

In memory of Ole-Johan Dahl and Kristen Nygaard

The Summer of 2002 has been particularly somber for the history of software in general and object technology in particular. Ole-Johan Dahl passed away on June 29. Only a few weeks later, on August 9, his old friend and colleague Kristen Nygaard died suddenly. In-between, as JOT readers have probably learned, another great pioneer, Edsger Dijkstra, lost his battle with cancer.

Dahl and Nygaard are the indisputed founders of object technology. Their collaboration in the nineteen-sixties led to the Simula languages, culminating with Simula 67 which created a whole new discipline, even if it took another twenty years for the industry at large to understand it. The inspiration of these two pioneers is without peer; the whole field – JOT included, of which Kristen Nygaard was a founding Advisory Board member – would not exist without them.

In this special section of JOT we pay homage to these intellectual giants. A particularly moving text is the eulogy that Nygaard himself composed in memory of Dahl shortly after his friend's death. We are grateful to Tony Hoare, Olaf Owe and the family of Kristen Nygaard for help in obtaining this text and permission to publish it here. The text brings out, in an often humorous way, some of the spirit of the collaboration that took place when everything was to be built. Some of us wish they had been in the room.

Next, Ole Lehrmann Madsen, a close collaborator of Nygaard, has kindly accepted to write an obituary for JOT. He recalls both the personal and scientific aspects of Nygaard's career. This touching text will make Kristen come alive for the readers of JOT who have not had the privilege of approaching him. I added a few reminiscences of my own about Kristen.

Note that a Web site devoted to the memory of Nygaard has been set up at http://www.ifi.uio.no/in_memoriam_kristen/

We mourn these two towering colleagues and hope JOT readers will benefit from these testimonials.

Bertrand Meyer
JOT publisher



Dahl and Nygaard at the time of Simula's development

Ole-Johan Dahl

A eulogy by Kristen Nygaard



Ole-Johan Dahl

Professor Ole-Johan Dahl, University of Oslo, Norway, died on 29 June, only 70 years of age. He was diagnosed with cancer four years ago, but the disease seemed not to be life-threatening at first. Last fall, however, the cancer took a turn for the worse.

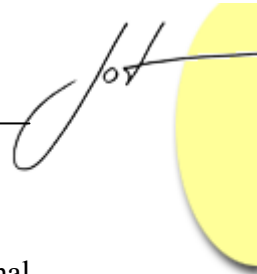
Ole-Johan Dahl is a member of the small group of scientists who will be regarded as founders of their own field of science. His field was informatics (US: computer science) and he won the two most prestigious international prizes specific to that field: The ACM A.M. Turing Award and the IEEE John von Neumann Medal. He was made Commander of The Order of Saint Olav by the King of Norway in 2000, and received a number of other signs of recognition.

When it became known that Ole-Johan Dahl was seriously ill, he received letters and greetings from researchers all over the world, and also from professionals who wanted to thank him for the programming tools they use in everyday work.

The British scientist C.A.R. Hoare – another founder of informatics – wrote this about Ole-Johan Dahl: “He is someone that I most admire as a scientist and as an educator and as a person. His whole life is a model of how life should be lived, right through to the end.” The US scientist Dave Parnas wrote about the Turing prize: “Finally, they have given this prize to people who have really made a difference. In more ways than I can ever explain, your work has changed the way people think about software and write about software.”

I got to know Ole-Johan early in the 1950s. We both worked at the Norwegian Defense Research Establishment, he at the “Computing Office” with me as his supervisor. Both of us were under the direction of Jan V. Garwick – the gifted and eccentric researcher who must be regarded as “the father of informatics” in Norway. Ole-Johan turned out to have an exceptional talent for programming. In addition he developed other characteristics suggesting a career as the absentminded professor. For this reason some of his friends were astonished when he won the hand of Tove, an exceptionally vital and wise wife. They had no reason to be, since Ole-Johan on closer acquaintance turned out to possess a warm and very vigorous personality. Their home, with two children and three grandchildren, became a meeting place for cultural activities for friends of Ole-Johan and Tove, to pursue their common interest in music.

He had a strong sense of humour, and at the same time he could be very outspoken in discussions – and that is an understatement. We shared the interest in music. He was an excellent performer; he once considered a career as a concert pianist, and was active in the chamber music life in Norway. As for myself, I was only a passionate listener. One morning after Ole-Johan had slept overnight at our home, my wife told me that she was unhappy because Ole-Johan and I had become enemies the evening before. I was



shocked, and had to explain to her that she had listened in on a perfectly normal conversation about a composer whom I loved and Ole-Johan at that time regarded as a charlatan.

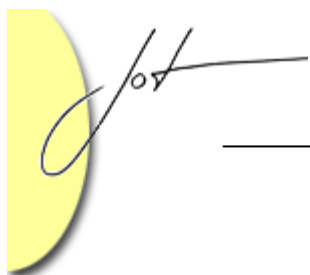
Another true story from the 1960s at the Norwegian Computing Center, when Ole-Johan and I developed the Simula languages and thus object-oriented programming: A new employee came running down to the switchboard office and shouted that two men were fighting in front of a blackboard on the first floor. The switchboard operator stepped out into the corridor, listened for a few seconds and said: “Relax! It’s only Ole-Johan and Kristen discussing Simula.”

In 1973 came the famous book “Structured Programming” by Ole-Johan Dahl, Edsger Dijkstra and C.A.R. Hoare, which had an immense impact upon the teaching of programming. From the 1970s on, the possibilities of proving the correctness of programs interested him most. In this field too he became an important researcher.

In 1968, Ole-Johan Dahl was appointed to the first professorship of Informatics in Norway. He built a team of colleagues who together with him created a high quality education in the subject. He also built a Department with an exceptionally pleasant human atmosphere, influenced by his generous personality. Few teachers are loved by colleagues and students to the degree that Ole-Johan Dahl was.

As for me I have very much to be grateful for. But, even if all of it is important, it is the sharing of research, the new challenges every day, the mutual inspiration, the steady building of new insights, and the joy of understanding that count most when I think back on the fifty years of our friendship.

Kristen Nygaard



Kristen Nygaard

by Ole Lehrmann Madsen

Professor Kristen Nygaard, University of Oslo, died suddenly of a heart attack on August 10. He was 75 years old. His sudden death was a great shock to his family, friends and colleagues. Although he had retired from the university he was still working hard on his new COOL project and during the last months he has been giving lectures in the US, India, and several places in Europe including Malaga. In Malaga he gave a keynote speech at the dinner party at ECOOP 2002 and I am sure that everybody attending will remember this talk. He was often able to say something deep and interesting and at the same time be very entertaining. Several spouses told me afterwards that this was the first time they enjoyed a talk at a conference.



Kristen Nygaard at ETH Zürich in September 2000

Life is full of strange coincidences. Just a few weeks earlier, his long-term collaborator Ole-Johan Dahl had died. In this issue of JOT there is an obituary for Ole-Johan Dahl written by Kristen Nygaard. Much the same can be said about Kristen as he writes about Ole-Johan. Kristen was a pioneer of his field: Kristen and Ole-Johan are considered the fathers of object-oriented programming, Kristen established what is now known as participatory design as a research field, and he is one of the few philosophers in informatics.

For his work he was awarded the ACM A. M. Turing Award, and the IEEE John von Neumann Medal. The American association Computer Professionals for Social Responsibility awarded him its Norbert Wiener Prize for responsibility in social and professional work. He was awarded an Honorary Fellowship by the Object Management Group. The King of Norway made him the commander of the Order of Saint Olav, and he was an honorary doctor at the Universities of Lund and Aalborg.

Kristen's main contribution to informatics was the development of Simula and object-oriented programming. His original field of research was operations research and he realized early on that computer simulations would be a useful tool in this field. He then made an alliance with Ole-Johan Dahl who – as Kristen writes – had an exceptional talent for programming. This unique collaboration then led to the first Simula language, Simula I, which was a simulation language. Dahl and Nygaard quickly realized that the concepts in Simula I could be applied for programming in general and as a result they



designed Simula 67 – later on just called Simula. Simula is a general-purpose programming language that contains Algol 60 as a subset.

For many people it is still surprising that Simula contains many of the concepts that are now available in mainstream OO languages such as C++, Eiffel, Java, and C#:

- Class and object. The class concept as a template for creating instance – objects.
- Subclass. Classes may be organized in a classification hierarchy by means of subclasses.
- Virtual methods. A Simula class may define virtual methods that can be redefined in subclasses.
- Active objects. An object in Simula may be the head of an active thread – technically it is a coroutine.
- Action combination. Simula has an inner-construct for combining the action-parts of a class and its subclass.
- Processes and schedulers. It is easy in Simula to write new concurrency abstractions including schedulers.
- Frameworks. Simula provided the first OO framework in form of Class Simulation. The simulation features of Simula I was made available through Class Simulation.
- Automatic memory management. Simula had automatic memory management, including garbage collection.

Most of the above concepts are now available in most OO languages. One exception is the Simula notion of active object with its own action sequence, which strangely enough has not been adopted by other languages. For Dahl & Nygaard, it was essential to be able to model concurrent processes from the real world when making simulations.

The ideas of Simula have been adopted over a long period. Before OO caught on, Simula was very influential on the development of abstract data types. As a result of the new ideas of abstract data types, Simula was extended with constructs like public, private and protected – originally proposed by Jakob Palme.

One of the observations made on the use of Simula was that in many situations people found that making a model of the application domain in Simula was often more useful than the actual simulation results. The process of describing the application domain in most cases provides a valuable insight in itself. This led Kristen to formulate one of his favourite messages: *To program is to understand*. Programming should not just be considered a low-level technical discipline where it is just a matter of *getting around it*. A program should reflect an understanding of the problem domain.

On the language side, this led to the development of the DELTA language – together with Petter Håndlykken and Erik Holbæk-Hansen. DELTA was not a programming language, but a description language – or specification language. DELTA was a further development of Simula, but extended with equations for describing state changes and the possibility of describing continuous changes over time.

In the late sixties, the Norwegian Iron and Metal Workers Union contacted Kristen. They had realized the need to build up competence in informatics. This led to a project where Kristen together with others developed courses and wrote books on informatics for the members of the union. This project inspired a number of other researchers in Scandinavia to start similar projects involving the users of IT-systems in influencing the design and use of IT-systems. For many years, this work was considered to be very political by the established researchers and companies, but eventually several companies realized that you get better systems and save money by involving the users in the design process. Techniques for involving users in the design of IT-systems is now known as *participatory design* and is practiced by many research groups and companies all over the world.

Kristen was also one of the few philosophers in informatics. This may be traced back to the early days of simulation. To be able to create a model of real-world phenomena and concepts, it is necessary to have strong conceptual means for how to understand and organize knowledge. In the design of languages the modelling capabilities were always central to the work of Kristen. One famous result of this is of course the subclass mechanism that was developed in order to represent specialization hierarchies. I had the pleasure of working with Kristen on the design of the BETA programming language. There were always two criteria for adding a construct to the language: it should be meaningful from a modelling point of view as well as a technical point of view. Following the tradition of Simula, BETA should be a language for describing models of the real world, but it should also be useful as an implementation language.

Kristen was of course thrilled by the enormous success that OO achieved. I remember when he came back from a visit to Xerox Parc where Alan Kay had showed him Smalltalk – he was very impressed and enthusiastic to see how the ideas from Simula had inspired the Smalltalk team. He was happy seeing these ideas adapted by languages such as Flavors, Loops, C++, Eiffel, Java, C#, and many others. He never participated in the critique of possible shortcomings of other languages – on the contrary he respected their creators and acknowledged their influence on the development of OO.

One of the negative effects of the success of OO seems to have been that everybody seems to be an expert in OO and everybody is teaching OO. Hundreds of books are now published written on OO. Kristen found that most of these books did not do a good job in teaching the fundamental concepts of OO. Many of us agreed with him, and last year we invited Kristen to give a number of lectures on OO for students and professionals. We



wrote in the invitation that whether you are a novice or an experienced OO-developer you will always get something out of listening to Kristen's perspective on OO and informatics. Kristen was very concerned on education and it was a great pleasure following his lectures. He was frustrated by the lack of quality in most books on OO and the goal of his last project – COOL: Comprehensive Object-Oriented Learning – was to develop first-class teaching material on object-oriented programming. He has just set up an international team of participants and was ready to start the work when he died. I certainly hope that others will be able to continue this work.

I came to know Kristen when he became a visiting professor at Aarhus University in the early seventies. When he came to Århus he has just completed his project with the Norwegian Iron and Metal Workers Union and he was working on DELTA. In Århus he initiated research on participatory design and object-oriented programming and thanks to Kristen both areas are well-established disciplines in Århus. While Kristen was in Århus we started working on the BETA language together with Bent Bruun Kristensen and Birger Møller-Pedersen. This led to a life-long cooperation with Kristen. We have all spent numerous hours discussing philosophical as well as technical aspects of BETA, languages and informatics in general. The impact from Kristen on my professional carrier has been enormous.

Kristen was not just a pioneer and researcher in informatics. He was engaged in several other aspects of society, including politics. He was the leader (1990-1995) of Norway's «No to EU» movement, which he led to victory in the 1994 referendum.

Kristen and his wife Johanna also became close friends of my family and me. We have enjoyed the hospitality and friendship of Johanna and Kristen at several occasions over the years. My wife Marianne and I had the pleasure of being together with Johanna and Kristen in June at ECOOP 2002 in Malaga. And we will still remember the last time we saw Kristen at a dinner at our apartment in Århus following ECOOP – where fortunately Birger Møller-Pedersen and his wife Kirsten also participated. For me Kristen was a father and mentor in almost all aspects of life. I will certainly miss him.

Ole Lehrmann Madsen

Professor at Aarhus University, Denmark, and director of the Alexandra Institute

I would like to thank Gisle Hannemyr and Birger Møller-Pedersen for useful comments on this text.

About Kristen Nygaard

by Bertrand Meyer

Let me add a few reminiscences of my own to Ole's moving comments. I have had the privilege of knowing Kristen since 1982, when as president of the Association of Simula Users I took part with him in a short program for Norwegian television on the occasion of Simula's fifteenth anniversary. This occurred during a Simula conference, where I asked for his opinion of my presentation. He said that I should watch my terminology, and not use "object" for "class". I certainly remembered the lesson, and since then have often wished others had heard it. It was representative of the intellectual rigor that marked everything that Kristen did.

In our community Nygaard is famous as the co-inventor of object technology, of course, but he was more famous in his country as a politician. He was very proud of having led his country away from an originally dominant pro-Europe stance to a resounding rejection of the European Union in the referendum that decided the issue in 1995. I could never come to terms with this, and over the years had endless discussions with him on the topic, ever trying to understand why someone whose views I so admired on technical matters could be so far away on a question that to me should have been decided by similar rational principles. My views never made more than a ripple on his certainties, but I think in the end I understood what his motives were: a certain attachment to an idealized view of Norway as an Agrarian society based on solidarity, and a rejection of what, a few years later, would come to be known as globalization, which he saw as threatening the social advances of Norway and the interests of the working class. Like his wife Johanna (who he liked to say was nicknamed "Johanna the Red") he was an enthusiastic member of the Left, establishing over the years a close relationship with the trade unions and, as Ole recalls, advising them with groundbreaking work on the effects of computerization.

He applied the same energy to defending these views as to extolling the merits of programming viewed as a modeling activity. This may be the biggest contribution of Simula: to show that to write a program is to produce a model of some system. The very first description of Simula 67, which he like to bring to recent keynotes such as at ETH Zürich in 2000, showed this very clearly, by mentioning, in its title, a "programming an *description* language". Too many people haven't understood this yet, and think that programming is some lowly coding activity, separate from the more noble tasks of design and analysis.

Kristen was a *force de la nature*. He suffused strength. He was tall, spoke loudly, and always had perceptive anecdotes. It's from him, for example, that I really learned what happened at the short-lived World Center for Informatics at the beginning of the Mitterrand years. He could be scathing in his criticism of short-minded or bigoted



politicians, but had a great sympathy for humankind both globally and individually. During that last meeting in Zürich where we both gave keynotes, we went together for an opera and then for dinner, and I was the one who suggested it was time to go home. He seemed never to tire, and was proud of it. I realize now that he was not as strong as he looked — and thought. That's too bad; he was as active as ever, and could have had many more years of contributions to the technology and to his country. He was larger than life, and our field is not the same without him.

Bertrand Meyer