ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S13MMI08

An Open Software System Based on X Windows for Process Control and Equipment Monitoring

A. Aimar, E. Carlier, V. Mertens
European Organization for Nuclear Research (CERN) - SL Division,
CH - 1211 Geneva 23, Switzerland

Abstract

The construction and application of a configurable open software system for process control and equipment monitoring can speed up and simplify the development and maintenance of equipment specific software as compared to individual solutions. The present paper reports the status of such an approach for the distributed control systems of SPS and LEP beam transfer components, based on X Windows and the OSF/Motif tool kit and applying data modeling and software engineering methods.

I. INTRODUCTION AND MOTIVATION

A. Equipment

Equipment for SPS and LEP beam transfer at CERN comprises systems like the SPS injections, extractions, targets, dumps, and collimators, and the LEP injections and separators. In total some 80 distinct systems spread over both accelerators and the fixed-target areas have to be controlled. At present, new control systems for the LEP beam dump and the LEP Pretzel separators are being prepared.

Although the functionality and the composition varies considerably between the different systems, all can essentially be characterized as 'slow controls': Reaction times as seen from the main control room are of the order of seconds; any fast responses, e.g. for beam dumping, are supported by special hardware. The amount of data exchanged between the main control room and the devices is small.

B. Equipment Software

However, comprehensive equipment specific software has to be provided to achieve the desired level of abstraction towards the main control room, to allow monitoring of the equipment performance, and to dispose of efficient tools for local and remote fault-finding to help keeping down-times low, in particular in view of the volume and the distances involved.

For this sake a lot of code has been written up to now, especially with the large-scale use of distributed processing. Due to the limited manpower there is a strong risk of bottlenecks in the treatment of requests for modifications or extensions which might arise from an evolving environment or an increased sophistication of use.

C. Tool Kit Approach

This experience has encouraged us to try a different approach by replacing our equipment specific programs by a general software system or tool kit which receives its individual functionality through a formal description of the equipment and the desired function in tables. If this idea is pursued rigorously almost full separation between code and data can be obtained, leaving only pieces of specific code behind which are uneconomic to parametrize.

As possible advantages we see, besides others, a more uniform appearance of the equipment, a more transparent specification phase with improved communication between hardware and software specialists, leaving less room for misunderstandings, and a shorter reaction time for developments and modifications.

Before starting the development we made some investigations in the commercial sector. At that time we came to the conclusion that a separate development could well be justified in view of the potential problems encountered when embedding and maintaining a commercial process control system in a given and evolving environment, disregarding any price argument. With time passing by, we might however come to a different finding.

To make our task more feasible we did not attempt to write a full package from scratch but rather tried to re-use a maximum of existing packages, tools, and mechanisms, combining them into the desired product.

One of the key goals was to arrive at a portable, thus platform independent, and evolvable software system which should be easy to adapt to changing environments or increasing needs. This becomes particularly attractive in combination with the X Windows system and the OSF/Motif tool kit since they allow to perform input or visualize complex results on a large variety of media without much adaptation work.

In the following we will first give an overview of the tool kit concept and describe its key ingredients, then the chosen implementation. Afterwards, we will present results obtained in a prototype application, followed by a status report and an outlook.

II. TOOL KIT CONCEPT

The layout of the tool kit and the way the user interacts with it is sketched in figure 1.

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S13MMI08

A. General Structure

The tool kit is built up in a modular way. Each module apports a well defined and limited function which contributes to a smooth progression and testing.

Its core module is a shared run-time data base (SDB), containing all data which are system-wide accessible, surrounded by a number of peripheral modules, some of them containing their own data bases (DB) necessary to perform the required function. A server (SDBserver) provides all modules with access to the shared data base thus permitting the inter-module communication. Besides that, there is a tool

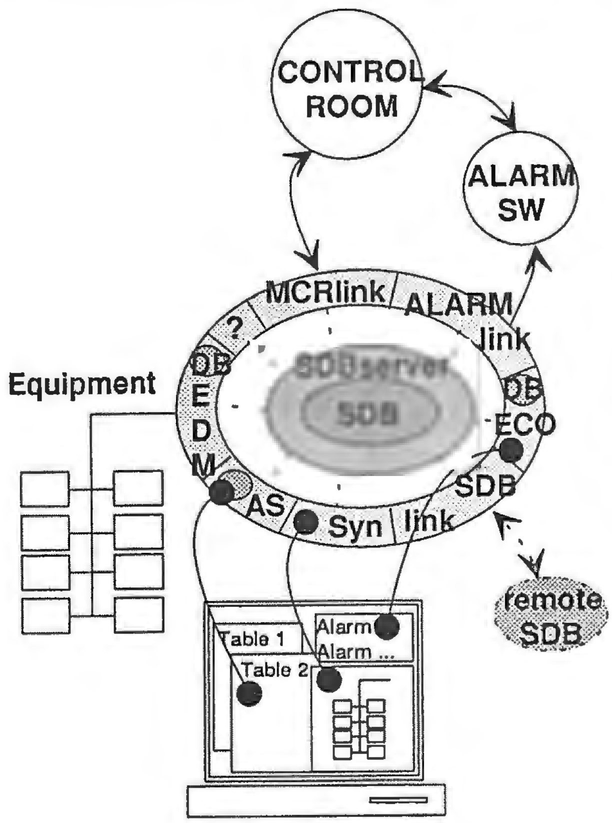

Fig. 1: The tool kit architecture and the user interaction

for accessing the different module data bases to configure the desired application and a graphics editor and animation system to link elements in the shared data base to screen objects. The most important items will be discussed below.

B. The Shared Data Base

The shared data base contains all data exchanged with the outside world at run-time, e.g. an image of all equipment channels, like voltages or temperatures, but also more abstract data like the last requested state of the equipment. Abstracting external data in terms of these values allows the system to be based on a unique and unduplicated representation of the context and to build tools for displaying, changing, and checking these values in a way completely independent from their external format.

C. The Shared Data Base Server

The use of a server for the book-keeping of shared data allows concurrent access and distributed computation in a straightforward manner. The interface to its services has been designed in terms of Remote Procedure Calls so that every client module can access the global data in a transparent and location independent way, preserving the important issues of openness and developmental possibilities.

D. The Equipment Data Manager, the Equipment Control Operation Module, and the Action Sequencer

The Equipment Data Manager (EDM), the Equipment Control Operation (ECO) module, and the Action Sequencer (AS) implement the more proper control functions.

The EDM keeps the shared data base consistent with the inputs and outputs of the attached equipment and takes care of the adaptation to special communication protocols. Special procedures like initialization routines are also located here.

The ECO checks for the occurrence of fault conditions and treats special events.

The AS takes care of the required sequences, i.e. all chains of actions which have to be performed upon a certain event or a requested state transition.

All modules are completely driven from their own configuration and run-time data bases. These contain, in case of the EDM, the complete description of the equipment, e.g. module addresses and transformation factors between electrical and physical values, or, in case of the ECO, alarms conditions, messages, and emergency actions.

E. The Shared Data Base Link Module

The definition of data which have also to be accessible from modules at remote sites is made through the Shared Data Base Link (SDBlink) module, which is in fact nothing else than a client to all involved data base servers and which has its proper data base containing the correspondences.

F. Special Interface Modules

Particular attention has to be paid to a clean functional implantation into the global accelerator control system [1] to allow a correct remote control and propagation of alarms.

For this sake two interface modules are foreseen. The first one (MCRlink; already exisiting) permits to access the shared data base through the RPC mechanism widely in use at CERN. This module will, in the future, also follow and implement the new control protocol guidelines as laid down in [2].

The second one (ALARMlink) is an interface to the

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S13MMI08

general alarm system. It transforms the alarms resulting from the ECO checks into the required format and feeds it into the existing software chain.

G. User Access Tool

An adequate tool is required to access the special module data bases, i.e. the tables which allow to tailor the application, in a proper and comfortable way. Several solutions are being considered at the moment, e.g. [3].

H. Synoptics

A tool permitting to generate synoptic images and animate them in correlation with the shared data base contents and changes (Syn), based on a commercial user interface builder, working with X Windows and OSF/Motif, and producing UIL (User Interface Language) or C code, is currently under investigation.

III. TOOL KIT IMPLEMENTATION

A. The Target Environment

The environment in which the tool kit should finally run is the process controller proposed as standard interface between the equipment and the accelerator network [1], in our case, a diskless industrial PC running under the real time operating system LynxOS, together with X Windows and OSF/Motif.

B. The Development Environment

To simulate the target environment we used a diskless DECstation 5000/125, with a remote workstation acting as file server. The use of a multi-windowing environment and the fact that all important tools were already available permitted to advance quite rapidly.

To simulate attached equipment electronics we used a) a HV power supply and a programmable timing unit built up in G-64, each controlled by a microprocessor and connected to the workstation through a terminal server by RS-232 links, and b) several commercial I/O modules on a Bitbus, linked to the workstation via a SCSI-Bitbus gateway [4]. No additional driver software was needed.

C. The Data Base

In an application like this where the reliable access to a lot of data is essential, the selection of a good data base system becomes a key issue. Our investigations for a non-commercial product which has proven its quality in similar applications has led us to the choice of ADAMO [5, 6].

ADAMO (= Aleph DAta MOdel) has been developed within the ALEPH collaboration at LEP where it is now used in the fields of data acquisition, detector description, event reconstruction, and data analysis [7]. More recently other experiments have started to use it, among those the ZEUS experiment at HERA [8, 9].

ADAMO is based on the entity-relationship (E-R) model of Chen [10]. This model adopts the view that the world consists of entities - objects that can be distinctly identified and have a fixed number of characteristics called attributes - and relationships, which are associations between entities. A data model [11], generally spoken, is a strategy for data organization which includes formally defined data structures, operators to act upon the data, and validation procedures to ensure that the data obey the imposed constraints.

The ADAMO system provides all these features in a form suitable for scientific computing where numerical algorithms are important. The choice of ADAMO is also supported through its proven portability to the major hardware platforms and operating systems used at CERN (in fact, it is based on the CERN software libraries).

D. CASE Support

CASE (Computer Aided Software Engineering) tools provide methods for describing graphically and consistently a software system in its life cycle through analysis, design, coding, and maintenance [12] thus speeding up the development.

Such a tool, called StP (= Software through Pictures, from Interactive Development Environment), has been employed for the construction of the tool kit, using an ADAMO-Interface to the Picture Editor of StP. This allows, starting

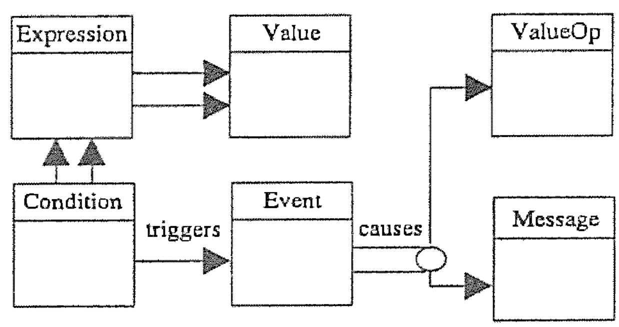

Fig. 2: E-R diagram of the ECO module (simplified partial view)

from the design of the data bases in E-R diagrams, to generate automatically the data structures in a format that can be accessed and manipulated from the desired programming language through a run time library.

As an example of the applied technique, figure 2 shows a simplified part of the E-R representation of the ECO module which can be interpreted as follows: A 'Condition' on which the system should check is made up from 'Expressions' with 'Values' as operands. Such a 'Condition', when fulfilled, can 'trigger' an 'Event' which 'causes' either 'Messages' or 'ValueOperations'.

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S13MMI08

E. Use of Standard Tools

To provide a maximum of portability, standard tools have been employed wherever possible.

The tool kit itself has been written in C. Since the UNIX world is our main target the internal communication layer through Remote Procedure Calls has been implemented using NCS (Network Computing System by Apollo Inc.) [13].

The windowing environment for all interfaces used for customizing or output is provided by X Windows and OSF/Motif.

IV. EXPERIENCE AND RESULTS

A prototype application has been realized which comprises about 30 digital and analogue I/O channels. On the DECstation 5000/125, the CPU time needed to perform a full cycle of readings and checks is currently about 45 ms (of this, about 30 ms are consumed by the data acquisition and mostly due to the way the equipment has been attached). The application performs very stably.

The allocation of virtual memory is around 6 MB. To have a safety margin when running in a diskless environment together with other processes a total memory size of 16 MB seems adequate.

V. STATUS AND OUTLOOK

The following elements of the tool kit have been commissioned until now: The shared data base, the data base server, equipment managers for Bitbus and multiple RS-232 links, the alarm module, and the RPC interface.

The most prominent future task is to transfer the whole system onto the desired target environment, a PC running under LynxOS.

Once this has been accomplished, we first plan to streamline the product in terms of simpler interaction (table entry and user interface), to add protection mechanisms against unauthorized use, and to take care of the display of actual trends, data logging, and re-play of stored data. One of the more long term goals is to provide a higher degree of support for fault finding (on-line help) through the application of expert system techniques.

The first test application in a larger project is planned for spring 92.

VI. CONCLUSION

The extensive use of existing tools, like the data management system ADAMO or the CASE tool StP has made possible to optimize the cycle between analysis, design, coding, and implementation and to arrive, in a relatively short period and with little manpower effort, at a satisfactorily working prototype system.

The use of recognized standards for the underlying mechanisms makes this tool kit a highly portable product which could find its application also in other areas where slow control systems are needed.

VII. ACKNOWLEDGEMENTS

The authors would like to thank the members of the CERN-SL/CO group for their support and for providing the necessary hardware and software infrastructure.

It is a pleasure to acknowledge the support received from the DEC Joint Project Office at CERN.

We are indebted to J. H. Dieperink, S. Fisher, P. Palazzi, and W. von Rüden for their contributions and A. Lovell for his help.

We would like to thank the organizers of this conference for their efforts.

VIII. REFERENCES

[1] R. Rausch and Ch. Serre, "Common Control System for the CERN PS and SL Accelerators", this conference.

[2] WOPRO Working Group, reported by G. Benincasa, "Control Protocol: The Proposed New CERN Standard Access Procedure to Accelerator Equipment: Status Report", this conference.

[3] A. Aimar, "VERDI, Visual Entity Relationship Data Interface", thesis, Department of Computer Science, Torino, Italy, 1989.

[4] A. Aimar et al., "SPS/LEP Beam Transfer Equipment Control Using Industrial Automation Components", this conference.

[5] M. G. Green," The ADAMO Data System, An Introduction for Particle Physicists", Royal Holloway and Bedford New College, UK, RHBNC 89-01 and CERN/DD/US/131.

[6] S. M. Fisher and P. Palazzi, "Using a Data Model from Software Design to Data Analysis: What Have We Learned ?", Comp. Phys. Comm., 57, pp. 169 - 175, 1989.

[7] Z. Qian et al., "Use of The ADAMO Data Management System Within ALEPH", Comp. Phys. Comm., 45, pp. 283 - 298, 1987.

[8] H. Kowalski et al., "Investigation of the ADAMO Performance in the ZEUS Calorimeter Reconstruction Program", Comp. Phys. Comm., 57, pp. 222 - 224, 1989.

[9] R. Loveless, "ZEUS Hardware Control System", Comp. Phys. Comm., 57, pp. 313 - 315, 1987.

[10] P. P. Chen, "The Entity-Relationship Model - Toward a Unified View of Data", ACM Trans. Database Systems, 1, pp. 9 - 36, 1976.

[11] D. C. Tsichritzis and F. H. Lochovsky, *Data Models*, New Jersey: Prentice Hall, Englewood Cliffs, 1982.

[12] D. J. Hatley and I. A. Pirbhai, *Strategies for Real-time System Specification*, New York: Dorset House Publishing, 1987.

[13] M. Kong et al., *Network .Computing System Ref. Man.* New Jersey: Prentice Hall, Englewood Cliffs, 1987.