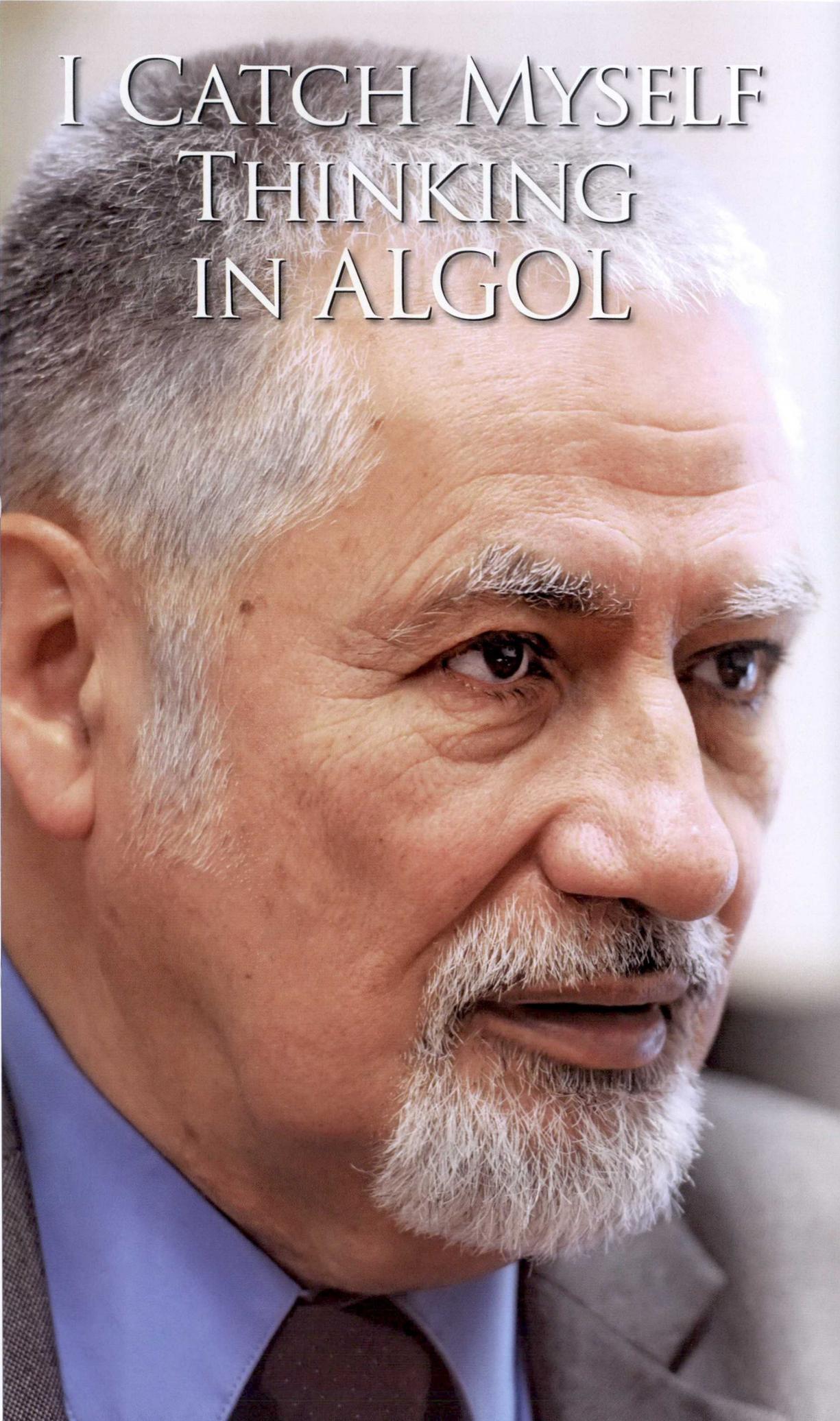


BRIEFLY SPEAKING

ACADEMIA

I CATCH MYSELF  
THINKING  
IN ALGOL



**B**ack in the 1960s, as a 21-year-old, he managed the first gigantic computer in Poland. Now the young people he gives win prizes in prestigious programming contests. We talk with **Prof. Jan Madey** about the history and future of computer science.

**ACADEMIA: Your field is software engineering. Which means what, exactly?**

JAN MADEY: Software engineering is about attempting to apply the methods of classical engineering to computer programming. For example, if we want to build a house we first work out a detailed blueprint, then construct it strictly according to the design, and afterwards double-check that everything has been done right. In computer programming, however, the “material” we are working with is more ephemeral, so it might seem like anything can be easily modified. But it’s like having a client suddenly changing his mind as his house is already being built – now he might want windows on the other side, now three floors instead of two. Even when large-scale IT systems are being developed, the requirements and needs are often not formulated very precisely.

The term “software engineering” first appeared back in 1968, when people began to realize that failing to approach programming in a way analogous to classical engineering was causing trouble, because already then the systems were getting very complex. Now, too, one frequently hears complaints that programmers have botched this or that job, but in fact it is generally not their fault, because what they were asked to do was not laid out precisely enough. We should remember that natural language is ambiguous and describing a problem in merely “colloquial” terms is not enough – generally there has to be a great deal of precision.

At the same time, IT systems are quite diverse. Some are mission-critical ones where very precise methods are needed, as people’s lives or wellbeing hinges on their proper operation (e.g. in medical or military applications). Here the description needs to be laid out in very great detail. But others, like a system for university administration, for instance, have a lot of elements that may be in constant change. Here the alternative approach is useful, where the process of IT development proceeds through continual dialog between the client and designers, and later the programmers.

We therefore have two vastly different approaches. These are very difficult, very much current issues. And they will be around for a long time, because technological change always necessitates a new way of thinking.

**New on what level?**

A generalization known as Moore’s law states that the performance of computer technology doubles about every two years: sizes shrink by half, processing capacities increase twofold, etc. This incredible progress is always

urging software engineers forward, because technology is constantly opening up new possibilities.

We should remember that on the lowest level, software is in fact still nothing but rows of zeros and ones, represented in the memories and registers of a computer. The first programs were indeed actually written in the form of such rows of finite length. Mistakes were very easy to make, and finding an error or demonstrating correctness were very difficult. Now we use programming languages and systems that conceal the true structure of software from us.

My first love was programming languages, not software engineering, because it did not yet then exist. ALGOL 60 was my first language, and I knew it like the back of my hand. I used to say in jest that when I woke up in the middle of the night, I would catch myself speaking ALGOL rather than Polish. The first three textbooks of ALGOL 60 were published in 1965 – one of them written by me.

**You graduated from what was then the Faculty of Mathematics and Physics at the University of Warsaw – now well known as “MIM” (the Faculty of Mathematics, Informatics, and Mechanics). So everything starts with mathematics?**

We should first point out that the notion of “informatics” (or “IT”) is not really very precise. It stretches across a broad spectrum, including research, applications, and various practical skills. Programmers (or “IT guys,” as they are colloquially known) are expected to be able to design a huge system, to write a well-functioning application, to have perfect knowledge of the latest version of system X, and be able to fix a printer. But there really are no such all-rounders. Also, on the research end of things, it is impossible to identify where computer science ends and mathematics begins. Some people start studying mathematics, then begin to be fascinated by theoretical computer science. The basic difference is that a true mathematician develops mathematics for mathematics’ sake. Without asking “what is this for?” – because that’s not the point. A computer scientist, on the other hand, even a theoretical one, is always thinking about the purpose of what he is doing (although perhaps subconsciously).

**How did it transpire that when studying math, you took an interest in computer science?**

I had mathematical inclinations while at school, though I was drawn to practical things. At Reytan

High School in Warsaw, I had very good math teachers who instilled me with a passion.

At university things were going well for me, although I was involved in many things at the same time, including a lot of sports. I liked mathematics, but I was still drawn towards practice. After the second year, a new opportunity appeared at the faculty – a department working on numerical methods, which I signed up for. There I had classes with, and got to know, Prof. Stanisław Turski (then Rector of the University of Warsaw).

His eldest son, Władysław M. Turski, an astronomer by his initial education, had taken an interest in computers. He found out that the Danish company Regnecentralen, employing excellent programmers and manufacturing a computer called GIER, was thinking of expanding into Central and Eastern Europe. Turski contacted its director and arranged for one GIER model to come to Warsaw for an exhibition in autumn of 1963. As a result, two individuals were to be sent to Copenhagen for a month-long internship. The obvious candidate was Dr. Andrzej Kiełbasiński, who ran the numerical methods department, but to my great surprise the other person chosen was me – then still a fourth-year undergraduate. Rector Turski evidently decided to invest in me.

That was a very successful visit, during which we learned a lot, got to know the top computer scientists in the company, and gained our first experience in programming the GIER. When the GIER made its appearance at the Palace of Science and Culture in Warsaw in November 1963, I was responsible for everything. The computer was made available to the University for research, practically 24-7. It was mainly used by astronomers, chemists, economists, physicists, and mathematicians. I was managing everything, and at the same time was studying at university, writing my master's thesis, being a host for visiting Danes...

#### **...you were then 21 years old.**

Yes, it was not easy to keep it all together. In January 1964, a decision was made at a very high level to purchase the GIER (for about \$140,000). After the exhibition, it turned out that I was the sole holder of the key to the room where the computer was kept. Our technician went to Denmark for training, and it was clear that no one could use the computer without technical assistance. No one but me. Then it was my own "personal computer," like today's PCs, only of a very different size.

It was beautiful. It looked like a piece of furniture, a three-door wooden wardrobe. The monitor was an electric typewriter. Paper printouts were the counterpart of what we now see on a screen, and the keys of the typewriter were the keyboard of the computer. Aside from that, there was a very fast paper-tape reader. No better one was ever made. It read 2,000 characters (or 5

meters of tape) per second. On 4 April 1964, not yet 22 years old, I passed my master's degree exam. The same day I was hired as a research assistant. Officially, on 1 February, the university's first-ever computer center (ZON) was created, in the form of an ancillary unit that could earn money.

The profits, if there were any, were divided along the following lines: 70% for buying research equipment, 20% for social spending, and 10% for bonuses. We earned a fortune. We set the price of 3,000 zlotys for each hour of use of the computer, or 5,000 if we were also doing programming work. For many years, we were like a goose laying golden eggs. That 70% went to fund hardware for many units of the university, especially chemistry and physics. Starting on 1 May 1964, still not yet 22 years old, I was made deputy head of the ZON computer center, with the university Rector as its head. I got myself in deep very early on.

#### **Did things go easily for you?**

No, there were many different problems. For example, the paper tape for the reader had to be of high quality. Such tape was only manufactured in one place in Poland in those days, but the paper was not good enough. In exchange for my ALGOL textbook, I received a dozen packages of tape from Denmark, but of course that was not a long-term solution. I was relieved only when Foreign Trade Minister Witold Trąpczyński personally called me up to tell me that he had managed to arrange a steady supply of paper tape.

For the Poznań Trade Fair in 1965, Regnecentralen hired me as an expert. At the official opening, Rector Turski and I were at the GIER stand. A government delegation was passing by, led by Prime Minister Cyrankiewicz. Cyrankiewicz broke off from the group, came up to us, and asked whether this was the machine that the Rector had fought so hard for. That shows on what level the decision had been made.

In 1975, the Institute of Mathematical Machines at the University of Warsaw merged with the Numerical Calculations Department to form the Institute of Informatics. Prof. Stanisław Turski was the first director, and when he retired in 1976 I briefly served as acting director. Then Prof. Władysław M. Turski took charge of the Institute, and I was made deputy director. In 1984 I took charge myself, and remained director – except for a two-year stay in Canada – until I became vice-rector of the University in 1996.

#### **How did you find time to write your doctorate?**

In 1967 I went to the UK on a British Council scholarship and had the chance to devote myself entirely to research (at the University of London). Soon thereafter I started working hard on my doctorate, on the topic of grammatical inference. To put things really simply, this is an attempt at modeling how a child acquires natural language. A child is not taught grammar, but acquires



One of Prof. Madey's keepsakes: a star made out of the perforated paper tape that was used to carry data and programs written in ALGOL

it through examples and counterexamples (being corrected by parents). I decided to try to have a computer “learn” in the same way.

I was given permission by the Faculty Council to be the thesis advisor for selected master's students, while myself still just a master's degree holder (something that was unprecedented and, I think, has never happened since). Five theses were written under my guidance. I worked out the methodology, and my master's students implemented selected fragments (modules) of the emerging software. My objective was to have a working system, not just a theoretical solution, and managed to do so. I defended my doctorate in 1972.

#### **You were already then working with foreign researchers.**

At the International Federation for Information Processing (IFIP) congress in Ljubljana in 1971, I met the American Jim Horning, who had somewhat earlier done his doctorate on grammatical inference. He became not only my friend, but in a certain sense my thesis advisor. He wrote me letters of recommendation for my first trips to the United States, then together we carried out a big Polish-American-Canadian project. I kept in touch with him until his death. When I got my doctorate in 1972, I knew about everything that had up to then been done in the world in the field of grammatical inference. Such work was not being done on it in many places: the United States, China, and Poland.

In 1973, I decided it was time to sever the ropes holding me down and go somewhere again to devote myself to research. I had a few interesting offers in foreign countries. The Technical University of Munich proposed a very interesting topic related to operating systems, so I chose it. When I arrived in Munich in autumn of 1973, it turned out that the only specification for what I was meant to work on had been in the letter of invitation. I was once again jumping in at the deep

end. But that proved beneficial. I managed to develop the topic and get to know the world's leading computer science gurus. Initially I worked myself, then I was given two people to collaborate with, and we initiated a system that was both theoretical and practical – the OS Kit (Operating System Demonstration Kit).

Returning to Poland, I developed the topic well enough that 19 master's theses and a few doctorates were written on it. I myself got my higher doctorate (habilitation) for work on that subject.

#### **Did you build a strong team immediately after returning?**

It consisted exclusively of my master's students. The subject became interesting enough that I was invited to the United States. In 1979 I went to the University of California, Santa Cruz. After martial law was introduced in Poland, that stay was prolonged to two years. There I mainly taught about operating systems and programming in Pascal. Things went so well that I was even in the running for “professor of the year” at the University of California. There I came into contact with the Wang Institute in Massachusetts, where I later spent a year working.

In 1989 I was intending to go to Santa Cruz again, when Prof. David Parnas arranged for me to go to Canada instead. He was a very well-known computer scientist who had gained his reputation in the United States back in the 1970s, but after famously coming out against President Reagan's “Star Wars” program he moved to Canada and had just completed a very interesting project related to the correctness of the systems designed to shut down nuclear power plants in the event of an accident. I worked with him for two years, with interesting results. They served as the basis for a Polish-American-Canadian project that gave rise to a big lab at the Institute of Informatics in Warsaw. The initiator of the project was Jim Horning, who I have

already mentioned, then working for Digital, which provided nearly half a million dollars for this.

I continued to develop the topic, taking to Canada various junior associates of mine, as undergraduates, then master's degree students, then doctoral students. And I myself earned the title of professor of engineering based on this work. My main results were not mathematical in nature, so I contributed not to mathematics but to computer engineering, where there is a hidden application.

**Simultaneously, teaching has been a passion for you the whole time.**

I have collaborated with young people at every stage. In 1982, when I came back from the States, I was found by Ryszard Rakowski, who in 1981, still under martial law, had persuaded Prof. Jan Szczepański, a member of the State Council, to create the Polish Children's Fund. With time, it focused on particularly gifted children. Ryszard Rakowski had very powerful charisma. I have been involved with the Fund for 30 years, and for more than a decade I have served as its chairman. Through the Fund's efforts, 21 years ago Poland began to take part in the European Union Contest for Young Scientists (EUCYS). I was made head of the jury in Poland, and began to travel to accompany Polish youths to the European finals. Now I am the Polish organizer of the competition. I have been accompanying the talented young people for 18 years, every year. Poland is second to only Germany in terms of prizes won. We really take the competition by storm, and this year was a particularly successful one. All told there were 35 countries taking part, there were 104 projects, and all the coun-

tries received at most three prizes. With one exception: Poland won six!

**How do you identify young talents?**

The Polish Children's Fund deals with school-aged children talented in all fields. Hard sciences, social sciences, humanities, ballet, music. Recruitment takes place every year, and only for one year. The winner does not receive money, but expert guidance and support. To continue participating in the program, they have to compete every year. The Fund's special event is a two-week camp at which there are lectures and workshops. I am always there. It used to be for the whole duration, teaching IT classes, but now I only go for a one-time lecture, albeit on a much broader topic. Some former participants, whom I got to know as 13- and 14-year-olds, are now professors.

I am really pleased that they feel no need to emigrate. Globalization demands and makes it easier to be a "citizen of the world." One has to travel. I myself spent about 10 years abroad, but my place is in Poland, my roots are here. It is here that I try to make something of it all. I really want these young people to do the same, and I am pleased that many of them indeed do. They are basing their careers in Poland – they spend years abroad, are well-known in the world, publish in the best journals, get invited to the best conferences, win world-class grants, but they identify with Poland.

**You are also known as a promoter of programming competition winners. How did that get started?**

In 1994, I found out that there was a university programming competition being held in Amsterdam. I got in touch with the organizers and was given a few days to organize a three-member team. Who did I turn to? Of course, to former participants of the Fund's program, who were then studying at the Faculty. I picked three of them and they went off not knowing anything about the competition, and they won by a knockout! The rules were that there was one computer, three team members, five hours, and about 10 tasks formulated as stories (in English).

And that is software engineering in a nutshell: there is a problem specified in an imprecise way, i.e. in natural language. There is cooperation, because the participants work as a team, and also limited resources, because they have only one computer. There is a tight deadline and working under stress, like in real life. This year, too, we have made it into the world finals of the competition, for the 22nd year in a row – no other university in the world can boast of such results!

I have contributed a bit to this, because I know how to work as a team. I invest in young people the way Prof. Stanisław Turski invested in me: by giving huge responsibilities, but also trust.

INTERVIEW BY

ANNA ZAWADZKA AND AGNIESZKA POLLO

PHOTOS JAKUB OSTAŁOWSKI

**Prof. Jan Madey** is a computer scientist, affiliated since the start of his career with the University of Warsaw. He was deputy head of its first-ever computer center, then a co-creator and longstanding director of the Institute of Informatics, later vice-dean of the Faculty of Mathematics, Informatics, and Mechanics and vice-rector of the University of Warsaw. He is currently the Rector's Deputy for Multimedia Education, the chairman or member of several scientific councils and program councils of different institutions. He has also worked at top centers in Europe and North America, including the University of London, the Technical University of Munich, the University of California Santa Cruz, the Wang Institute in Lowell, Massachusetts, Queen's University in Kingston, Ontario, and McMaster University in Hamilton, Canada. He authored the first textbooks in Poland on the programming languages ALGOL 60 and Pascal. For years he has been working to provide guidance to exceptionally talented children and youths. He is the Polish organizer for the European Union Contest for Young Scientists (EUCYS), chairman of the Polish Children's Fund, and also coach of the teams of students from the University of Warsaw who have for years advanced to the World Finals of the ACM International Collegiate Programming Contest. He has been decorated with the Officer's and Commander's Crosses of the Polonia Restituta Order, and with the Medal of the Polish Minister of National Education.