

## I N T R O D U C T I O N

Although the proceedings of the TAPSOFT conference are published in two volumes, there is, hopefully, no clear distinction between theoretical and practical contributions to this conference. In fact, there are several papers in both volumes which try to bridge the gap between the theory and practice of software development. Most of the authors, as well as the program committee members, are aware that there is such a gap, and most of us have already started to build bridges, but it is awfully difficult to reach the other side. I remember an excellent lecture by Jim Thatcher in Nyborg last year, where he explained his own difficulties in trying to bridge gaps within a big computer company, where the practical side was not only represented by people working in software development, but also by the management of the company.

Two years ago we began a seminar at the Technical University of Berlin to try and study the role of formal methods in software development. The participants were people from theoretical computer science working in the field of algebraic specifications, and people from software engineering partly using algebraic and partly using informal techniques for software development. Even in this small group of people, especially for Christiane Floyd and myself, it was most difficult to understand each other's terminology and main aims. But all of us were convinced that it would be most desirable to understand the aims of the other group and to launch joint projects.

The main joint project started at that time was the organization of this TAPSOFT conference. To begin with, we only wanted to combine the Colloquium on Trees in Algebra and Programming (CAAP '85) with a corresponding Colloquium on Software Engineering (CSE). It was Maurice Nivat who convinced us that we should have a third part, an Advanced Seminar on the Role of Semantics in Software Development, and Jim Thatcher, who enthusiastically took up these initial ideas which eventually led to this joint conference on Theory and Practice of Software Development.

After discussing the possible integration of CAAP into TAPSOFT with Maurice Nivat, Max Dauchet, André Arnold, and Giorgio Ausiello at CAAP '83 in L'Aquila, we were all convinced that combining CAAP with more practical aspects of computer science might help to bridge the gap between theory and practice. We all hope that at this 10th Colloquium on Trees in Algebra and Programming in Berlin 1985 people from theory and practice are listening to, and learning from, each other. This is only possible if all participants, especially the speakers, try to explain their ideas and results in simple terms and avoid the temptation to present too many of the technical details given in their papers.

However, I am aware that not all of the papers in this volume and not all of the corresponding lectures at the TAPSOFT conference will focus on bridging the gap between theory and practice. Actually, there is also a great need for the development of new theories in a mathematically precise and consistent way, without looking all the time at practical applications. Working in theoretical computer science, we have to make use of all kinds of mathematical theories, constructions and results in order to obtain a powerful theory from the mathematical point of view. But it is equally important that it should also be powerful from the point of view of computer science. This means the theory should be helpful in solving problems in practical computer science.

At this conference we want to focus principally on problems in software development. What are the main problems? How can they be stated precisely? Which of them can be better solved using formal methods and languages? Let us take a look at one specific question: In what way can formal specification methods and languages improve the software development process? Before we can try to answer this question, we have to make sure we know what kind of improvements we have in mind. What are the desirable concepts for software development, and how can these concepts be realized? It is difficult to get more than some informal ideas from people in software engineering in answer to these questions. Even basic notions such as "requirements specification", "design specification", "interface of a module" or "user interface" seem, as yet, to have no precise meaning in software engineering. They are interpreted differently in different papers, and even one and the same author offers a variety of interpretations.

For a theoretician, a simple solution in this case is to define all these notions as he pleases and start constructing his theory. Although this sometimes seems to be necessary, in most cases it turns out to be the wrong way: Theory and practice then diverge. It seems to be much better, although sometimes frustrating, to keep in contact with people in software engineering during the development of the theory. Similarly, it is advisable for people in software development to remain in contact with the users during the development of the software system in order to integrate this properly into the user environment.

In addition to the problems mentioned above, there are even bigger ones when considering the development of software for distributed systems, including aspects of nondeterminism and concurrency. On the one hand, there are powerful mathematical theories for concurrency and nondeterminism, using techniques from topology, modal logic, algebra and category theory. On the other hand, there are complex languages including concurrency, like ADA, and even more complex computer systems and networks which have been installed, although nobody is able to understand these systems in detail or even prove their correctness. Certainly, the need for interaction between

people in theory and practice in such fields is even more important than in other areas which are at present better known.

Last but not least, several of these problems, especially the development of software which may be used for the control of critical processes or military purposes, should also be discussed with respect to their social impact.

We hope that this conference also constitutes a lively forum for discussions of this kind.

Berlin, March 1985

Hartmut Ehrig