ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S21CS01

ISSUES in ACCELERATOR CONTROLS

A personal view, from a distance and in soft-focus

(Conference Summary, ICALEPCS-91)

Berend Kuiper, CERN, Geneva, Switzerland.

Dear colleagues,

The fact that I am standing here in front of you at this moment of the conference should issue two different signals. To you it should signal that the more frivolous part of the meeting has started; to me it signals that I must have reached a certain age....

The second personal comment which I want to make is that, at the lunch meeting in San Francisco, when the International Scientific Advisory Committee, ISAC, were discussing the ICALEPS-91 programme, my friend Shin-Ishi Kurokawa suggested that I should say the "closing words". Of course I was very flattered and, since that looked to me like an affair of ten minutes, I promptly accepted. By the time I received the final programme, two weeks ago, I found myself put down for a "conference summary" of 40 minutes. Of course I am still very much honoured and - noblesse oblige - I shall try ..... But what I shall really present to you, will be in the form of a, somewhat hand waving, overview of a number of main controls issues, as I think to see them, from a certain distance and in a soft-focus, with occasional side comments on what the conference gave. So it is neither of the two, or both, however you like to see it.

When attempting to make conference summaries of this kind, one is always tempted - and possibly even expected - to "discern" and then to point out the "great lines" of evolution of the subject and then to make predictions, "far sighted" if possible. Of course such an activity is jolly risky since at the beginning of any such trend, a few discernible examples and implementations of one sort, or a new product here and there, do not necessarily make a trend. By the time the developments have really taken on, however, the "great line of evolution" has become obvious to just about everyone and chances are that the trend is already approaching its end and that some other trend - at that point with hardly decodable patterns - is already infiltrating the old situation which - since it is by now known - has become comfortable and homely and - thank God - at long last more or less efficiently usable.

What, then, is there - other than the technical novelties themselves, which are so disconcertingly complex and changing - what is there to guide us in deciding our directions, to decide what course we shall choose, what we shall buy, etc. Well, ladies and gentlemen, it may sound like a platitude, but the sole constant factors in all this - on the long term - are, on the one hand, human nature with its penchant for comfort and simplicity in doing one's job and, on the other hand, the omnipresent limitations in our resources, in other words considerations of economy.

Now I am not trying to deny that some future technological breakthrough can bring an enormous benefit and change radically the way we think about our problems and that, in doing so, it may hit us from an ambush or so to speak pull the rug out under our feet. We have lived through that before. What I want to say is that this will only happen if that breakthrough brings convincingly more comfort and/or economy. Vendors are slowly learning to bring their new products and trends in a more constructive way and not any more with the sole aim of wiping out the competition and thereby possibly also the very customers they are trying to court. The reasons for this - recently more rational - behaviour is human nature and economy. The human wish for comfort blows up the software packages - systems and applications (just think of transmission protocols, graphics, etc.). The investments are then becoming so enormous that frequently learning new systems and porting software become insurmountable barriers. And creating new such barriers becomes increasingly unpopular with the clientele, who are constantly growing and, through user groups and other

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S21CS01

mechanisms, start becoming increasingly significant pressure groups. Some of the recent new technologies are therefore only taking on so rapidly since there exists a real need and because they were presented with some simultaneous adoption of standards, which warrant some degree of continuity and portability. Conversely, OS/2 (for example) has not made it since apparently there was no direct need in the field where DOS was being used (no greater comfort, just new learning) and because there were no clear signs that there will be a standard. There, where OS/2 could have made a difference, UNIX was already taking up the field and so one may say that OS/2 was too late, irrespective of its qualities which some people are praising.

But let us return to controls. In the accelerator world most controls people are still licking their wounds from the recent Wild West situation in which they were married to all kinds products whose manufacturers then either changed their mind or disappeared. We now try to start all over again and are attempting to extort money from our managements under all kind of pretexts and inventing all kind of euphemisms for the one simple message "scrap it all and build it anew". By the way, I see with pleasure that the slogan "rejuvenation" has also caught on at KEK! I assure you, dear colleagues, that it is not by mere frivolity of just wanting to have the next gadget: we are far beyond that state of mind. It is by the conviction that a new era has started in which we get really new value (meaning user comfort) and really more guarantees (meaning economy). And this is the background of many of the upgrades, revamps, rejuvenations and what have you, of which a number of interesting papers are reporting at this conference. And you may have understood, ladies and gentlemen, that with these phrases I consider having done that part of my duty which is summarising the upgrades of this conference. But this is just an introduction, as you may see.

Next come the large accelerator projects in statu nascendi like SSC, RHIC, HERA, UNK and - let us hope - LHC. Where shall they go? What will their choices be? It is interesting to speculate on the so-called trends on the basis of such more or less concrete projects. Some of them, like RHIC and LHC, are somewhat constrained by their prehistory and previous investments (economics!). They then have to choose between some continuity and homogeneity and more radical innovation. They most probably will try to steer a middle course but they will not really be able to resist the new things, because of the long project times. Then there are SSC and UNK, who both start practically without previous investments that need being safeguarded. Both are starting in a green pasture, in a certain sense, and for a certain time. SSC is confronted with a plethora of possibilities and, at least at present, apparently matching resources. UNK depends to a large extent on the USSR industrial market, which is presently still narrow, but recently they organised some hard currency influx through a collaboration with CERN. Both projects will suffer - for different reasons - from the very long project times. Sticking to one technology will prove an illusion: in the time-span of around 10 years, technology may typically change once or even twice. What, then, should guide the controls people in making their choices, which ones are meaningful and which are not, how can they avoid disaster, where shall they place their ambition, make their mark? Again the only two usable guide lines will prove to be human nature and economy of resources.

Traditionally there exists what one used to call architecture, a complex of problems on which, in the recent past, numerous experts have hotly debated and written innumerable papers at conferences and otherwise. Recently - and, dear colleagues, here comes my first controversial statement - architecture, in the conventional sense, is an issue that seems to be on its way out of the accelerator world. And we should not be too sorry for that since, to be quite honest, architecture (with all respect for my architectural friends here) is not a main issue in itself but rather it used to be a constraint which kept us from what it was really all meant for, that is controlling accelerators. It is my personal experience, which has been confirmed by looking at other labs, that the investments in architecture - which I admit were necessary in those times - have eaten up the larger fraction of the totally available controls resources and the real accelerator controls work, that which some of my friends disdainfully call "only applications", came always too little and too late. It was as if we were constantly building a piano which then kept us from really making music.

But fortunately it seems that we are on our way out of this vicious circle because of the enjoyable fact that industry, helped by the researchers in many places, are getting closer and closer to offering complete turnkey computer networks in which a very diverse collection of computers, equipment and gadgets may be simply plugged in and which then can communicate with each other in a user

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S21CS01

friendly way. The technological intestines of the computers and networks are becoming more and more invisible to the end user and even to the applications programmer. The higher end of the control system starts more and more to become a black box with a number of user related and user understandable functionalities. Although the investments in development of these products are gigantic and will remain so for some time to come, it will be the task of industries (and - conceptually - universities) and it will seize to need our (the accelerator people's) development. It will soon suffice to make a judicious choice of what building blocks to buy and then to do the integration. Moreover, the recent and continuing efforts of standardising at all levels on protocols and other interfacing conventions, makes that the plugged-in equipment and gadgets may be exchanged for newer versions, using entirely different internal technologies, which may then increase their performance but without fundamentally changing their functionality. We may actually have reached something like a "standard model" today, but the point is that the model may even evolve without wiping out our investments.

You may have noticed that at this point I think that I have dealt with architectural topology and with networking.

Of course some of you may remind me that I have not mentioned the new ideas of Rob Parker and others at SSC (and, in a sense, of Steve Magyary at LBL), that is - naively speaking - mapping the whole process address space in one huge memory of a huge and fast central computer and constantly updating those data with a fast data pump straight from the front-end, through fast multiplexed optical fiber links. Conceptually very simple, the only problem being technology. But let us not fool ourselves, this is nothing new, this is how it all began twenty years ago, this is where one starts thinking in the first place and I vividly remember my first primitive thoughts when I was parachuted into controls around 1975. At that time, that scheme was spoiled by the growing size of accelerators and the slowness of what had to be reliable transmission (CAMAC and all that). That had as a consequence, the necessity of making an explicit choice of what data really had to be fetched to the control room, since getting more than a very modest choice was simply not possible then. Today, with the new, overwhelming possibilities of data transfer, the simple "over-kill" scheme, mentioned above, may again draw within reach, even for the largest size accelerators. Then, once all the data are - physically fresh - in a central memory (preferably even the last so many of them, in a rolling buffer, well labeled with their cycle number and time-stamp), then all you need is a central machine with enough parallelism and you can do almost anything you want. And conversely, one may act back on the process globally with response times which will be vastly more "real time" than previously. But again, no illusions, the fastest global feedbacks shall always be dedicated and bypassing the central machine. In my mind there is no doubt that this scheme will now become technically feasible. The question is only whether it will be economical and - above all - whether in the large accelerator (and other) developments projects, which spread over many years and involve many separate teams, the distributed networks are not more acceptable sociologically since they allow a more natural decoupling between those teams, which in turn allows each team to proceed according to their own style and working rhythm (remember: "my car, my wife, my computer).

No doubt you will now start saying: Berend, since you are busy pooh-poohing all our cherished main issues on which generations of accelerator controls engineers have made their careers, what then do you consider a real issue? Well, one good example is certainly the man-machine interface. Not so much the workstations as such, which again may be bought from the shelf and plugged into the black box, but the way in which we are using them. It is certainly nice to have the windowing techniques on our screen and possibly on remote screens, it is nice to have many windows and to shuffle them to and fro, to the foreground or otherwise, just like papers on our desks. But the analogy goes further: many of us are used to live with a mess on our desk and having panicky moments when looking for something in the geological strata (although there are favourable exceptions, I admit). The same is of course possible on the workstation screens, only those screens have a much smaller surface area, which adds to the chaos. Nor is it attractive to constantly shift windows up and down and sideways, to the foreground or to icons, when you are in a hot operational or machine experimental phase. In the end, it is my firm belief, what we need and what cannot be replaced by windowing alone, is simply more square meters of screen and more pixels, so that a judicious choice of displays may be shown simultaneously without the need for interaction just to find the appropriate sheet. Remember that it is still infinitely simpler to flip your eyes from one screen to another than to take the mouse

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S21CS01

and call another window. So here comes my second statement: we need vastly more pixels, not developing ourselves but making the correct signs to industry, who will surely react sooner or later, since that need is not specific to accelerator controls but to human nature.

But more pixels immediately confront us with the already marginal speed of these splendid new devices. The beautiful and powerful graphics packages have vastly increased the quantity of data and code which is being manipulated and thus even our present higher end workstations could easily accept a factor of 10 improvement in throughput, without us becoming unduly spoiled. But when one aims, as I recommend, to a factor of between 10 and 100 more pixels, then it is fairly safe to say that we are still looking for a factor of around 100 in throughput before the workstations will really give us all the comfort we can use. The recommended throughput is of course not proportional to pixels, partly by algorithmic tricks, but also partly since more parallel screens displayed make for less manipulation which latter would eat most of the resources. The process data displayed will, in comparison, have more modest requirements by that time, even when we permanently display a choice of refreshing oscillograms coming up through the networks. Fine, you will say again, but all these fascinating developments can only logically be done in industry. Where can we, the accelerator controls engineers make our mark? The answer is: in the design and layout of the display and interaction surfaces, essentially a thing related to applications development, to which I shall return in a moment. It can be a fascinating work, requiring knowledge of the system, of the accelerator problem, of psychology and above all a good sense for proportions, i.e. common sense.

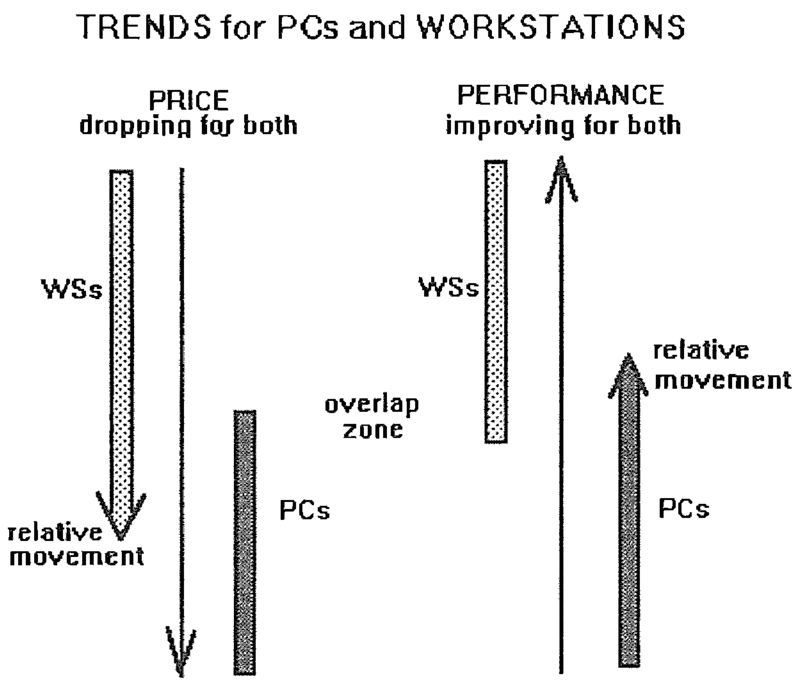
Now we happen to be on the subject of workstations, it occurs to me that some individuals at CERN wage an intensive action, both technically and politically, with the doctrine that PCs and DOS be the panacea for all controls matters. Let me say right away that I do not mean Alberto Pace, who is always rather balanced in his statements. And in fairness it must be said that, after Magyary's publication at Vancouver, a recent pilot project at CERN, i.e. the controls for the experiment, of which a paper is presented at this conference this morning, has handsomely and very convincingly confirmed that, if one wants, then one can, in a number of contexts, conveniently use the combination of PC - DOS - Novell - Some commercial applications - PC-I/O-cards. Earlier in this session Magyary filled us in about the present status of his system. I hope you have managed to attend: "Cela méritaitait le détour", as the Guide Michelin would say. But since the polemic persists and seems to keep both controls people and management at CERN off balance, it may be worth while dwelling a few moments on the subject. The question is whether this "PC versus Workstations" is a real issue, or even, whether using PC & DOS all-over-the-place is a breakthrough in Western thinking... Let us see...

The arguments fielded in favour of using PCs all over the place, for the next generation of control systems, are more or less the following: (1) there is a plethora of high quality commercial applications software with good documentation...offered by many vendors, according to one standard...and, in some cases, with good mutual integration; (2) PCs are cheap, since they are mass produced...and there are many vendors for the same standard (conveniently forgetting the Eisa-Microchannel dichotomy!)...(3) there are excellent networking products like Novell, but also other ones...(4) there is a large collection of all sorts of plug-in cards for Input/Output and other purposes, again by different vendors. Finally, (5) using PCs for all controls allows office work and developments on accelerator controls to be done from the offices on one and the same station. All this constitutes the kit of building blocs which - remember what I said before - makes the issue of architecture redundant for accelerator controls people. One just makes the choice of what to buy, then plugs it all together, installs the commercial applications and - bingo - starts controlling.

Now that looks all fine! There are, however, also other argument to be made:
(1) The plethora of commercial software products mentioned are all running under MS-DOS, a single user operating system. For the more sophisticated networking in larger accelerator projects, however, multi tasking operating systems are a must...and in that case why not UNIX, or the like?....Moreover, the advantage of this wide choice of applications should not be overstated, because only a minute subset (the spreadsheet and one or two others) is of any practical potential use in accelerator controls... and, even so, recently the business world (who were traditionally DOS oriented) have started discovering UNIX (which previously was a hobby of universities and science freaks), so the UNIX market is now in accelerated expansion and the same software houses as referred to above are

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S21CS01

now porting their products (like Lotus 1-2-3, Wingz, Mathematica, etc..) to UNIX. Although the difference in number (all counted) is still great at the moment, the difference in available number for the accelerator controls-relevant candidates is modest and shrinking.
(2) It is true that PCs are cheap....if one takes the cheap ones. In this comparison, the PCs should be compared with the combination of workstations at the top and, say, VME based front-end processors. Taking the workstation level first, we may immediately observe that, when we compare equal things, meaning equal processing power and equal monitor screens, or, in other words, high end PCs with medium workstations, then the differences in price are not great and one should also state that today the monitor tends to be the larger part of the cost of the total package, hence by definition equal price for equal performance. Moreover the tendency is converging, as Alberto showed this morning:



At the front-end level, it is true that one PC crate is cheaper than one VME crate, but, by the time one has industrial quality and made the calculation per slot (a PC has 5 and a VME crate has 21), the difference is dwindling...and the individual Input/Output boards typically (now still unequal in price) will be converging further for similar functionality, simply by the component count and square meter price...there are no miracles at this level.
(3) No problems for networking between workstations and VME crates, the TCP/IP possibilities, with software and all, are commercially available and in future one my migrate to ISO standards and other developments, when these will become a practical reality, and all that is likely to stay entirely transparent to the user.
(4) The choice of functionalities for VME cards is already by now comparable to that for PCs (and growing fast), since from the start the industries became interested in it for their automation projects and the PC has only invaded that field coming from administration.
Finally, (5) it is now perfectly possible to call an X-window under MS-Windows on a PC screen in an office and to work on it.

So, in order to cut a long story short: the signs are that the two worlds, the PCs, on the one hand, and the combination of Workstations with VME, on the other hand, are converging and that the "PC versus Workstation" issue is a non-issue....it has been overtaken by the evolution of the market. There is of course nothing against the PC in the appropriate context (with DOS or with UNIX), but there exist no plausible arguments to use only PCs.

Now, after this frivolous interlude, let me gently move to where I think the real issues will be in the near future. It is my feeling that it will be in two broad areas: (1) one is the so-called front-end, the embedded equipment controllers (far front-end) and their connection to the upper layers (FECs) and

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S21CS01

the face these present to the applications and to the operator. (2) The second area is the applications software, in a broader sense.

Let us start looking at the front-end first. In our sort of organisations, it is more and more the tendency that the embedded intelligence is not the responsibility of the controls group and - if anything - this tendency becomes stronger. There is also some logic in that situation since there is a stronger binding between that intelligence and the device (say a power supply or beam transformer) than there is, or should be, between the control system and that local intelligence. Moreover, interfering there by the controls group would also be inefficient by the mere numbers in question (at least in the larger installations), and by the tendency of that local intelligence to grow into more or less autonomous subsystems which - in addition to their operations-oriented functionality - have an intricate internal life which is jealously guarded, and may just about be managed, by a good number of dedicated groups, each with their own electronics wizards. Thus we must henceforth resign ourselves to the idea and the fact that internal life escapes our (the controls people's) detailed understanding and control.

But we are well advised to take appropriate measures to safeguard the upper part of the control system from importing the intricacies and diversities of the far front-end, because these imports would constantly require adaptations and patches over patches, which in the end will make the system non-understandable hence unreliable. Even when not explicitly imported, these intricacies tend, over the years, to diffuse upwards, mostly under the pressure of an urgently needed functionality which would otherwise be much more complicated to implement. How then can we safeguard ourselves against these imports? Well, the old methods of agreed boundary conditions (today called "protocols") between the two domains are still as valid and as effective as ever, provided that both parties honestly use the agreed boundary conditions only and do not "hack" into the other domain. When sticking to such a discipline, both parties may arrange their internal lives more or less as they want, at least if this defensive aspect were the only one.

With somewhat more ambition, one may speculate about interchangeability of devices of the same category, without the need of changing the software above the boundary. That requires, in addition, the standardisation of the functionality of devices inside the categories.

And the ultimate speculation may be of connecting every process device through a standardised plug at any point into the control system which latter then interrogates the device and, if it is recognised (meaning that it exists in the control system's data base), configures itself accordingly and initiates the device. This needs standardisation at the electrical and connector level as well (and a few other things, of course).

Now the first level, the level of defense, has been reached long ago in hardware and there, where only hardware was concerned, the result was always quite good. But since devices today are a combination of hardware and software, the possibility, hence temptation, of cheating is much bigger and therefore the danger of the upward diffusion, mentioned above, is a real one. It can only be contained by human discipline, unless one squeezes the physical connectivity down to an unpractical level.

When pursuing the second ambition, the one of hardware interchangeability and at least partial applications portability, one must join the present endeavours around the so-called controls protocol, which was the object of the workshop this morning. This work uses the fact that an accelerator consists of only a small number of categories of process devices and that, within each category, there is a strong similitude of those functionalities which are relevant for operations (as opposed to pure engineering and service functionalities, which depend on implementation, i.e. technology and the taste of the device engineer). With some goodwill on behalf of the device engineers, these operations-relevant functionalities may be made to conform to one model. This means that the behavioural model of the device can be standardised and consequently also the software interface towards the applications programs and the appearance to the operators on the consoles. This in turn allows conserving the applications software when devices are interchanged for versions implemented in different technology. It would also allow moving devices between accelerators and also porting

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S21CS01

parts of applications from one accelerator to the other, but that is a bit more complicated and I shall come to that a bit later.

This protocols work has long suffered from conceptual and semantic misunderstandings between the parties and of laboratory-specific political difficulties, but it is now starting to make some progress inside CERN since it has finally received managerial support. But the issue is presently far from exhausted and substantial work must still be invested, including conceptual work, and it must be supported by examples of implementation, before this chapter will have reached maturity and thus the economy will start paying off the initial investments.

At the far front-end, there has since years been a hesitating, but now probably accelerating, penetration of industrial Programmed Logic Controllers, so-called PLCs, in particular for the more industrial like support services, like power, cooling water, gases, radiation protection and access control, who previously were not always integrated with the accelerator controls. It is not excluded (and there are examples) that these techniques will also diffuse into things like power supplies and vacuum, but less likely into beam instrumentation. These PLC devices come in a kit which allows configuring a range of controls functions and allows simplified applications development with a minimum of conventional coding. The material is robust and reliable and often comes with the fitting patch panel material which allows organised interfacing to the sensors and actuators in the field. Although dubbed "expensive" for many years by the accelerator controls community, there is the dawning realisation in the laboratories that, all things counted, they may in the end be cheaper than the home built controls at that level. Now it should be clear that the influx of all this new technology does not change human nature, and thus the specialised groups, will - even with such a kit - keep building their own subsystems with all the required local diagnostics and with some measure of stand alone capacity. So what I said a bit earlier about the need of well defined boundary conditions is likely to remain for some time to come, since that is largely a consequence of human nature.

Coming now to applications, it is not too surprising that, since in the past so much energy went into creating the architecture, that the applications software was always too little and too late. The causes of this were, first, the necessity to create the architecture before applications could start, secondly, that the management was not wanting to see the real cost of controls, and, last but not least, the relative programmer-unfriendliness of the system, i.e. unfriendly already for the professionals and consequently so much more so for the uninitiated operators, equipment engineers and accelerator physicists.

Although there is considerable progress since then in the basic environment and tools for program development, meaning operating systems, compilers, debuggers etc., the problem of applications development is still as severe as before since in the meantime also the user expectations have grown in step with the sophistication of the market and so one may confidently say that for any of the large accelerator controls projects, mentioned earlier, an effort of the order of 100 man*years is required for an applications package which is supposed to give some real comfort. This canonical value seems to be invariant in any transformation. Over the life cycle of the system this is considerably more. Not yet included in this are the various accompanying upgrades and rejuvenations which, depending on their ambitions, may add large fractions of the mentioned effort to the bill. At this point I am therefore already speaking about the order of 500 - 1000 man*years due in this decade, for the world's five largest accelerator centres alone. If now we also include the world's medium and small accelerators, then we may, again conservatively, triple the bill to say 2000 - 3000 man*years, so 200 - 300 Million SF or, if software houses are involved, 0.6 to 1 Billion SF. Now, try to mix into the argument the still growing level of user expectations, which is nurtured by the beautiful applications (in non-accelerator topics !), available for little money on the PC-DOS vehicle, and then there seems to be no end to that game, say 1 to 2 BSF or, to quote a nice round figure, 1 BUS$ for the decade.

Having arrived at this point one may ask the question: is there any analogy with the topic of architecture as discussed previously, is there any hope that in the next five years or so the industrial software products move equally massively into the field of applications creation, I mean applications which are relevant for the more sophisticated accelerator control?

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S21CS01

Obviously the answer must be at least partially yes..but let us see.. There are three categories of applications software which are relevant:... First, there is a number of generic applications software packages, developed for conventional process control (say chemical plants), which are already successfully being used for controlling utilities, general services and even vacuum. Second, there exist a number of generic applications packages, which actually grew out of the accelerator field, and therefore may be called more or less accelerator oriented. Examples include: the Vsystem, the EPICS system, equally born at Los Alamos, and the CEBAF system. And then, at this conference Le Goff presented his interesting ideas about a generic control system for large physics detectors, of which I have the sneaking suspicion that it may well be adaptable and extensible to the accelerator field proper... Finally, there is Rol Johnson at Maxwell Brobeck, proposing to sell the physical applications of Fermilab (in a manner he will no doubt divulge in due course). Third, there are a few commercial programs which have not been made for that purpose but which may be used for accelerator control, in certain contexts. Examples are spreadsheets like Lotus 1-2-3, Mathematica and others, available under DOS and under UNIX.

So far so good, the industrial packages can do a good part of the job and, with some goodwill, their range of reasonable usabilty in accelerators may be somewhat extended. The programs originating from the accelerator laboratories obviously go a longer way towards our needs, but they seem not yet to be covering the field for the larger installations and more sophisticated applications. Programs of the third category often do not have the handles by which they may be easily connected to the control system, although, with some effort, it may be possible and often worth while. Programs of all three categories each cover a certain field and the ranges are to a certain degree overlapping. But data exchanges between them are in the best case awkward, which means that they lack integration. Once I have pronounced that word, let it be said loud and clearly that what we really need in the end is a degree of integration like in the MacIntosh. And then the question, which I asked previously, may be reformulated as follows: is there any hope that industry will, within the next few years, provide us with the packages which are accelerator oriented and highly integrated, like they are in the MacIntosh?

And to that question my answer would be: not jolly likely! And I think that there are several reasons. Firstly, the market is too narrow for them to deploy the huge efforts which have, for example, been the pre-requisites for the really nice and powerful software under PC-DOS. By huge efforts I mean the sum of several competing firms. Remember that the high quality was rarely achieved by the first manufacturer. For example, of the leading spreadsheets MS-Excel is the third and Borland's Quattro the fourth. And then there is of course no chance at all that industry develop our specific applications for each individual operation (unless at cost plus basis). At best they would offer us tools and a few selected generic packages, on condition that these stay close enough to the needs of the much larger industrial market. The second reason is that, in order to achieve a good coverage and relevance and quality, a large enough number of practicing accelerator people must be involved in the relevant industrial firms.... and that is not the case.

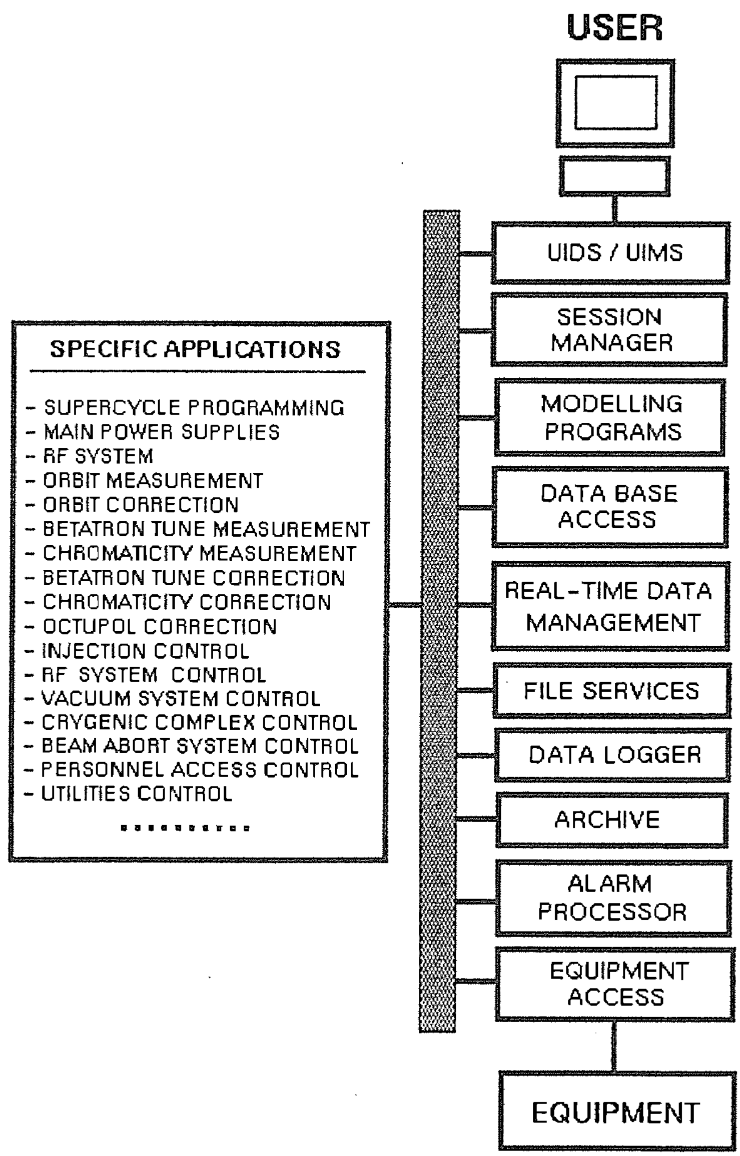
We can then do either of two things: either we resign ourselves to the situation as it is, which means that we keep muddling on with a mixture of what the market delivers today and knit it all together the best we can, accepting the limitations and frustrations, or we take the bull by the horns and try to help in the good direction. But how could we do so?

Well, first an foremost we must specify what we need. Now some people tell me that every new accelerator is different, so we cannot specify controls for a future accelerator. But we have seen that a lot of functionalities, mostly those which support the specific applications, do recur in every new accelerator in some flavour or other. There are now plenty instances to prove that point. By now that collection is called the applications environment and there are people (including me) who suspect that these program codes may be made generic and thus accelerator independent, but configurable for a range of individual accelerators. Examples of the software functionalities in the applications environment are cited in the right-hand column of the picture on the opposite page. It is from Nikolai Trofimov's talk.

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S21CS01

The UNK Control System

Specific Applications and Application Environment



ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S21CS01

Note that the practical implementation of such a scheme will meet fewer technical barriers today than a few years ago, through the present trend for convergence of the architectures (Standard Model). But it is essential that this specification be done by the leading accelerator laboratories together, for several reasons. The first reason is coverage and relevance (maximum input of different requirements and experience). The second is political acceptance (those who participate tend to accept). The third is economy: only when a sufficient number of laboratories agree on these specifications, will the size of the market be large enough and will one be able to afford the price; conversely, only when the same product can be sold to a number of the bigger clients, may it be attractive enough for industry to produce it. Obviously we will never reach the price levels of Borland's Turbo Pascal at US$ 99,95 per licence, but after a few years our software bill will be vastly cheaper than it is today. And we will gain other advantages, like time to concentrate on real accelerator control problems, in contrast to spending our time on coding the same functionalities over and over again in different, incompatible flavours.

Such a collaboration could have a number of positive spin-offs. First, we are obliged to compare notes in some depth and not only by presenting each other our brilliant solutions during conferences. Second and related, the quality of the product will in general be better (if we can avoid the camel which, remember...is a horse designed by a committee). If, on the basis of these common specifications, the bigger laboratories together issue a call for tenders and negotiate a common contract, then, third, we may profit from a joint software maintenance organisation and, fourth, we may at long last get good updated documentation. Fifth, many smaller labs will then certainly join, which will stabilise the setup and improve the service. And so on...Alternatively the labs may set up a consortium of their own, but the industrial version seems to be the wiser one.

Even if we do not in the end make a common contract, the common specification exercise will be a highly interesting and rewarding one. All parties will be coming out enriched. For one thing it will become clearer what are the essentials and what is "couleur locale". I would therefore propose to make an exploratory workshop, somewhen not too late in the next year (1992!), involving around 20 or so people with experience in operation, machine physics and controls, with the aim of exploring what common ground there be....

I shall stop my speculations at this point, but you now see why it is my feeling that there will always be an interesting job in the applications (even if we keep muddling on like we do today), most certainly in the specific ones and then there will be a very challenging one in specifying, developing, updating, adapting and extending the mentioned generic software concerning the applications software environment..

Ladies and gentlemen, coming to the end of my long palaver, I wonder whether, by all those words, you may have lost the thread of he argument I am trying to make. Summarised in a few short phrases it goes as follows. A number of issues, hitherto main fields of our endeavours, which have been at the center of our attention and consumed most of our resources, will henceforth be industry driven and not any more our - the accelerator controls people's - field any more. That field, a bit broadly and hand-wavingly, may be called architecture. If you want to stay in that field, then go to industry: a lot of the advanced work will be done there. The two fields where - in my opinion -. the accelerator people will now have to concentrate their attention, where they can still be creative, where they still can make their mark, - and where they are really indispensable ! - are: first, at the front-end, in particular its connection to the henceforth industry-supplied upper architecture, and, second, - of course - the wide field of applications. But, in order to deal successfully with either or both these fields, the controls people must become much more interested in the accelerator proper (or telescope, or tokamak, ...) and that is not a bad thing. Not bad, since in this way we have moved a bit closer to what we are being paid for in the first place: that is controlling accelerators.

Thank you for your patience!