Session I: discussion (1st part)

SESSION I

DISCUSSION (1st part)

October 6, 1982 - Morning

Chairman: J. BLAAUWENDRAAD (The Netherlands)

M. FANELLI - I would like to ask from Dr.Haas whether he could spend a few more words about the integration of the quality control process into the automated management system.

W.R. HAAS - The quality control was not part of the system. We didn't do anything like numerical control of the machines,or quality control;we only provided the means to plan and to supervise the process of producing the prefabricated members. For example,we provided means to keep track with the actual production, but the supervising engineer had to look at the prefabricated member and to decide whether a prefabricated member was of good or poor quality. If the prefabricated member was of poor quality, he had to enter data in the system which indicated that the prefabricated member of poor quality had to be fabricated once again.

M. FANELLI - Anyway I suppose all information that came out of this quality control process was stored into the data base.

E. ANDERHEGGEN - I have a question to Prof. Werner. I want to know at what level this system is working, within the operating system of the computer. Are you talking about a new operating system, which is not based on an other one (like UNIX),or is it just a job superimposed to an existing operating system? Another question is: on which hardware do you implement your system?

H. WERNER - First to the last question: the computer we are using is the HP1000L with 512 kbytes and the RTE/XL operating system. On the input side,we have three different devices: Televideo, Televideo with graphics and a HP graphic terminal. On the storage side,we have a ten Megabyte harddisk and the database management system Image 1000. The output side consists of a matrix-dotprinter and a small HP plotter. In the methods-base there is a well proved software suite for structural engineering.
Now to the first question: Syrakus is not a new operating system. We take it that an operating system is given. Our software works between the operating system and the application programs,written e.g. in FORTRAN. We have to write some small layers for adapting it to a special operating system like RTE, UNIX or CP/M. The operating system RTE/XL allows the user-communication-module to work in the foreground,while the more time-consuming calculations are done in the background of the computer.

D.D. PFAFFINGER - I have a question to Dr. Haas or to Dr. Werner, I don't know. The system you explained is basically a draughting and construction system. I am wondering on the feasibility of extracting from such a system the data which are necessary for the structural analysis, which is a further step of idealization. Could you comment on this?

W.R. HAAS - In the project which I described, the structural analysis had already been done so we didn't make any effort to close the gap between the design data and the input data for the structural analyses. In principle, it should be possible, of course. We investigated it in one special case. We constructed walls and columns and above the walls and columns a flat slab. We then semiautomatically extracted the structural system for the analyses of the flat slab from the geometrical data of the walls, columns and slabs. That's the only case that we did such an investigation.
In other fields of application, there are links between CAD systems, or modelling systems, and systems for structural analysis (for example finite element analysis). The CAD system produces a solid model which is subdivided in a finite element mesh by a mesh generator and then analysed by a finite element program.

H. **WERNER** - I think your question was a question on specific applications and this specific applications belong to the base of the methods.

H. **PIRCHER** - I have two questions to Dr. Haas. The first question is: if you should have a similar project, how much manpower would you expect to save to do it? And the second is: what is the minimum size of a building to use such a system?

W.R. HAAS - Well, to the first question, we have now a lot of experience which we would use in the second project. We also have now a better 3D modelling system and we also have many programs for generating lists for bills of quantities. If a similar project would come, we would use our experience and the existing program library and I think we would now be able to produce a softwere package, specially tailored to the new project in much less time, I would say 1/3 of the time needed for the first project.
The second question concerns the project size. I think it is not so much a question of size but of type of the buildings. I think rather small projects of prefabricated members can be handled with our system, whereas for projects with cast in place concrete buildings the software must be changed to a great extent.

D.P. GREENBERG - Question to Dr. Wagter. You made some statements with respect to how difficult it was to use some three-dimensional modelling systems in civil engineering practice. There may be 45 minutes computational times for some of the modelling routines. I wonder if you could comment on some of the causes of the inefficiency in the computation behind the systems which you investigated.

H. **WAGTER** - We stated that we looked at the systems as users, we did not get involved in algorithms and efficiency of the computer program itself. We just tried to define, I think, a few simple programs and to look at how that works out in three-dimensions. The only difference we met was the difference between wire models, surface models and solid models and that has various connections to the algorithm of course. We found out that nearly all the complications in solid modelling took a lot of computing time for the algorithm itself. Doing all kind of manipulation in surface models, you have to use a hidden line algorithm; this problem is automatically solved in solid modelling. It takes a lot of time and I did not see any wire model that can use a solid model as yours

has to look quite through it. So I cannot give an exact answer to your question because I can't give an answer about the efficiency; it has directly to do with the amount of information that has to be taken into account and this directly depends on the type of model examined and the algorithm used internally.

D.P. GREENBERG - I have seen and had similar reactions to many of the commercially available systems, what I am trying to find out are the causes of inefficiency, which really gets down to either data structures or the algorithmic method, which you use. I am not sure what they are.

H. WAGTER - One of the main reasons of inefficiency mostly occurs when different users are working on the same system. For this reason we were happy to be able to test several times as an only user on a rather big system, and even then we got high response-times. It must be said that the number of users and the kind of work they are doing on one system have also big influence to it. It just must be stated that solid modelling is very complicated; it needs complicated computations.

P.J. PAHL - Both the development and the maintenance of software systems of the types which you described for workstations cost considerable effort. I would like to ask all four gentlemen what type of organization do you see as preferable for the development and maintenance of such software?

J. BLAAUWENDRAAD - This is a very nice question. You ask for organization for two purposes: the development and the maintenance.

W.R. HAAS - I think such systems should be developed at Universities with one experienced programmer, because it takes a lot of effort and the commercially working software house is too expensive to do such a thing; we also are doing that, because we don't see any kind of this development in Germany. Maintenance and marketing should of course be done by an experienced software-house.

R.W. HOWARD - I shall reply from my backround, which is a user association which has had to do many things in its time, and succeded with some and failed with others. When we were first formed - about ten years ago - people thought we would act as a focus for joint-development of software, that groups of members would come together, that they would specify their requirements and we would develop software. Now the problems of joint-specification and development are considerable in that everybody asks for any possible requirement that they might want; and the specification becomes so large that it doesn't meet anybody's requirement sufficiently. So I think the best place for development and specificatiion of software is very close to the user. You always have to get one user to say I have a problem, rather as - I think - RIB have done in their example. Having solved some particular problem, you can then generalize it for other users, but they start with a very clear specification of the need in mind, so I think development has to happen very close to a single user and very close to a particular need. Maintenance: well, again I think that the people who develop the software really have to be responsible for the maintenance. Another thing we were asked to do in the past, was to collect together a library of software and provide support and maintenance for users. Now, you have to know a lot about a piece of software, to provide adequate maintenance and, if you are

trying to do that for a range of systems from different sources,it's impossible; so again I think that the maintenance has to come from very close to the developer, who should be very close to the user and that means an engineering company, or software-house, specializing in the construction industry or engineering.

H. WAGTER - I agree in this difficult question with Mr. Howard. I think that the main part of the development should take place at the engineering offices for they are the users. It has been stated that a lot of problems in computer aided design have its origin in the fact that people that develop the software are not the users. Besides that - and I know it is a bit of a danger to say it in an audience where 50% consist of professors - the experience is that the programs developed at universities are mostly very bad documented and do not have a very efficient background of people who made it. It is the same for maintenance; I think and I agree again with Mr. Howard that it should be done at engineering offices who developed the software. An other aspect of that is the maintenance of the hardware that should be carried out by the hardware firm of course.

H. WERNER - I think we have two main, partly overlapping,phases in software development; the first one is the development of new methods, e.g. Finite Element methods. I think especially universities should be involved particularly in this task. But that does not mean that specific application programs, which are part of the second phase, can not be developed at universities. A close connection to the practice, e.g. to an experienced conductor, is necessary in that case.

P.J. PAHL - Let me comment first on the question of university involvement. I think it has been correctly stated that it depends essentially on the degree to which the universities are willing to accept the task of documentation. In several comments it was stated that those who wrote the program should also be involved in maintenance. Universities - as a rule - are not in position to do this. Therefore the transition of software from the universities to the maintaining organizations is one of our problems, at least in Germany. I wonder if others have experience in this area which they would care to share.

G. KRUISMAN - I have some questions to Mr. Haas. The CAD system you described seems to be a production preparation system; production in the sense of building a building. In mechanical engineering you would call such a system a CAM system. Have you considered to name it a CAM system? It would be nice to have in the building industry CAM system too. The CAM system seems to be rather product bound and, in this context, I would like to ask you: was the building method programmed into the system? Was it the building method of the contractor? And was the building designed before the contractor came into the picture, or was it designed by the contractor?

W.R. HAAS - Why didn't I call it a CAM system? It is more than 50% a CAM system, you are correct, but building means modifying and so frequently some parts of the buildings were modified and redesigned and that is the CAD part of the system: so we have a mixture, we have some CAD capabilities and some of CAM capabilities. The original design was done by an American consulting engineering office. The design, however, was not detailed enough for the contractor,for example

the consultant did not do a complete typization of the prefabricated members. So the more detailed design had to be done by the contractor, for example the complete typisation of these prefabricated members. It was done with the aid of this system.

S.J. FENVES - I was going to ask a question to Dr. Werner, but I would like to comment on the previous discussion. One thing we have to learn from computer science, which is widely practiced in that profession, is the idea of throw-away programs. The majority of the programs, especially those in universities, are intended to be thrown away at the completion of the student's study. If there is something more valuable in the student's thesis, it should come out later, through the marketplace and other mechanisms. It is totally unrealistic to think about universities converting themselves into software-houses.

W.R. HAAS - I think the 3D-modelling situation shows very clearly what I mean. What we need are basic methods and algorithms, for example for calculating intersection lines, for all the boolean operations of the volumetric oriented primitives. A lot of work must be done in this area which should be done by universities. We would like to have the same situation as we have in the finite element field, where we can take SAP or any program of this kind and build an application layer around it, which fits the situation inside the market where we act.

S.J. FENVES - I agree with you that the concepts have to come out of universities, but the program that embodies that concept need not be the final program that runs in a production mode, on a variety of equipments and in a variety of enviroments. Very few SAPs, ICESes, etc. are restricted today to the one particular hardware and programming language enviroment in which they were originally developed. They were good programs and therefore it was worthwhile for somebody to implement, improve and translate them.
I would like to know more about the database part of your system, and particularly its interface with the application programs. Did I understand you correctly that only two interchanges occur between the database and the application program; when all the data are in, they are moved to the application program and, when the application program is done, it dumps its results in the database?

H. WERNER - That's not all: at first the database contains information describing the masks. Its second task is to store the input data and the relations between them. The third one is to deliver on request these input data to the application programs. The output of an application program is given to an output file routed by OUTCOM. Information on that output and data relevant to further activities are recorded by the database.

M. KUWAGATA - I would like to ask Dr. Howard: you show us three drawings of three levels of workstations, on each drawing I can find a telephone handset on the desk. Is that used for simple telephone communication or is it used as acoustic coupler for data transmission? I am not sure which purpose it is used for, so please teach me.
An other question is as for the network of data communication. A work station constitutes a network, so we have to solve the communication network problems. That is, for example, data communication network architecture. So have you solved such a problem or not? For solving this kind of problems, we need some

counterparty engineer, I mean telecommunication or datacommunication engineer. I'd like to ask you what type of counterparty you have.

R.W. HOWARD - We did not specify a telephone on the desk. I think this wasn't a specific instruction to the person who did the economic study and I suspect it is for the user to ring his wife to say he is going to be home late. The connection to the network would obviously be an hardware connection and we were a bit cautious about which network, because I think that is very much up in the air at the moment, but obviously there are some standards which are becoming very strong contenders, as the standard network configuration. But basically, apart from looking at the relevant standard in the Telecom area for the public network side of things, again we were rather evasive there; we think at the lowest level something like Teletex, which is the international Telecom standard for linking wordprocessors, is going to be quite useful for exchange of documents; but, obviously, for sending larger quantities data and the standard for sending graphic data between systems, apart from looking at DKS, IGES, etc., we stayed a bit on the fence, as far as some of these standards were concerned, but we are aware of what is going on in that area.

J.P. RAMMANT - I have the impression that there is a war situation created here between software houses and universities. As a software house I would like to ask some naughty questions to the University of Münich concerning Syrakus. I have a serious doubt that this system really can do anything. The first question is: it's hightly modular, very well, I like that, but I fear it will be very very slow in application because there are a burdain of things you take with you, which are, as far as I believe, unnecessary. Second question: you are using ten megabytes. On ten megabytes you cannot do a lot of things, so please convince me and give me some examples of applications which you have done. I would appreciate that.
So, the first question is: it is highly modular and you are using database techniques. Is this really necessary? All these methods, all these burdain of things make the programme very heavy and slow.

H. WERNER - What I wanted to show is that in future it is necessary to use application-independent software; Looking at the practice you see programs written for a specific computer with a specific input form. If they are to be installed in an other computer or if another input form is to be used you have to rewrite these programs.
The overhead of our independent software is not big. Look at the message which defines the input form or look at the message which transmits the input value. You have to write small layers which convert the message into the specific mask on the screen or which convert the second message into a specific input form of an application program. When using another application program this last converter has to be changed.

J.P. RAMMANT - As software developers, we also use standard routines for input and output and layout of databases.

H. WERNER - Can you tell what is the standard input?

J.P. RAMMANT - We are using straight forward subroutines which are, I think,

rather simple; the subroutines manage the input and the screen outline. But I think the principal statement I would like to make is: I don't care about hard ware; if an engineer wants to buy our system, he buyes the hardware along with the software, because the hardware is much cheaper than the software. Also it is clear I don't think you can make programs really independent on hardware; there is always some dependence. If you want to have a nice and good interactive program, you are hardware-dependent; that is my experience.

H. WERNER - That's quite write, it is impossible to have independent software which runs all the computers. Our aim is to narrow the adaption work.

J.P. RAMMANT - The last question is on your applications. I am still not convinced. What applications have you done?

H. WERNER - The system is still in development and the first issue will be a reduced version without a database running on a CP/M operating system. The connection to a database is part of the second stage. In the first version we use a throughput of the input data to the application program. We are just finishing this issue and we are going to present it within three months.

H. PIRCHER - I would like to add a question to this discussion. I think we all agree that there is a need for standardization and, if somebody creates this standard, the first question is how to convince the others to use it and also - I am a software developer - if they would like to use it. Therefore the question is how and when Syrakus will be supplied to other developers and how to convince them to use it. That is a big problem of time scheduling and organization.

H. WERNER - With this question we come back to the question about the relation between universities and software-houses. Our aim is to give tools to the software houses. To your question about time scheduling: I think the whole systems will be ready in the middle of 1983 and then we can offer it to the software houses. To the question about convincing the software houses to use it: we will demonstrate one working system; Syrakus is not an abstract model. Software houses cannot be convinced by abstract models,they want to see a working system. We will be able to show a short version in a quarter of a year and the whole system in one year.

R.W. HOWARD - I have a general question. Which sorts of standard do get adopted? I think we have one successful example which is the CP/M operating system for microcomputers. Nobody ever designed it as an official standard, but it was made available al low cost for the people developing 8 bit micros; they couldn't afford to develop their own operating systems so it is very widely adopted and now even IBM gives an option of the CP/M operating system. And so I think official standards always have to exist, but the ones that really have an effect on the way things work are what are called today "de facto" standards and I would reccommend, if Prof. Werner wants Syrakus to become a general standard, he makes it available at very low cost, makes sure that people are aware of its existance and hopes it gets taken up for very practical reasons.

W.R. HAAS - I would like to comment on the question of standardized input conven

tions. We have established such a standardized input convention in 1976 and it is rather widely used in Germany because 24 program developing institutions stick to it, but it has one shortcoming which reflects the computer situation at the year of 1976: it is not interactive, it is batch oriented,it has powerful possibilities for generating data but it has no prompting. So we now have some difficulties at the market. We managed to establish the"de-facto"standard by selling a software package at very low costs to the software developing institutions which does all the free format handling.
There was a second question concerning portability. I think portability is not so difficult to achieve. If you stick to ANSI FORTRAN IV of 1966,you can achieve a high degree of portability. We have one program running on 16 different computers in an almost unchanged form.

H. WAGTER - The subject of communication and standardization between systems has also been a subject of investigation I talked about and I must say I am very pessimistic about any development in this field. At this moment, it really is a disaster to link systems. I think it is necessary to do developments in this field together with hardware development. There is one system, one development going on in the United States, called IGES. Studying the IGES reports you soon will find out at what a low level the IGES specification is written. The only things all systems have in common are just lines and points and you have to write thousands of all kind pre and post processors to achieve a practical value. This is a result coming out of the existing situation and most of the people here are talking about standardization in a future situation. About CPM, what is called a degector standard, that is correct, but there is also a CPM2 and and a CPM2.2 and a CPM 2.24 and that illustrates a little bit that also here we will see problems. Nevertheless I agree that we shouldn't stop doing work on this field.

C. NUTI - I have a general question, I heard speaking of time but what do we mean for time? CPU time or general time? Because, in general, in engineering the important thing is not the CPU time, but the cost is due to the total time.

H. WERNER - I would like to answer your question in terms of the system Syrakus. If you have an interactive input to a big computer via one terminal you can correspond with the computer directly. You have the computer for yourself all the time. If there are three, four or, may be, ten terminals corresponding with the computer and each of them expects a quick response on each wrong character like "This chapter is wrong" then it is possible that a big computer can ba over loaded just by these activities. That way of employing a computer is not effect ive, I think. The better way is to have some local intelligence within the terminals for checking the input and for giving a quick response. So, if you use software suitabled to the coming network solutions you can save computer time for more number crunching activities. That way the relation between CPU time and total computer time can be improved.

R.W. HOWARD - I see a number of examples in recent use. Companies have gone out and bought some very small computer, because people always look at the hardware first and buy the hardware and then think how they are going to solve their problems; entirely the wrong way to do it. But it has resulted in some very ingenious solutions; people who found the limitations of a small computer when

faced with large analysis-means, either then you take a long time to solve the problem in elapsed time, or take a lot of data on and off disks and in some companies, perhaps small companies, where the computers are not needed continuously, people write programs, which they start off one day and they come back the next morning and it is finished, that may suite a certain sort of working enviroment. So I think you have to design the system to suite the type of office: if it is one man-office,then he is quite happy to leave it to work all the night and get results the next day, provided, as Prof. Werner said, he checked the data to start with, so that he knows he is going to get these results. In an other company there might be people waiting to use that machine and it must work fast and therefore time is very critical.

H. WAGTER - I just wanted to say that you should not underestimate the frustration limit coming out of long response-times. When I am talking of time in my paper, I just mention CPU time, although I understand that the turn around-time during the project may be much more. For a simple handling on the computer, an operator designer just expects the system to response in zero seconds. He has a very complicated job, he has some understanding for the difficult time the computer might have that moment, he accepts some ten seconds. If it is reasonable or not when it is longer, the man gets a little angry about it and the effect of this should not be underestimated.

B. BONI CASTAGNETTI - I have four questions. The first one is for Mr. Howard about the tube he suggested as a raster tube for level-three workstation. For what is my experience in construction and engineering works, the storage tubes with dynamic refresh capabilities are more useful,since the resolution is higher and the screen is larger too. I would like to have your opinion about this fact.

R.W. HOWARD - Well, I think we are very much going from recent surveys which reckon that, probably supported by the television industry, 95% of displays will be raster in four or five years and - once the television industry gets itself set up to sell raster screens, and once the resolution improves I think they would become more acceptable. We felt that they would probably be adequate for level two - level three - we said that storage tubes should also be available, because we now have a lot of people, a lot of users who just won't accept the lack of precision of raster screens at present. People like Tektronix and I think they are coming back to raster screens with refresh buffers and colours. It certain will be interesting to see which technology wins, but I think there is an enormous investment behind raster technology and the resolutions will improve and become more widely acceptable by the users.

B. BONI CASTAGNETTI - At the present time, for what is our experience, 1024 resolutions of the ruster tubes are surely not acceptable by designer, practical designer in construction.

R.W. HOWARD - I think that,if you have the right facilities for zooming,you can get round some of the limitations of raster graphics, but the resolution really needs more time.

B. BONI CASTAGNETTI - The second question was directed to Mr. Haas. Have you any productivity-ratio conventional figures versus computer aided design about your project?

W.R. HAAS - No, we don't have any figure concerning our project.

B. BONI CASTAGNETTI - The usual three to one, ten to one.

W.R. HAAS - I know published figures which begin with 20% increase of productivity up to a factor of 10. But in our project I don't know the figures because the design wasn't done in parallel in conventional way. We got the job because we could convince people that it could be done much quicker with less engineers using the CAD system.

B. BONI CASTAGNETTI - The final and most important question is about the choice between what we can call open system and a close system. Open system is a work-station integrated in general purpose computer and close system is the so called tanky system. For what is my experience, close systems are less useful in a computer aided design, than the open systems because design is a chain, I think: at the beginning a conceptual phase, which is the analysis, then the draughting phase and then the bill of quantity, the bill of material phase. In an actual application, these three phases can be carried out only if you can dialog between the graphic-software and the application-software in a Fortran environment - for instance - which usually are application software, written directly by the final user. I would like to ask the opinion of Dr. Wagter, or Howard.

H. WAGTER - It would have been the beginning of the questionnaire, because it is a very complicated question. Shortly, I agree with you in your opinion that the open system, as you called it, is much more flexible. In a turnkey system you get a lot of software you never use and a lot of software you use is not available. This is a general remark of course. The question is too complicated to be answered with just "yes" or "no".

R.W. HOWARD - Well, again, I am not quite clear about your distinction between open and closed, but I think certainly systems should be open to allow one to exchange information between different systems, and of course a lot of standardization has to be done to make this more possible, at least it should be possible to get inside the system to see how it works in order than this might happen. But, in terms of what might be called closed, your work-station at least having local intelligence, has to be under the control of the user. I think that, with the distributed processing, you can have a lot of intelligence at relatively low cost locally. The problems of response on current day small systems are limited, but he has through his network perhaps access to a greater processing power, if he needs it for large analyses, on occasions.