BUILDING THE NEXT LARGE ACCELERATOR CONTROL SYSTEM

Chairman: S. Kurokawa (KEK)
Panel: R. Bailey (CERN), G. Baribeau (CERN), H. Brandis (GSI), G. Daems (CERN), D. Gurd (TRIUMF), P.G. Innocenti (CERN), R. Keyser (CERN), P. Lucas (FNAL), S. Schaller (LANL), I. Wilkie (CERN)
Animator: B. Ruiper (CERN)

Animator's Introduction

Ladies and gentlemen, in this last session you might think finishing to a bit more frivolous tune. The substance, however, may prove considerably more serious than the title indicates.

I have deliberately chosen somewhat controversial themes. They are so much more fun to talk about! In the past, several of us had difficulties to start a controls project or we had difficulties in the middle or at the end. The origin of the difficulties have mostly been at the level of common sense and, also, in what some choose to call sociology. So I have included a few of these not too technical topics. Above all, however, there are a number of topics, some advanced, in various stages of exploration in the labs. Other ones older but still controversial since the - sometimes ideological - discussions around them remain very animated. I invite you to concentrate and comment on their potentialities, their possible impact, trends and possible difficulties around them.

I have deliberately avoided some highly technical things like transmissions or buses, operating systems etc. Although these are very important, whole words of problems in themselves, the precise choice of these has rarely been decisive for the success or otherwise in making controls systems for large physics installations. Besides, most of those have been treated in the preceding workshops of this conference.

A quick summary of chosen topics.

1. Controls early and with full support. One major problem with controls is that they usually have been an afterthought. People designed an accelerator and at a certain moment they discovered they needed controls. Now controls is such an important part today, that you simply cannot do it that way any more. You must start early and with full management support.

2. Major new requirement/trend/breakthrough. I said earlier that we have rarely fallen flat on our face as a result of purely technical problems. Nevertheless, a major new requirement, trend or technology breakthrough may pull the rug out under all your plans.

3. Operational & control protocols. Large systems tend not to be as homogeneous as we would wish for. Due to different design teams, different manufacturers. Even more so because of improvements, upgrades, using new technology, which cannot be retro-fitted to the whole system. Every change then creates a nightmare of adaptations spreading out over the system. The only hope for limiting the impact of change is defining and adhering to logical interfaces, i.e. protocols, between subsystems.

4. Autoconfiguration. Related to interfacing, but a bit more advanced hence controversial. Imagine a system which has identical outlets at relevant points in the field and into which outlets you can just plug any process equipment. The controls system then interrogates the equipment you have plugged in, finds out what it is, configures itself for dealing with that equipment and gets going. Would be nice and the impact could be enormous.

5. Operational aspects. I have said that controls have often been an afterthought. After those controls were implemented, the next afterthought then was to use them for operations. I feel that it is desirable to have the operations people in early, in order to define what they want. Whether they can, we will see. In many cases they could.

6. Data bases. Any control system is a collection of larger and smaller data bases, which interact. Should we not see this as a whole for once, or even henceforth, and deal with it on a system's basis and early on?

7. Application structures. One can make a controls system in which applications are single entities, written in a boy-scout way by individuals. And you can make such a control system work. But as applications get more complicated and as their number and size grow, you may benefit by structuring them and by providing an infrastructure, meaning an environment in which you can make applications more efficiently and more reliable. In order to be economic this must be done early on.

8. Controls system exploitation. Running a controls system requires substantial effort but exploitation also uses to be an afterthought. On the other hand, the collective experience in exploitation may well help us to build better control systems. We should therefore bring this expertise in early on.

9. Automation. Probably it is true that what can be automated should be automated... well, within reason.

10. Modeling and expert systems. Finally we have these modern buzz words. These two are not unrelated and, although their use is only in the early stages, it is by now clear that the impact in the coming years will be strong.

Discussion

Controls early with full support

R.K.: Let me now fire the first question at the audience: do you think one could get managerial support, meaning that the resources be allocated in reasonable proportion for starting on controls in a significant way as soon as the first outline of the accelerator (or whatever it be) become visible?

Dohan, TRIUMF: TRIUMF have submitted their proposal for a kaon factory and in that proposal, at the outset, there is a very strong controls component both in terms of manpower and in money as a percentage of the total cost.

R.K.: Do you want to say that you have actually spent money and manpower in parallel with the accelerator study?
Dohan, TRIUMF: In the controls study two or three man years have been invested, probably like for the other studies, beam physics and so on.

Dohans, CERN: In a well managed organisation, official support will be subject to economics considerations. Can we quantify what we will get back from a proper control system? Cost/Benefit estimation will be of great importance.

Knott, ANL: Argonne is putting together a proposal for a 7 GeV Synchrotron Radiation Source, like DESY in Grenoble. We have put in an effort of 4 to 5 man-years, comparable with TRIUMF and it has enthusiastic support. It is considered to be a difficult job, since concerning a difficult storage ring.

R.K.: Maybe things are actually changing? I think that my friend Schaller has prepared a relevant statement.

Schaller, LANL: It occurred to me last evening that our job of selling a control system is somewhat simpler now than it was say 20 years ago when Lampf was being built. Lampf was one of the first machines to have an integrated computer assisted control system. The leap of faith that must have occurred in the early sixties must have been large compared to what we’re talking about at this point...

We have been looking at possible designs for a control system for the Advanced Hadron Facility which may or may not be built at Los Alamos. We have come to the conclusion that we CAN get early participation if we can sell it as a control system providing an integrated environment for the entire accelerator and that such an environment would be to the common good of everyone involved in the machine. For the Large Hadron Facility, we decided to approach the controls in terms of aspects that are not directly hardware related. I put the following question to the floor: how far can one carry these control system studies independent of specific hardware?

Allison, KFA Julich: We built a system in Canada for the Varennes Tokamak and we had colossal difficulties because no hardware whatsoever was specified. We had to go on and build a generalised control system for an undefined machine. You can go ahead and start building but when it comes to commissioning everyone points the finger to you when it does not do all the things it has to do.

Benincasa, CERN: We are presently doing the design for a new European Hadron Facility, which is comparable to the TRIUMF Koon Factory. The design for the control system is just a course one because there is only the conceptual design of the facility. Nevertheless, a certain number of characteristics can already be foreseen. For example, the machine is planned for very high intensities and a lot of monitoring and computing is necessary to prevent damages. It is of course not sensible to attempt a detailed design of the control system hardware but the main characteristics in real time, or for preventing damages, may already be specified.

R.K.: In addition, it is probably not too complicated to make a course estimation of the price because you have an idea of the number of channels required by the machines and, extrapolating from existing machines, you can make a fair guesstimate. But a definition of detailed hardware and software implementations does not make sense because this machine may come in three years, maybe five years, and the commercial market evolves rapidly.

Debinson, CERN: As chairman of the controls design study at TRIUMF, I could identify two problems. First, it was not at all easy to convince management to invest time and effort in controls early. It took much longer than the heads before the would officially sanction it. I am afraid Dohan presented the situation too easily. The other thing is that we really had problems to understand what the true requirements were from the accelerator physicists. I do not agree that you can get a good estimate of the price, because you can get an idea of the precise number of channels. You have machine people grossly overrating their needs and others grossly underestimating them.

R.K.: It seems that we cannot escape that fact that in this sort of inquiries into the controls ’needs’, we need people who have a judgement on both controls and on accelerators and a good dose of common sense in addition. The surest passport to incongruent system’s definition is when each party know only their part.

Benincasa, CERN: You can make some cross checks. We estimated roughly between 10000 and 20000 channels for the European Hadron Facility. This makes sense when comparing with the CERN FS, the AGS and others. The CERN Booster has 6000 in comparison. I estimate the precision at 20 or 30%. I do not think the error is a factor of two. This sort of hadron facilities requires additional channels for instrumentation and, although this is not easily assessed, it may be allowed for in a generous reserve.

R.K.: The channels are important because probably half the total money is in the front end. Clearly, higher up in the architecture you have a variety of choices which can influence the price. But a factor of two there corresponds to 25% in the total, so Benincasa’s point seems reasonable.

Debinson, CERN: If you wait until the acceleration is completely specified then you may end up with zero control systems when the accelerator starts running. It must be an iterative process. Positive in our case was that by proposing something, by asking dumb questions and in particular by then writing it all up, we provoked a lot of the machine physicists to think hard about what they really needed.

Schaller, LANL: I think that this discussion of exactly how many channels is a bit premature. I would much rather see emphasis on generic controls functions such as Peter Clout’s functions from the automation session. I would like to see analysis of knowledge sources available in a particular laboratory where the accelerator is being proposed and the activities that are necessary to acquire knowledge, not just to build the machine but to actually run it. Knowledge you need for the design, knowledge for the commissioning phase, knowledge for the operations phase. And, I do not quite know how to do it, but I would like to get engineers and accelerator physicists involved in this thinking of knowledge that is needed in a systems context, not the number of channels.

R.K.: I agree, but you probably need both. You do have to estimate budgets!

Dohans, CERN: I’ve gone through such an exercise for the ACOL machine recently. I did not have too much trouble with the machine people. We only started when the machine was entirely defined but the controls were ready in time all the same. In fairness one must say that the ACOL controls were only an addition to the existing control system of the AA.
Knotl: ANU: Processors are becoming more powerful for the same price and networks are doing the same. This works for designs that are expandable in different directions. The number of channels may then always be adjusted later. Unfortunately these developments do not affect the manpower cost and the software generation cost which we need to know early for funding by DoE. The number of channels is almost a non problem. It is the cost that is the big problem.

B.K: I still think that half of that very cost is proportional with the number of channels, so a coarse estimate of that number is necessary early on.

Russe, HMT: We may not forget that CERN needed a long number of years for gaining the experience which now enables them to define an accelerator and its control system the way they do. In contrast, people just starting from scratch are in the position as was pointed out before.

B.K.: The collective experience in the world of accelerator controls is very large and if we think of collaboration... all have got a chance.

Rousch, CERN: A statement about a different kind of support: after the essential decisions on the architecture, methodologies, etc. one should have the official and consistent support of management to implement what has been agreed and that divergence of philosophy and dispersion of effort may be kept to a minimum. So, for economy in resources in large projects it is essential that management supports the controls group as the leader of what has to be done over the whole area of controls that we covered in our first workshop.

B.K.: Occasionally, management must actually manage! (laughter)

Allison, KPA Juelich: In a number of smaller projects (compared to CERN accelerators) there is no controls group but rather a modest instrumentation and controls team. Examples are Varennes and Lausanne and others. For reasons of pure workload (quantity and complexity), it is therefore essential that management have the process adequately detailed early on.

B.K.: Sure, but on the other hand, in a small team you have often the advantage of a very direct contact with the people who design the machine. In case the instrumentation/controls people have a fair interest in the process itself (op. one of my earlier remarks) then this organization has a definite advantage over the larger ones, where problems may be accentuated by segregation of activities and knowledge, hence more complicated human communications.

Crowley-Milling: I think there is a question of context. A small lab starting a new machine or a larger lab starting in a green field site may have this problem. But big labs who have a constant succession of accelerators being built, have an inbuilt knowledge, they have people who, whether officially or not, are working on the controls side of a project even before the project is approved. One should make sure that management give the maximum amount of attention to the control side, and that it thinks the controls are important in any context. But I do not think you can say definitely that they should start before the accelerator, or halfway through, it depends upon the context and the knowledge and upon what facilities the people have available.

B.K.: I see your point, but it is not in contradiction with the statement that one should preferably start early and with full support.

Crowley-Milling: I was trying to say that the stage of the design study at which the controls work should start is dependent upon the type of laboratory and its previous experience. If it is a lab starting in the accelerator field, that has had experience before, then the emphasis should be much more on getting the management to give a lot of manpower and effort to build up the expertise in controls right at the beginning.

B.K.: Two points we have not touched upon. First, some difficulties in the recent past have arisen from the fact that the pre-study was done by one group, whereas a different group was made responsible for building the system. The latter then completely redesigned it. Management should bring in early the people who are actually going to do it, because they are likely to stick to the liquid line and do not reinvent it. Second, this gains time badly needed by the controls group in order to define, together with the people of the process equipment, the rules of the game at the far front end. If the controls group miss this early moment, then the process groups take their own decisions, resulting in diversity which for large projects implies a gigantic waste of resources.

Even if we are early enough, but we do not look out, we can be hit by unexpected things. They can be technological and can pull the rug out under you, if you have missed them. May I invite my friend Peter Lucas to say a few words on this?

**Major new requirement trend/breakthrough**

Lucas, FNAL: Something I noticed, when I was being interviewed for my job at Fermilab, was that every physicist's desk had an open thermodynamics book on it. Everyone was trying to learn as much as they could about temperature and pressure, and radiation. This was the major new development on the physics front at the Tevatron. From the controls point of view, one of the most difficult new things was developing the closed loop controls of certain parameters in the satellite refrigerators. In the SSC, we are building a new accelerator and control system and it is likely to be more challenging of that type. In that presentation on the SSC, we heard on Monday, it was pointed out that all the controls hardware is going to sit in little holes bored into the ceiling of the tunnel. From the point of view of a network maybe the new thing in the SSC may be seismograph which may have to feed back into power supplies.

B.K.: Personally, the points you mention do not strike me as having a dramatic impact on the general controls options. The satellite refrigerators seem substandard as much affecting the central controls: the alignment monitoring possibly feeding back into the power supplies must be a very slow process, a small fraction of the total load on the network system. I wonder whether people can shoot in another example of a thing that may hit us. I wonder whether my friend Pier Giorgio Innocenti has not occasionally got the creepy feeling "an I sure that I haven't missed a major point?", or "is not that new development making my system completely ridiculous even before startup?"

Innocenti, CERN: The only thing I can do is touch wood!

B.K.: Any examples of developments that only appeared at an advanced stage of a controls project and have threatened to cause a major hangup?

Dohan, TRIUMF: Something bothers me about the TRIUMF koon factory. In a traditional complex like CERN, the accelerators are rather decoupled, they even have their own controls systems. At the TRIUMF koon factory, the high beam intensity will couple accelerators much more dramatically. When we do
bunch to bunch transfer between the different rings we suddenly beam load the
cavities of the target machine fully. The accelerator people always say that
our general control system (ethernet on top with console computers feeding in
and a bunch of frontends coming out). I wonder whether we have enough
bandwidth to do the necessary intra beam traffic. We almost need analog
bandwidth to correct things in another ring for things that happen earlier on.
Can our ethernet handle the load when we do not even know the physics of all
these resonances? Do not forget, we can tolerate no more than 0.1X beam loss,
which today is not even done at low current machines.

R.K.: From what I understand, your accelerator people (the ones you quote) may
be right. You should solve the problems where they arise and not import them
into another ring or into the centrals system (where they do not belong there).
Systems engineering becomes feasible only by defining subsystems, decoupling
them to a maximum and having precise conventions about their interaction. I do
not see an obvious bandwidth problem in your case, but I do see a truly
managing need for constancy and reliability. lest you splash the beam right
over the machine, which then you will be unable to service. This last one
may be the sort of thing which I was hinting at.

This brings us to the operational aspects. It is not really the next item on
our list, but the sequence had nothing magic. So why not have a statement of
my friend Roger Bailey here on operational aspects.

Operational Aspects

Bailey, CERN: I can offer in the form of a challenge a summary of some of the
things which have already been said this afternoon: on Tuesday I mentioned
that the the SPS we have something like a hundred many years of experience in
operating the accelerator and of course there is a similar wealth of experience at
places like Fermilab, there is even more at the PS and Brookhaven because they
have been around longer and then there is SLAC, DESY, LANS, etc. If I sit at Fermilab correcting the closed orbit. I am doing much
the same thing as I would be doing at CERN correcting the orbit.
But if I, as an SPS guy wrote a specification of what I would like the
orbit correction software to do, I would guess that the guys at Fermilab would
not be a hundred percent happy with it. There is an overlap, but of course
there are some differences. So the challenge which I would hold out is this:
can the community of operators and machine physicists on a world wide basis
get their act together long enough to really specify what they want
applications software to do? That is sort of a long question, but if you read
it in green, you get the gist of it.

R.K.: I might add: not only application software to do, but controls to do, an
estimate of data rates for instance.

Southeren, CERN: About the challenge to make specifications world wide, I will
ask another question. There is about one century that cars are being built. I
wonder why there are so many models you can buy and by what criteria you make
a choice? I believe that even for a simple problem like closed orbit
correction there will be several methods and each person likes particularly
what he has invented.

Bailey, CERN: The point is that in spite of the differences in the makes of
those cars, their controls have de facto standardised on a steering wheel one
or two pedals, a few knobs and meters. And it is this very fact and the
standardisation of traffic rules that have made the car such a success.

Phinney, SLAC: There are several accelerators at CERN and can the entire CERN
community agree on specification of what they want? (laughs...). If you fail
in one place, it may be hopeless otherwise.

R.K.: In the past we have often failed, but let us say, sociological reasons.
But the question is: can we in principle? We cannot ignore the sociological
aspect, but in our understanding of the problem we should be able to clearly
distinguish between engineering and sociology. I feel that presently there are
some hopeful signs....
The answer is yes but it depends from what level you talk. Certainly at the application level there are a lot of similarities between the Fermilab pbar source and what we do at CERN. There has been a lot of useful collaboration. I think on the collider operations as well. I know that people like Chris Saltmarsh and Lyn Evans have been there very often and I think there are a lot of similarities.

**R.K.:** New accelerators are predominantly new combinations of principles and problems, existing already in previous accelerators, and the relevant total expertise in the world today is broad and rich. It must therefore be possible to make an operational aspects study for a new accelerator which hits 80 to 90% right. One may allow flavours for different machines but one could start from an integrated view. The first for a new accelerator (a) this study is rarely done and (b) when it is being done, then collective experience is usually not drawn on.

This brings us to the contribution of my friend Ian Wilkie.

**Applications Structuring**

**Wilkie, CERN:** I fully expect to get shot down in flames for what I am about to say but hopefully I shall provoke a reaction. What we have been working on in the SPS has been the upgrading of the application software to make the SPS into an injector for LEP. The first thing we did, was completely divorcing ourselves from the details of the controls system hardware. We did not worry about the operating system we were going to use of any of the details of the bus standards and so on. We took a step back and we decided that the thing that was needed to do was finding out first of all how we, the operation group, wanted to view the system. What we wanted to do with it, what procedures we wanted to use, and so on. What we then built was the beginnings of a system that in essence is completely divorced from the SPS itself. It is a pulsing synchrotron control system, it is data driven, it is very generalised. Our part of the system works on data that is generated by model programs, the configuration of the state is variable and can be adapted to handle different sorts of machines. The supercycle generation guys tell us that their software in principle will work on any accelerator any synchrotron, and not just the SPS. They specifically took out SPS dependent features. But then, when I was asked to say something here, it occurred to me that really the requirements for an accelerator are not how you want to operate it, but the end user requirements. The requirements to deliver beam to the physics is with a particular intensity, a particular energy and particular types of beam, and it really doesn’t matter how you operate it as long as you do it. To a certain extent it is a matter of preference and history in the organisation that uses the accelerator so. In the SPS there is a history which we have adopted, the way that you want to view the machine, and at the SPS, I am sure, there is similar view and there certainly was in the ISR. So what I am suggesting is that in the not too distant future you might be able to conceive of a system which will operate any machine and what I have here is an advertisement for such a system (see figure). We propose a system that can control any pulsing synchrotron, version 1.0 later versions will control colliders. We used about mid-1983, say when LEP comes up. The question I want to ask is: does anybody think that such a thing is possible and if so would they buy it? I should say that the system should run on both VMS and UNIX and probably on OS/2 (laughter).

**R.K.:** Inscription is now open...

**Lister, EPFL:** I would say that you are lacking in ambition. I would ask what there is in your system that is so specific to an accelerator that you would not be able to seriously control a tokamak with it. Why cannot you make it generically capable of running something that is not generically conceived in the cycle fashion of an accelerator?

**Wilkie, CERN:** I guess I have not any experience of what it takes to run the tokamak, its possibly quite true that we could do that but before handing in my resignation at CERN I will have to do a bit of market research to find out what the potential is.

**TX, YL:** Does it also handle rapid cycling machines, say 50Hz?

**Wilkie, CERN:** In principle yes, but then implementation constraints may start becoming involved. Now I do not want specifically to make an advert for our SPS system. I am trying to push the idea that maybe in high level control software a lot of ideas are frequently reused without people realising it and then the wheel is reinvented many times over. I think our system would work on any reasonably slow cycle synchrotron and maybe it works on other machines.

**Phinney, SLAC:** In fact, rapid cycling machines have a beam concept that is not so logically different from a supercycle concept but the constraints are different. I mean if you would just extend your logic a little and try to make sure you have done it not too tied to the sort of segments of the supercycle, you might get quite a way.

**Wilkie, CERN:** Yes, implementation comes in very strongly but also we have to extend it to cope with the collider in the SPS. We thought about it a little and, if we made some changes in the way we think, maybe it could be extended to other things.

**Phinney, SLAC:** One other comment is that all colliders are not round.

**Wilkie, CERN:** Does anybody know what this acronym stands for? We first made the word and then thought of an acronym. We wanted a word that sounded like UNIX but wasn’t.

**R.K.:** I would like to ask Michael Crowley-Milling: if you were to make a new control system, would you go seriously into this direction?

**Crowley-Milling:** Yes, I think that one would very seriously consider it. Already we try to decompose at the hardware level, the intermediate level, the higher level. What Wilkie is proposing here is a universal or semi-universal magic area which you could plug the other things into. Of course one of the difficulties of general systems always is that if its really general then you can always design a specific system that is smaller, better, faster and cheaper. If you do not treat the design cost. Therefore, if you choose an existing generalised system, you have to show that it does give savings compared with doing an adhoc system.

**R.K.:** Like for these general commercial software packages, you must share the cost over a number of users and the accounts must be made over a number of years.

**Crowley-Milling:** It seems that at present there are more and more medium sized accelerators being used in the world and therefore this possibility might exist as long as each individual one only has to bear a fraction of the total cost. But the big accelerators are on a longer time scale generally, so that new ideas may come in between one accelerator and another. LEP is being built with one technology, one set of ideas on software. SSC may be full of expert systems and so on. That is a multi-year jump and I do not think that one
generalised system could survive sufficiently. But if we are going to have many more medium sized accelerators, nowadays that is about the size of the PS, it may well be worthwhile looking forward to a collaboration on a generalised software kernel to which the individual networks and lower levels of hardware and so on are connected according to the actual needs of the lab concerned.

Boutheon, CERN: Knowing the evolving nature of our machines, their developments and modifications, I must ask whether Wilkie’s private firm intends to provide updated versions and if yes for which price, and do they propose on-site maintenance?

Wilkie, CERN: Of course you would update such a system and an idea that occurred to me just now is that maybe the major updates are made when a major new accelerator with different constraints and different ideas comes into being. Thus the benefits of the understanding, gained when you do that, can be passed back to existing subscribers. So perhaps each new lab, each new accelerator pays for the upgrade and everybody else benefits. As to on-site maintenance well, our software has no bugs, so it does not need any maintenance. [laughter...]

P.K.: I would like to ask my friend Pier Giorgio Innocenti: if you were in the happy situation of coming in early and having a free field and taking your decisions right from the beginning, when the accelerator is not too far advanced, would you dare to go into this direction?

Innocenti, CERN: For sure, no doubt, but I would not buy the source code, in order to avoid the temptation of creativity. [laughter...]

P.K.: Somewhat on the same lines of cutting large systems into layers, and trying to structure them, there is the topic of protocols and I would like to ask my friend Guy Barbaud to tell us what he thinks all that means.

Operational and Control Protocols

Barbaud, CERN: In a typical controls layout, we have the applications in the middle, we have the consoles and operators on one side and the devices which are part of the machine on the other side. Every device is either acting on the beam, such as power supplies, or getting data from the beam, such as pickups. What we aim for, when establishing some control protocol, is to define the connections between the applications and the devices. So the objective of a control protocol is to describe on both ends of this connection how the device can be seen. This should not be confused with communications protocols e.g. in the sense of ISO/OSI. Controls protocols are situated above the OSI layer 7 and they should be able to make use of different communications protocols. One question which is often raised: could and should we look for only one protocol for all the devices? In accelerators we have many types of devices. Or, a somewhat different approach, should we look for some methodology to build up and upgrade consistent protocols. Does the floor want to comment?

P.K.: What is your own opinion?

Barbaud, CERN: My opinion is that we should look for a methodology which allows to build consistent protocols. I think that building only one may prove to be rather difficult and inefficient. If you compare a power supply and a semigrid, it seems difficult to make them look alike. Possibly they will be different but they should be consistent.

P.K.: Do you mean that if you cannot standardise the thing, you may at least standardise the ways to get to it?

Barbaud, CERN: I think what we should standardise at least is the functionality of the devices, rather than the technology involved.

P.K.: Does anybody think we could get a protocol which, say for a number of years, is the operational protocol for orbit measurements, and applicable to a number of accelerators? Who makes a bid? Hello there, are you all sleeping?

Sicard, CERN: In my opinion it is not very easy to do that. First, for sociological reasons. One recent example at CERN: the orbit measurements of LEP and of EPA have been done in completely different ways. People never thought about discussing these two protocols together. Second, technically you may be able to define what a beam position monitor is but you have trouble defining exactly the conditions in which you want to read them and these are usually dependent on the machines themselves. But yes, it is probably possible by investing considerable effort.

P.K.: There is another expression for the word “problem”: some choose to call it a “challenge”.

Rebay, CERN: I think this problem of instrumentation is something that could be common to quite a number of accelerators. I worked on instrumentation in the Booster previously and now I am involved in the LEP instrumentation. I do not see any reason why the two things could not be defined with common primitives and so my answer is YES.

P.K.: If I ask somebody whether you could possibly make power supplies look alike, I probably get a very good statistics of yes. However, if I move to orbit, people are scratching their heads, so it is a gradual scale. You probably cannot say flat no to the orbit. One must look at the problem more closely but with an open mind.

Potier, CERN: I can say YES and it is an experience, it is a reality: we use the same closed orbit program for both CPS and EPA, though there are slight differences because in the EPA we have no correctors, but the correction protocol is the same. So we try to go further.

P.K.: So they are similar, are they? They are not identical but they are similar?

Potier, CERN: Yes, they are similar. They differ only in the interactive program since that has to show different machines. But that is mainly due to a primitive way of coding the program. The modelling programs which are used are the same.

P.K.: I think the following question should always be asked: if things are so similar, are the remaining differences really significant? Or can they, with a little bit of goodwill, come closer together?

Phinney, SLC: I simply cannot see why the upper layers of that would not work for anybody’s system. It seems to me that you just have to plug in the different data base and data acquisition.

Chohan, CERN: I think the answer is YES and the problem that Sicard is mentioning is a sociological problem, if it is a problem at all. Maybe Peter Lucas can tell us about the Fermilab Beam Profile system which is the same through all the machines, including the accumulator ring.
Lucas FNL: I could just answer that we manage to make BPM systems that are really different down in the machine to some extent look similar when the operator interface level. I consider that somewhat of a success actually.

Phinnex SLAC: If you look at our machine it is really like a collection of tiny little machines, like the PS complex. We were not successful in convincing the engineers to make BPMs in one area of the machine to look like the BPMs in the other area. But for the person on the console the software presents them all identical. The same goes for the power supplies and for everything else.

R.K.: If you would present a really convincing scheme early on, then you may have more of a chance than if you come with this idea when the hardware people are halfway through their developments and would not change it anymore.

Crowley-Walling: The whole idea of the data module approach was to do precisely this. You adapted this module to suit different sorts of hardware, but the interface into that module from the rest of the system was consistent for a power supply, whatever power supply, for a BPM whatever sort. If that approach is adopted, then there is no problem at all. The difficulty is that due to the usual not invented here syndrome, what was called data modules in one place is called equipment modules in another and now has to be called something else for LEP, only because it is a different group of people who is concerned. It is a question of management. If there is a perfectly satisfactory way of doing it then at least the protocol should be the same, the names should be the same and then you have no difficulty in shifting from one machine to another or even from Lab. to Lab.

R.K.: These notions are indeed not new. Since we are on it: what would you choose to call a data module in your system, would earlier have been called a handler or driver in other places. But all that is a bit besides the point. We would like today to go a bit beyond that level. Can we normalise the messages between the power supplies and the control system so that we may simply plug in another supply and keep using the same software?

Steiner GSI: It is funny that we could agree on unification of the interface and protocol for power supplies. Five watts stepper, as well as hundred milliperm second bump kickers power supplies, etc., but that we cannot agree on beam monitors. I think the main reason for this discrepancy is that we buy power supplies from industry and we make beam monitors ourselves.

R.K.: There is a lot of truth in that. There is less ego attachment to power supplies, hence less tendency to declare them "special". Now let us move on to a related next issue: Autoconfiguration. Maybe my friend Henning Brandsis tells us about developments at GSI.

Autoconfiguration

Brandsis GSI: Let us assume that each device has its own microprocessor in it. That is not so futuristic. Then we can build in a very easily a self-identifying mechanism so that this device can tell its name and type on request. We can then connect it to any outlet and the computer servicing that outlet finds out which device is connected to it. By this way the date path to that device within the network is built up automatically. In addition the servicing computer uses the type of the device at that outlet and configures to this device type automatically. We apply the same technique on the console side, with console elements which identify themselves and we could use that to give the application software the information about what the console contains in the actual stage. Well, this last point may be of less interest in future when we go to workstations to control the systems, but let us take on the first steps. In this control system we tried to circumvent the problem of maintaining the data base, describing the data paths to the devices by hand. So we built in an autoconfiguration scheme which is not so ideal as I said before, because we have devices without microprocessors and so we could build in the automatic construction of data path but we could not build in such free way of cabling. We can, to a certain extent, connect the cable to different outlets but not so completely and we had not this automatic configuration of the device software. My question is now: I had no reactions to my post on that theme and so either this problem of finding out where the real device is has been solved already or it is no problem at all. What does the audience think?

Rebany CERN: My comment on this autoconfiguration is that we must be very careful. I think that there are two ways of attacking the problem: one is from the top, the other is from the bottom. Today we should attack this kind of problem from the top, i.e. starting from the data base. When you decide to install something, you should first put the info in the DB. Thereafter, that information can then be extracted for all purposes including control. This is a much better way that introducing it from the bottom. I understand that in some cases during the operation you have things which can change from one configuration to the other and then the autoconfiguration can bring the solution to the problem.

Brandis GSI: The database has the description of the device. But if you have no identification of the device itself, you have to describe the way to that device correctly and you have to do your cabling correctly. If you can use the identification of the device, then you can route your messages to the place where that device is at the moment. If the device has been plugged into a wrong outlet, the system will find out the way to that device without the necessity to change the data base describing the way to it. That is what I mean by autoconfiguration. So the data base has only the description of the devices without the path to them.

R.K.: I think the data base should know by interrogation mechanism what is connected to which outlet and update itself accordingly. So I agree with the essence of Rebany’s remark. But it is not incompatible with the principle of autoconfiguration.

Perilliat CERN: I have two comments on this subject. The first one is about the hardware. Twenty years ago, there was something like that included in the CAMAC specification, this autoconfiguration information, but I have never seen any CAMAC module providing this information, neither any software using or somebody thinking to use these facilities. The second comment is that perhaps this is a very expensive scheme to set up. DEC has something like that for distribution of new versions of their operating system. They spent hundred or several hundreds of manyears to provide something like that. But then they have to send around the world something like a thousand configurations per day. I am not so sure that the accelerator community has to set up a hundred or thousand accelerators per day to justify such an investment.

Ziegel FNL: All CAMAC modules designed at Fermilab do follow a numbering scheme. For this reason it has been a very powerful diagnostic tool for us as far as CAMAC crates and links are concerned. We know each module we expect to see in certain slots. Then there is a diagnostic program which will give the full description of what is in the crate. In this context I may add a question: everyone seems reluctant to speak about Multibus II which has been called AT&T in an identity to every module. If we really think this is such a great idea, then how come that people aren’t really flocking towards Multibus II?
B.K.: The man who would have answered this question (Klaus Mueller), seems to have left this meeting a few days too early.

Steiner, GSI: It should be emphasised that this autoconfiguration is not just an academic fancy. We have built a control system for a synchrotron with a cycle time of 0.2 seconds, and we had to apply this same control system to a Storage Linac. The set values are refreshed each pulse, pulse-to-pulse modulation (PPM) in CERN jargon. We typically have of the order of seven to ten pulsed power supplies hanging on one equipment controller. When we got into bandwidth trouble we redesigned the configuration. We then just plugged the power supplies into the outlets again, made a restart and things worked, without editing without going to a terminal and generating a new configuration table.

Zagl, FNAL: Last week, the magnet test facility guys at Fermilab told me that they have no way of identifying their DVMs. If somebody plugs in a different DVM, they can lose track of the particular magnets. So people are looking for the identification numbers.

Rausch, CERN: Autoconfiguration is also relevant to the transmission. On the LEP token ring, when you insert or remove a station, a workstation, the whole management of the network itself is automatic, so that is autoconfiguration. Also, on the low level, on the equipment network, we have foreseen a polling mechanism which detects when you insert or remove an element. Finally, at equipment level, if something goes wrong, say for some vital systems, you may need to have a backup system. On the MIL 1553 there is a facility of automatic switching to a spare network, which again is a form of autoconfiguration.

Rabany, CERN: I agree that the possibility of identifying a module or the configuration is an advantage but this does not contradict my earlier claim that the introduction of the data base should be done at the data base level FIRST. If you do so, then you can check whether the configuration is correct, using this information as a reference. This is also for should be the natural sequence of working. You do not compile with a suitcase full of hardware and just plug them in at random. Supposedly you make certain considerations before placing your modules. In the data base you then enter all relevant information. The same holds for cabling. Before ordering, you install it in terms of software at the data base level, then you can extract very nice lists.

Similar arguments hold for maintenance. You may say: just plug into the other outlet, or whatever it is, and it works again. Yes, but if all people are doing that without updating the data base, then we are heading for anarchy. So my plea is: FIRST describe what you want to do in the data base. If one does so consistently, then, given a good communication medium, any person designated to maintain or to make an intervention can know the correct configuration.

Phanup, SLAC: I am entirely in agreement with Rabany. We do have serial numbers, we use them to keep track of modules that could be problems of identifying measurements that have been made on the module. We do not use them to identify the devices and I would much rather see it solidly described in the data base, not flexible, so you really know where you are.

Innocenti, CERN: I think that we are speaking of complementarity between the two treatments, bottom up and top down. They may be complementary but they are of very different power. DB management systems offer powerful facilities. So, not only can you use the DB information for purposes of checking the modules' identity, but you may use that information to associate your software or the geometry of your machine or whatever. So I think that FIRST entering the information in the data base is much more powerful and it should be enforced.

B.K.: I am convinced the data base should be the starting point and that it must be fully updated at all times. But I am still wondering whether we should fill in the DB off line and use it only as a reference for checking the identity and configuration, or that we should allow actual autoconfiguration to happen, but then use that very mechanism updates the DB automatically. That would also guarantee its correctness.

I see that we have already slid into our next topic: data bases. Maybe we will listen to my friend Dave Gurd, who prepared a statement.

Data Bases

Gurd, TRIUMF: Much of what I wanted to say has perhaps already been said during the previous discussion. The main point that I wanted to make is that a correct data base design is absolutely fundamental to the design of any control system and I would make the point this strongly that it is not possible to design a correct control system unless you start from a correctly designed and complete data base. This is perhaps even more true because the increasingly complex accelerators imply larger and perhaps increasingly complex data bases. We may have to be very wide ranging in our understanding of what should be included in the data base. What we have been talking about here is one end of the spectrum, the static descriptive data base, which explains where the equipment is and what it does and hopefully it includes some information about the physics of the equipment and the physics of the accelerator. But the data bases range from there, through dynamic data bases, which keep track of the current state of the machine, and on with a long list, to the manuals, instruction manual, drawings and perhaps even a stores catalogue. One important issue is then: to what extent should all of these varying kinds of data bases be integrated (not centralised) under one system. Another issue is the importance of separating the roles of the data base management system, from the data base which is actually accessed in real time. For the audience, I selected two questions. The first one: do people here agree with the very strong feeling that I have about the fundamental importance of a correctly designed data base in order to make a control system work? The second question: is it not possible, given that I believe the contents of the data bases for most of our accelerators is very similar, that a collaboration could result in some standard data base structure that could be used between control systems?

Plant, LALN: A good rule may be: expect the unexpected. One unexpected thing in our world is the use of permanent magnets, quadrupoles, etc. and how to adjust their effect on the beam. The static data base would usually contain the locations of magnets along the beamline, which you would need for modelling codes. I know a machine where the effect of the quadrupole is adjusted by moving it backwards and forwards along the beamline, so that now becomes your set point for that magnet. Expect the unexpected.

Gurd, TRIUMF: Also at TRIUMF we have quadrupoles which slide up and down the beam line, but I see no reason why one cannot design a structure that describes that kind of device in the data base just like anything else.

B.K.: Nor does it strike me as too fundamental a thing.

Rabany, CERN: On the second point of Gurd, I would say that I am prepared for a collaboration in the direction you mention.
B.K.: I suppose you do not really oppose a duty trip to Vancouver! (laughter)

Corbis, CERN: I certainly think that all static data should be somewhere on a relational data base, preferably on a main frame. But in a real control system you need the data about everywhere and you cannot address a mainframe from a routine. Somebody agrees that you must load these data in whatever form is necessary for the process. Now the other part is the dynamic data. While you are operating the system, they can exist in whatever register they want. If you want to save them, you must put them into archives, otherwise they may be lost at the end of the run. The question is: how do you do this? Interleaved? I think that the static data base should at least know where the dynamic information is. So that means an index to the archives, as a minimum. As to the moving quadrupoles: their data are neither static nor dynamic, or both. If you change them during the operation, the problem is: how do you get them into the static data base?

Perrillat, CERN: During this conference, I have seen two ways to attack this problem, which may or may not be entirely divergent. The first one is the data base approach, in which we have to concentrate a priori on the data structure. The other one is the object oriented approach, which is aiming at hiding completely the data structure. What is your feeling about these two approaches? Can these converge later or will they completely diverge?

Welland, RPM: At Brookhaven we consider the data base as the corner stone of the software part of the control system. As to Perrillat's question: emphasising object oriented programming with a high level introduction into the data base, allows one to separate where one is getting one's data from, depending upon object. In essence that is one of the things we are developing. So we have a choice, for instance, a magnet position, for another type of magnet the data may be in the static data base, accessible by query, for another type of magnet the system would read the set point of a device. To the high level software that should be completely transparent.

Schaller, LAWL: After a lot of looking at the high level controls, we consider extending the idea of a data base to a knowledge base. I mean the object oriented capabilities of some knowledge bases in the context of an expert system. This would allow capturing information not only about the individual devices but also the relationships to other devices. If we have a knowledge based system, as a foundation, then there is also a chance of eventually capturing knowledge about what operators know about how devices interrelate. So we now consider starting with that sort of a basis. It may allow to slip in some machine reasoning underneath everything else, without having to adapt the top.

Corbis, CERN: A question to answer the one of Perrillat. Where should the static data be, in the static data base or in the objects. I think we can have both, you should construct your objects dynamically from the data in the static data base.

B.K.: The data base should know the objects and the static data in them.

Corbis, CERN: Off line you should know what is in the objects, once it is installed in real time you should not know any more what is in it.

B.K.: We have been talking a long time about all kind of principles, systems and gadgets. While we are inventing and designing, we are creating problems for the people who will run and maintain all that. The experience of these people may help us make better controls systems. Let us therefore listen to my friend Gilbert Daems.

Exploitation

Daems, CERN: I think we all agree more or less about the future layout of control systems. There will probably be a certain number of workstations on a local area network, there will be one or more data bases. A certain number of workstations or minicomputers will be on the network and those crates are connected to process equipment, fully equipped with local intelligence. So what is then the maintenance problem? If we assume, for the sake of the argument, that the system is delivered as one integrated whole, hardware and systems software, say by Hitachi, then hardware maintenance will be a small problem. So then we have to concentrate on software maintenance. Presumably, the major items would then be the data base, a moderate number of some detailed operational and control protocols to interface all this process equipment. That process equipment is of course assumed to be under the responsibility of the process specialist. I also assume that there is an application program framework, in which some environment the applications are developed. So I think that in the future the maintenance problem will essentially concentrate on the applications software. The latter would mainly be developed in house by the users (Operations Group and accelerator physicists) and by the control group. Therefore, my question to the audience is: how to make in house produced software reliable, in other words what about the software quality? How can we implement it, what is the cost of it and could we really do it?

Dehan, TRUMP: Part of the quality problem comes from what some call the key scout style of programming. What often happens is that an application writer goes to his neighbour who knows writing some similar code. He then adapts that code and puts in its environment. Could not this thing be put properly into the data base? I am intrigued by the notion of putting some things that are so called predefined programs there. I want to write some kind of language for writing an application. I want to use a process which knows about the interrelation with the physics of the beam and/or with the time measurement. Would it not be cute to retrieve this from data base rather than leaving it up to the individual application programmer, who is an experimental programmer?

Kaye, CERN: May I ask for a clarification of what you mean by reliable software in this context, is it since it doesn't meet specifications?

Daems, CERN: There seem to be three components to software quality. First of all it must do what it must do (specified functionality) and that has the highest priority. Second, there are qualities like execution speed, error handling, robustness. We may call these the intrinsical quality. Third, there is conformance with the existing standards, existence and quality of documentation. The first category (functionality) is probably relatively straight forward to test, by exercising through the all specified functions. The third category (standards) may be checked by inspection. To some extent this part could be forced by a program development environment and installation tools. The second category (intrinsic) is the most difficult to test, partly since it is difficult to define. Execution speed depends on load. Error handling used to be a stepchild, hence a constant source of worry and confusion. Robustness, the resilience against rough environment, is a general hence vague notion. Some programs suddenly hang up after having run successfully many times. Now, how are you doing for these categories? I think that for obvious reasons the first component usually scores best: the programs mostly end up by doing what they should. But, thereafter, mostly programmers do not have or are not allowed enough time, and they are under pressure when installing the software. Thus probably not all tests which could be done are actually done. Sometimes they cannot be done since there is no test environment appropriate for the test. It must then be done on-line, with the known limitations when there is
beam. All this goes at the expense of the intrinsic quality. Standards and documentation also suffer from lack of time.

B.K.: Who else? Come on, is everybody here only building things? Nobody is making the things run! That would be one explanation for the sorry state of affairs (laughter...). What do people think about software reliability? I think that even in the limit that we buy Hitachi, there might be some problems left in hardware. What about diagnostics? Do you not have the problem of localising things, even if you cannot pinpoint them? If you want to localise faults, hardware or software, then you need diagnostics at the boundaries between the modules of your system. There must be questions of diagnostics at interfaces between modules, but also at interfaces between human parties.

Danesi, CERN: On the layout I postulated, which you seem to agree with since you did not protest, the diagnostics is not too hard to implement. I think. Assuming it is really properly modular and interfaced with standard protocols, then you can easily embed test features at the interfaces, in the hardware and software. However, I am perplexed that nobody has reactions on software quality. If I speak with people outside this room, they all seem to agree that there is a problem about quality tests. Maybe it is that we don’t know how to implement them or they cost too much.

Casaleno, CERN: The industry is at present making a very very big effort to supply us with tools to make tests and I think in the near future we will be able to buy these tools for a good price. They are putting a lot of money in it, because they have discovered that seventy percent of resources are spent there.

Danaels, CERN: Maybe we could ask the question differently: are we prepared to pay for software quality? I recently read an article about the Safeguard system which was developed by the Bell laboratories for the DoD. Superficially, as seen, their productivity is several times lower than ours. Now I do not think they are worse programmers than we are, I just think they spend more effort in providing good software. So a software project of 125 man-years in our context would take 500 man-years when done in the Safeguard style. I am sure it will be of better quality, we are prepared to pay that price? When you want to drive a Rolls Royce you have to pay for that. If you drive a dext chaux you play less but you get something else.

Routhen, CERN: As to Keyser’s question: we are not saying that the programmers make bad software or buggy software. The problem is that each program, tested alone, may be perfectly working, but that there are some spurious faults that may block the execution in certain contexts. The context is never the same, even when we think that we have reproducible processes. Very often what we call “software is not enough reliable” can be summarised by the fact that we run some programs one day and the day after the same program does not run or hangs up. How to improve it, I have no idea. And I wonder if better development tools will really improve this aspect.

Kuhn, CERN: This year, a team around Lyn Evans and Roger Baily have made a valiant effort for getting good software going. But then we had a miserable start of the SPS, for reasons unrelated to their software. They then never got a fighting chance of commissioning their software. Management was scared stiff of having the physicists all over their office, complaining about lost minutes. Often people get discouraged from making a fuller automation because they may get no chance of commissioning their product. I agree with Daems: software has its flaws partially because quite often no sufficient time is made available for testing it properly on the machine.

B.K.: Time is pressing, so we shall move on to the next point. We shall skip the automation item of our list. We have treated that in some detail in our workshop this morning. We go straight to modelling and expert systems and my friend Dennis Weygand has the floor.

Weygand, DNL: Anyone who has listened to me this morning knows already what I am going to say. I am pretty sure that the next generation of accelerator control systems will have embedded expert systems. In the environment of more powerful CPUs, added resources of distributed parallel systems, together with enhanced AI environments, a model based expert system will take a more central role than real time device control. Also sub tasks such as alarm filtering and fault diagnosis require some intelligence, but such things are already well established and will certainly take their role in future control systems developments. The question which I put to the audience relates back to software reliability: how may artificial intelligence and expert systems aid in the design and development of reliable control systems?

Danesi, CERN: This morning we saw that making an expert system is very, very manpower consuming. Are we prepared for diagnosis and maintenance to pay a price of ten or twenty man-years if not for a product of which we don not know teh benefits?

B.K.: As to potential benefits, they seem pretty obvious, the question is really at which price. Like for any new product or procedure, the gain does not come immediately and it is unclear early on. It needs hard work of pioneers! One must bootstrap. A pilot project should help defining a next one with better cost/benefit ratio, and so on. As Axel Danaels put it: are we willing to pay the price?

Weygand, DNL: One possible line: since generally expert systems are implemented at a relatively high level, the possibility of collaboration is rather good. So you might distribute that development and thus write off the cost over several institutions, particularly for fault diagnosis and alarm filtering.

Roffel, CERN: I do not have any experience with collaboration for expert systems but I have a lot of experience for collaboration around modelling facilities and certainly it is a field where collaboration is very easy across various laboratories. For the time being one way is also to go ahead, start from this one, and expand the field to the other facilities.

Weygand, DNL: Immediately one recognises that a very early expert system eventually grew up to be an expert system shell, like MICYN grew up into KEE. So already there is collaboration and maybe the rules are not so specific. Certainly in shells and in inference engines as such there is already significant collaboration.

Xue, INP, Beijing: I think an expert system is a sort of linear system, so maintenance cost will decrease with time while the IS itself will learn to maintain itself.

B.K.: To start with, the expert system will cost a lot of effort. But that should not discourage us.
Building the Next Large Accelerator Control System

ATTEMPTED SYNTHESIS

Controls early with full support

In spite of some favourable examples recently, controls are often an afterthought. Yet, starting early is essential for three reasons. First, for budgeting. Second, for reaching a balanced system's design. Third, for negotiating the boundary conditions at the interface between process equipment and controls system properly.

Starting early is not easy because of the volatility of early machine designs and the reluctance of machine designers to spend time on controls while their ideas on the machine have not yet settled. If pressed, they may either grossly overstate their needs or underestimate them. Getting clear cut controls specifications out of machine designers may be an illusion anyway, even when the machine design is finally emerging. Some accelerator knowledge in the controls team is therefore essential. Only with such an overlap of expertise can one reach the balanced trade-offs which are the basis of a good system's design. Therefore, using reasonable assumptions, one may present machine designers with written proposals, which then will trigger them to respond.

A number of generic aspects, not directly accelerator or controls hardware dependent, can and should be considered early. e.g., data base structures, architectural considerations, programming environment. However, freezing decisions on these should be postponed to the appropriate dates. First, since these decisions should be made in an overall context, after trade-off considerations with all major aspects. Second, since the market for commercial products evolves rapidly. A definite calendar of latest allowed decision dates should therefore be composed and kept updated.

Cost estimation early is essential since controls for modern machines may amount to 10 to 20% of the project budget. The generic considerations (cp. above) define the general infrastructure, not too strongly dependent on the number of parameters or control channels. This may very roughly concern half the controls budget. The half is far from trivial, and so one cannot advert to a number of control channels, with the cost of controls for existing accelerators are important.

Beginning early is more important for small labs who start from scratch and larger labs starting in a green pasture. Labs with an ongoing success of machine projects (hence, also half of the control projects) are helped by existing experience in parameter and other resources. Even so, starting early helps negotiating agreements on the interface between process equipment and controls system.

Failing to do so early on may result in irrelevant diversity hence - in large projects - enormous waste of resources. Due to increasing use of local intelligence, specific process electronics presently consume as many resources as the control system proper.

Finally, the team designing controls should know that they also must make it true. Changing the team between design and implementation may result in re-marketing the controls completely. Controls responsibility must therefore be defined early on and management should enforce adherence to the negotiated agreements.

Major new requirements/trends/breakthrough

Certain novelties can have a hitherto unexperienced impact on the controls problems. Examples may be workstations and (possibly) the extreme reliability required at high intensity hadron facilities. An early assessment of these is mandatory since potentially they may influence the design approach, even back to the actual machine. As a general rule, problems should be dealt with close to where they arise. Often their impact may thus be contained in a subsystem, without influencing the overall design. In general, though not always, this means as close to the front end as possible. In no case should problems be imported into other places without overriding reasons.

Operational aspects

The main users of the control system are operations and machine physicists studying the beams. Their experience and ideas should then be brought into the design early. This can be explicitly, by collecting their desiderata in a formal specification, or implicitly, by squarely joining the design team. Difficulties in formulating desiderata may be comparable to those mentioned earlier; volatility of machine design and reluctance to think controls. Yet accelerators, in particular their controls, are predominantly new combinations of principles and solutions in existing machines. Many controls issues are only weakly or not dependent on the machine or controls hardware. The relevant experience integrated over the world is rich and may be tapped. Formulation of these desiderata is therefore possible and meaningful. It may be interesting to abstract those aspects whose validity extends to a number of machines in different labs. For example, could a closed orbit display be composed of identical primitives for different machines at different labs?

Applications structures

Applications software is the last link in the controls system. But without applications there is no control. It is like a concert piano without music. Applications are usually late and not always of the quality wished for. In part this is due to the difficulties of early controls definition, in part to a chronic understaffing of this activity. In part also since it is done in an individualistic way by casual programmers. This situation is typical for most laboratories. The suffering parties are operations and machine experimenters and, as a consequence, controls people themselves.

The early definition issue is dealt with in the preceding sections.

There is a growing awareness that in applications one is solving the same or similar problems over and over again. In this awareness, at most places a rationalisation is achieved by providing the applications developers with libraries, common components, reusable runtime code. These are both at better productivity and better quality. They also alleviate exploitation, by reducing diversity. Although it is agreed that this helps, the total impact is insufficient at most laboratories.

In view of the persistent applications software malaise, inter laboratory collaboration is again mentioned. If the problems are so similar, must they really be different, or could one agree on common frames, which can be tailored to local favours? In controls jargon: can one find a complete set of common primitives from which most accelerator applications software can be made at different machines? One is probably helped by the fact that most applications software is structured in layers. The higher levels are not very
machine dependent or controls system dependent. Several laboratories use their own unique higher level applications software structure for different machines in their accelerators complex. Going one step further, there is one example of an applications package (LAN) that has successfully been ported to several accelerators at different laboratories.

One may then speculate about the possibility of a private firm or an inter-laboratory consortium as the donor and thereafter mail and enhancing the package used by a number of laboratories for different accelerators. The system may be applicable to machines other than accelerators, say tokamaks, etc. Although an interesting notion, this raises a number of questions. In the first place there is the issue of return on the investment. The degree of generality pertains a high investment, whereas the total number of licences will be limited at first. Then there is the question whether one can long stay abreast of developments. Accelerators evolve and the market of electronics and computers even more.

Operational & control protocols

Process equipment (devices) of one and the same category, e.g. power supplies, are often different in implementation and thereby present a diversity of faces to the controls system, irrelevant for operations. Standardisation of that interface works for a uniform interaction by the operator, reduces the diversity of software, alleviates trouble shooting and eases replacement of devices and reconfiguration. This standardisation goes by protocols.

By protocol of a device, in this context, we mean a formalised agreement as to how relevant properties of the device are to be changed by the operator and by the controls system. The protocol must be based on an agreed model of the device. Close to the device the protocol takes the form of an agreed message format. Somewhat higher up it presents a software interface towards the applications. At the console it may present a formalised way of display & control & interaction. Operational & control protocols should not be confused with the seven layers of communications protocol in the sense of ISO/OSI, although part of the latter may govern the actual transmission to and from the device. The control protocol is situated above layer 7. One and the same protocol message can in principle use different communications protocols for its transmission between controls system and device.

The utility of these approaches has a general, albeit somewhat hand waving, agreement. In most laboratories a number of conventions have been adopted at the device level, at least for certain series of identical power supplies. But that is of course the easy part. Not very much of the sort exists for beam instrumentation. At the console level some uniformity in presentation and interaction has been achieved at some laboratories, in spite of diversity at lower levels. The question is how far can one go and how far should one go? Can power supplies (or at least 90 % of them) at one laboratory all look the same to the applications software? The stakes in terms of software economics and flexibility are obvious. Could this go beyond one laboratory? Can closed orbit displays on two machines look alike?

Introducing standardised controls & operational protocols in an existing system would mean a major upheaval and should usually be avoided. At best, one may try to gradually converge by introducing them for new additions and major retro-fits. One must then accept cohabitation of old and new for a number of years. For new projects, however, standardisation of controls & operations protocols is one of the most fundamental and paying things one can do. It must happen early on, before the devices people are too far into their local implementations.

Autoconfiguration

A further step in the direction of standardising device interfaces is autoconfiguration. Imagine, in extremis, a controls system in which all components in the devices are made through one common type of universal outlet and corresponding connector on the device cable. Given some intelligence in the device, which is the present trend anyway, the controls system may then interrogate the device and configure its software accordingly. The scheme presupposes a strict categorisation of devices and that the corresponding software modules and parameter sets are readily available on the data base. It also presupposes a strict protocolisation at the interfaces (cp. previous section).

The advantages of such a scheme seem obvious. Autocconfiguration is already rendering convincing services in a number of other contexts. At least one laboratory (GSI) has implemented autocconfiguration systematically for controlling a major accelerator complex, albeit at a somewhat more modest level than sketched above. Besides its attraction, the scheme raises some fundamental issues. Some hold that the data base must be the source of everything and that autoconfiguration may result in losing track of the wanted situation. They feel that every modification should first be inscribed in the data base after due reflection, whereby these data may then be used to any purpose including control. On the other hand, it is conceivable that the autoconfiguration mechanism itself first updates the data base. It is also conceivable that the data base contain the wanted, "reflected" situation alongside the actual situation, displaying the differences, with possible warning signals.

Finally there is the economical question: how much benefit brings the scheme for how much investment? The GSI example may indicate that the price need not be excessive.

Data bases

It is generally agreed that a correct data base design is absolutely fundamental to the design of any control system. This point becomes more important because of the increasing complexity of new installations but also because of the increasing involvement in all aspects of the machine. Data bases spring up in a natural way for all kinds of applications and the question arises in how far these should be integrated (not centralised). Distinction must be made between the static general data base management system and the dynamic data base which keeps track of the actual state of the machine and is accessed in real time.

There is seeming contradiction between the spirits of data bases and of object oriented design. The first exposing the data structures, the other hiding them. One may orient the design of the data base. With an object oriented design and a high level entry into the data base, one may choose the source of the data according to the object in question. For example, for one object the data may be in the data base, for another object the system would read the set point of a device. Extending the data base to a knowledge base with object oriented capabilities, may eventually allow to capture also relationships between devices.

Again, even when using a commercial DBMS, the investments in design of these data bases are huge. But given the similar nature of the needs, accelerator control data bases seem a topic par excellence for inter laboratory collaboration.
**Controls Exploitation.**

While we design and implement new facilities, we are creating further work that must be done in running, trouble shooting and improving those new facilities. For good reasons in many places the exploitation is mainly done by dedicated teams. The effort, integrated over the life cycle, deployed by these teams in dealing with insufficiencies in controls hardware and software, is enormous. It is therefore desirable to reflect on how to improve the quality of the facilities and for that purpose, it may be worth while to use the exploitation team's experience.

Supposing that the controls system fulfills the functional requirements, then exploitation is composed of three aspects. First there is the actual running, comprising starting up, shutting down, backups, preventive maintenance, etc. It is the only aspect if the system is fully reliable. Even so, running can be more easy or more cumbersome, depending on available utilities and intrinsic qualities of the system. The other two aspects are trouble shooting, on the one hand, and repair — preferably prevention — on the other hand. Trouble shooting starts from the clue of what happened or what did not, using a strategy and diagnostic tools. Since the main strategy is localisation, diagnostics are most useful at the interfaces (hardware and software) between subsystems. This is particularly true at interfaces joining responsibilities of different parties. By this strategy, trouble shooting benefits from, indeed requires a well structured control system, in which irrelevant diversity is kept to a minimum by standardisation. A well reflected error message system can be of great help, but is usually a source of confusion since a stepchild. Trouble shooting experience should therefore be fed back into future designs.

A special place is taken by spurious bugs, mostly in applications software: a program which worked one day, may not do so or hang up a next time. These consume disproportionate effort, since difficult to find and hence recurring. They are related to the intrinsic program quality. This quality is related to three aspects. First, many applications are developed by non professional programmers. Second, for that public, programming tools and frameworks are necessary, but are often lacking. Third, adequate time for testing and debugging is not always allocated: often the pressure for having the next facility delivered is greater than for having the previous one brought up to reliability.

Software quality can be improved. For space probes, for nuclear energy reactors, etc., the usual accelerator software quality would be completely inadequate. For these applications, sophisticated tools and deliberate procedures are used to ensure a specified level of quality. This requires a multiple of the resources which are usually invested at accelerators. The question is: are we willing to pay that price? One point of hope: industry is deploying enormous resources on programming environments and affordable packages may hit the market in the coming few years.

**Automation**

For this topic we refer to the attempted synthesis of the dedicated workshop.

---

**Modelling and Expert Systems**

Modelling in accelerator control, meaning on-line use of models for setting machine parameters, possible abstract ones, is starting to be used at some places. There are a number of pilot projects using expert systems to some purpose. Combining these two is the obvious next step. Model based expert systems may eventually take a more central role than real time device control. However, very much work needs still to be done for all three and it will take a number of years before these come to some measure of maturity.

Again we are hitting the resources barrier. Like in many new fields, the pay-off is unclear, uncertain and no time scale can easily be fixed. With the background of existing priorities and delivery commitments, management is not exactly jumping at these new ideas. This situation is not new. Initial experiments have been "underground" almost everywhere, made in evenings and weekends. When something more or less convincing was achieved, one then tried to sell it to management. We are at this very point now: some managnes have bought the notion, but support (in terms of resources) is not always convincing. Because no numbers of benefit/investment can seriously be advanced. One must then bootstrap. One pilot project should help defining a next one with better cost/benefit ratio. It is clear that substantial investments will dominate for the next few years. But the potential is there, so one way or another it shall continue.

Again, because of resources, collaboration is an obvious thought. Since expert systems are largely implemented at a high level, the a priori possibility for collaboration is good. The same holds for modelling and that has been demonstrated by the successful collaborations on porting models from one machine to the other.

**Question:** will expert systems also help us designing better control systems?