EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN/PS/91-52 (CO)
November 1991

Workstations as Consoles for the CERN-PS Complex, setting-up the environment

P. Antonsanti, M. Arruat, J.M. Bouché, L. Cons
Y. Deloose, F. Di Maio

Paper presented at the International Conference on Accelerators and Large Experimental Physics Control Systems

Tsukuba, Japan, 11-15 November 1991

Workstations as Consoles for the CERN-PS Complex, setting-up the environment.

P. Antonsanti, M. Arruat, J.M. Bouche, L. Cons, Y. Deloose, F. Di Maio.
CERN PS - 1211 Geneva 23 - CH

Abstract

Within the framework of the rejuvenation project of the CERN control systems, commercial workstations have to replace existing home-designed operator consoles. RISC-based workstations with UNIX® , X-window™ and OSF/Motif™ have been introduced for the control of the PS complex. The first versions of general functionalities like synoptic display, program selection and control panels have been implemented and the first large scale application has been realized. This paper describes the different components of the workstation environment for the implementation of the applications. The focus is on the set of tools which have been used, developed or integrated, and on how we plan to make them evolve.

I. INTRODUCTION

The current control system of the PS complex is based on 16 bit computers which will be replaced because of obsolescence of the hardware and system software. A rejuvenation project is in progress for upgrading the different parts of the control system: hardware interfaces, process computers, communications and operator consoles [1] [2]. UNIX has been selected for both the console layer and the process layer.

During this first 3-year period (1989-1991), a workstation infrastructure has been set up and the basic building blocks of the programming environment have been provided. The first large scale application, the hadron injection process into the PS, has been realized for the 1991 PS complex start-up and extended during this time [3]. The workstations have been installed in the control room and new man-machine interface had to be defined [4].

In the context of such an evolution, the first task is to compose a base environment whose major parts are: hardware, system software, data-base management system, equipment-interface and user-interface. From experience, we were very concerned about getting as many functionalities as possible from this layer in a "safe" way: we wanted to minimize system development and be confident in the future of the environment.

The second task is to provide generic applications to support functions like console management, error handling and all direct interface with the equipment: synoptics of parts of the machine, parameter tables and control panels. This has been achieved through collaboration between the controls group and the operation group.

The third task is to integrate into the environment additional user-oriented tools for the production of specific applications and for simplification of generic tools. These tools are mostly from the commercial market and therefore it is certainly the more fastest changing part.

II. BASIC ENVIRONMENT

A. Hardware infrastructure

For operation and for development, DEC™'s RISC-Ultrix™ workstations are used (about 50 in 91/92). We use common configurations with only network interface (i.e. no direct VME, CAMAC or GPIB) and local disks for virtual memory and temporary files only.

The central facilities consist of servers providing the following services: workstation system files, user files, data-base and time-sharing servers. Central time sharing servers are used mainly for resource hungry software (hardware or administration) which are transparently available on office workstations by means of the X-window network facilities.

Each local sub-network includes a regional server supporting local workstations, DSC[1]s and data. These servers are high-end workstations with SCSI disks.

One important characteristic of our current architecture is that in order to cope with man-power resources for exploitation, we opted for a very homogeneous environment. Every operation critical system is, for the time being, from a single vendor and covered by a single maintenance contract (hardware and software). This has been very efficient. However, for the sake of real vendor-independence, software portability and commercial relations, mixing vendors would be profitable, especially for tasks which are not exploitation-critical.

Another characteristic of this architecture is that our newest servers are enhanced workstation configurations instead of "mid-range" systems with high performance bus, fast dual-ported disks, etc. This is due to the increasingly faster obsolescence of the hardware and the fact that Ultrix does not

---

® UNIX is a registered trademark of UNIX System Laboratories.

™ X Windows is a trademark of Massachusetts Institute of Technology.

™ Motif is a trademark of the Open Software Foundation.

™ DEC and Ultrix are trademarks of Digital Equipment Corporation

[1]DSC are VME-based process computers with Real-Time UNIX.

provide yet full benefit of DEC's special hardware such as BI and DSSI.

B. System Software

The base system software is the common Ultrix distribution: UNIX programming environment, TCP/IP, NFS, etc. with minimal additions from public-domain or commercial sources. To remain vendor-independent, we try to follow the evolution of industry standards as close as possible, especially OSF[1]'s products. We have replaced DEC-windows™ with Motif since 1989. We did not anticipate the distribution of the OSF/1™ operating system nor of DCE™ (Distributed Computing Environment), but we will use the vendor's versions.

The basic programming environment on our platforms consists of a C programming environment, composed of the UNIX tools (compiler, debugger, Make, SCCS...) with the addition of GNU Emacs[2] and of interactive tools, like DEC-Fuse™ .

C. Data base

Historically, the first major additional component to the base system software is a data management system. We are using a commercial relational data-base: Oracle™ which is the standard database in use at CERN. Most of the control system data is managed through it: equipment definition, alarm codes or computer definitions, for instance.

A client interface is available on every system. Forms and SQL are used for the various software exploitation tasks: software module additions, equipment updates, etc. Programs manipulating complex or archivable data, like the beam's cycles editor and the error servers use a central data-base via embedded SQL.

A dedicated server is providing "on-line" Oracle service for the two accelerator divisions. In addition, critical and read-only data which require efficient access, like equipment description, are distributed from Oracle into dbm[3] data-bases on the different sub-networks.

D. Equipment Interface

The next building block is communication with the hardware. An equipment access interface is available on process computers: old ones (Norsk Data® and PDP11®) and new ones (DSC). Remote procedure calls (RPC) are used to communicate with the process computers from the workstations.

The equipment interface in the CERN-PS complex is a well-defined, strict syntax interface. The arguments of the main procedures are: equipment identifier, "property" (selector inside a fixed set), event condition (CERN-PS timing) and data. The data is constrained to simple types (char, int, double, etc.) and to one dimension only. There is also a very limited set of procedures.

All the equipment definition (name, class, host computer, number, description, etc.) is maintained in a central data-base.

These two aspects have many advantages for integrating equipment access into any environment: small number of procedures and simple argument types make it easy, even in a commercial tool like a spreadsheet. In addition, once all the different communication channels are supported, any equipment defined in the data-base is accessible, without explicit knowledge of the network topology.

The equipment interface in the workstations is a local dispatcher to the equipment interface in the process computers. It provides network abstraction: the equipment identifiers have no correlation with their physical implementation (network, host...). This is mandatory in order to be able to move equipment from old computers to new one without interfering much with the applications.

An additional software layer is provided for generic applications which need to operate on a wide range of equipment types without integrating equipment-specific code. For example, how to display the status of a power supply or how to control it, need not to be re-defined in each generic application. All specific data and code related to the different equipment class, like labels, control words or transition functions are maintained in this library layer.

E. User Interface

The last major part of base environment is the user-interface.

This part is mainly composed of the Motif tools: interaction objects ("widgets") like buttons, menus or dialog windows, general libraries and a user-interface definition language (UIL). Motif is an additional layer on top of X-window and we try to ensure that programmers need only use this upper layer (i.e. minimal use of X-lib).

The basic widget set in the Motif distribution address most of the user-interaction issues. However, it does not provide many tools for data presentation nor for complex parameter control. These two parts requires programming at a lower levels: X-window (Xlib) or toolkits "intrinsics".

One solution in avoiding the duplication of such functions by different programmers is to add widgets for unsupported functions. Adding library functions like graphs in the form of additional widgets has the benefit of providing a uniform final programming environment. As no widgets were available as a common well-accepted solution, we developed our own widgets for these functions.

---

[1]OSF : Open Software Foundation

™ DEC windows is a trademark of Digital Equipment Corporation

™ OSF/1 and DCE are trademarks of the Open Software Foundation.

[2]GNU Emacs is a programming oriented editor from the Free Software Foundation.

™ DEC Fuse is a trademark of Digital Equipment Corporation

[3]dbm is a simple data-base system in UNIX

We provide programmers with widgets for data presentation: alpha-numeric or graphics which also support some major features in our environment like high refresh rate without blinking or variable parameter types.

We also provide a single widget for parameter control with dedicated functionalities like familiar interface (compared to previous hardware devices) or multiple input modes (mouse, arrow keys, numeric key-pad..).

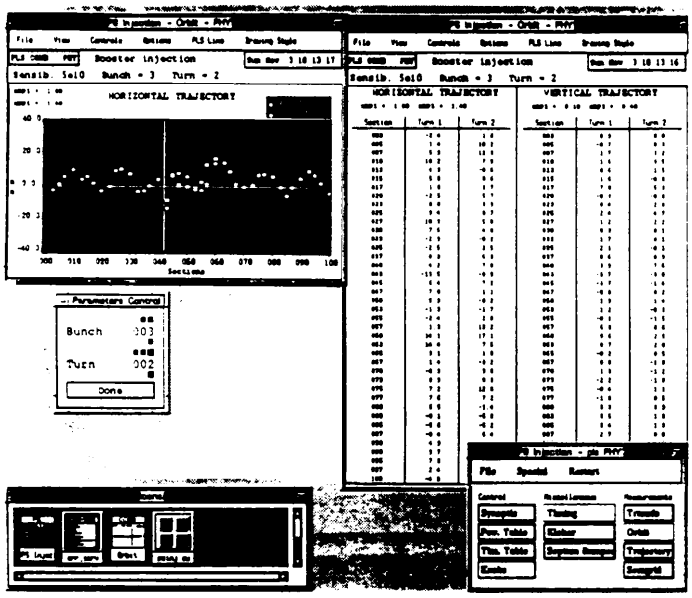

Figure 1. Motif application using home-made widgets

The development of widgets is a rather costly process which requires some proficient programmers but developing these functions as widgets has simplified greatly most of our Motif applications.

For end-users, the Motif programming environment is a very complete environment but requires some specific training. Courses have been organized and a support activity has been provided.

We are now introducing interactive editors for producing user-interface definition in a Motif-standard format: UIL or C. Such tools are available from various sources, like DEC or Siemens. If strictly based on Motif, without any specific data-structure, library or language, they have limited prototyping facilities but produce tool-independent applications.The major work in integrating these tools in our environment is to include our "user-defined" widgets which is an extra cost to their development.

III. GENERIC APPLICATIONS

Generic applications are programs which are not bound to a specific operation nor to a specific piece of equipment. The two main categories are general console utilities and equipment oriented applications.

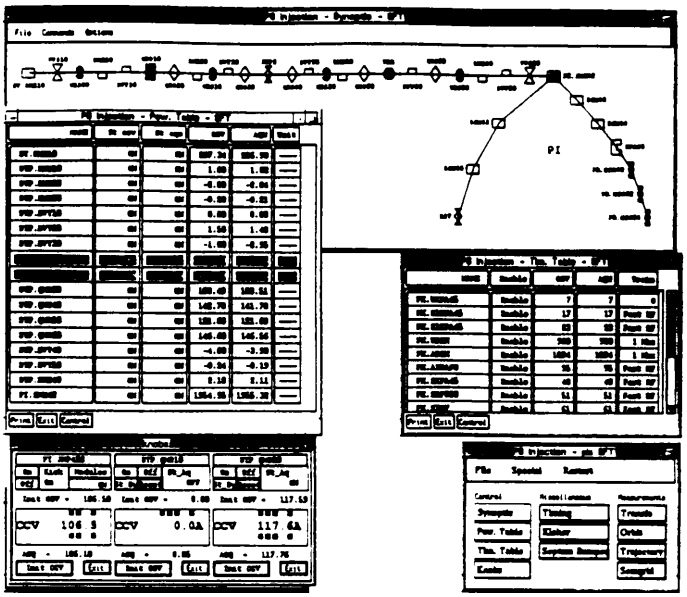

Figure 2. Generic applications

A. Application management

A console utility is required for handling the different functions available when the user logs in: program presentation, global condition selection and general facilities control (alarms and errors).

Some window management is also required in addition to the functions which are already available via the Motif window manager. The initial position and size of the windows need to be managed in order to distribute the application windows on the screen properly. The windows of the different programs which belong to the same operation context (same accelerator, same condition) also need to be handled as a set: common identifiers in the title bar and global iconification/raising/destruction. In addition, starting an application, bringing its window to front or "de-iconifying" it are connected to a unique interaction element.

One major issue in this area is window overlapping. In our context, the set of concurrent application which an operator will activate is not well defined; it is then difficult to define a fixed layout. Therefore, screen zones are defined mainly through program types. For example: control panels should appear in the bottom left corner of the screen, while equipment tables are on the right part.

B. Parameter tables

The first equipment-oriented generic application is a dedicated application for displaying operational parameters in a tabular format.

Each piece of equipment (power-supply, timing, etc.) belonging to a user-defined "working set" is presented along with a list of equipment-related values like status, control/acquisition value, unit and a color indication. The tool can be set to refresh the data every machine cycle if necessary (1.2 s).

The functions of this tool are multiple: to present the list of equipment, to support the selection by the user of a particular piece of equipment, to provide an interface for simple commands (On, Off ...) and to activate control panels on request.

C. Synoptics

Another implementation of the same kind of functionalities are synoptics presentations of a part of the machine.

Synoptics present the major and critical control element of that part of the machine and indicate their status by means of the color. They also present the position of some beam monitors. Like parameter tables, they include selection, simple commands and control panel activation.

These displays provide less detail than tables (no current values) but in a more synthetic way. They are used mainly for beam-transfer sections and not yet for circular parts.

The major problem for their implementation is that synoptics require user-interface information, which is not part of the equipment description available in the control system data-base. In our context, two types of information were not registered in data-base: the topological information which is used for positioning the icons and the type of physical element which is necessary for selecting the icon to display.

We opted for a data-driven solution : topological information and equipment specifications are entered in a synoptic-specific file. Another synoptic application realized for the control of the Proton Linac [5] includes a graphic interactive editor for entering this information.

Interactive editing, based on home-made tools or on commercial ones (data-presentation packages or user-interface editors), provides the end-user with a complete solution for building the synoptics themselves. Otherwise, one more maintainable solution is to exploit central data-base information shared with other programs, like AutoCAD™ or MAD but such a database does not exist yet in our context.

D. Control panels

A third important generic application is the direct interaction with pieces of equipment by means of "control panels", which are dedicated dialog boxes.

The user can activate individual control for a particular piece of equipment (power-supply, kicker, gun...) from different parts of the environment: from the application management layer, via buttons, from the parameter tables and the synoptics, via double clicks on the icon or via menus. Individual control activation can also be implemented in user-tools, like spreadsheets by means of an external function.

Wherever the request is made from, the control panels are handled by the same server application. It was implemented in this way for the sake of homogeneity and interface simplicity. As an additional homogeneity feature, all the control panels activated from the same operation context appear in the same main window (i.e. "shell window"). The exact layout of the control panels depends on the type of the equipment; however, parameters are usually presented in a similar way : name, buttons for discrete commands, display of the status, a "wheelswitch" widget for controlling the continuous parameter, the display of the parameter acquisition, etc.

We are modifying the tool in order to be completely data-driven : we do not want to have to extend the tool for each new type of equipment. Therefore, all the data and code which are necessary for the user-interface are moved to the equipment-interface library.

IV. USER-ORIENTED TOOLS

The basic Motif environment is very suitable for producing interactive applications. However, end-users as well as generic application makers can greatly benefit from the integration of higher-level or more user-friendly tools.

Most of these tools are from commercial sources and, as Motif is a rather recent market, the majority of them are just starting to be available in production version.

The integration of a commercial tool usually requires an interface to our specific functions, like equipment-interface and to provide support to end-users if possible.

A. Nodal and Console Emulation

The first user-oriented tool is CERN-sourced. Nodal is an interpreted language which is widely used in control applications at CERN.

The workstation version includes the equipment interface and some user-interface facilities based on Motif [6].

In addition, most of the facilities which were supported on the previous consoles (touch-panels, alpha-numeric displays, knobs) are available by means of emulation through compatible functions.

Nodal was the major programming environment for the consoles and is very familiar to CERN's staff. The Nodal environment on the workstations provides a very valuable way of porting existing applications and introduce many users to the workstations.

Having emulation facilities is critical in order to be able to transport, in a short time period, the whole set of applications of one accelerator from traditional consoles to workstations while focusing only on a few critical parts of them. It also adds some familiarity in the end users context.

B. Data Presentation

In the X-window/Motif environment is now possible to acquire commercial tools, like DataViews™, which includes a lot of data-presentation functions, like multiple 2D and 3D graph formats. Some also include a Motif look-and-feel (buttons shadows...) which provide consistent presentation and

---

™ AutoCAD is a trademark of Autodesk Corporation.

™ DataViews is a trademark of V. I. Corporation

interaction for end-users. Such tools usually support only a sub-set of the Motif facilities for handling the user interaction but are very powerful for data-presentation.

Such a tool can be very useful for the implementation of synoptic-like applications, i.e. applications with dynamic colors, shapes or graphs but without much computation or many options. This can provide the end users with a very productive solution for synoptic-like applications.

We also plan to integrate PHIGS as soon as there is a need of a high-level graphics library.

C. Motif-based

Another major category of tools are user-interface building tools with dynamic programming support such as interpreted languages. We'll provide such a tool (UIM/X™) for fast-production of simple applications.

The environment is a graphical editor for building up the user-interface of the application and a C interpreter for testing the application from the editor. By-passing most of the UNIX compilation process, the development of the application is much improved. Programmers still need to be trained to Motif but can use a very efficient environment.

One limit of such solutions is that applications are tool-dependent instead of just Motif-dependent. However, run-time support is usually not a big financial issue and it can be envisaged to re-work the application when it is stable enough in order to transform it to Motif-only (some tools include this facility).

D. Spreadsheet

High-level spread-sheet programs, like Microsoft-Excel™ are every-day tools of much of the scientific staff. These tools include many presentation facilities and embedded functions. However, up to a recent date, the Motif market was rather poor in this area. As products are appearing now, we are integrating them (we have started to do this with Wingz™)

These tools have many applications: quick prototyping of control algorithms by the specialists themselves, dynamic treatment and presentation of the data, etc. One major area for them are the machine development applications. The control of the CERN Isolde separators is strongly based on such tools and this application of a spreadsheet is very spectacular [7].

We are porting one off-line PC application to on-line. As one production version of this program has been already realized in the plain C-Motif environment, we will be able to compare the two solutions in area like flexibility, development and maintenance cost, user-interface efficiency or robustness.

E. Mathematical package

Mathematical packages including symbolic computation and graphical data-presentation are becoming popular tools among our scientific users. Mathematica™ is a good example and we are providing this tool on workstations.

Such a package can be used for prototyping algorithms in a very efficient way provided that the specialist is used to it or is assisted by an expert in the package.

One interesting feature of these tools is that a production version of the program may be implemented in an operation-oriented environment supporting standard facilities (instruments control, timing conditions, archives...) with communications to a separate mathematical process supporting the computation.

Fortran is also available and the MAD optic program is currently being ported.

V. CONCLUSION

One main objective of this first period dedicated to set-up the workstation environment was to provide a complete production environment with low costs in development and maintenance of the system software and with small exploitation man-power. The current situation is rather satisfying from this point of view. The environment has proved to be convenient for application production, although there are still a lot of improvements to introduce, both in performances and functionalities.

The two more important events which occured during this set-up phase are the suddenly increasing range of solid industrial standards supported by the major hardware vendors and the jump of the software vendors into the Motif market.

There are probably two directions where we will improve our environment and expertise. The first one is the integration of analysis and design tools to enhance production and maintenance of complex "made to measure" programs. The second one is to continue the effort of implementing user-oriented tools for making wider the population of application-makers or to enhance (replace?) the specification process.

V. REFERENCES

[1] The PS and SL Control Groups, PS/SL Controls Consolidation Project, Technical Report, CERN PS/91-09 (CO), CERN SL/91-12 (CO), April 1991.

[2] R. Rausch, Ch. Serre (Editors), Common Control System for the CERN PS and SPS accelerators, Nov. 1991, these Proceedings (ICALEPCS, Tsukuba, Japan, Nov. 1991).

[3] M. Arruat, M. Boutheon, L. Cons, Y. Deloose, F. Di Maio, D. Gueugnon, R. Hoh, M. Martini, K. Priestnall, J.P. Riunaud, "New Controls for the CERN/PS Hadron Injection Process using Operating Tools and High-Level Modelling Accelerator Programs", these Proceedings (ICALEPCS, Tsukuba, Japan, Nov. 1991).

[4] M. Boutheon, F. Di Maio, A. Pace, General Man-Machine Interface used in accelerator Controls: some applications in

™ UIM/X is a trademark of Visual Edge Software Ltd

™ Excel is a trademark of Microsoft Corporation.

™ Wingz is a trademark of Informix Software, Inc.

™ Mathematica is a trademark of Wolfram Research, Inc.

CERN/PS Control system rejuvenation, Nov. 1991, these Proceedings, (ICALEPCS, Tsukuba, Japan, 1991).

[5] J.M. Nonglaton, U. Raich, Porting Linac Application Programs to a Windowing Environment, Nov. 1991, these Proceedings, (ICALEPCS, Tsukuba, Japan, 1991).

[6] F. Perriollat, G. Cuisinier, A. Gagnaire, "Nodal", PS/CO Note 91-20, November 1991.

[7] R. Billinge, A. Bret, I. Deloose, A. Pace and G. Shering, "A PC Based Control System for the CERN Isolde Separators", these Proceedings (ICALEPCS, Tsukuba, Japan, Nov. 1991).