect state. Here the corresponding SSP uses the tables not only to poll the state of the equipment every few minutes, but also to define the relationship between the data received and the FS definitions.

The aim of the SSP is to arrive at a description of the problems concerning the equipment for which it is responsible, using FS descriptions. This result is stored in its AL and all changes are passed to the next software level, the LAS. All software concerning flat files, maintenance of the AL, acceptance of FS's from ECA's and passing the result to the LAS is contained in an alarm library. This, together with a SSP 'template' is offered to assist in building SSP's.

D. User Surveillance Programs

A USP offers users more independence to survey their system and often concerns equipment not connected to ECA's. This approach is used in computers dedicated to surveillance as

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S06SA03

well as in PCA's. Wherever it is used, the strategy is the same. Each USP surveys its system and arrives at a result which is transmitted as a SAV1 string to a corresponding SSP. Periodically, contact is made between the SSP and USP to align the AL of the SSP with that of the USP and to verify that the USP is still functioning.

E. Local Alarm Server

All FS's generated or dealt with in a computer will finally end up in the LAS, having passed via a SSP. The contents of the AL of a LAS represents the overall state of all equipment surveyed by that computer. This is in contrast to the contents of the AL of a SSP which only represents the state of the equipment monitored by that SSP. To ensure that the contents of the LAS AL equals the sum of the AL's of all SSP's in that computer, the LAS periodically demands the state of each SSP, correcting any discrepancies found.

It is the LAS which provides the external connection to the CAS, via a Remote Procedure Call (RPC) [6]. All changes of state in the LAS are immediately sent to the CAS.

There were 2 reasons for introducing the LAS level. The first was to reduce the number of physical connections to the CAS, as there were multiple SSP's at the level below. Secondly, it was to perform logical analysis on the FS's concerning all the equipment connected to that computer. For the moment no further analysis of FS's is performed at this level.

VII. The Central Alarm Server

The CAS [7] represents the hub of the alarm system in that it receives FS's from all computers performing surveillance tasks via LAS's and distributes them to the various control centres for display to interested parties. Within the CAS there are a number of processes running, all communicating in a standard way using messages. These processes are concerned with: the management of the AL, which represents the state of the whole process; providing a 'backup' to each LAS to ensure AL consistency; access to the central alarm database which runs on-line within the CAS; archiving all FS's received; and finally dispatching relevant FS's to consoles.

It is the database which is the centre point for the processing within the CAS. All FS's known to the alarm system are present in the database. Any FS arriving from a LAS which is not known is placed in a trace file for further investigation.

When a FS arrives at the CAS it is in the form of a SAV1. The triplet FF/FM/FC is the key used to access the database, the remaining information in the SAV1 representing the dynamic part of the FS description. A number of relational tables are accessed and 2 types of static information are retrieved. One concerns FS details such as: person responsible, location address, installation concerned, in all, 10 fields. The second concerns information relating to who might be interested in the FS. This organisation is done within the database by grouping each FS into one or more 'categories', which represent areas of interest of the various users of the system. Examples of category definitions are: one for each equipment group, safety, machine operation, technical services etc. A match of a FS to one or more categories will return 6 fields of independent information for each matched category. This information concerns: description of the problem; action to be taken; priority, very serious, serious, and warning etc. All information is tailored to each category. For example a fire alarm attached to the safety category would have in its 'action' field: 'Immediate Intervention', since it is the safety services which deal with that type of problem, whereas the same FS associated with the machine operation category would be more for information or in some cases require the beam or equipment to be switched off depending on the location of the fire. Naturally FS's associated with machine operation would not be attached to the safety category.

As a result of this database access, a list is made of all the categories associated with the FS. It contains the SAV1 and both parts of the static information for each category. This represents all information known to the alarm system for that FS.

Any user who would like to receive alarm information at a console must run an initialisation which asks what categories of information are required. This operation informs the CAS and thereafter any FS received by the CAS which is associated with any of these categories will automatically be sent to that console. All communication between the CAS and consoles is made using RPC's.

VIII. Reception of Alarms at a Generalised Console

For the 'Injection Tests', a very simple, dedicated console was built using a 286 machine to receive FS information. It was soon found to be inadequate and it was decided to incorporate FS reception into the generalised console manager which was under development for workstations used in the control centres.

The Console Manager (CM), completely manages a workstation from 2 dedicated lines of 'icons' at the top of the screen, referred to as the CM banner. They provide information about LEP and allow the execution and control of multiple application programs, including the reception and display of errors encountered by these programs. To interface to the manager it was decided to dedicate an 'icon' within the CM banner and define it as a toggle with 2 states described by 'icon' texts: 'SHOW ALARMS' and 'HIDE ALARMS'. When a console is first initialised, only the CM banner is visible with the alarm 'icon' indicating 'SHOW ALARMS'. In this state no application programs are running.

To initialise the console to receive FS's the 'SHOW ALARM' 'icon' is selected which brings to the foreground 2 alarm windows and changes the 'icon' to 'HIDE ALARMS'. If the "HIDE ALARMS' 'icon' is selected, the CM removes the 2 alarm windows, but keeps them active in memory. One window has a single row of alarm 'icons' at the top and

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S06SA03

displays the AL; the other displays the instant faults. An alarm 'icon' called 'configuration' is used to configure the console for FS reception. This displays all the categories existing in the CAS. A selection of categories is then made which corresponds to those FS areas of interest. An 'apply' function then sends this request to the CAS, where a search is made for all AF's corresponding to that selection. All information concerning these FS's is then sent to the console. Thereafter any FS changes within these categories will automatically be sent to the console.

Alarm software within the console receives these states, places them in an AL, displays them in the active FS window, and manages a local archive. Information displayed is the date, FF, location address, FM, and fault description, all of which occupy 1 line. FS's are displayed in order of priority and time, with each priority having a different colour. The default is to display the most recent, highest priority FS's, but scrolling is possible.

Using the alarm 'icons', various operations can be performed on a displayed FS such as: display the dynamic and static information relating to that FS; acknowledge a FS by displaying it in inverse video; inhibit a FS, which sends a command to the appropriate PCA and SSP to flag and terminate that FS; etc. Another set of alarm 'icons' allows FS's to be re-enabled; the screen to be printed; scanning of the local archive; and creating a 'Test' alarm which checks most elements of the alarm system, finally arriving as a displayed FS in the active alarm window, and terminating automatically 20 seconds later.

Periodically the console alarm software requests a 'backup' to the CAS to verify its AL. If the console cannot access the CAS for any reason, it colours the alarm window blue and prints a message that the console has lost contact with the CAS. This is the final link in the chain which verifies the correct functioning of all parts of the alarm system from the point of generation of the FS in an ECA, right through to the display of that FS in the control centres.

To allow an operator to use the console to run other application programs, but at the same time be informed of any FS changes, the SHOW ALARMS 'icon' is used to indicate the arrival of any new FS.

Instant faults are not categorised. A console can either be initialised to receive all or none. The display is again 1 line per FS and works in 'roll over' mode with a scrolling facility.

Any console in the system can run the CM and initialise to receive FS's. This provides a very flexible method to connect to the alarm system. In practice we run with 6 permanent connections and up to 6 temporary ones.

IX. The Alarm Database

A. Overview

The database is the key to the overall system. Without it management of the system and interfacing to the equipment groups would not be possible. All FS's that can be generated, including all static information used to describe these states, are contained in the database. Relationships between FS's and categories, LEP states and equipment states are established. To enable all this information to be maintained, and at the same time use it in a coherent fashion, 2 relational databases with the same internal structure are used, one running on a centrally maintained VAX and the other in the CAS. It is the VAX database which is used for maintenance and interfacing to the equipment groups and is considered the 'master'. Archiving of all FS's arriving at the CAS is also maintained using both the database on the CAS and VAX.

B. The Master Alarm Database

Each equipment group interfaces with the alarm system through a standard flat table called the 'Interface Table' (IT), which was defined by the alarm team. It contains a complete description of all FS's generated by the equipment group. Roughly 90% of the column definitions are common for all equipment groups. The rest concern equipment specific information like data and FS relationships which differ widely. A set of scripts has been built by the alarm team which checks the consistency of IT's both with respect to themselves and the alarm database. The scripts must be run by the equipment groups before the IT's are used by the alarm system. Responsibility for these tables and the way they are interfaced to the equipment databases, lies with the equipment groups.

Preparation for a database update consists of: verifying any changes made to IT's; using these tables to generate all necessary flat files for those SP's affected by the changes; and updating the master database from the IT's. At this point the CAS is stopped, the database loaded, and restarted. All affected SP's are also stopped, internal tables initialised from flat files, and restarted. This operation takes about 15 minutes.

C. The CAS Database

The database running in the CAS is used only in 'select' mode. For each FS arriving, the database is accessed to: check its existence in the database; append all static information known about the FS; and classify the FS according to the user 'categories' of interest.

D. The Central Archive

Each FS that arrives at the CAS is archived. The archived information consists of the elements of the SAV1 string which is stored in a flat table. This means that to complete the description of a FS in the archive, access must be made to the alarm database. To avoid the management problem of storing archived information, a minimum archive is stored on the CAS. Each day, the previous day's archive is automatically transferred to the VAX, and removed from the CAS. The VAX manages 2 alternating archive tables which are 'record' limited. When the current table reaches its limit, it is copied to a file and the other table becomes the current archive. In this way a

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S06SA03

continuous archive is available on-line. Facilities to access this archive from consoles initialised to receive alarms are available.

X. SPECIAL REQUIREMENTS FOR SAFETY

In order to satisfy the safety requirements, 3 main areas had to be considered. First, all FS's associated with the safety of personnel had to be categorised according to the action required. These actions were: immediate intervention by the safety service, approximately 500 FS's have been defined for this category; immediate intervention by the technical service; and intervention by the technical service during working hours.

Secondly, the transmission of FS's to the SCR had to consist of 2 independent paths: one using the control system to transmit details; and the other, called the 'redundant' channel consisting of 1 summary state per LEP point transmitted via a 'hardwired' line to a synoptic panel in the SCR.

Thirdly, all the active components in the computer and hardwired transmission paths to the SCR had to be powered by secure electrical power. To reduce this equipment to a minimum, 2 computer networks around LEP were installed: one for the machine and one for safety including the services.

XI. EXPERIENCE WITH NEW TECHNIQUES

As a continuation of our efforts to explore the possible uses of ES's within the LEP alarm system, a pilot project [8] was launched to build a prototype surveillance system for the power converters of LEP. This work followed a thorough investigation of the commercial market for products which were suitable for our environment, and which could do as well, hopefully better, than our running conventional system.

We were looking for a modular, 'real time' ES running in a UNIX environment with the ability to link to standard commercial software. The 'real time' aspect was not so much the response time, although that was important, but rather the possibility to reason over time using temporal constructs. Modularity was important because we already had equipment data acquisition, networking, etc.

Finally we found a system which seemed to have all desired features, as well as an interface to a graphical package which we were already using. It was an American product and perhaps because it had no established European agents, we found it impossible to get the necessary technical support to allow us to continue with the product.

This was a major set back, but we decided to visit, for the second time, a commercial exhibition of ES's in France. A European product which seemed ideal was found. Training was arranged for 2 of our personnel, and a prototype prepared for the LEP power converters to be used in an evaluation of the product for one month. The first month was spent in trying to load the product, and thus little work was done on the prototype. An extension of 1 month was arranged for the evaluation and when work finally started on our prototype it soon became clear that the product was not operational and, apart from numerous bugs, a number of the advertised facilities either did not work as described or did not work at all. Subsequently we found that the product was no longer on the market. This concluded our work in this area to-date.

XII. CONCLUSION

The LEP alarm system today provides an important, reliable facility for machine operations, and for technical and safety services. It has shown itself to be flexible to equipment changes and capable of accepting upgrades gracefully. The components from which it is built are well interfaced and can be independently changed. Capacity of accepting new FS's has been demonstrated recently by connecting the complete technical services of the Meyrin Site, some 5000 states, representing 6% of the current CAS total.

The decision to pursue the conventional approach was correct and there remains doubt that the system would have been ready in time if the alternative approach had been taken.

XIII. ACKNOWLEDGEMENTS

I would like to thank the many people who have contributed to this project, especially: Tuanhua. Huang*, B.Marchal*, and L.Mantegazza for the database; J.M.David*, F.Rodor*, and J.M.Ortholand for the CAS and MMI; J.P.Forestier* for the PCA; and C.A.Cameron for research into new techniques and continued development on the PCA.

* people no longer at CERN.

References:

[1] J.Fuchs,P.Skarek,L.Varga,E.Wildner-Malandain,Integration of Generalised KB-Sytems in Process Control and Diagnosis, 'SEAS' Conference, Lausanne, Switzerland, April 8-13, 1991.

[2] F.Chevrier, Safety Controlled By An Expert System On Experimental Sites In High Energy Physics, Pergamon Press, Operational Expert Systems Applications in Europe, 1991.

[3] J.Ancelin,J.P.Gaussot,P.Legaud, A Real Time Knowledge-Based Alarm System EXTRA, AI & other Innovative Computer Applications in the Nuclear Industry, 1987 Snowbird Utah, Plenum Press, New York & London.

[4] P.G.Innocenti, The LEP Control System, Conference on Accelerator & Large Experimental Physics Control Systems, Vancouver, BC, Canada, October 1989.

[5] J.P.Forestier, The PCA Alarm Software Environment & Surveillance Software, CERN/SL/CO/Note/91-15 (unpublished).

[6] P.S.Anderssen,V.Frammery,Tools for Remote Computing in Accelerator Control,Conference on Accelerator & Large Experimental Physics Control Systems, Vancouver, BC, Canada, October 1989.

[7] F.Rodor,The LEP Central Alarm Server & Console Interface,CERN/SL/CO/Note/91-7/8,(unpublished).

[8] C.A.Cameron,LEP Alarm Group Expert System Project Status Report,CERN/SL/CO/90-122,(unpublished).

Trademarks:

UNIX is a trademark of AT&T Bell Laboratories.