The Continuous and Seamless Replacement of a Running Control System Succeeded

G. v. Egan-Krieger, R. Müller*, J. Rahn**

Berliner Elektronenspeicherring-Gesellschaft für Synchrotronstrahlung m.b.H. (BESSY), Lentzeallee 100, D–14195 Berlin, FRG

Abstract

A continuous replacement and upgrade procedure for the control system of the running light source BESSY has been proposed in 1990 [3]. Today we can present the components of the 'final' system. The concept has shown to be practicable and flexible. Technically improved solutions unforeseeable at start time of the conversion and not available before 1991 could be utilized even in the main turnover phase. The intended goals have been achieved: Beam time has not been affected by the conversion process. Developmental and operational experiences show that the system will supply us with key components for the control of BESSY II.

Introduction

The 'Berliner Elektronenspeicherring-Gesellschaft für Synchrotronstrahlung m.b.H.' (BESSY) operates an 800 MeV storage ring dedicated to the generation of synchrotron light in the VUV and soft X-ray region [1].

A cautious and smooth modernisation scheme for the aged control system [2] of the fully equipped and continuously running light source BESSY has been introduced in 1990 [3]. Since then step by step the electronically cascading interface system attached to a central minicomputer system has been exchanged 'in the fly' by a DCE (*distributed computing environment*). Application programs have been ported to a modern workstation platform. Progressive replacement of all system components led us to the current phase of finishing and clean up.

Prerequisites and schedule of the modernisation concept [3] as well as the features of our graphical user interface management system (UIMS) [4] have already been described. Purpose of the present paper is to point out those modifications of the original approach, that have shown to be advantageous and describe specific solutions, that could be of general interest.

Configuration Overview

The new system consists of four stationary operator consoles (See fig. 1). Two of them are the workstation console displays of HP Apollo 9000/730 machines and two are fast, but otherwise 'dumb' 19" X-terminals. One mobile console, an elder 17" X-terminal, can be used at the location of equipment that has to be repaired or where diagnostics have to be performed.

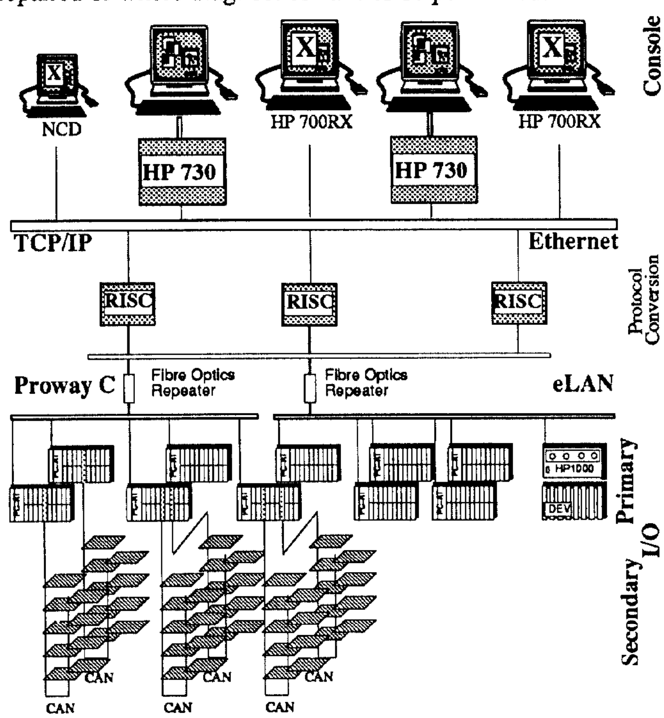


Figure 1: Sketch of the Hardware Configuration

Three unexpensive RISC workstations with minimal configuration provide protocol conversion from TCP/IP (IEEE 802.2) on the presentation and command level to Proway C (IEEE 802.4, IEC 955) on the process field bus 'eLAN' (See [3]).

Attached to three segments of the Proway C fieldbus are about twenty micro processor nodes (crates) with 312 slots. Node masters are IBM–AT compatible boards (See 'Interface Node Master'). On the parallel bus (ECB) they control both standard I/O interface cards and gateway controller boards to lower level field bus systems, the *Controller Area Networks* (CAN) [5]. In total eight CAN configurations allow to equip the electronics of about 50 power supplies with an embedded controller (See 'Low Level Field Bus').

The new hardware configuration not only replaces (nearly) all aged interfaces. In addition, a 20% increase of the number of remote controlled devices is achieved. About 50 pieces of

*eMail: egan@acc.bessy.de, mueller@bessy.de, rahn@bessy.de

0-7803-1203-1/93$03.00 © 1993 IEEE

equipment formerly handled by local buttons, potentiometers, wheel switches and digital voltmeters are now attached to the computer system.

Software Architecture

The completely transparent transition from the old control system to a flexible, modern one imposes severe restrictions on any possible solution. More software modules are needed and the architecture is more complicated than in a newly designed system.

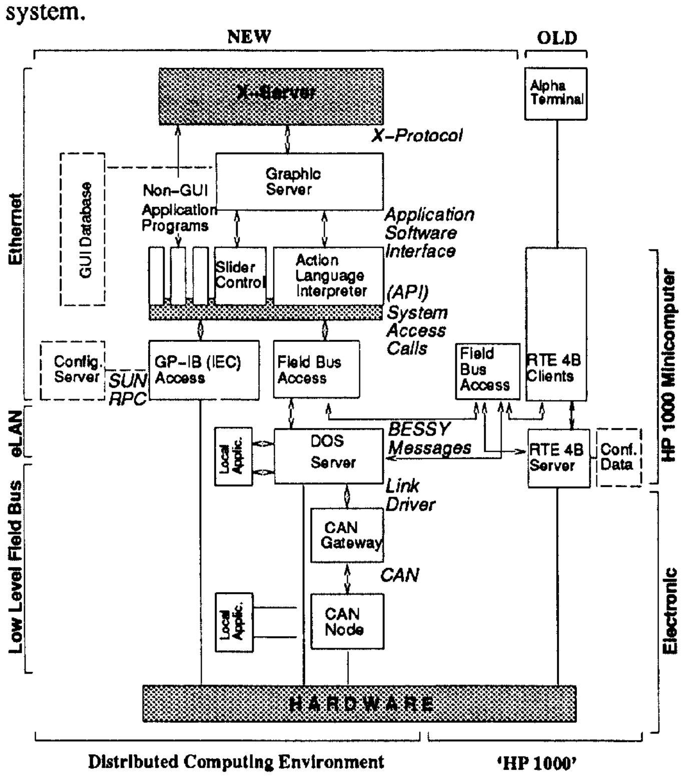


Figure 2: Sketch of the Software Architecture

The new control system is based on six very different protocol layers (See fig. 2). As the operator clicks a button of the *Graphical User Interface* (GUI) at one of the console displays up to seven server programs have to communicate until that event is converted into the hardware action the operator had in mind. Nevertheless the total response time stays within the order of 100 ms. This period of time is basically due to the moderate performance of the field bus 'eLAN'.

Despite that complexity any program that has to drive or control a piece of equipment has to use an application program interface (API) from a library of system access calls. That central software interface allows to keep the application code transparent and independent from the current field level configurations.

Programmable Control Application

Complex driving programs written in a sequential language (BASIC, FORTRAN) and programmed by sets of configuration data are difficult to modify and maintain. Properties of the hardware and exception handling are usually spread all over the code.

We try to minimize these problems with an object oriented approach [7]: According to hardware specifics and driving methods devices correspond to classes in the code, usually derived from the obvious generic structures (e.g. power supply, valve, etc.). A built in interpreter section allows to attach device name to device class, form lists of devices, do simple arithmetical and logical operations, parse input and write output files. That provides the desired flexibility to the sets of configuration data.

Interface Node Master

As a new technical solution a board equipped with an Intel i386DX processor became available in April '91. It is 100% IBM-AT compatible and runs MS-DOS 5.0. It fits with its ISA and ECB bus connectors into a dual bus crate.

This board implies so many advantages that our developments for the system envisaged so far (MC 680x0 running OS 9) [3] have been frozen.

- The equipment server software existing for CP/M (Z80) systems [3] is much easier and faster ported to the more similar DOS (i386) system than to OS 9 (MC 680x0).
- In addition to the ECB bus IO cards standard PC hardware becomes accessible by the ISA slots. This makes it easy to utilize inhouse expertise for PC systems. Solutions from outside the controls group can be integrated into the control system with little extra work.
- More ECB bus slots per crate (26 instead of 16) provide better flexibility.
- Whenever required the hardware allows to replace DOS by a realtime UNIX kernel or another powerful realtime operating system like Lynx OS.

The equipment server code fits into modern object oriented concepts (object description, data, methods). Software interfaces within this server foresee to treat local application programs and requests from the fieldbus on an equal basis. Vacuum and high frequency surveillance, lifetime monitor etc. that run presently on stand alone PC-AT systems will be integrated into the server at this point.

Low Level Field Bus

The fast and robust CAN (*controller area network*) [5] has been developed for trucks and cars and is available as unexpensive firmware [6]. Its efficient protocol (Iso/DIS 11898) is based on priority drive bus arbitration. Messages are not sent to specific addresses. Nodes broadcast data objects that are received and evaluated by all participants where the objects are meaningful. A mechanism to notify data requests is part of the protocol. CAN supplies very flexible solutions for the common problem to treat analogous signals at the location of the devices and transport the data to the higher level micro computer.

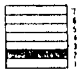
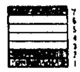
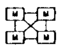
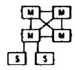
CAN	Profibus
ISO/OSI Level 2	ISO/OSI Levels 1, 2, 7
	
Multimaster	Multimaster/Master-Slave
	
Messageoriented	Point to Point
unlimited user 32 user for RS 485	32 user per Segment
1 Mbit/s for 40m (approx. 40 KBit/s for 1000 m)	500 Kbit/s for 200m (93,75 KBit/s for 1200m)
CSMA/CR	Token Passing with Polling
RS 485, fibreoptic possible	RS 485, fibreoptic
Extremely reliable datatransfer	Excellent for large amount of data.
Free priorities	Two priorities
Deterministic at the highest priority level. Very fast.	Deterministic but not always suitable for fast realtime applications.
Simple Wiring	Simple Wiring
Decentralized mounting of modules possible	Decentralized mounting of modules possible
CAN - Controller Area Network Bosch/Intel	BMFT-Project German DIN 19245 standard
Low-Cost	Low-Cost
Excellent choice for semi-open systems	Easy system-expansion. Excellent choice for open systems.
Many Chip-Manufacturer have comitted for CAN.	No chip-set available

Figure 3: CAN compared to MAP conformal Profibus

We have defined and implemented a small protocol layer between DOS Server and CAN gateway that allows to establish a stable and fault tolerant link between Proway C node master (i386) and CAN gateway (Z 84xx) (*Link Driver* in fig. 2). A device server running on the embedded controller (Z 84xx) of a CAN node has been designed and realized. The required communication protocol (on top of the CAN bus protocol) between CAN gateway and 'slave' node has been implemented (*CAN* in fig. 2). Matching of these modules has been tuned and the Proway C-CAN microprocessor system as a whole runs stable.

The determination of the beam postition for the fast orbit measurement and correction system currently under development at BESSY will take place locally at the pick up stations. The application software that converts signal strength to the electron beam position information will run on the embedded controller of a CAN 'slave' node (See fig. 2).

Measuring Devices

Varying command sets and limitations to bus cable lenght require specific solutions for the control of measurement equipment on the GP-IB bus (IEEE 488, IEC 625). We use separate GP-IB controller units that give bus access to any computer equipped with a serial line interface (RS 232 C).

Serial line interfaces on the console computers attached to a GP-IB controller unit are handled by dedicated server programs. A configuration server (See fig. 2) supplies the actual mapping of device names to serial line server responsible for the appropriate GP-IB segment. The API library routine then opens a *rpc*-connection to the GP-IP access server that mediates the exchange of commands and replies with the device.

Summary

With the new system maintainance will be no problem for a long period of time. Nearly no extra machine time has been needed for installation, testing and debugging. From graphical entities to device servers object oriented programming methods have been used at all levels. The modern man machine interface based on the X window system has been intuitively learned by the operators and is widely accepted. Application programs could be drastically improved and are much easier maintainable on the modern platform.

During the start phase of BESSY II we are immediately in a position to control hardware at the level of the magnet or vacuum laboratories. For the operation of the light source BESSY II it is at least a core system. Superfluous modules that have been necessary for the current control system conversion (like the Proway C field bus) have to be removed.

Synchronisation of application programs and database structures have to be introduced. Simulation programs have to give decision and control support. Sophisticated driving programs have to compensate correlated interactions (e.g. of undulators scanning at different speed). For the storage ring components different levels of privacy have to regulate access permissions and protection.

References

[1] S. Bernstorff et. al., Physica Scripta, 36, 15 (1987)

[2] G. v. Egan-Krieger, W.–D. Klotz and R. Maier, IEEE Transactions on Nuclear Science, NS-30, 2273 (1983)

[3] G. v. Egan-Krieger, R. Müller, Proceedings of the 2nd European Particle Accelerator Conference, Nice, pp. 872–874, 875–877 (1990)

[4] R. Müller, H.–D. Doll, I. J. Donasch, H. Marxen, H. Pause, Conference Record of the 1991 IEEE Particle Accelerator Conference, San Francisco, pp. 1311–1313 (1991)

R. Müller, Proceedings of the 3rd European Particle Accelerator Conference, Berlin, pp. 1167–1169 (1992)

[5] Road Vehicles – Interchange of Digital Information – Controller Area Network (CAN) for High Speed Communication, Document Iso/DIS 11898, International Standardization Organization (1991)

[6] INTEL 82526, Serial Communication Controller, Architectural Overview, Order Number 270 678-001 (Jan. 1989)

Philips PCA 82C200, Stand-alone CAN-Controller, Product Specification (Oct. 1990)

[7] R. Lange, Diploma Thesis, FB 20, TU-Berlin, in preparation