CAN WE MAKE A CONTROL SYSTEM DO WHAT THE OPERATOR WANTS ?

Panel Discussion on Cyclotron Control Systems and Operator Interfaces

Chair: W. BUSSE, *Hahn-Meitner-Institut Berlin, Glienicker Str. 100, D-14109 Berlin, Germany*

Panel Members: Th. BLUMER, *Paul Scherrer Institut, CH-5232-Villigen, Switzerland*

J.C. COLLINS, *Indiana University Cyclotron Facility, Bloomington, IN 47405, USA*

L. DAVID, *GANIL, BP 5027, F-14021 Caen Cedex*

P. KROON, *Kernfysisch Versneller Instituut, Zernikelaan 25, 9747 AA Groningen, Netherlands*

A. LASSILA, *University of Jyväskylä, Accelerator Laboratory, POB 35, SF-40351 Jyväskylä, Finland*

M. MOUAT, *TRIUMF, 4004 Wesbrook Mall, Vancouver, B.C. V5A 1S6, Canada*

1 Introduction

W. Busse

An accelerator control system can be characterized by the widely accepted generalized view of a system composed of three levels:

- equipment level (lowest)
- transaction level
- presentation and application level (highest)

The *equipment level* interfaces the upper two levels to the accelerator equipment. It combines interface modules and field busses where it applies. Although the accelerator equipment proper is not regarded part of this level it has to comply with the rules of its standardized input-output (i/o) ports and their corresponding standardized i/o- protocol.

The *transaction level* is the transport and conversion level, in general set up as a computer network, and it interconnects to one or more configuration data base(s). It converts the more physics oriented language and human parameter names of the higher to the more hardware oriented language and addresses of the lower level and vice versa.

The *presentation and application level* is the interface to the human operator. It is at this level where the operator or machine physicist gets the look and feel of the accelerator. This level comprises the reasoning if required and the tools for automation and /or appropriate machine operation.

The three levels described are isolated from each other by a standardized programming interface, which prevents changes on one level from having an impact on others.

The panel discussion will concentrate on the uppermost level described and will have to face the fact that control systems are used at different times by different personnel with different and often conflicting requirements, e.g.

- towards the end of the constructions phase and later for detailed investigation of faults, extensive hardware and software test facilities and utilities are needed for the corresponding specialists,
- for the commissioning and later on for machine development, a wide variety of utilities to measure and manipulate the beam is required,
- in routine operation a comprehensive suite of applications software should allow for long-term exploitation of the machine.

With this in mind and to stimulate the discussion the panel members will present aspects and problems related to

- control systems growing over decades and ever facing new requirements which they have not been designed for in first place (cf. chapters 2,3)
- controls which have to integrate home made and commercially available control systems (cf. chapters 3,4)
- off-the-shelf industrial control systems (cf. chapter 5)
- automation (cf. chapter 6) and
- the evolution of control systems since computers have entered the field (cf. chapter 7).

2 The PSI accelerator control system Th. Blumer

Short history of the PSI control System

Taking the PSI accelerator control system as an example I should like to start my illustration of evolutionary steps by a list of historic milestones of the PSI accelerators:

- First beam through both cyclotron's 1974.
- First beam through Inj2 and Ring cyclotron 1984
- First beam of 1.5 mA through Inj2 and Ring cyclotron 1995
- First beam on the spallation source 1997

During all this time the control system was constantly improved and enlarged. To allow system development in parallel with machine operation we have chosen a highly distributed and modular system.

First generation ACS

The original controls equipment consisted of a hardware communication system with manual control stations used mainly for DACs and ADCs. In addition an IBM 1800 computer provided online access for specific tasks like 'save-restore', 'probe analysis' etc.

Second generation ACS

In 1980 with the construction of Inj2, a major upgrade for the control system was planned. The system consisted of three PDP11s with CAMAC interfaces and computer controlled operator consoles. The central VAX and later a dedicated MicroVAX was used for CPU intensive jobs.

Third generation ACS

For the spallation source SINQ, beam intensity had to be increased to 1.5mA. This and the general development in technology once more needed a complete revision of the control system. The goal was: an increase in throughput of more than a factor of ten, the introduction of workstations and the access to all parameters independent of their location. The first feasibility tests for the present distributed system were made in 1989. The proposal was presented at the ICALEPCS in Vancouver.

Present state of the control system

Operator interface

OpenVMS workstations, form the operator consoles. The GUI is Motif. A central machine acts as a disk server for common variable data. Programs and all static data are resident on each workstation. (300 ksFr)

Frontend (FEC)

HPrt 743, VME based RISC processors, serve serial CAMAC loops. The FECs receive lists of logical requests for i/o, execute them according to the implemented hardware and return status and result. This software is based on POSIX to provide portability. (200 ksFr)

Communication

The hardware path is a single collision domain of 10Mhz Ethernet. We have implemented an asynchronous protocol, based on connection-less messages, implemented both in basic Ethernet and UDP/IP. (50 ksFr)

Size of the control system process interface

In total there are more than 110 Crates with more than 1300 modules and 600 fieldbus sub-assemblies. There are a total more than 2500 devices, 8000 analog values and 5000 status signals.

Software

Over 450 programs and procedures are directly accessible from the console. Many of these are generic programs that are configured by a multitude of data files.

Investment in the control system

The above listed hardware price figures (in brackets) are estimations of the replacement cost of the presently used equipment and at their original prices. This figure of over 4.5 MsFr does not include cabling, terminals, racks etc. Also the development of components and the cost of equipment that had already been replaced is not included.

Manpower

Since the beginning in the 1970's there were on the average 3 persons doing software for the core of the system and 2-4 writing specific applications. The set of presently used software was completely created or transformed with an overall investment of approximately 50 man years. This figure includes development. The hardware manpower investment for the control system, without beam-diagnostics, has been also been in the order of 5 man per year since the beginning. But 75 man years for the present system could be a representative figure. These numbers are only rough estimations since it is not possible do discriminate between maintenance, development, replacement of obsolete equipment and other not directly accelerator related work. Also the definition where the control system hardware ends and where other groups responsibility begins is not easily defined.

3 The GANIL accelerator control system

L. David

The Ganil control system has two main characteristics:
First, it is a home made control system, and, second, the control system as a whole was never upgraded at the same time, but only part by part, which means that we always had to pay attention to the integration of old devices and interfacing crates. Today we have 3000 devices in 50 crates, with new VME crates but also with very old CAMAC crates and with a lot in between.

From 1981 to 1991, we used black and white, text oriented user interfaces, with CAMAC crates along a serial loop and we had no field bus. Then the new projects (THI, SPIRAL) needed a more secure system and it was necessary to change the user interfaces and to move from the single 16-bit computer to a distributed control system with workstations and real time crates connected to a network. However, we did not have enough time, man power and money to change all the system layers in only 3 months, during a winter accelerator shutdown.

From 1992 to 1995 we used graphical user interfaces which were just Motif based, with CAMAC crates connected to the network, and without any field bus. We did not change the CAMAC modules interfacing the accelerator equipment but only the controllers to move from the serial loop to an Ethernet coaxial network.

By this time, we mainly developed device oriented programs, like the one still used for tuning beam losses when injecting into the cyclotrons: The operators have to look at the beam shape or intensity, then they change the current in a steerer, look at the beam again, change another set of steerers and so on. Later on, we had to abandon software development for beam tuning, to be able to spend more time for experiments.

From 1995 onwards, we started using commercial tools such as XRT widgets, added to Motif, and an INGRES database which is mainly used to store all the beam parameters. We have also started using VME crates instead of CAMAC and we are on the way to change all our CAMAC interface modules, which turned out to be difficult to maintain, by cheaper and more powerful VME cards. The network has been updated to twisted pair, the crates and workstations are being connected to network switches instead of concentrators and we have set up a field bus for power supplies, motors and PLCs. With this new hardware we have started to develop new object oriented software to make new programs available faster than before.

Today the operators can use automatic beam tuning programs such as beam centering along the beam lines, or transverse matching. These programs calculate theoretical beam parameters, measure real beam characteristics such as the center of gravity and then calculate and set new parameters for a better beam automatically.

Since this winter, GANIL has a new large control room, only equipped with workstations. With the new hardware and software tools, we will be able to provide high intensity ion beams, stable enough for the Spiral tests with exotic beams in early 1999.

4 The KVI accelerator control system

P. Kroon

At the KVI we use two different control systems, a home-built system for the AGOR cyclotron and a partially commercial system for the beamline complex.

AGOR control system

The cyclotron control system was built in Orsay (France) during the construction of the AGOR cyclotron. In Orsay, a group of 4-5 people developed the software for the control system. Simultaneously in Groningen, 2-3 persons worked on the development of fieldbus (Bitbus) hardware and firmware. The development started in 1987 and completed in 1994 when AGOR was moved from Orsay to Groningen.

The system uses KAV-30 MicroVAX CPU's as front-end computers. Each of these front-ends controls one or more Bitbus fieldbusses that interface to the accelerator equipment. For the presentation layer OpenVMS workstations and X-terminals are used. All systems are connected via Ethernet. The front-ends are based on the VAX/ELN operating system. The only commercial component of the control system software is the graphical presentation layer, which is based on SL-GMS (Graphical Modeling System) (and, of course the operating systems VMS and ELN). All the rest (database, logging, hardware drivers) was written in-house.

Beamline control system

The beamline and experiment control system is based on the same hardware and operating system configuration as the AGOR control system, namely KAV-30 CPU's with VAX/ELN and Bitbus, OpenVMS workstations and X-terminals. The same suite of Bitbus modules was used that had already been developed for AGOR. However, the software is largely based on Vsystem from Vista Control Systems.

Work on the beamline and experiment control system started in 1992. This was shortly after Vista had rewritten their Vsystem for the X-windows environment and also had a version available for the VAX/ELN environment, thus fitting exactly to the hardware configuration we used for Agor. After a fairly long evaluation period we decided to use Vsystem for the beamline control system.

Most of the development work consisted of understanding the Vsystem peculiarities, and designing and writing the hardware specific software, i.e. the software that connects the Vsystem database to the Bitbus hardware. This was done by 1 person doing the software and on the average 2 technicians taking care of interfacing the beamline equipment to the Bitbus fieldbus. Most of the beamline equipment could be controlled through Vsystem when the first beam was produced in 1996.

Status

Currently both systems are maintained by 3 software persons with incidental support by 1 or 2 persons from the electronics group.

The operator consoles are formed by 5 OpenVMS workstations and on the average 4 X-terminals. There are 6 KAV-30 Front-ends, controlling 220 Bitbus nodes. Each Bitbus node controls either one or a few devices directly, or indirectly via 19 PLC's. These PLC control the vacuum, cryogenic and gas-regulation equipment.

Comparison

A quick comparison between the AGOR control system and Vsystem:

The Vsystem based beamline control system is more stable and reliable than the AGOR control system.

The AGOR control system is difficult to maintain, debug and expand. This is partially due to the fact that none of the persons involved in programming the system is available for support.

For a non-control system specialist, who wants to write pro grams in C or Fortran for e.g. measurement procedures, interfacing to Vsystem is easier and more efficient than interfacing to the AGOR control system.

In both the AGOR control system and Vsystem, the update/refresh rate of the presentation screens is sufficient. In Vsystem, the time to activate a new control window is experienced as too long. This is mainly due to the large amount of data exchanged between front-ends and workstations during the window setup phase and the limited performance of the front-ends.

When creating control screen pictures with Vsystem, one is limited to the set of control tools supplied by Vista. With the AGOR control system and the SL-GMS Graphical Modeling System, there is an much greater flexibility in creating animated control objects.

The Vsystem database organization is simple, not very flexible and proprietary to Vista. Also the communication protocol between the presentation workstations and the front-ends is proprietary.

The main drawback of Vsystem follows from the latter two points: in some cases the control system group has to disappoint the users as a request simply can't be fulfilled because Vsystem doesn't support the requested feature, while the controls group can't change it.

Plans

We have decided to migrate the AGOR control system to Vsystem. The main reasons are the simpler maintenance and greater reliability of Vsystem, and the reduction of the number of different systems to support.

We also plan to replace the KAV-30/ELN based front-ends by something more state of the art because the KAV-30/ELN combination is quickly becoming obsolete.

5 The control system at JYFL

A. Lassila

When the decision to build the new accelerator facility at Jyväskylä was made in the mid 1980's there was plenty of time to make plans for the cyclotron and its peripherals. The choice of a control system was not a hasty one. Starting from 1988 when the first offers were asked to the final decision at the end of 1989 several different candidates were evaluated and judged. Top level reliability and the cheapest offer were the main reasons for us to choose the ALCONT control system.

Pros

Today, the choice of an industrial control system has proven to be the right one. The hardware reliability is outstanding, because the system is designed to work in extreme conditions in process industry. Modularity makes it easy to expand and change and overhaul. Hot swapping of process interface cards makes quick repairs possible if they become broken.

The main reason why the total cost of the system was the cheapest are the application development tools. Normally these systems are delivered as turnkey projects for process industry, so there is very little to do for the final customer in these cases. In our case the system provider had no idea what a cyclotron is nor of how to control such a thing. Therefore they offered tools and training instead. With these semi-graphical tools the programming is really easy. All programs can be simulated prior to loading, which can be done on-line without stopping the processes. Loaded programs can be monitored on-line and control parameters can be introduced and adjusted while the programs are running.

The control system is easy to operate. We use student operators who are able to operate the cyclotron in steady runs after a two weeks training. If things get complicated the regular cyclotron crew can operate and help remotely. The system provider also has remote access to the system. System diagnostics and occasional fixes can be applied remotely.

Dynamic data-exchange between processes is an additional good and useful feature. Third party programs (e.g. mathlab) can be integrated to control devices. Information from the control processes can be gathered for reporting to Excel spreadsheets. New toolkits for various purposes are introduced to the market every now and then.

And cons

System providers tend to keep the unit prices high. The process industry is ready to pay anything for reliability. Accelerator laboratories on the other hand are very different customers. Price negotiations each time we buy something are tedious, but worthwhile. For bigger systems than ours it is much easier to negotiate the prices down to a reasonable level. However, if we include manpower in the total cost of the control system, then the prices are cheap.

The system software of the new XPR-processor cards only supports expensive SCI-Serial communication interfaces. SCI-cards are actually independent computers that communicate with the host XPR via the baseplate bus. On the other hand, the card has its own memory space, so the application does not occupy the host XPR as much as the old and cheap 8-channel MCL cards did.

When fast actions are needed, the 50ms polling cycle per 16 interface cards could be considered slow. Event based program execution and fast cycles are possible though, but the speedier the application the heavier the load on the host processor. Separate modules and intelligent interface cards should be dedicated for fast applications if needed.

The editor tools are not quite at the level of mainstream windows programs. On the other hand, the fewer the commands there are per menu the easier is the learning of their use. The occasional system reports of errors etc. that give no hint what actually might be wrong also may annoy the user. Some of these messages are meant for the system personnel only, and getting more information from them is sometimes difficult. Perhaps the information is classified and the system providers fear that critical information could fall into wrong hands.

Summary

During its 8 years of operation the control system has required no more than about 12 man years of work. Most of that work was done during the first two years when the accelerator facility was under construction. During recent years the control system has been working more or less on its own, only some application changes were made and scheduled services done. From our point of view the system has saved us lots of trouble and it has saved us time and money as well.

Reference

A. Lassila, E. Liukkonen, Evolutionary steps of the control system at the Jyväskylä accelerator facility, *Contribution to this conference*

6 Is there a need for automation

J.C. Collins

As in many areas of human endeavor, cyclotron controls are subject to ideas which rise quickly, get lots of attention and then fade away. The question for discussion with the audience is whether automated controls, using artificial intelligence techniques, was such a fad, is still a fanciful idea or will be the future of control systems. By "automated controls" is meant source-to-target beam tuning without human intervention. "Artificial intelligence" is meant to include rule-based reasoning systems, fuzzy logic and neural networks.

This question arises partly from personal recollections and partly from examination of proceedings of past cyclotron conferences. The first identifiable references to automated control are a review paper by W. Busse and a description of UNILAC tuning procedures by L. Dahl at the Caen conference in 1981. One can find a proposed automatic control system by A. Jankowski (Tokyo, 1987), a second review by W. Busse (Berlin, 1989, " . . . expert systems have not yet found many significant applications . . .") and a beam transport system using fuzzy logic by J.S. Chai (Cape Town, 1995). One could conclude from this list that artificial intelligence techniques have proven notably unsuccessful over the past 17 years.

There are continuing efforts to realize AI style controls. Vista, Inc., has conducted trials at the Brookhaven ATF and Argonne ATLAS. A commercial concern is developing a system at SLAC and Trieste ELLETRA is reported working on AI tuning. It seems fair to say that these projects have met with varying degrees of success.

One may ask why so little progress has been made in 17 years. Some possible reasons upon which the audience might comment are:

1.) Many cyclotrons operate in a single particle, single energy mode which, with minimal care, is easily reproduced. Simple hardwired feedback on the main B field and/or a critical steerer or two is sufficient for normal operations.

2.) Diagnostic elements are critical to AI control but they cannot be located where they are most needed (a corollary of Murphy's Law), their signal/noise ratio is too low and their outputs are used in incomplete or inaccurate models to calculate effector values.

3.) AI is not Object Oriented enough to get the attention of today's computer professionals.

4.) Accelerators actually need the fuzzy thinking, rule breaking, ill-logic of a human operator.

7 Evolution of accelerator control system implementations

M. Mouat

One perspective on the evolution of accelerator control systems allows them to be grouped according to the manner in which the software is developed. This approach leads to central control systems with four types of software. In almost all systems prior to the early 1970s, there were no computers and thus no software. These systems were "hardware only" (type 1 - all hardware) and all control was exercised by devices such as knobs and buttons with various types of displays (meters, storage scopes, chart recorders, etc). In these systems, many devices could not be remotely monitored or controlled. When computers were introduced to accelerator controls in the early 1970s the software in the control system was an in-house custom design and implementation (type 2 - in-house custom). As time passed there was a desire to try to save money and to follow industrial practices so industry was contracted to provide customized software to meet the requirements (type 3 - commercial custom). The latest trend is to use extensible toolkits to meet the requirements (type 4 - toolkits, a)commercial and b)freeware). A toolkit is a collection of application programs and infrastructural software that allows a user to configure a control system, given acceptable hardware, with a minimum of programming. There are both commercial and non-commercial toolkits and their beginnings occurred at approximately the same time. The most common non-commercial control system toolkit is supported primarily by an international collaboration of research facilities. Not only has there been an evolution in the control systems but in the developers as well and both have an impact on the end user.

In hardware only systems (type 1) the developers were generally physicists, engineers, and electronic technicians. Because control capabilities in these systems were vastly more restricted and less complex than today, the end user could quite easily participate in the design and implementation of a solution not just in defining the requirements. In the hardware only environment the end user could get the level of satisfaction he/she desired by doing the work himself.

Moving on to in-house custom systems (type 2), the software is developed by in-house programmers that generally work from requirements given by end users. This system allows a great deal of flexibility because the staff programmer can tailor an application very closely to what the user wants, even if that means providing rather eccentric or non-standard user interfaces or features. Such tailoring can provide very efficient or convenient interfaces and accommodates new ideas if the programming resources allow the development to occur. This approach can consume as much programming resources as money will support and these resources have been the target of cost reduction.

In an effort to save money by reducing the number of staff programmers, control system software has been outsourced to industry. In commercial custom systems (type 3) the development is done by programmers working for a company under contract and using a very specific requirements document. Experience in this area seems to indicate that the initial cost can be lowered but there are several problems that arise. One problem is that specifying the requirements so precisely is difficult for most end users. Another significant issue is that after delivery of the control system software, most accelerators are very dynamic and changes are required often. Having a commercial supplier provide changes is very expensive and typically not timely or convenient. A third issue is that having the software outsourced means that the laboratory does not have the same level of programming support on site. This means that there is less exploration of new software technologies and less support for related but not specifically control system applications.

In an effort to keep the software costs down but still provide a better level of flexibility in ongoing developments and changes than in the commercial custom systems, toolkit systems (type 4) have evolved. The developers of the commercial toolkits are basically the same people as those providing commercial custom systems, that is commercial programmers. In the case of the primary freeware toolkit the developers are an international collaboration of laboratory programmers. The toolkit approach does not provide a delivered control system as in commercial custom systems. Toolkits require laboratory staff to learn the tools and then do the system configuration. Because the tools are numerous, flexible, and complex, the learning curve is considerable. This means less initial saving than the commercial custom approach but hopefully during the life of the project, there is a significant saving on changes that occur. For toolkits there is also a compromise from the commercial custom approach of no staff programmers to having some programming and configuration support but less than the in-house custom approach.

The current trend is to reduce the number of programmers and instead do configuration using flexible tools. This will likely lead to a greater expertise in tool knowledge and less programming expertise in the controls group. If this occurs, there may be an impact on other areas of software development because the controls group, at least at TRIUMF, has also helped other groups with programming applications such as databases and diagnostics.

Typically, toolkits support a wider variety of applications than will be specified from a requirements document and should provide a more generalized approach to requirements. This should help to satisfy the user's needs in a timely fashion because more requests should be able to be handled without additional software development. Experience shows that some users will become fluent with the tools and can help themselves in doing developments. One caveat is that users may not be so free to specify how they would like to accomplish a task. They will be able to get their result but they will have to do it the way the tool does it. Changing a tool's functionality has several implications. In commercial toolkits the company owning the toolkit would have to be convinced that the changes were desirable and then usually paid to make changes. In freeware toolkits the local controls group would want to remain standard and thus has to convince the collaboration that the changes are desirable. At that point they either do the changes themselves or find someone with the time and resources to make the changes. Changes to toolkits will occur more like the release of a new version of an operating system, where changes occur less frequently but you get a number of enhancements at once.

Currently, many new facilities and major expansions to existing facilities are using the toolkit approach but both new commercial custom systems and new in-house custom systems are common. New commercial custom systems are more likely to be sub-system controls, PLC systems, or turnkey systems on small dedicated accelerators. Most existing facilities are not replacing their existing control system software but continuing to maintain what they have because after years of use it is very likely meeting their general requirements. Knowledge on the life-cycle of toolkit systems is still in its relative infancy but currently facilities using toolkits give very favorable reviews of their experience. In the future, the impact of web based applications can be expected to be significant but so far the impact has been limited.

8 Summary of questions and answers

Topic concerned with accelerators for industrial or medical applications

A control system should "run" the machine, we do not want to "operate" it. It should include fully deterministic tuning methods, you should design in automatic procedures for all operations. The system should include diagnostic knowledge and present informative alarm messages and propose actions to be taken. All personal flavors of the accelerator should be taken away or should at least be hidden.

While this may be good for industrial production machines it is not sufficient for research where constantly new methods are wanted. Here the operator needs a high performance interface to the process to cope with the different situations.

Concerning automation, PSI uses three stages in setting up their machines. First best known values are set. Then beam properties are measured and corrected to theoretically correct behavior, (beam position, beam optic, cyclotron phase, etc.).Finally the operator will tune the beam to best transmission (remember PSI runs 1,5 mA and has to reduce every possible loss).

Topic concerned with safety interlocks

What is your policy concerning interlocks?

The personal safety system is completely independent from the control system. Although it is computer based, it incorporates its own computer solely dedicated to this task. It acts like a hard wired system.

Equipment safety is in general guaranteed by hard wired interlocks. The overall machine protection system is part of the control system, most signals can be disabled by the operator, they are then clearly shown as disabled.

Topic concerned with hardware or software obsolescence

Old systems tend to have obsolete hardware after some time. When do you change your hardware ?

In general control systems for accelerators in the research field keep their hardware as long as it can be maintained at a reasonable price. If new requirements come up, one will have to compare the cost of a reasonable new investment to the drawbacks that might follow from staying with obsolete hardware and the as-is hardware concept. However, present control system implementations are generally modular enough to allow for new hardware when new requirements are coming up, i.e. in general new (and may be only new) requirements entail new controls in the research field. This may be different for industrial applications.

What would you answer if your management told you to change to EPICS?

We are happy with the present system and I see no advantage in a switch. I would have to ask for

- 4 MsFr for new hardware
- 50 man-years for the implementation of the software
- 0.5-1 year of beam time for implementation and testing.

This corresponds more or less to the effort deployed for the SLS control system.