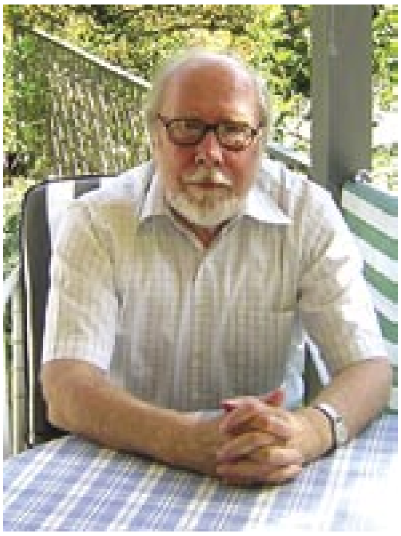
Conversations on the History of Informatics (I):

An Interview with Niklaus Wirth

Informatics is a young science and during the first decades computing scientists were too busy with the basics to have time for reflections. Informatics seems now to come to its adolescence. It is getting more and more independent from its many parents such as mathematics, systems theory, electronics and even economy, and it starts to pose questions concerning its roots, its deeper sense, and its role in society – in short, it starts to concern itself with its own history. However, while many agree that such a history is important, few agree on what informatics is and how its history should be investigated.

Therefore, the OCG (Austrian Computer Society) and ÖGIG (Austrian Society for History of Informatics) have decided to conduct a series of interviews with outstanding computing scientists, asking for their views on both the general issue of the history of informatics and for their personal reflections on both the past and the future role of informatics in our society.[1]

The following interview with Niklaus Wirth is the first of this series.



Is Informatics old enough to start discussing its history?

The word informatics is itself rather young, dating back to about 1970 and created by Dreyfuss. It's a contraction of "traitement d'information automatique", an expression which emphasizes that something automatic is going on. Using this sense of the word, informatics would exclude anything before automatic machinery. For me 'automation' implies electronics, so I would leave out the mechanical calculating machines of the 19th and early 20th centuries. With this definition, a history of informatics would begin with electronics, or perhaps with the electro-mechanical computing machines of the 1940s. We are looking at only 60 years, but we move during that time from an unwieldy tool for a handful of mathematicians and engineers to a digital 'context' which frames the global economy and, increasingly, western society. If we consider history to be a study of change, then I'd say we have sufficient material.

But it is material that many of us have lived. Would a formal study bring anything new to the table?

If you are living within a particular 'context', you may not be able to see this context without going outside of it. Much of today's world lives outside the umbrella of the computer, networks, or even the telephone. We tend to forget this because we are so firmly positioned within the paradigm of informatics as a way of life. In looking at the history of informatics, we step outside this context and look at ourselves.

What would you include within the scope of a history of informatics?

The computer itself is important but the computer alone does not define informatics. There are many other dimensions. The way the computer is used also tells a story; in the 50s the computer sat in a laboratory and a scientist would sign up days in advance to have this huge machine all for himself. Then there is the user community which has mutated from the military and the atomic research community of the 50s to the people today who are telephoning, playing, studying, and working. The scope of application has moved during this time to embrace robotics, games, shopping, even the opera. Advances in language design have allowed for abstraction, which of course defines the boundary between virtual and real. And so on. Informatics history deals with the co-evolution of the computer and these other dimensions.

What was the first step from the early computer towards this world of informatics?

I would definitely say reliability. The vacuum tube-based computer had a meantime between failure in the order of hours: each (high quality) tube had a MTBF of about 10'000 hours. Hence, in a computer with, say 5.000, tubes, some tube would go out every two hours and with it the entire computer. This meant that a program had to have a runtime of less than two hours if you were to have a reasonable chance of its completing before the next breakdown. The introduction of the transistor, around 1960, changed all this.

But the transistorized computer remained the plaything of a small population of specialists for quite some time. What was happening?

For one, many were learning to program. A new kind of (written) language was needed and this has taken a long time and is still in development. The computer was a tool which demands great attention from the tool-bearer. To the degree that the programmer has to focus on the underlying computer's details, he is distracted from concentrating on the task at hand. A good language provides a better platform, computing model, abstraction. For a long time computer science remained the domain of the engineers who built the machines, and of the mathematicians who used them. Programming languages

[1] *Thanks go to Prof. Laszlo Böszörmenyi, former student of Wirth, for the idea and impetus to set this series of articles in motion.*

were the first topic that came from neither of these two camps. Among languages, Algol 60 was the first with some measure of mathematical formalism and rigor. This layer of abstraction was a catalyst for programming methods, language design, and algorithms. These topics attracted new interests, users and ideas about what to do with a computer.

Who was driving the research agenda?

Initially it was the military and then business. Despite the interest and involvement of a new community, computer science remained for some time an academic stepchild; it was in the house, it was useful, but it had no status. The programmer was homeless, commuting, as I did at Berkeley, between the engineering and mathematics buildings. But in1965 Carnegie-Mellon (Pittsburgh), MIT (Cambridge), and Stanford (Palo Alto) each conferred upon computer science the status of an academic discipline in its own right. Each of these universities established an independent computer science department, thus giving recognition, official status, attention and financing to a sector much broader than computer technology per se. This meant a tremendous shift of focus.

But it took many universities, including the ETH in Zurich, another fifteen years to grant full recognition. How were you impacted?

The chain reaction had started. With or without a computer science division, the informatics community at ETH and many other universities and research institutes coalesced around the changing agenda.

Somewhere we moved from a closed community of computer specialists to a global community of informatics aficionados. What factors and players brought us to the tipping point?

Xerox PARC's Alto appeared on the scene around 1974 and this powerful workstation eventually spawned the ubiquitous personal computer, although the PC of 1980 was still a long way behind the Alto. PARC with its Ethernet and the DoD with Arpanet were driving forces in the development of networks. Now it has become difficult to find a research project in computing that does not involve networks. These developments profited from advances in language design, but also contributed to it. For example, I introduced Pascal in 1970, but Pascal attained its real acceptance after 1977. This was due at least in part to the personal computer. Along came a generation who had ready access to computers and, most importantly, who had not to unlearn old habits, whose first interface to the computer was a high level and structured language. This was the first generation which was free to focus on what you could do with the computer, rather than on the computer itself.

Where will informatics take mankind? Are there important (historical) lessons that we should keep in mind?

People believe that they can no longer live without the mobile telephone, the PDA, the internet, computer-based entertainment. The computer has simply become a part of the fabric of life, just as mobility has. But how many people understand the relationship between mobility and our dependency upon fuel resources? The capability of mobility has been abstracted from the physical world. Rather than deal with the abstraction, people can choose to simply live within it. So too with informatics. I often hear people proudly boasting that they know nothing about computers, or about

Niklaus Wirth was interviewed by Ann Dünki, a former Ph.D. student of him.

physics. But a sound education in natural sciences is essential to an understanding of our world; how does a gear, a pump, an electric motor, a wing, a gas turbine, a computer work? What are its underlying principles and laws of nature? The computer can provide capabilities and information, not more. But it seems that many people believe that the computer makes understanding superfluous, "because you can look it up in the Internet", if needed.

Would you pose a question to the readers?

Is the computer a food or a drug? Do we use it to stimulate the way we teach, deal with, learn and think about the increasing abstractness of our world; or do we use it as a sedative, to make us drowsy, and unaware of the real world around us? The choice will be a historical one. ■

Niklaus Wirth is one of the most influential scientists in the area of informatics. He made essential contributions to Algol-60, and thus to the establishment of informatics as a science. He designed a series of programming languages. Early works on Euler, PL/360 and Algol-W were followed by the extremely successful and significant Pascal language. After Pascal and a longer visit at Xerox PARC in California, he devoted himself to the efficient coupling of hardware, language, compiler and operating system design. This led to the simple and clean programming languages Modula(-2) and Oberon(-2), the Lilith and the Ceres computers and the Oberon operating system. Wirth is one of the most quoted authors in computing science. Many of his books and papers belong to the evergreen classics, one of the best known being Algorithms + Data Structures = Programs, which helped generations of students to understand how algorithms really work. He was honored with numerous awards, among others, in 1984, with the ACM's Turing Award, the highest distinction a computing scientist can achieve. Wirth has retired from his professorship at the ETH Zürich, lives in Zürich, actively follows the development of informatics, and reflects on it with a critical view – not without his fine sense of humor.