REPLACING A POPULAR ACCELERATOR CONTROL SYSTEM

R.P. Mannix

Rutherford Appleton Laboratory, DIDCOT, Oxon., OX11 0QX, U.K

Abstract

The ISIS computer control system has been both effective and well liked by its users. It is obsolete and very limited in power. The struggle to build a new system, using commercial software, that is at least as good as its predecessor (in spite of the advances in technology!) is described.

I. INTRODUCTION

ISIS is a 200µA, 50Hz, 800MeV proton synchrotron/heavy metal target spallation neutron source. The present control system has been used both during commissioning (1980-85) and operation. The design current of 200µA has recently been achieved.

A control system can be considered in two parts:

1. The environment, comprising-

computer system(s)
local control microprocessors
equipment interface (i/o systems)
man-machine interface
system software
equipment-specific software
database organisation
running environment for control programs
commercial packages
graphics language
programming language(s)

The environment is always provided by or via a "controls group" of some sort and they alone make changes.

2. The implementation, comprising-

control program specification
control program authorship
control program testing and installation
control program availability
implementation of commercial packages

The implementation is done to some extent by the users and from 0-90% by the controls group, depending on local preferences and precedence.

II. PRESENT SYSTEM

1. The environment.

computer system(s)-	GEC 4000 (1970's 16-bit minis)
local control microprocessors	Intel 8080
equipment interface (i/o systems)	Direct CAMAC/MPX,AWS
man-machine interface	CAMAC displays, touch screen, programmable knob
system software	GEC DOS (obsolete)
equipment-specific software	BABBAGE (GEC assembler)
database organisation	Rudimentary from BABBAGE
running environment for control programs	Semi-compiled interpreter
commercial packages	None
graphics language	) GRACES (interpreter)
programming language(s)	)

There are 5 GEC computers, connected via a CAMAC based star network (Figure 1). There are approximately 15000 lines of data module (equipment routine) source code. The equipment interface (MPX) consists of 700 modules in 70 crates. There are several hundred control programs in use. The Analogue Waveform Switching (AWS) system is run separately, via touch screens/PCs and CAMAC, to switch ~260 oscilloscope signals to the control desk.

The computers are very modest in power and are very limited in storage capacity. The operating system allows us to control hardware from a number of concurrent interpreter processes on each processor. High priority processes communicating with hardware completely lock out all others. Communication with other systems is non-existent, peripherals such as floppy disks are obsolete and unsupported, backups require shutting down the control system and maintenance is expensive (24hr cover is essential). The various branches of the interface hardware are tied explicitly to particular processors– reconfiguring after a processor failure is impossible– a serious disk drive fault can, and has, shut down the accelerator for a few hours. An average ISIS experiment lasts two days, within which several data taking runs, all of which are essential, must be performed. These sorts of breakdowns, although infrequent, are highly undesirable.

2. The implementation.

There are basically two schools of thought with regard to control system displays. The "crew" school like the use of colour, mimic diagrams, meters and dials etc. The "machine physicist" school like pages of numbers in monochrome with maybe an x-y plot for light relief every now and then, regarding all else as puritans regard the theatre- the devil's work. You have to allow both!

The guiding principle is "informed anarchy". The controls group rarely gets involved in implementation now (although we did start things off). Changes are made at will by machine physicists and crew members. The (dis)advantages of this approach are:

Advantages-

> Extremely effective control system
> Users "in touch" with machine
> Changes to programs can be specified, implemented, tested and installed in under 5 minutes (for small changes) at any time by the person concerned.
> Small (ie cheap!) controls group
> Secure (due to use of idiot-proof interpreter)

Disadvantages-

> Inefficient use of resources
> No-one knows what all the programs do
> Complicated programs difficult to maintain in author's absence.
> Language limited to interpreter (slow, lacks facilities, needs to be learnt)

A final consequence- lack of well defined standards- is either an advantage or a disadvantage depending on who you talk to!

The decision was made to upgrade the environment with minimum impact on the implementation.

III. THE NEW SYSTEM

1. The environment.

computer system(s)-	DEC workstations
local control microprocessors	STEbus systems
equipment interface (i/o systems)	CAMAC/MPX,AWS (as now), connected via Ethernet.
man-machine interface	X-terminal
system software	Open-VMS
equipment-specific software	C functions
database organisation	Commercial package (see below)
running environment for control programs	Open-VMS process
commercial packages	Vsystem[1]
graphics language	Interactive screen generation OR port of existing commands.
programming language(s)	BASIC/FORTRAN/C

The Vsystem package provides a fully distributed database driven control system with a graphical interface over a number of DECnet nodes. Each control or monitoring object is referred to as a *channel* and, by using the *channel name*, control windows can be generated with an interactive draw package which interact with that channel via a number of control and monitoring "tools". Alarm handling, on-line database editing and logging facilities are also provided.

Access to the control and monitoring channels is also available from normal user programs written in VAX BASIC/FORTRAN or C. Such programs can be invoked from a terminal window or from an interactively generated control window.

The use of channel names, database and handlers (equipment routines) maps very well on to our existing system based on Data Modules. The graphical interface provided with the new software should enable the functions of 75% of the control software to be replaced without recourse to writing code.

The system-wide nature of the databases and the networked nature of the CAMAC driver crates makes access to all equipment possible from any processor– something which was not previously possible. Up to 256 CAMAC crates can be accomodated on one Ethernet segment (Figure 2).

New and replacement microprocessor systems will be connected directly via Ethernet (at present they are connected to the MPX system.

Terminals with any level of access to the control system can easily be added anywhere on site (if desirable!). A control system for a new beam line can be provided by buying a workstation, another CAMAC controller and local interfacing hardware. The incorporation of this into the existing system is automatic (subject to the licensing agreement).

The current operator interfaces (touch sensitive screens , tracker balls, colour displays, knobs etc.) which are all obsolete or nearly so, will be replaced by high quality mouse/tracker ball and keyboard driven workstation type displays.

2. Implementation changes

The major changes the user will encounter (apart from the environment changes) will be in the specification, testing and installation of control programs and in the availability of the control programs.

The interpreter approach has been dropped and there is therefore no alternative to the edit/compile/link/run cycle, however streamlined. Choices have to be made as to whether the task is of sufficient complexity to warrant user code, or whether an interactively generated window would do the job and, if code is to be used, which language is appropriate. The specificity of devices on the control desk will disappear- all programs will be available from everywhere.

The requirement for users to update displayed information in their own programs has been dropped (at least for the interactively generated windows) database "readers" continually update the database values, changes being displayed as they occur.

No new system can look or behave as the old one did. Users are familiar with the old system and will be reluctant to change. Any shortcomings in the new system will be picked on and amplified, shortcomings in the old system having been assimilated years ago. Use of the new system from a user-written program is provided via a simple interface provided by the controls group.

IV. PROGRESS

Ten databases have been written, those for the injector magnets, the injector timing system, the injector general purpose status modules, target systems and extracted proton beam magnets– a total of 2000 channels so far. Handlers for the programmable timing modules, magnet power supplies, function generators, equipment microprocessors and status modules have been written – conversion of BABBAGE to C is not too difficult. Control windows for the timer modules and magnet power supplies and the target station monitoring systems have been prepared and a suite of hardware test programs written for test access of interface modules via the ECC controller (external to the database).

A significant amount of time has been spent deciding how to modify the standard usage of the Vsystem software to suit our needs and in moving to new versions. Now this has been done, production of software should speed up.

V. OTHER APPROACHES

Well funded organisations can adopt the approach of doing everything themselves. The cost of this approach is high because of high level skills which need to be purchased- is the job of the controls group of a neutron source (for example) to push forward control system theory or to provide an engineering solution?

Various approaches have been described, using exclusively mass produced software on PC platforms (spreadsheets, DDE, LabView, etc). This approach may have many benefits, especially in smaller projects – I suspect such approaches are seldom glamorous enough to be given the full backing of their parent organisations.

VI. CONCLUSIONS

The choice of commercially available software must be correct for those establishments where lack of effort and staff turnover are problems. Good local support and constant updating of the product are also essential, as is the ability to feed requirements into the supplier's development plan. Even so the effort involved in changing to a new system is always underestimated, both by the customer and by the supplier. There is a lack of flexibility inherent in commercial systems- things have to be done "their way". This can be overcome by always having the option to tie in user programs in a variety of languages to the package concerned.

The licence and maintenance costs for such software, especially if not mass produced, are usually substantial, but must be compared with the staff-year costs in an organisation and offset against smaller staff requirements. Maintenance costs in our case equate to approximately 0.3 staff-year per year.

No commercial product will allow the easy assimilation of an existing control system. Whatever practises and methods prevail in an operating machine are the "right" ones by virtue of the fact that they are in use and familiar to those who use them. Any new system must be modified to fit what exists– not the best way to proceed. It is also clear that the only timing information of any interest to the user are the times taken to (1) present the control window required on the screen and (2) to operate a piece of hardware and see the effect on the screen, what happens underneath being irrelevant.

We are happy that the new control system will meet all our expectations with regard to extensibility, reconfiguration and communication and will perform as well as or better than the existing system. It seems to be an unwritten law of control systems that, as the power of the processors increases, the complexity of the software rises until the perceived response time rises to the maximum acceptable. We would hope that the extra complexity in the new system will be achieved without a rise in perceived response time.

The suitability of Ethernet (or any general purpose network system) as the main i/o channel is unclear. On the one hand emerging computer systems are frequently only supplied with an Ethernet port and moving away from this becomes expensive. It is truly distributed and configuring the system and expanding it become trivial. On the other hand, the generality of the system means that it is slow. It is clearly unsuitable for a fast data acquisition system but should be well suited to a supervisory control system such as ours.

The key advantage over our old system (and over Q-bus, turbo-channel, SCSI controllers or even PCI interfaces) is the logical separation of the interfacing hardware from a particular processor and the similar distributed nature of the databases- this is a truly distributed system (aside from one disk which must always be available to the system). Directly connected systems force all access to be through the parent processor, whose failure will isolate that equipment entirely. It also allows the integration of PC's, CAMAC, VME, STEbus, and other systems in a controlled manner. Together with distributed software such as that described, it provides a system which is easily reconfigurable should a processor fail. The price paid for this flexibility is speed.

REFERENCES AND ACKNOWLEDGEMENTS

1. "Vsystem" is acknowledged to:

Vista Control Systems, Inc.
134 B Eastgate Drive
Los Alamos
New Mexico 87544

GEC Computers are now GPT Datasystems Ltd.

DEC, Open-VMS are registered trademarks of the Digital Equipment Corporation.

Ethernet is a registered trade mark of the Xerox Corporation.

LabView is a registered trademark of National Instruments.

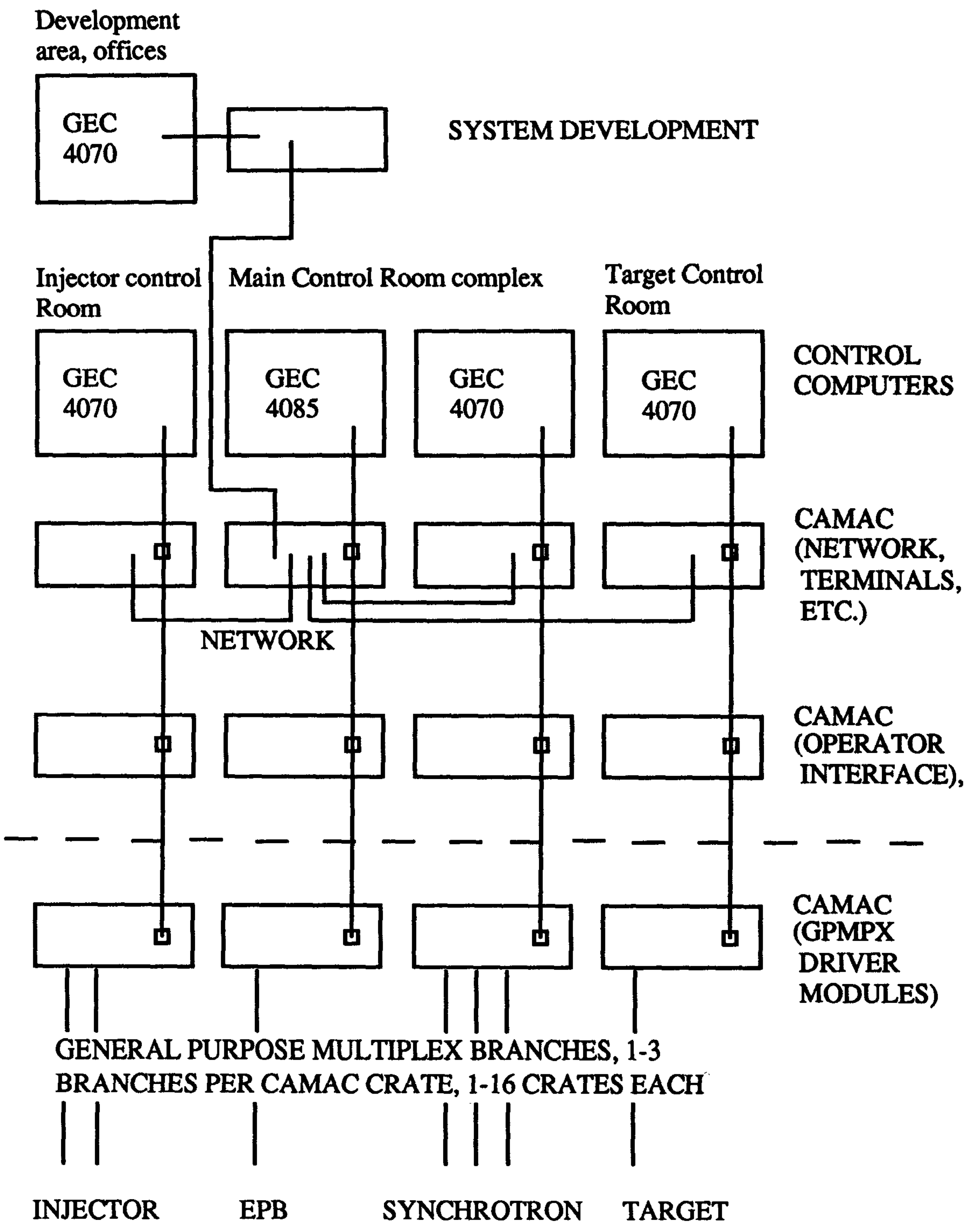
Development area, offices
GEC 4070
SYSTEM DEVELOPMENT
Injector control Room
Main Control Room complex
Target Control Room
GEC 4070
GEC 4085
GEC 4070
GEC 4070
CONTROL COMPUTERS
CAMAC (NETWORK, TERMINALS, ETC.)
NETWORK
CAMAC (OPERATOR INTERFACE),
CAMAC (GPMPX DRIVER MODULES)
GENERAL PURPOSE MULTIPLEX BRANCHES, 1-3 BRANCHES PER CAMAC CRATE, 1-16 CRATES EACH
INJECTOR
EPB
SYNCHROTRON
TARGET

FIGURE 1. General arrangement of present system

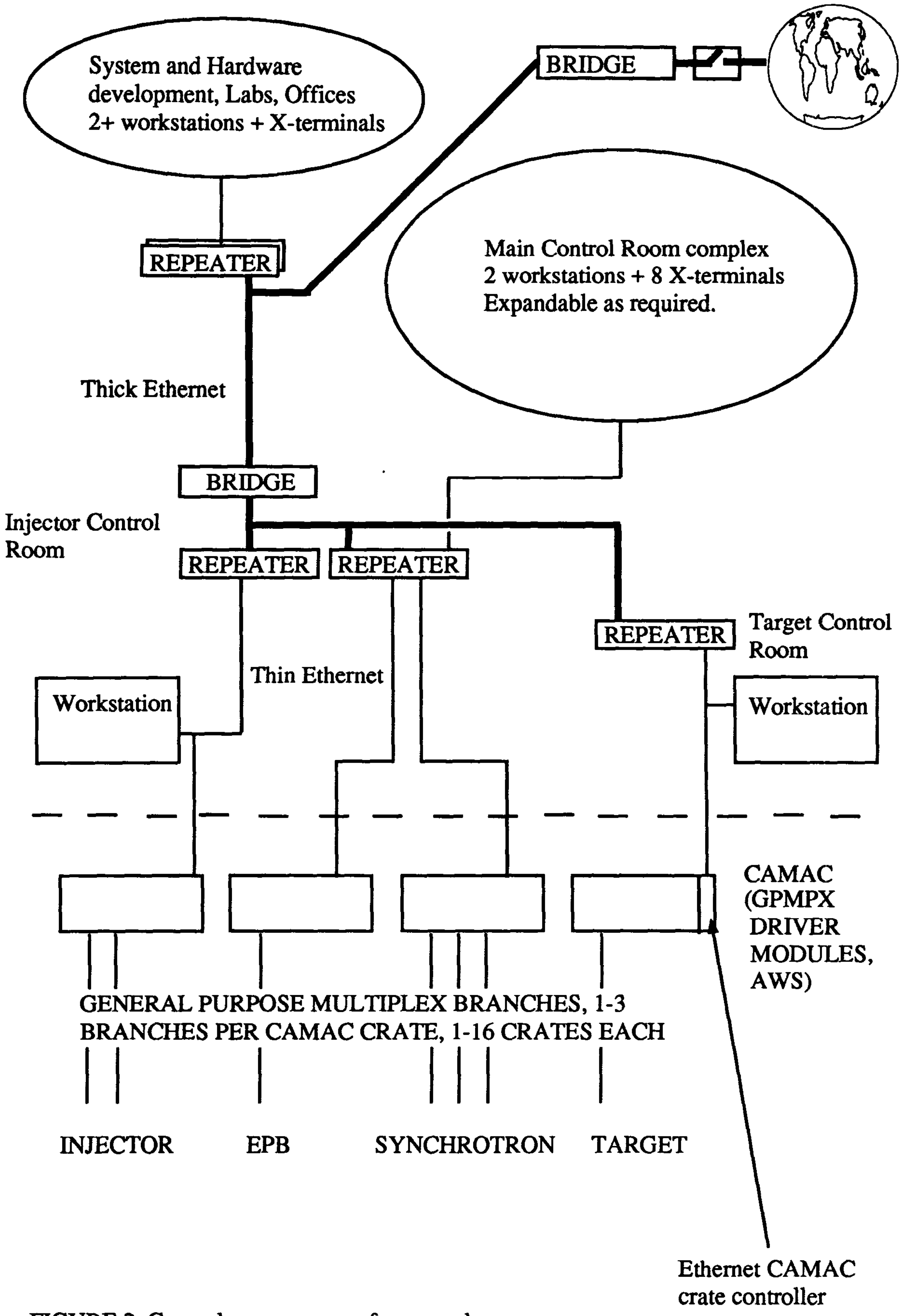
System and Hardware development, Labs, Offices 2+ workstations + X-terminals
BRIDGE
Main Control Room complex 2 workstations + 8 X-terminals Expandable as required.
REPEATER
Thick Ethernet
BRIDGE
Injector Control Room
REPEATER
REPEATER
REPEATER
Target Control Room
Thin Ethernet
Workstation
Workstation
CAMAC (GPMPX DRIVER MODULES, AWS)
GENERAL PURPOSE MULTIPLEX BRANCHES, 1-3 BRANCHES PER CAMAC CRATE, 1-16 CRATES EACH
INJECTOR
EPB
SYNCHROTRON
TARGET
Ethernet CAMAC crate controller

FIGURE 2. General arrangement of proposed system