Session IV: discussion

SESSION IV

DISCUSSION

October 8, 1982 - Morning

Chairman: G. DEPREZ (Belgium)

J.P. RAMMANT - I do have a question to the last two speakers concerning checking programs. I have got experience that, for a non-linear analysis, I had to struggle against some Regulatory Commissions and they proved that my program was functioning wrong, because their program was functioning all right. Is this a valid method? I ask you because you didn't mention it for checking. Programs should not be used to check other programs. I think it is a serious question.

D.D. PFAFFINGER - May I have a short comment on that question? I happened to have the same experience. We were forced some time ago to verify a program by recalculating against another program, which has been accepted by the Authorities. We were just forced to do that. I think it is not a very good way to do it, because, if you compare the results from several programs, even in the simplest case, you usually get a slightly different result from every program. So I would say that it is not a good practice of checking. But there is little you can do about it.

M. KUWAGATA - Is your questioning point a way of checking a program? Usually we use a benchmark test, because there are many similar function programs in Japan, so we can compare the results on the same model of a structure. And that is a most reasonable way. A second way of checking programs is to chose a simple model. In that case we can get the result of mathematical analysis methods, so we can compare the result of analysis and program computation result. Of course there are many other ways, but these are most frequently used method for checking a program.

M. FANELLI - Speaking about program validation, or verification, I think that a way, that is both very sound in principle and very appealing to the practicing engineer, is the validation against experimental results, if the experimental results are properly obtained of course. And, in this connection, I would like to mention that it seems to me that there is a trend now in all branches of engineering, where experimental data are both precious for validation of theory (and so also computer programs) and difficult to obtain; a trend, say to try to establish, through International Organizations, international databases of experimental results, which can be put at the disposal of people interested, for instance, in validation on computer programs, trying to simulate the thing that is done by experimental results. This has been done, for instance, in the field of fracture mechanics that I know of, and I am sure it is being done in several other fields. At least in one instance, I am directly concerned, not in the field of civil engineering but in the field of hydraulic engineering. I feel that this could be a good way to share the knowledge and to promote workshops or benchmark tests to compare different programs and to validate them. If you do that, sometimes you get some very unsettling experience, even on very simple cases. Different programs give very different results and you discover that the programs

can be very sensitive to the basic models that are incorporated into them, or to the basic analytical procedure that you use, expecially so in dynamic problems. So I would plea for a wider diffusion of this practice on international databases of experimental results,connected with the practice of organizing international workshops,or benchmark tests to compare and validate computer programs. If someone would comment on that, I think it would be useful.

E.ANDERHEGGEN - To Prof. Fenves: I was impressed by the fact the you could convince the old people of the Committee to use computer programs and somehow to produce codes, if I understood you correctly. Could you tell us exactly how this program works, what is the input, what is the output? Finally you have to produce a text; do you have text editing facilities incorporated? How does the thing really work in practice?

S. FENVES - Concerning your first comment, if you show people that you can help them, they are usually quite willing to do so. Specifically, in the case of the AISC steel specifications, it was George Winter (whom many of you know through IABSE) who looked at our first document - a formal representation of the specification that was just passed in 1969 - and who came to me afterwards and said, "This was a nice dry academic postfacto exercize; what can you do to help us while we are drafting the specification?". Much of my work since then has been the direct result of that comment.
To the second question, we haven't started text writing yet, but I have a couple of students who are interested in doing that,purely as an exercise in artificial intelligence. All you need is a random number generator that generates a few simple variants on the sentence "should not exceed", "shall be less than","shall be limited to", and you can produce texts.
The program that is coming out of the U.S.National Bureau of Standards accepts descriptions of data items and of decision tables. For the decision tables, it can generate trees and identify missing, redundant or contradictory rules, and simply reports that result; the user can go back, add rules, change entries and redo the analysis. At the network level, it accepts, as part of the definition of the data items, the local precedence among them and assembles all of that together (it is a cannibalized CPM program, nothing more than that; a partial precedence ordering among branches of a directed graph) and checks to see if the graph is connected and is acyclic. If the graph is not acyclic, the program outputs the list of nodes that comprise a cycle. The user can look at it, go back, change the definitions and so on. For outlining purposes,you can ask for a display of a spanning tree. Any spanning trees of the network is an ordered sequence; all the links that are not in the spanning tree become cross references in the text, pointing to things that have been previously defined. You can always order the tree so that everything points one way, so that you don't have the usual shifting in the text where something will be defined fifteen pages later, while some other thing has been defined seven pages earlier. Finally, at the outline level, classifiers are attached to individual provisions, so that you can generate trial outlines, and change them around as you like it. That's all the program does at the present. As I said, some people are interested in expanding it into production of text.

J. BLAAUWENDRAAD - A question for Prof.Fenves and a second one for Dr.Pfaffinger. However firstly a comment. We heard that there were roughly two reasons to do the work that you explained us. One was that several States in the United States have so different codes, I understood (at least there are differencies, and you try to cover that) and, on the other hand, just to make your codes better and more complete. That would even hold if you had one code for all States, isn't it? I think we have similar problems in Europe but we try, stimulated by the European Communities, to harmonize our codes first and then try to get a better text. We have a committee which is studying this. You said you are in contact with CEB. I do not know if they are feeding in the European Community Committee, but it would be nice to have you in contact with that Committee. And my question is: these codes are growing and growing and get more and more detailed. There may be a danger that you make it easier to go even further in this way? What do you think about it? Do you stimulate it by using decision tables, or don't?

S. FENVES - That question has been brought up before. You can talk to any Committee member that has been through the exercise that we put them through when they ask us to cooperate with them, and they will be the first ones to vote for simplicity, conciseness and compactness, because we make them work twice as hard as they would normally work. Maybe Brook's chart is correct, writing a specification is much like writing a program. To do it our way may multiply the work by a factor of nine, as Brook indicates about programs. The people that we have contact with and that we have educated by us would be the last ones to be tempted to add more regulations.

D.D. PFAFFINGER - To Prof. Fenves. The final result will be written text to the public. Will you also give the decision tables to the programmers, who have to convert the written text again into Fortran statements?

S. FENVES - In the previous studies that we did, such as AISC69 and AISC81, the decision tables have been published as separate documents. At that time, we could not get the original committees to review and approve the tables, but a lot of people are using them directly as a source document for coding and a lot of students are using them. I understand that after a very complicated lecture on buckling provisions and columns in the steel specification, the students sneak down into the library and look at the decision table to find out what the lecture was about.

J. BLAAUWENDRAAD - Another question to Dr.Pfaffinger. I liked your presentation and I think it is very useful. My question is, is it all validation on the user's level when using a program? Or is part of it in fact a check on the validity of the program at the moment when it is brought to the market? A couple of things may be done, especially you plea for inserting automatic checks and things like that.

D.D. PFAFFINGER - Infact it's both of it. I would say: most of the validation checks have to be asked by the user, but the programs have to provide the means for those checks and things, that can be done automatically, should be done by the program; so the ball is also with the software developer. But as the things are now, most of the validation has to be done at the user's level.

H. PIRCHER - My question is how clear is the responsibility in case of bad mistakes and in case of big damage, and if somebody can find the mistake in the program and the calculation. How clear is the responsibility?

D.D. PFAFFINGER - The general position is that the only responsible man is the engineer. He has the final responsibility. There is no responsibility on the data center, or the software developer or whatsoever. The final responsibility is of the engineer and, if there is something wrong, he will be the one who has to defend himself.

H. PIRCHER - Is it clear also for the Law? In the case of a damage what will be the happening for the law?

D.D. PFAFFINGER - It is clear in ordinary cases, with the exception of gross negligence. If it can be shown that there has been gross negligence then you will be in bad shape, but that's usually not the case. But I am talking like a lawyer, I am not.

G. DEPREZ - It is sure that the engineer is responsible of results. He has to supply correct computation and good design. The informatic field has a responsibility of means. Now it is clear for everybody that it is impossible to give guarantee that a program is safe and reliable. At present we can be more sure, but not absolutely sure that a program is reliable. For the lawyer, from time to time these ideas became more clear, but it is sure too that all the firms which sell programs has to let know clearly every detail to their customers. If they don't use this normal way, they can be considered like people who wanted to sell something different from what they sell. They could have the responsibility to have not informed their customers on what they exactly sold them.

H. PIRCHER - What can we do, so that also persons out of the engineers community get this opinion?

M. KUWAGATA - As for the responsibility of computation results, when I read a paper of the proceedings of the colloquium held in 1978, I had a very strong impression. I feel sorry, I forgot the name of the author of that paper. It said that there is a famous sentence on the program manual face: "This program has been tested to the best of our knowledge, any responsibility in connection with the use of this program must however be declined". Such a sentence, or such philosophy is not valid in Japan now. But, anyway the responsibility is very clear, great and heavy to the program owner. Therefore, we and other big software developers, like Japan IBM, have the limitation of the responsibility to the computation charge.

H. PIRCHER - I have to say that all this is my opinion too, but I have some experiences that this opinion is not a common opinion to others. I have an example: we had to do with a very unexperienced client and we had to do calculation and, at the end, the calculation was wrong and stupid due to three mistakes. A first mistake was that the client prescribed nine prestressing cables for a box-girder bridge (and we know that the number of prestressing cables should be two, four, six, eight or ten, not nine); this was the first mistake. The second mistake was a normal mistake in input prepared by us and the third mis-

take was an error in praparing data for loading, done by the client. So we had three mistakes and the calculation was very very wrong, but the client needed three weeks to discover it and they said they had a three weeks work to be remade and it was necessary because our input data was wrong. So we had three mistakes, three weeks occuped and we needed insurance to fight against them. It was very difficult to manage the situation and I think the common opinion is not so clear, and I think that it is a problem for the user of programs and especially for smaller programs. That is also a problem for education.

S.J. FENVES - I have found that the opinion stated was very common. The case that you described about a court judgement in Japan, I cannot see how that kind of a judgement could be rendered by any Court in the United States, awarding damages based on a second party misusing, or not knowing, how to use the tools that he contracted to deliver. I cannot see that in a similar situation any Court would exonerate the designer from mistakes in the program.

D.D. PFAFFINGER - May I just say one sentence? I think that situation is not a question of opinion but a question of formulating a contract with the client correctly and you have explicitly to state what you are liable for and what you are not. Then you are covered in all ordinary cases with the exception of gross negligence.

G. SCHMIDT-GOENNER -If you are not able to use a pocket calculator you should not use it. You cannot make the man who sold you a pocket calculator responsible for the bad results you get out of it. I think, if you bring it down to that level, it should be clear who is responsible for the results of an engineering task.

P. LENGYEL - My problem is - I fully agree - that the user should have the responsibility for using programs, but, if we are thinking of the basic goals of structural engineering, then, by this mean, only one goal is achieved and I am thinking of two aims. The first one is to achieve all results always with less effort that is something we can achieve this way: by using computer programs and checking the results with traditional methods, with results achieved by traditional methods or by other programs. However, I am not sure we can achieve this way the other aim, which is to get more economic structures, that is that all new results during the development of new models will in any cases differ from the previous results and, if he takes this solution, it is not guaranteed. I think the way out may be what Prof. Fanelli has advised for achieving this aim and this would be partly a question also if I may quote it. Doesn't Prof. Fenves think that somehow the judgement of computer sooner or later, may be later, must be included and regulated in the standards because of the second aim?

S.J. FENVES - If you are talking about standards in the European terminology, definitely yes. If you are talking about standards - namely performance specification in U.S. and CIB terminology - the answer is no. What designers want in performance specifications is less and less prescription. A pure performance specification cannot possibly address the tool that you use to derive the results. A performance specification says the light intensity in this room shall be so many lumens per square meter. How you achieve that level, is nobody's con-

cern. Whether you use a computer to calculate that intensity has nothing to do with that performance standard. So, if you are talking about performance standards, there is no place in them for mentioning the computer. If you are talking about procedures or prescriptions on how to do things, then computation is definitely involved.