From Object-Orientation to Formal Methods:
Dedicated to the Memory of Ole-Johan Dahl

Volume Editors:
Olaf Owe, Stein Krogdahl, Tom Lyche
Preface

After Ole-Johan’s retirement at the beginning of the new millennium, some of us had thought and talked about making a “Festschrift” in his honor. When Donald Knuth took the initiative by sending us the first contribution, the process began to roll! In early 2002 an editing group was formed, including Kristen Nygaard, who had known Ole-Johan since their student days, and with whom he had developed the Simula language. Then we invited a number of prominent researchers familiar with Ole-Johan to submit contributions for a book honoring Ole-Johan on the occasion of his 70th birthday. Invitees included several members of the IFIP 2.3 working group, a forum that Ole-Johan treasured and enjoyed participating in throughout his career. In spite of the short deadline, the response to the invitations was overwhelmingly positive.

The original idea was to complete the book rather quickly to make it a gift he could read and enjoy, because by then he had had cancer for three years, and his health was gradually deteriorating. Kristen had been regularly visiting Ole-Johan, who was in the hospital at that time, and they were working on their Turing award speech. Ole-Johan was gratified to hear about the contributions to this book, but modestly expressed the feeling that there was no special need to undertake a book project on his behalf. Peacefully accepting his destiny, Ole-Johan died on June 29, 2002. Kristen, who was 5 years older than Ole-Johan, was still very active. Quite surprisingly, he died 6 weeks later of a sudden heart attack. During this short period, Norway lost its two greatest computer scientists.

We are grateful to all who helped with the book, especially all the contributors and the anonymous referees. Credit is due to Ellef Gjelstad, who helped with the preparation of the manuscript.

University of Oslo
December 2003

Olaf Owe,
Stein Krogdahl,
Tom Lyche
On 12 October 1931, Ole-Johan Dahl was born to the family of a ship captain in Mandal, the southernmost city of Norway. In 1952, three years after beginning his studies at the University of Oslo, he was introduced to computers at the Norwegian Defense Research Establishment (NDRE) where he fulfilled his military service obligation. Jan V. Garwick was responsible for the field of mathematical analysis and calculations, and Ole-Johan was placed in the “computing room” led by Garwick’s assistant, Kristen Nygaard. It is quite likely that, in this setting, Garwick, Nygaard and Dahl were the first in Norway to develop programs on “large” digital computers. In the years to come, NDRE cultivated a scientific collaboration with the pioneering computer group at the University of Manchester and the electronics company Ferranti. As a result of this tie, NDRE got the first version of Ferranti’s Mercury computer in 1957.

Ole-Johan’s next project was to develop and implement a “high level” language for numerical computation, called MAC (Mercury Automatic Coding). While Kristen changed focus to the area of operations research in the mid-1950s, Ole-Johan became Garwick’s main collaborator. Together they formed the first real programming group in Norway.

In 1958, Ole-Johan completed the Candidatus Realium degree at the University of Oslo in mathematics. His thesis entitled “Multiple Index Countings on the Ferranti Mercury Computer” [94], was formally in the area of numerical analysis, but actually in computer science and programming, a field that had not yet emerged. In fact, he was one of the first to acquire a relevant and modern education in computer science.

In 1960 Kristen had become research director at the Norwegian Computing Center (NCC). He decided to make an attempt solve two main problems in operations research, namely, the lack of concepts and language for description and communication about large and complex systems, and the lack of a specialized programming language for simulation tasks. Realizing that he could not do this by himself, he looked to Ole-Johan, a specialist in programming language, as the obvious collaborator.
They started working together in 1961, and in 1963 Ole-Johan joined Kristen full-time at NCC. Together they created the Simula 1 language (1961-1965) and Simula 67 (1965-1968), introducing the concepts of class, subclass and inheritance, virtual binding of operations, and dynamic creation of objects. The Simula concept of quasi-parallelism reflected that objects may in principle be independent processes running in parallel. Implicit forms of information hiding, through the subclass mechanism, were later also complemented by explicit forms of information hiding.

These concepts, which constitute what today is called object-orientation, have greatly influenced modern programming languages, programming methodology and modeling languages, including UML. The class related concepts in Simula were clearly ahead of their time; it took some 20 years until they gained understanding and popularity. Languages such as Smalltalk, Beta, C++, Eiffel, Java, and C#, have directly adopted Simula’s fundamental concepts about objects and classes. Object-orientation is today the dominant style of programming, description, and modeling.

To quote Bjarne Stoustrup: “Simula isn’t just an innovative programming language. From the beginning it came with a philosophy of design based on modeling that has had impact far beyond the realm of programming and programming languages.” The object-oriented philosophy also underlies the modern use of windows and the graphical interfaces that we all use.

How could Ole-Johan and Kristen, at such an early stage, design a language with all the mechanisms that today form the “object-oriented” paradigm for system development? An important part of the answer is obviously that they were extraordinary talented researchers in the field, and with rather different perspectives and personalities, which frequently led to confrontations and heated discussions. There is a famous story illustrating their style of working: A new employee at NCC came worriedly running down to the receptionist and cried out: “We must do something! There are two men fighting upstairs in front of the blackboard.” The receptionist listened for a moment, and replied: “No don’t worry. It is just Kristen and Ole-Johan discussing Simula!”

In addition, it was probably very fortunate that they first designed a language for simulation (Simula 1), and later generalized the concepts in a general purpose language (Simula 67). Ole-Johan has expressed it this way: “A reason for this may be that in developing a complicated simulation model it is useful to decompose it in terms of ‘objects’, and to have an explicit mapping from external objects to program constructs. A natural way of developing real systems is not much different.” Kristen emphasized that an essential motivation behind Simula was system description, and the need for a language to model real world concepts.

The final recognition of Ole-Johan and Kristen as the founders of object orientation has been established through two prestigious awards, both given during their last year of life: In November 2001, they were awarded the IEEE John von Neumann Medal “for the introduction of the concepts underlying object-oriented programming through the design and implementation of Sim-
ula 67,” and in February 2002, they were given the A. M. Turing Award by
the ACM “for ideas fundamental to the emergence of object-oriented pro-
gramming, through their design of the programming languages Simula 1 and
Simula 67” [tribute].

Earlier, in June 2000, they were awarded Honorary Fellowships for “their
originating of object technology concepts” by the Object Management Group,
the main international standardization organization within object-orientation.
In August 2000 they were appointed Commanders of the Order of Saint Olav
by the King of Norway (one of the highest national awards), and in 1999
they became the first to receive the Rosing Honorary Prize, awarded by the
Norwegian Data Association for exceptional professional achievements. Both
were elected members of the Norwegian Academy of Science. Ole-Johan was
also a member of IFIP Working Group 2.3 and of Academia Europaea, and
he received an honorary doctorate from the University of Lund, Sweden.

A characteristic of Ole-Johan as a researcher was his quest for simplicity,
elegance, and purity rather than ad hoc solutions with their associated excep-
tions and compromises. This is reflected in the class concept of Simula. From
an educational point of view, its elegance, generality and simplicity make
Simula well suited for teaching object orientation.

We include below a version of the last paper Ole-Johan wrote [36],
which provides a detailed summary of the research leading up to Simula, as
well as afterthoughts and viewpoints of its cultural impact. Other sources
of information on the development of Simula are [23], and more recently
[Hol94, Kro03, Bös03].

Research at the University of Oslo

In 1968 Ole-Johan was invited to be a full professor to create the discipline of
computer science at the University of Oslo. He spent enormous efforts building
up a curriculum in computer science in the late 1960s, writing textbooks at
night while teaching during the day, and supervising up to 20 graduate stu-
dents at a time. For 10 years from 1968 he was the only professor in computer
science in Oslo. For many years, he had only one lecturer and a few research
assistants helping him. The courses he designed met the highest international
standards of the time and for many years to come. Most of the courses were
offered for 20 years or more with only minor changes; some are still being
taught.

After beginning his career at the university, Ole-Johan deliberately stopped
working on further development of Simula as such. In his new position he felt
that it was essential to build up computer science to be an accepted academic
discipline, and establish a theoretical foundation for basic concepts of com-
puter science and programming languages. He made important advances in
programming methodology, introducing techniques for program structuring

1 References labeled with numbers refer to the bibliography of Ole-Johan.
and conceptual modeling, based on the experiences of the design and implementation of Simula. Early works in this direction include the papers [1, 4, 10, 13, 14, 16, 17, 63] with his work on Hierarchical Program Structures as the best known and most influential.

Inspired by Tony Hoare’s logic for program reasoning [Hoa69], he continued research in the area of program architecture, programming and specification languages, as well as verification methods and techniques. Most of this work is related to the area of formal methods, where the idea is to use mathematical methods to specify, develop and reason about programs. Because of his computer science background and education, his theoretical work was accompanied by concern for practical relevance and usefulness. Exploiting the advantages of both, he advocated combined use of top-down development and bottom-up development.

This led to research on abstract data types, a concept inspired by the class concept of Simula and Tony Hoare’s paper entitled Proof of correctness of data representation [Hoa72]. In particular, Ole-Johan focused on generator induction, inspired by the work of John Guttag and Jim Horning, subtyping, inspired by Simula’s subclass mechanism and by the work on order sorted algebra, and integration of applicative and imperative class-based reasoning. Based on mechanizable evaluation and reasoning, he developed a theory for “constructive” types and subtypes centered around a long term project, called Abstraction Building, Education Language (ABEL), which served as a research testbed and source of student projects [5, 21, 25, 26, 33, 34, 43, 44, 46, 53], and resulted in several dr. scient theses supervised by Ole-Johan [Owe80, Nos84, Mel86, Lys91].

Ole-Johan was teaching formal methods, and their practical use for 30 years. He believed that computer science students should know the principles of program reasoning, and that this would make them better programmers even without performing detailed verification. The course work has resulted in the book, Verifiable Programming [5], which includes much of his own research and results. He supervised a large number of students, ten of whom became university professors.

**Formal methods, object-orientation and concurrency**

Work towards the understanding and formalization of what is now called object-orientation, was carried out already in the early 1970s [15, 18], and with the thesis of Arne Wang, one of Ole-Johan’s first students [Wan74].

Ole-Johan’s early approach to reasoning about object-oriented systems builds on the idea of limiting direct access to attributes of an object from outside, either disallowing all remote variable access or allowing access to some variables (seen as an abbreviation for implicit read and write operations). This means that one can give local invariants in a class and prove that the class invariant is established and maintained by looking at the text of the class
itself and possibly super-classes. When subclassing is restricted so that super-invariants are respected, reasoning about objects can be done without looking at the global state space: Hoare style reasoning can be done locally in each class.

Early on it was recognized that this kind of class-based reasoning was fruitful for the Monitor concept of Hoare [Hoa74] and for aspects of operating systems concerning process control. The co-routine and monitor papers were part of this research direction [15, 18, 19, 20, 21, 29, 35, 64] and the thesis of Stein Gjessing, who had been supervised by Ole-Johan [Gje83].

In contrast to those object-oriented approaches where “everything is an object”, Ole-Johan believed in object classes side by side with data types. Ole-Johan’s view was that objects should reflect mutable data structures, handled by references, and data types should reflect immutable, but copyable, data [28]. This called for user defined data types. In ABEL, a functional sublanguage was defined for definition of abstract data types, whereas classes were defined in an imperative style. Thus Ole-Johan considered functional programming as a complement to object-oriented programming rather than a competitor.

According to the original ideas of Simula, an object would in general have its own activity, as well as data and procedures. Objects with activity were reflecting “independent processes”, and objects without activity (but still with data and procedures) were called passive. In Simula 67 these ideas were realized by allowing objects to be co-routines, the natural way at that time to imitate concurrent processes and a useful simulation mechanism. In today’s world a natural adaptation would be to let objects be concurrent processes, and one would obtain a distributed system by a set of objects running in parallel and interacting by remote method calls (only). This is the approach taken in Ole-Johan’s later works. See for instance [26, 36].

Ole-Johan’s work on abstract specification of concurrent objects by means of histories presents a techniques for “black box” interface specification of process classes. The abstract state of a concurrent object is represented by its communication history, i.e., the trace of all visible communication events involving the object, such as method calls. As the abstract state at any given time is reflected by a finite history, Ole-Johan developed specification techniques based on finite traces, and a central idea was to use right-append as a trace generator in order to describe new actions in terms of the current history, thereby avoiding recursive process definitions as found e.g. in CSP. Reasoning about concurrent objects in terms of such histories is compositional and integrates well with object-orientation. Ole-Johan developed a style of history specification where specifications of a certain form can easily be refined into an imperative object-oriented implementation. The use of histories was initiated in the early 1970s and remained an important research topic for him throughout the 1990s [21, 22, 26, 41, 58]. When he retired, Ole-Johan was writing a book on concurrency based on research in connection with a course on concurrency and process control [69, 54].
The above principles for object-oriented programming, specification, and reasoning, constitute what we may call “the Dahl School.” A further introduction is given in the paper by Johnsen and Owe in this volume.

**International visits**

Due to Ole-Johan, several prominent researchers visited Oslo. In particular we mention the one year stay of Donald Knuth from Stanford University in 1972 to 1973. Knuth had a great impact on the computer science development in Oslo at an early and crucial period in time.

Knuth, who immediately understood the benefits of the Simula ideas, gave up work on his own simulation language (SOL), and became a supporter of Simula. He could have made Simula quite well-known by teaching it at Stanford, but when asking for an inexpensive compiler for academic use at Stanford, he was unfortunately turned down by NCC (despite the strong arguments of both Kristen and Ole-Johan, who fully understood the importance of this opportunity).

In the late 1970s Reiji Nakajima, a post doc at that time, made a one year visit to Oslo, which led to a number of interesting discussions around abstract data types and ABEL. In the early 80s, Neelam Soundarajan visited Oslo for a year, working together with Ole-Johan on reasoning with histories. Ole-Johan enjoyed this cooperation very much, appreciating Neelam’s clarity. Neelam came for a second year in the early 1990s. Zhenjia Lei, University of Xi’an also visited Ole-Johan, which resulted in a return visit by Ole-Johan in Xi’an in the mid-1990s. Apart from this visit, Ole-Johan had only one sabbatical leave during his career, which was spent at Stanford University in the late 1970s.

Ole-Johan also enjoyed the many shorter visits by a large number of scientists, including Tony Hoare, Hans Langmaack, Dines Bjørner, Eugene Kindler, Cliff Jones, Manfred Broy, David Luckham, Jean-Pierre Jouannaud, Pierre Lescanne, and Willem-Paul de Roever, most of whom also enjoyed music evenings and dinners in Ole-Johan’s home.

**Personal interests**

Ole-Johan was a music lover and an excellent amateur pianist. In fact, he was one of the best “prima vista” amateur pianists in Norway. He knew the world of classical music well, and that of chamber music in particular. Much of his free time was filled by music, and he was a central member of the Board of Oslo Quartet Association and a driving force behind the yearly chamber music courses at the Nansen School in Lillehammer. He regularly arranged house concerts at the department, often playing together with visitors who happened to be musicians, or with his wife or daughter.

In addition he enjoyed and excelled in many kinds of games including chess and bridge, and spent much time as a student pursuing these interests.
At conferences he was known for his skills in classical billiards (3 balls) in addition to the piano. A personal bibliography is written by his wife [Dah03].

References


[tribute] The Department’s page of tribute: http://www.ifi.uio.no/adminf/tribute.html with links to Ole-Johan’s and Kristen’s own homepages (Kristen’s is more rich on information). For Kristen there is a memorial page at http://www.ifi.uio.no/in_memoriam_kristen/ See also http://www.jot.fm/issues/issue_2002_09/eulogy
Ole-Johan Dahl – a Bibliography

This bibliography is based on information collected by Berit Strange, from the Library of the Department of Informatics, University of Oslo. Ole-Johan’s homepage www.ifi.uio.no/~olejohan contains links to some papers.

Books


Papers in Proceedings, Journals, and Books


   (Presented at the Object-Oriented Programming Workshop, June, 1986.)


Reports from University of Oslo

Early reports are from Department of Mathematics, section D (for numerical analysis and computer science). Later reports are from Department of Informatics (IFI), which was formed in 1978. Reports marked by a star (∗) represent early versions or revisions of papers mentioned above.

Research Reports


Compendiums and Lecture Notes


Reports from the Norwegian Computing Center (NCC), Oslo
73. The SIMULA Storage Allocation Scheme. O.-J. Dahl. NCC Doc. 162, Nov. 1963 (9 pages)
74. SIMULA Status Report. O.-J. Dahl. NCC Doc. 1.1, 1964 (10 pages)
75. The SIMULA Data Structures. O.-J. Dahl. NCC Doc. March 1964 (23 pages)
78. SIMULA status report. O.-J. Dahl. NCC Doc. 1.10, 1964 (24 pages)
80. Basic Concepts of SIMULA – an ALGOL Based Simulation Language. O.-J. Dahl and K. Nygaard. NCC Doc., 1965 (17 pages)
81. SIMULA, an ALGOL Based Simulation Language. O.-J. Dahl and K. Nygaard. NCC Doc., April 1966 (26 pages) *
97. Mac Bulletin II. O.-J. Dahl. NDRE report (S-17), 1960 (3 pages)

All versions of the common base language definition assume knowledge of Algol 60. A final and complete version of the SIMULA 67 language definition without this assumption is found in: Standard SIMULA, as approved by the SIMULA Standards Group Aug. 26, 1986. (176 pages) ISBN 91-7162-234-9.

MAC was an acronym for MERCURY Automatic Coding, a high level programming language for the MERCURY computer, and a compiler.
Video Recorded Talks

101. Introduction to SIMULA (talk given in Norwegian), to be available through the department homepage: www.ifi.uio.no.