Zeitschrift: IABSE reports of the working commissions = Rapports des commissions de travail AIPC = IVBH Berichte der Arbeitskommissionen

Band: 31 (1978)

Artikel: Discussion

Autor: Bokeler

DOI: https://doi.org/10.5169/seals-24904

I SESSION

DISCUSSION

August 30, 1978. Morning.

Chairman: BOKELER (German Federal Republic).

BOKELER - Now we start the discussion. I hope we have a lot of sentences and statements in the last few papers, that we have a very interesting discussion. Please, if you start your contribution to the discussion, say your name first, ladies and gentlemen, please.

FANELLI - I was very interested by the contributions by Dr Pfaffinger and Dr Anderheggen and I have to ask both of them a question which is related to what each of them has said. Dr Pfaffinger showed us a very complex example of a complete analysis of a structure with a discretization having many thousands of loads and so on. Now, this kind of analysis is conceivable only for final design. In the stages of preliminary design, we should rely on a more simple idealization of the structure in order to be able to do many runs at a reasonable cost and introduce alterations in each reasonable final design. This is, other types of programs come in, related to much more simple structures, of such, for instance, the in-analysis program that was illustrated by Dr Anderheggen. So I asked, in fact, to start the discussion about how should we view these two kinds of approaches, one for the preliminary stages of design and the other for the final stages. Clearly, we cannot use the same approach in the two cases; we could not use a very complex analysis involving many thousands of equations in a preliminary stage, for we are bound to make many alterations and many runs of the same analysis over and over. So, it seems to me that one of the points we must focus on is a distinction between methods for preliminary design, which must be cheap, easy to run, fast and very simple for the user, and methods for final analysis, which can be as complex as needed.

PFAFFINGER - I perfectly agree with your statement and, of course, one did not start with this big analysis from the very beginning. As a matter of fact, I shall see some of the preliminary analysis we did, so the other preliminary analysis done on frame structure is very simple, just to get the problem in hand and see what is going on.

The same, indeed, is true for the evaluation of the results, to check them, so that also small problems run on a computer to verify that the general flow of forces is correct. I might stress out , in my opinion, you always should start with the second analysis first. Usually it is hard to set up the conceptual model, so you could approach the problem of computer aid, starting with the simple conceptual model and then looking for the results of a more adquate model.

ALCOCK - I am delighted with Dr Anderheggen's thesis of the syntax diagrams: the concept of this describing program with short user's manuals is a subject which can be dissociated from linear analysis, and it is on that aspect that I have a contribution here. We also believe very strongly in syntax diagrams and we re_ wrote the manual of the program STRESS using them. The result, I am happy to say, is a sort of 30 world-wide systems, including one in Austrialia, and also in a similar way, we have a tendency to find another similar system in France and another in Brazil, where we took the common language and an afternoon' s work to convert it from English to Portuguese and from English to French. I had valued Dr Anderheggen's comment on that matter because it seems to suggest some difficulties with STATIK in this respect. We said we had no trouble. And inciden_ tally as far as 40/30 world-wide system and we have no trouble. The organization consists of myself and my partner and one 18 year-old boy: that is our organization and there should be no trouble in supporting this. We do have a couple of com_ puting equipment on which to do it. Now, I really must come in on the system GENESYS, which is an English system. We have syntax diagrams very similar to the GENESYS ones. So, I believe that is good thing to describe things this way. I have really got one more thing here, i. e. to understand that machine independence. When we deliver GENESYS, we deliver it simultaneously on 5 different ways, so this can be put all in FORTRAN; and it' s a pity, I think, that STATIK was not written in low level and has not be tempted by the wonderful facilities of this equipment; but perhaps it would not be too difficult to think now to turn it back into elementary FORTRAN. A final word about output which was a remark to start the output as just a list of things. The output from GENESYS is automa_ tically put into a form which conforms to the syntax diagrams and can be reinput. We did not use the term "integrated system" at all - we used the term "integra_ ting system", so that if you wish to communicate from one subsystem to another subsystem, your job has to use GENTRAN to define the syntax diagrams of the output of one subsystem. You then automatically get the input to the other. So I think integrating system is better. Welcome are the comments and the answers on what you think of this approach system.

ANDERHEGGEN - About the languages, this will not be an afternoon work, but it may be a couple of weeks work. It would not be too difficult, expecially the input would not be too difficult (the output could be more difficult).
In the syntax diagram, I always used the first letter of each work, which makes it possible to shorten each word down to the first letter. Why so? Because we run into some difficulties sometimes, but this was a good decision, so if you have to change language, it is easy to change a little bit of it. The reason why we chose, I explained a little bit better in my paper, perhaps because we do not feel to be able to support a large work. At the University, as you certainly know, people are changing all the time. We do not have a commercial organization of this kind, we have as little a commercial organization as possible. We would have liked to write the program to get the ideas and then leave this to somebody else, which might be different from what you are doing. We are not really making money . . . we leave the making of money to somebody else. Of course, our program is successful only if money can be made out of it, this is clear. So, the problem of language is a rather minor one.

In Switzerland, there are people speaking French and other people speaking German . Integrating or integrated may be not too well aware of the GENESYS. I thought , however, this was a kind of off-spring of ICES because they have still some idea which is similar, haven 't they? You have a Gentran, which is something quite strange: I am not sure how it works. What I did not like in ICES is the idea of working with the same data base for extremely different applications like structure analysis or project management: this is something which is inhuman. It would be interesting to speak about integrating system. Of course, it makes a lot of sense to have a program producing output in a form which can be read and also can be used as an input for the next program.
Another program concerns program portability. We have used a CONTROL DATA computer and it is a big temptation to pack a lot of information on one word; then when you go to a machine with smaller word length, you run into trouble: this is the main reason. Another thing is also that CONTROL DATA has quite an exotic version of FORTRAN, which is quite different from IBM. With the new FORTRAN standard coming out, some of these problems might be solved. We will try in the future to write programs which run on smaller computers, medium-size computers for this kind of purpose.

DUTERTRE - I would like to thank Dr Anderheggen for his talk about syntax diagrams; I think if we could, in this colloquium, at least agree that syntax diagrams can be used by everybody, I would ease every one and enable you to use the other' s program. Everybody is being talking about his program, and no one is being talking about someone else using his program, and that is a bit tricky, unless you have a common way to express how to get in the program, and syntax diagrams are one way to express how to get in.
I have particularly appreciated the fact that your work started from STRESS and worked up from there. Following exactly the same pattern, the new STRESS, we have used another program called TITUS in France, which is very similar, as in the input we put the two together, in order to get syntax diagrams compatible, we worked from there and used a common syntax diagram for all our programs. I insist on that: if we could agree that syntax diagrams including simple rules can be used by everybody, there would be a big step forward .

MILSTON - I have been working as design engineer for about 30 years, which means that half my career was before the computer age, and I was really interested in Dr Pfaffinger's paper, where he gives requirements of structural engineering programs and gives conceptual analysis model of the real structure, the representation of results, the interpretation of results.
They are absolutely identical to those we used in structural design before computers were even thought of, and I think it is practically the same as Prof. Hardy-Cross of Main Distribution published in his work in 1932, nearly 50 years ago. I find exactly the same terms: first of all you have a real structure, then you have a conceptual model, then you do a numerical analysis, which gives a solution; then you represent these results and then you interpret the results.
The question is, of course, if there are essential differences between his requirements of structural engineering computer programs and requirements of structural engineering design.

PFAFFINGER - Thank you very much for your comment. I first have to say: I do not know the book by Prof. Hardy-Cross; secondly, if you read the introduction of this paper, I try to stress that the situation now has basically not changed for the structure engineer, having the computer and not having the computer. The only difference is that what we did by hand in the old days, we are doing now on the computer. We do it with more precision, we are able to set up more sophisticated models and we are able to approach reality closer than we could do by hand. The difference between the old days and now really is this: we are enabled to set up more realistic models, bigger models, more sophisticated models than we used to solve, but in the general flow of the work and in the general approach of the work, there is no basic difference. If you read the introduction, I have tried to stress this point.

BOKELER - Is there any other question to the statement "using syntax diagram is a way of standardization" ? Any question to that? Are you sure that everybody knows that a syntax diagram is what we mean? Would you like to say something Dr Pfaffinger?

TOMINO - I would like to ask the real meaning of syntax diagram. Well, I do not know if it may hold some kind of ambiguity, but many people are using that in Industry. The definition is quite odd to myself too.

BOKELER - Can you give a sort of definition, if possible?

ANDERHEGGEN - There are different kinds of syntax diagrams. Very often recursion is used, i.e. there are symbols which represent another diagram and there will be other symbols representing other diagrams, and these diagrams can contain themselves ... this is complicated. Now, the one we use, I think it is rather simple: you just have to follow the arrow and from there it gives you the allowable sequence of input data. So, by following the arrows, you know what the sequence of input data is, but you also see immediately what the program can do and which data are needed for instruction input. Reading the input description of a program which is working with this kind of diagrams, you see that all the input structure of tridimensional frames, very general with curved members or straight members, can be completely described in any detail on one page, for the whole STATIK program. You describe the input which you need for the preparation of your input card, and you also see immediately what the program can do, because there are just a few words, which tells you what it is all about.

BOKELER - The syntax diagram is a sort of checking list, with some sort of definitions for the language. It defines the syntax of the language but in the same time it shows really what you can do with the program.

TOMINO - To my understanding, the syntax diagram is a kind of diagram to define the control of the program: this is really a useful working from a program to another one. Of course, I hope to see in the future that I can use some other

people ' s program very easily, but for this I need some tool which makes possible to do it. This could be the use of a common language. Another could be the creation of a program, a system or a module for consulting program, providing so a kind of communication between one user and another.
In your paper is said it has a kind of memory manager, I mean, a kind of program which manages the numbers on the basis of high memory or auxiliary memories. If we can have such a kind of tool which can manage their transfer, then we can realize such kind of communication.

ANDERHEGGEN - May be I make it somewhat clear. We are talking about users' program, from the point of view of the users, since it really does not matter in which language the program is written. There is a question of communication between a user who doesn't know anything about programming language and the program. Now, our program is not such that you can say: use half of it and then write your own program and use the data - the data are all ready substantially somewhere, - but we are not going to tell you where, so you have to use our program as it is. I think it is a different level we are talking about; I am talking about engineering, not about the question of programming.

ALCOCK - I come back on the syntax diagrams and the levels at which they are used: the basic language is reasonably able to use 25 statements. You can draw syntax diagrams to find the whole of the basic language on one sheet of paper; it's one syntax diagram. Following the arrows through print you can write the names and the variables separated by commas or semi-colons, and this is shown by diagram you have to look. So, if we do it at the language level, we can also do it at the level as illustrated in our publication, showing how to prepare the different instruction programs, the same sort of syntax diagrams of the same kind of rules, but there is no reason why one should stay out and not permit people to get down one level from preparing the data to define the storage of data in the store, by means of the use of the syntax diagrams, whereby you can get them out and transfer them.

ANDEREGGEN - We also have another program called FLASH, for general plate- bending, plate-stretching and shell analysis; where the whole input is described in just two pages with a very compact syntax diagram of this kind; it' s just an elastic shell program, very general, and the whole thing is described in two pages.

SCHWARZ - I would like to give a short comment on this. The question was that every input user program should be standardized by such a type of syntax diagram. I am not quite sure whether this should be very good or not, since syntax diagrams are excellent methods which run in defined environment . . .
but if you have more introduction between the user and the program, this type of input should not be, I think, the best one at all. You might use many techniques to get input and direct the program what it should do, and so I think standardizing of these things should not be the best way to promote program usage at all.
I am afraid that syntax diagrams are rather frustrating for people standing away from computer usage and not familiar with the methods of computer programming.

BOKELER - Mr Schwarz, could we close this first statement? It is so important because you said these syntax diagrams are not usable for an average program. Can I say this in a short abbreviation? Syntax diagram for a normal batch program is too complicated: it might be better to use manual techniques, or, say, input sheets or something like that - is it correct ?

SCHWARZ - I just want to mention alternatives to your own interpretation. Expecially in the field of interactions, there are surely other methods, better than syntax diagrams.

ANDERHEGGEN - I don' t completely agree with you, because even when you use interaction time-shearing system, either the computer asks a lot of questions, the coordinates of point number one, and you write them, the coordinates of point number two, etc. ... Either you have to answer the question of the program by one single piece of data - which might be nice but I don' t know if it is the best way of doing it - or then you have some more less complex input statement, which have to be described somehow; so I think that a reasonable way of having a time-sharing input system is to write the whole line with a certain number of different data, and the syntax of this line can be described by syntax diagram, then the program checks the whole line and if it is bad, you have another line. The alternative, if you have a standard format, is this: you fill in the sheet with the pencil where your data have to come and then your secretary punches it. First, when you use a terminal, I think it is a very bad way of having the computer know what you mean by the position and the line and I think a Secretary can even understand our syntax diagram and what you mean by this syntax diagram.

BOKELER - Is there any other question or short comments? As you say, the only alternative to syntax diagram is coming to a formula; I do not think it is correct. There are several different methods.

TOMINO - In this discussion we all agree to come to a standardization in this problem; if this standardization is only on one page or two, it must be written, of course; if we are writing programs, we have to write also the manuals, we have to declare the used algorithms only if we accept that we can use such one or two pages for everyday' s use. The main result of this discussion is understanding the standardization problem. Maybe we come to some clearer and more concrete remark in future acknowledgments.

BOKELER - Another question on this statement should become towards standardization on the input form, on how to write a user's manual, how to write those sheets - is there any contribution to that? I am reading a paper which is describing an entirely different way of describing programs. I believe in syntax diagrams, it' s a marvelous thing, but I hate to think of reducing these as the only way of doing something. There are other ways for more complicated syntaxes.

DUTERTRE - Maybe we could simply say that we need a standardization of users' manual, and so we can understand others' manual. But we said we need standardization. Are we convinced that we are able to standardize? I think that

is a question. We cannot say we need this when we are not convinced we are able to do it. I am convinced we are able to do it from the experience in this field. People who have looked at the question of input have come up with the same answers, roughly, very compatible answers so that is a good point to me. There is a way to come out of this. It must not be the best way but we can look into it and surely there is an answer to it.

UHERKOVICH - I think it is not the question whether the standardization of the diagram is the only way or not, because each one is working, or a lot of people are working with these input diagrams; it would be very nice if they all used ours: writers are using the same language, the same tipe: this is standardization as I understand it.
Second, what I would mention is that we should tell who should do this work, who should make this standardization, because we can say: "ya , ya, this is very nice", but the real problem is who should do this. Another question: we hold an International colloquium here. Is there any tool by means of which we can transfer a program from one country to another one easily? This is a statement from Prof. Anderheggen too, who has written his program first for Switzerland. Is there any use for program transferring to another country, or is this a national standardization?

OBERNDORFER - The calculation of the reinforcement in the concrete here is quite different from the other countries, but the calculation of forces or deflections in a program should be done in a way that the program can be transferred from one country to the other one: however, the design of the concreting and of the steel members is a national problem, which it is not necessary to set up in a way that the program can be transferred.

FANELLI - Well, my comment to Prof. Takino' s paper is not really a question. I only wanted to give expression to a feeling of great admiration for the very advanced system that Prof. Takino presented to us. It seems to me that this is really an instance of how the possibilities of computers can be integrated in the fabric of an advanced society, an advanced industrial society, and how some of the questions that we are striving to answer can be answered in practice.
He in particular illustrated how the programs are reviewed by a specialized body, who in some way certifies them, and how the human appraisal is still put into the analysis in the end, when he speaks of the hand-written pages of comment at the end of the analysis. These are all important points and the solution that he illustrates is an example that these problems can be solved.
May be this is not a final solution, but it is a solution that has brought the work about, and this solution has taken into consideration, it seems to me, all the aspects that are important: the interaction with authorities, the interaction with codes of buildings, and so on. It shows the way, in a sense, to what we are looking for. So, I think that in this field, our Japanese colleagues are possibly one jump ahead of us and we could take all the possible profit from this experience to see how it was, what can be improved on it, if it can be transferred to another type of structure analysis besides building analysis and all things like that, and on all these aspects, I think we could have a very lively discussion.

KLEMENT - I would ask Prof. Takino how the telephone company does, as I think it is very interesting to hear that a telephone company gives computer calculations to so many engineers in the country. I have a question: are there other fields of computer calculations, may be commercial computer calculation, which this telephone company is selling ? Then, I want to know at what prices will the computer calculation be sold. Is there a monthly amount which is fixed to be paid for being with a terminal on the computer, are there other licenses for special programs which are used? I want to know if you are able to say whether this program system is a profitable one for the company.

TAKINO - Our corporation is a telephone company and has several kind of computer services. Many and many branches in Japan are subscribers to our system: exchange service and other offices of banks, medical agencies, distribution firms, etc. Our company is a half-governmental company , so we have not much money. We are going on well because we have many services, but in Japan, there are similar companies in competition: therefore, our service charge is not so high, it's a reasonable one. Subscribers pay a fixed charge, telephone and computing charge, however there is no difference if you use one or another program, there are no different licenses.

KLEMENT - Who pays these programs? And who pay their development and assistance?

TAKINO - Our company develops them and looks to the assistance of several software firms or manufacturers. The cost is included in computing cost.

HAAS - I have two questions to Prof. Takino of more commercial kind, about the reduction of manpower, which can be achieved when we use such structure analysis program. Here are my questions: how did you get these numbers ? Because if we have these numbers, you will promote the selling of those programs briefly. A second question: if we use those structural analysis programs, could we get, for example, a reduction of the reinforcement?

TAKINO - We investigated how a program may be used among subscribers by picking up 40 engineering services: we ask a question in each case and we distribute a form to fill in. Using your example, even if the solution is a real reduction, I suppose that normally they do not reduce the reinforcing bars, because first they choose on their experience and then they use our programs. Usually they do not repeat computer calculation to minimize the cost of the building.

TOMINO - We have run in the use of the system and also we have some contact with Mr Takino' s organization for the development of that program. If we were in a country (e.g. Pakistan) different from ours and we had to investigate a building from an antiseismic point of view following fixed rules, we could investigate it more and more in detail, using a well-chosen program. We can get some information and such a kind of calculation will give us some ideas and results. About Mr Takino' s mention of the advanced stage of the use of computers in Japan, I can' t be so optimistic as he is.

BOKELER - I think we must stop now. Thank you all for this very interesting discussion. We shall meet in the afternoon.

I SESSION

DISCUSSION

August 30, 1978. Afternoon.

Chairman: BOKELER (German Federal Republic)

FANELLI - I was willing to contribute an observation on the last remark made by Prof. Castellani about the possibility to extent his conclusions concerning two-dimensional structures. This is quite a detail point, and it is not much in the theme of our colloquium, but since the point has been raised, I think it worth while mentioning it. We made some non-linear analysis on cracked buttress dams, taking them as two-dimensional structures.
We studied cracks and made several analysis, some with through cracks and some with not-through and extended to the whole height of the dam, the apparent rigidity under hydrostatic load was almost unaffected in a two-dimensional situation, so valid deviations of the apparent stiffness begin to appear only with the completely through crack and cracks that extent at the whole height of the dam.
This could be an interesting indication, even in a two-dimensional situation, that agrees with your paper too.

BZYMEK - My comment on the paper of Mr Tagnfors is that the language presented in the paper is very clear and, as a matter of fact, not knowing the manual of this language, it is easy to understand this problem of oriented language. I think this is something similar to the idea presented by Prof. Anderheggen, because the language is divided in blocks, and this could suggest us to draw a syntax diagram from this program as well. As a matter of fact, about the way of presenting the interface between computing and design, I think that perhaps we would not be able to follow syntax diagrams, but at least we could recommend some criteria. One of these should be that the manual is presented very easy and, as far as I am concerned, I like very much the presentation of the manual based on the syntax diagram presented by Prof. Anderheggen.
At this point, I see some similarity between the paper of Mr Tagnfors and Prof. Anderheggen. This is the first comment of mine.
The second comment is on the paper delivered by Mr Oberndorfer from Austria. This was a paper which was very nicely presented, and , from my point of view, has some good conclusions, that means - it gave some criteria that should be met by good software, and it gave also some other points of concern, and I think this is very valuable.

LANG-LENDORFF - I want to come back to the paper presented this morning by Mr Takino. Mr Takino told us something about an evaluation committee working in his country. The task of this evaluation committee is as follows, I understood: to examine the different programs in order to be used in different calculations for housing and so on.
My question is: how does it work? Can you give us some features about working principles of this committee and how do you examine a program? You can't, in my opinion, divide the programs into good programs and bad programs. It all depends on what kind of examples you are calculating.

TAKINO - It is very difficult to examine the programs. The committee examines the programs in three or four meetings. At first, the committee examines manuals and the possibilities of their use by users; then they examine the manuals, and if there is some possibility that the user makes a mistake, the committee recommends to write things more simply in order to avoid misuse. Then, the committee proposes its data to inspect the correctness of programs. In Japan social contingencies have to be remitted to the building inspection officers; the building inspection officers have to understand and to examine the results of the programs. If these are not truly understood by the building officials, the committee recommends to quote the output.

BOKELER - If there isn't any other question, I would like to come back to the question if it is necessary to standardize the input documentation of programs and if we are able to specify the interface between computer and design structures engineering. If it is so, how shall we do the first step to do for this standardization?
The organization we have to go to, is it IABSE, is it EAB, is it FIP or the standard international organizations, or other institutes? In my opinion, all of us, must think about the answer.

FANELLI - I Would like to say a word of thanks to Dr Boekeler for his excellent chairmanship and now those who are interested in the visit to ISMES Laboratories can proceed to do it.