CONTROL SYSTEMS UNDERWAY TO BE IMPLEMENTED AS EXPERT SYSTEMS

Winfried Busse

Hahn-Meitner-Institut Berlin GmbH

Glienicker Str. 100, 1000 Berlin 39, Germany (West)

1. ABSTRACT

In their early days accelerator control systems have been described on within the context of the accelerator they had been designed for, i.e. within conferences on cyclotrons, tandems, ion sources etc.. Since the early 80's the control system design and implementation tends to evolve as a discipline of its own generalizing the view of process control systems to make them independent of a specific process environment. The paper intends to report on the state of the art and the direction which control system design is (necessarily) going to take.

2. INTRODUCTION

An accelerator control system acts as the interface between a human operator and various machines in varying operational states generally called the process. Control systems are computer based and cover a whole range of applications such as surveillance, man-machine interaction and even closed loop control. Solutions exist for small as well as largely extended systems acting through computer networks.

Given today's state of the art the hardware can be designed to be modular, highly standardized and with such local intelligence that standardized control protocols can be implemented even at the base level. As a result these systems are reliable, easily maintainable, flexible with regard to the requirements and still extendable. Most components are commercially available. Being computer based modern control systems have opened a new field of applications which cannot be made available in hard-wired controls.

However, accelerators are mostly one-off machines with specific requirements. They often also enter a new technological field. Therefore, control philosophies may be transportable to new accelerators, but a specific implementation normally is not. New implementations are underway which might help to get around this problem, mainly since intelligent workstations have come along with powerful tools to alleviate applications programming. Nevertheless, the right tools to generate control systems to given specifications are still missing, and introducing 'expert systems' into this field does still not appear to be feasable.

3. EXPERT SYSTEMS

Expert systems are a branch of artificial intelligence (AI) which in turn is a discipline of computer science. An expert system uses computers (processing ability) to perform a task which requires human expertise. The list below shows some of the many advantages of expert systems over conventional systems[1].
Expert systems
- combine the best capabilities of many human experts into a single 'best operator',
- make the 'best operator' available at all times,
- make knowledge explicit (codify expertise),
- can be used as a training tool for operators,
- have consistent behaviour,
- test the expert's notions,
- force participants building the expert system into a greater understanding of the task.

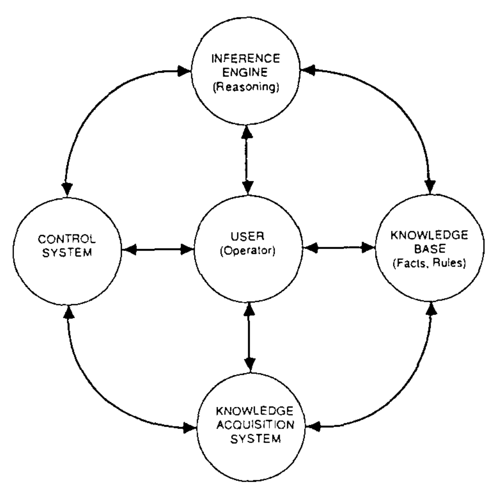

Fig. 1 Organization and interrelations of components of a typical expert system

A schematic view of the organization and interrelations of the components of a typical expert system[2,3] is given in fig.1. An expert control system can be organized into five elements and their eight relations:

- the (conventional) control system
- the user (operator, machine physicist, ...)
- a knowledge acquisition system
- a knowledge base, containing facts and rules
- an inference engine (the reasoning part)

In general a conventional process environment has only two of the elements, the other three reside in the users' head. The latter three are extracted by the expert system and then put into a computer. One of the biggest difficulties with constructing these systems lies in the so-called 'knowledge-acquisition bottle neck'. In traditional knowledge-acquisition a programmer interviews an expert to 'extract' his knowledge. These programmers usually have little or no training in interview methodology and the entire process is laborious and inefficient. Learning systems provide partial answers to this problem by automatically codifying knowledge that would otherwise have to be manually elecited and encoded.

In the accelerator field expert systems have been proposed or are being applied to numerous problems[4,5] such as data interpretation, monotoring, design, diagnosis, trouble-shooting, configuration, equipment-tuning and accelerator beam-line fault-finding. Other fields include e.g. medical diagnosis (MYCIN) and identification of chemical compounds (DENDRAL).

4. ACCELERATOR CONTROL SYSTEMS: STATE OF THE ART AND PRESENT TRENDS

Up to now the process environment of accelerators mainly consists of two components of the expert system of fig.1, the control system proper and the operator. In general the knowledge gathered by the accelerator designers is lost to a far extent when the accelerator moves to the beam production phase, only a few facts and rules are collected in so-called application programs to aid the user in operating the machine. Facts and rules gathered in the design commissioning and operating of one machine have to be regathered for the next generation accelerator.

Nevertheless, standards (e.g. design and operating standards) have evolved within the 'control system community' which can be found in the majority of the running systems. But they mainly still exist as uncoded knowledge and it is felt necessary to make them commonly available through tool kits.

4.1 Control System Architecture

For all, but the smallest systems, distributed computing has become universal. Nearly all distributed systems can be devided into three logical hardware layers

- an upper layer made up of the operator interface and the application computers
- a middle layer of processors distributed around the accelerator and dealing with sets of equipment
- a lower layer of processors in the interfaces to different types of equipment.

The layers are interconnected by a communication network as the backbone. In most cases this is a local-area network for the upper and middle layers with multidrop or individual links connecting the middle layer processors to those of the lower level.

The software architecture splits up in a similar way into

- an upper level comprising the application processing, including the operator interface,
- a middle level, transferring operational actions into hardware interface actions and vice versa, in general data driven,
- and a lower level of equipment driving and hardware access routines.

A (distributed) data base is the backbone with its access routines from and to the upper and lower levels. It allows to separate the upper and lower levels from each other and to make them independent of each other by operational protocols which only deal with 'meaningful' operational parameters and which are transparent with regard to distribution and specific hardware.

4.2 Software Methodology

The common hardware architecture is already suposted by generally accepted standards with written specifications, the required hardware is commercially available to a far extent. It goes without saying that in turn increased emphasis is now put on techniques for designing software in a formalized way to reduce the increasing cost of ever more sophisticated programs.

Structured-Analysis / Structured-Design tools (SASD) are being discussed and have been proposed or adopted for some systems. Contributions to the 1987 International Conference on Accelerator and Large Experimental Control Systems at Villars[6], Switzerland, showed considerable interest in Object-Oriented techniques both for programming and for system design. As a matter of fact Object-Oriented Programming (OOP) is a formalization of a principle which has been used for quite some time in control system implementations by introducing Data / Control/ Equipment structures.

All these techniques usually demand a greater investment of time and effort in the early stages of the design, but they promise rewards in a better overall understanding of the design problem, in a common language among designers, in shorter debugging times and easier maintenance as well as the possibility of reusing all or part of the results.

In addition to using SASD and OOP techniques an approach is made to simplify application programming by the use of sophisticated graphical editors or systems which allow the dynamical attachment of symbols on the screen to the control elements they represent. Sometimes they even generate the necessary data structures to provide for their connection to the corresponding process parameter.

4.3 Automation

The following example may serve as an illustration of the latter technique. It was presented by P.Clout et al. on the occasion of the 1988 Accelerator Control Toolkit Workshop[7] as a tool-kit proposal to implement a control system for the fully automated Ground Test Accelerator (GTA). It must be emphasized that full automation typically requires an expert system, whereas the present GTA approach is a first step towards an automation toolkit.

Automation[8] is the intelligent application of a model to a process to consistently produce the desired output. This can be conceptualized in seven levels (cf. fig. 2):

1 - Data Acquisition

All process data is collected through the control system. This data is used to monitor and study the operation of the process as well as develop the control algorithms.

2 - Supervisory Control

Manual remote control of the process is available to the operator through the control system.

3 - Model Support

The control system incorporates modeling programs that allow the operator to test proposed changes against the model. The model is also used to suggest corrective changes. The operator still controls the process manually.

4 - Continuous Control

The set points of the control loops are available to the operator through the control system and the maintenance of the set points is performed by the control system. Steady state operation is accomplished through this mechanism.

5 - Sequential Control

The transitions required to change the state of the process are defined in the computer. The operator, the state of the other subsystems, the state of the components in this subsystem or a master sequence can initiate these changes. Automatic startup, operation and shutdown are accomplished through this mechanism.

6 - Fault Recovery

Sequential control is applied to automate fault recovery. These sequences restore the operation of the process, perhaps in a degraded mode.

7 - Optimization

The operation of the process is optimized by set point adjustment and possibly state changes to achieve optimum performance. This level requires extensive study of the process parameter space. The solutions which are algorithmic may be described by continuous control mechanisms. The solutions which require some sequence of steps may be described by the sequential control mechanisms. The solutions requiring adaptive learning will require some artificial intelligence (AI) mechanism.

According to this conceptualization the GTA group proposed the 'magic tool box' which is visualized in fig.3. By the use of the graphical editing techniques mentioned earlier each tool generates the appropriate data structures which are processed by the 'engine' to run the envisaged application.

4.4 Advanced Techniques

The requirements on the beam quality of modern and mostly complex machines can only be met by using computer based machine-modelling and beam-simulation techniques to find the optimum operating conditions. Model-based accelerator control systems allow the operator to test proposed changes against the model, preferably off-line by simulation. The model can also suggest corrective changes. There are recent accelerators which are told to not have run without the extensive use of fast digital feedback based upon models.

One of the successful examples is the LANL/Stanford/SLAC developed GOLD program which takes

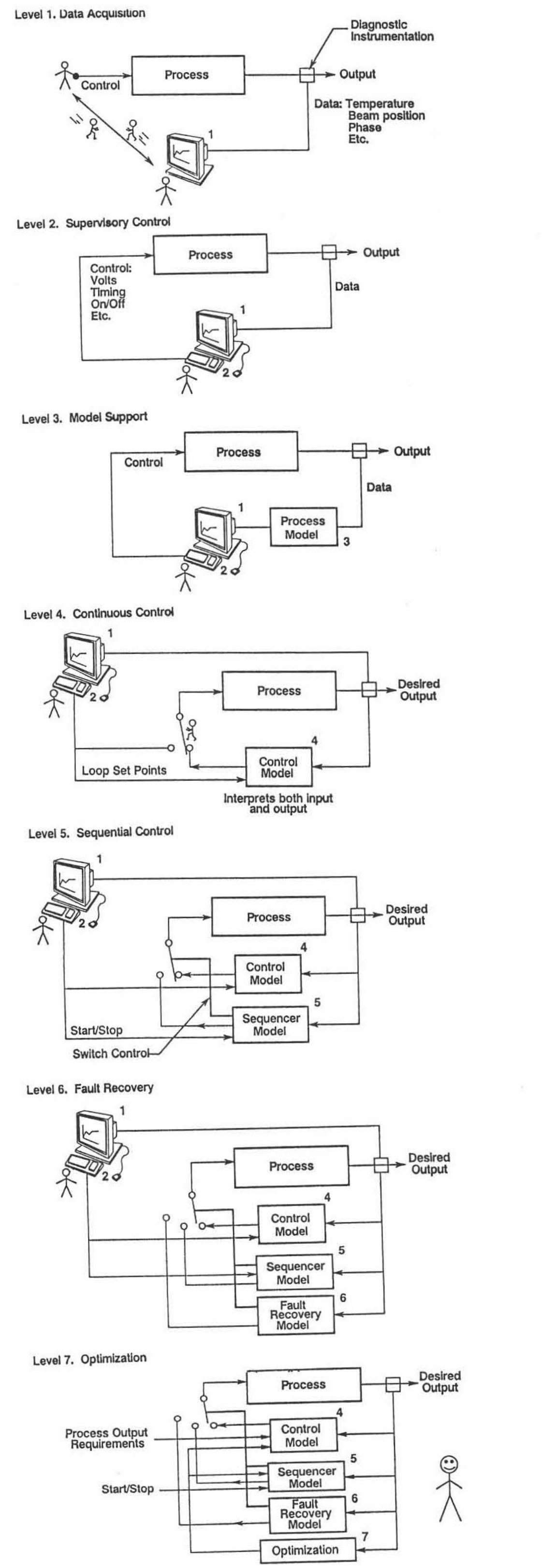

Fig. 2 Automation conceptualized in seven levels of control

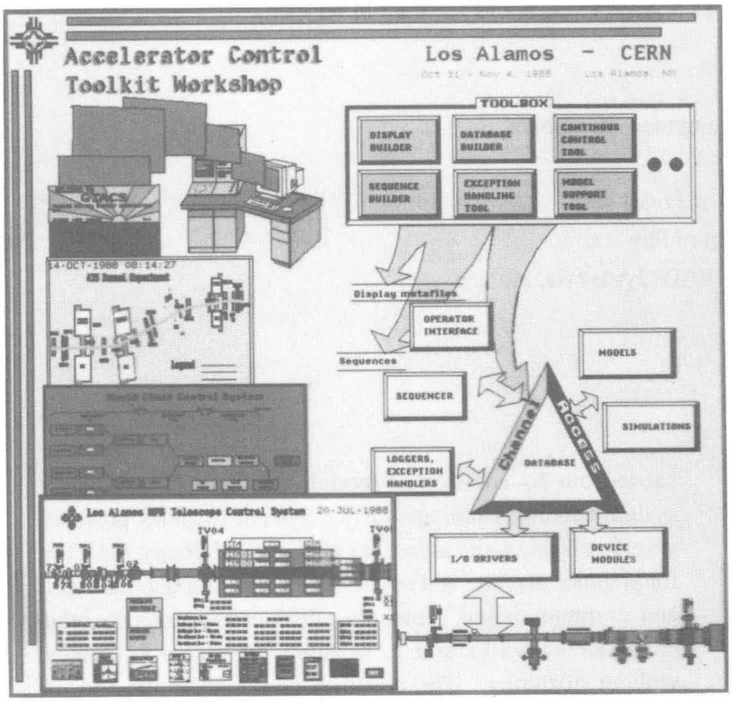

Fig. 3 View of the magic tool box

beam position measurements and suggests changes to the beam line (e.g. setting values, element properties) by using the code COMFORT[9]), which is an optimizer and a mathematical model of the beam line.

The operator interface is now dominantly provided by the new 'workstations'. The workstation is a powerful processor running a sophisticated graphics system. Although many graphics standards are in use, general agreement seems to aim towards a single standard as a basis for future work (X-windows or eventually MOTIF). These workstations have so much computing power that they can even be used for running machine modeling programs.

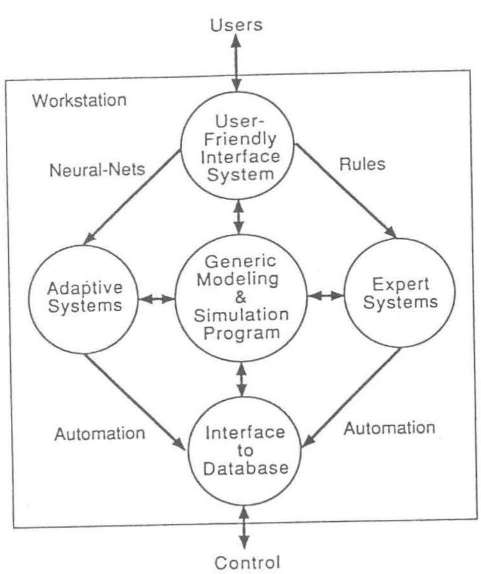

Fig. 4 Schematic diagram to show how neural networks, expert systems, modeling and simulation codes, and a graphical user interface can be combined into a single system

M. Lee et al. have designed a graphical environment[9]) for model-based control on a DEC VAX station with the intent to port the graphical interface to other workstations. Their proposed WORK station Solution is being developed in a portable trajectory analysis and correction system which can be used in any control system. Fig. 4 shows how neural networks, expert systems,modeling and simulation codes, and a graphical user interface can be combined into a singlesystem.

The self-learning property of neural networks may provide solutions to problems (non-linear, multidimensional interdependencies) we do not yet know how to approach.

5. CONCLUSION

Although a considerable step towards formalization of control system design has recently been taken and although more advanced techniques have successfully been introduced, expert systems have not yet found many significant applications and so far only fairly trivial cases have been attempted. To obtain the knowledge base from appropriate experts seems to be the most difficult problem. No tools exist to help with this job. In view of the increasing collaborative activity within the accelerator control community and the tremendous progress made in the field of smart human operator interfaces it will be very interesting to see the reports of the next International Conference on Accelerator and Large Experimental Control Systems to be held in Vancouver towards the end of this year.

6. REFERENCES

1) Clearwater, S., Papcun, G., Clark, D.,1985 Acc. Controls Workshop, Nucl. Instr. and Meth. A247 (1986) 193.
2) Buchanan, B.G., Shortliffe, E.H., "Rule-Based Expert Systems", (Addison-Wesley, Menlo-Park, 1984).
3) Buchanan, B.G., Smith, R.G., Ann. Rev. Comput. Sci. 3 (1988) 23.
4) Clearwater, S.H., Lee, M.J., Proc. 1987 IEEE Particle Accelerator Conference (Washington D.C.), IEEE Publishing, New York (1987) p. 532.
5) Selig, L., Clearwater, S., Lee, M., Engelmore, R., Proc. Second Workshop on AI and Simulation, Seattle WA, 1987.
6) Proc. of Int. Conf. on Accelerator and Large Experimental Control Systems, Villars (1987), to be published as CERN Yellow Report.
7) Accelerator Control Toolkit Workshop, Los Alamos, Oct. 31-Nov. 4, 1988, organized by the Los Alamos National Laboratory and CERN, no proceedings
8) Clout, P., "Towards an Automation Toolkit", contribution to the Accelerator Control Toolkit Workshop (ref. 7).
9) Lee, M., "A Graphical Environment for Model-Based Control", contribution to the Accelerator Control Toolkit Workshop (ref. 7).