Cambridge University Press & Assessment
978-0-521-51825-3 — From Semantics to Computer Science
Edited by Yves Bertot , Gérard Huet , Jean-Jacques Lévy , Gordon Plotkin
Frontmatter
More Information

**From Semantics to Computer Science**

Gilles Kahn was one of the most influential figures in the development of computer science and information technology, not only in Europe but throughout the world. This volume of articles by several leading computer scientists serves as a fitting memorial to Kahn's achievements and represents the broad range of subjects to which he contributed, whether through his scientific research or his work at INRIA, the French National Institute for Research in Computer Science and Control.

The authors reflect upon the future of computing, discussing how it will develop as a subject in itself and how it will affect other disciplines, from biology and medical informatics, to the web and networks in general. Its breadth of coverage, topicality, originality and depth of contribution, make this book a stimulating read for all those interested in the future development of information technology.

# From Semantics to Computer Science

## Essays in Honour of Gilles Kahn

*Edited by*
Yves Bertot
*INRIA Sophia-Antipolis Méditerranée*

Gérard Huet and Jean-Jacques Lévy
*INRIA Rocquencourt*

Gordon Plotkin
*University of Edinburgh*

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

Shaftesbury Road, Cambridge CB2 8EA, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

103 Penang Road, #05–06/07, Visioncrest Commercial, Singapore 238467

Cambridge University Press is part of Cambridge University Press & Assessment, a department of the University of Cambridge.

We share the University's mission to contribute to society through the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org
Information on this title: www.cambridge.org/9780521518253



First published 2009

*A catalogue record for this publication is available from the British Library*

ISBN 978-0-521-51825-3 Hardback

# Contents

# List of contributors

Roberto Amadio
*PPS, Université Paris Diderot*

Nicholas Ayache
*INRIA Sophia Antipolis Méditerranée*

François Baccelli
*Ecole Normale Supérieure*

Jean-Pierre Banâtre
*INRIA/IRISA Université de Rennes*

Alain Bensoussan
*University of Texas at Dallas*

Pierre Bernhard
*I3S, Université de Nice Sophia Antipolis and CNRS*

Yves Bertot
*INRIA Sophia Antipolis Méditerranée*

Luca Cardelli
*Microsoft Research, Cambridge*

Giovanna Carofiglio
*Ecole Normale Supérieure and Politecnico di Torino*

Olivier Clatz
*INRIA Sophia Antipolis Méditerranée*

Thierry Coquand
*Chalmers University of Technology and Göteborg University*

Bruno Courcelle
*Institut Universitaire de France and LABRI, Université de Bordeaux 1*

Pierre-Louis Curien
*PPS, CNRS and Université de Paris 7*

Hervé Delingette
*INRIA Sophia Antipolis Méditerranée*

Mehdi Dogguy
*PPS, Université Paris Diderot*

Serguei Foss
*Heriot-Watt University, Edinburgh*

Pascal Fradet
*INRIA Rhône-Alpes*

Frédéric Hamelin
*I3S, Université de Nice Sophia Antipolis and CNRS*

Laurent Hascoët
*INRIA Sophia Antipolis Méditerranée*

Andrew Herbert
*Microsoft Research, Cambridge*

Gilles Kahn

Yoshiki Kinoshita
*National Institute of Advanced Industrial Science and Technology (AIST), Japan*

Paul Klint
*Centrum voor Wiskunde en Informatica and University of Amsterdam*

Emmanuel Ledinot
*Dassault Aviation*

Edward A. Lee
*University of California, Berkeley*

Jean-Jacques Lévy
*INRIA and Microsoft Research-INRIA Joint Centre*

David B. MacQueen
*University of Chicago*

Grégoire Malandain
*INRIA Sophia Antipolis Méditerranée*

Eleftherios Matsikoudis
*University of California, Berkeley*

Robin Milner
*University of Cambridge*

Tobias Nipkow
*TU München*

Bengt Nordström
*Chalmers University of Technology and Göteborg University*

Christine Paulin-Mohring
*INRIA Saclay - Île-de-France and Université Paris-Sud*

Xavier Pennec
*INRIA Sophia Antipolis Méditerranée*

Yann Radenac
*INRIA/IRISA Université de Rennes*

G. Ramalingam
*Microsoft Research India, Bangalore*

Aarne Ranta
*Chalmers University of Technology and Göteborg University*

Thomas Reps
*University of Wisconsin, Madison*

Erik Sandewall
*Linköping University and Royal Institute of Technology, Stockholm*

Maxime Sermesant
*INRIA Sophia Antipolis Méditerranée*

Makoto Takeyama
*National Institute of Advanced Industrial Science and Technology (AIST), Japan*

Christian Urban
*TU München*

Jean Vuillemin
*Ecole Normale Supérieure*

# Preface

Gilles Kahn was born in Paris on April 17th, 1946 and died in Garches, near Paris, on February 9th, 2006. He received an engineering diploma from Ecole Polytechnique (class of 1964), studied for a few years in Stanford and then joined the computer science branch of the French Atomic Energy Commission (CEA), which was to become the CISI company. He joined the French research institute in computer science and control theory (IRIA, later renamed INRIA) in 1976. He stayed with this institute until his death, at which time he was the chief executive officer of the institute. He was a member of Academia Europaea from 1995 and a member of the French Academy of Science from 1997.

## Gilles Kahn's scientific achievements

Gilles Kahn's scientific interests evolved from the study of programming language semantics to the design and implementation of programming tools and the study of the interaction between programming activities and proof verification activities. In plain words, these themes addressed three questions. How do programmers tell a computer to perform a specific task? What tools can we provide to programmers to help them in their job? In particular, how can programmers provide guarantees that computers will perform the task that was requested?

## Programming language semantics

In the early 1970s, Gilles Kahn proposed that programs should be described as collections of processes communicating through a network of channels, a description style that is now known as *Kahn networks*. He quickly proposed a format for the description of these networks and addressed the question of giving a precise meaning to this format. This is how Gilles Kahn came to be interested in programming language semantics. Part of the work to define this language of communicating processes was done in collaboration with David MacQueen, then at the University of Edinburgh.

In particular, Kahn networks could contain loops, which invite infinite behavior. The data on the communication channels could then appear as infinite streams of data and Gilles needed to refine the existing theory of domains to provide an adequate treatment of potentially infinite data structures. This led to the description of *concrete domains*, in collaboration with Gordon Plotkin, also at the University of Edinburgh.

The work on concrete domains followed the style of denotational semantics, where the meaning of each program is described as a function from some domain of input data to some domain of output data. Denotational semantics requires that a meaning should be ascribed to each construct of the programming language: this meaning should then be a function mapping the semantics of the sub-components to the meaning of whole expression. In any meaningful language, this implies that one should be able to find solutions to recursive equations, in other words, equations in which the defined object appears on both sides.

With this work, Gilles Kahn became recognized as an expert in the domain of programming language semantics. For instance, Gilles Kahn and his student Véronique Donzeau-Gouge were invited to work on the formal semantics of Ichbiah's LIS language, which was to become the Ada language after winning a competition from the American Department of Defense. Moreover, the theory of concrete domains was to have a wide impact, through the contributions of Gérard Berry and Pierre-Louis Curien on the definition of *sequential algorithms* and then on game semantics and the study of fully abstract models for sequential programming.

Gilles Kahn then set out to change the general point of view for describing programming language semantics: instead of looking for a meaning for programs, he proposed the provision of a clear description of how programs operate. He started from an earlier proposal of Gordon Plotkin, known as *structural operational semantics* and suggested considering complete executions of programs. In parallel with denotational semantics, it is then necessary to associate with each programming language construct and each possible behavior of this construct a description of an execution of this construct as a composition of the execution of its sub-components. Gilles Kahn noted

Cambridge University Press & Assessment
978-0-521-51825-3 — From Semantics to Computer Science
Edited by Yves Bertot , Gérard Huet , Jean-Jacques Lévy , Gordon Plotkin
Frontmatter
More Information



the similarity of this approach with the concept of natural deduction used in proof theory and proposed that this description style should be called *natural semantics*.

**Programming environments**

From the middle of the 1970s, Gilles Kahn was also interested in developing practical tools to enable programmers to perform their job more efficiently and more reliably. The first aspect, studied from 1975, was the structured manipulation of programs, in which the elementary constructs of programming languages were directly understood by the program development tools. With the advent of graphical workstations, structured programming environments became very powerful: the mouse could be used to select whole expressions in one gesture and the display of programs could benefit from varied colors and fonts and from the enhanced readability of bi-dimensional layout. Several versions of interactive programming environments were developed, the first one was called Mentor and was developed in Pascal.

An offspring of the work on Mentor was the development of Mentor-Rapport by another of Gilles' students, Bertrand Mélèse. This tool was specialized for the production of structured documents. Mentor-Rapport was later to become Grif and have an important influence on the development of SGML and XML.

The next generation of programming environments was developed in Le_Lisp, another output of INRIA's research. This version was called Centaur. The development of Centaur started in the middle of the 1980s. Again, the tool was based on the structured manipulation of programs, but the project had the ambitious goal that a programming language should be entirely described with formal specifications of its syntax and its semantics; from which all programming tools could be derived. In particular, the language of inference rules that was already advocated by Gilles Kahn and Gordon Plotkin to describe operational semantics had also been in use for some time to describe type systems: this language could be used as a universal tool to describe many aspects of programming languages; it could also be given a computational behavior, through a translation into Prolog. This language was integrated in Centaur under the name of Typol, thanks to the work of Thierry Despeyroux. Typol was used to describe a wide variety of programming languages and tools for these languages: interpreters, compilers, partial evaluators, etc.

Generic programming environments are now part of the programmer's palette of tools. The Eclipse environment, produced by IBM is a well-known example. More characteristically, the advent of XML as a widespread communication standard also shows that the industry and the academic world are now convinced of the importance of structured data, as opposed to plain text. This predominance of structured data was already asserted in the Mentor tool, developed around 1975 by Gilles Kahn and his team.

## Proving environments

Structured manipulation as found in Centaur could be used in other domains than just programming environments. Gilles Kahn was soon interested in applying it to tools that support mathematics on the computer. In particular, Gilles Kahn thought that *real theorem provers deserve real user-interfaces* and the Centaur technology was a good instrument to implement powerful man–machine interfaces for these tools. The readability of mathematical formula could benefit from the strong layout capabilities, a structure-based translation could be used to transform formal proofs into natural language mathematical text, and moreover the selection of positions in logical formula could be used to guide the proof process. The latter capability was called *proof-by-pointing* and attracted some notoriety to Gilles Kahn's team.

From proof environments, it was a natural step to get involved in the theme of computer-aided proof, in particular with the Coq system that was developed at INRIA, initially under the supervision of Gérard Huet. Gilles Kahn contributed in many ways to the growth of this system, studying formal descriptions of his own theory of concrete domains or some aspects of constructive geometry. Gilles also contributed to a tutorial to the Coq system.

In 1996, Gilles Kahn was appointed to the inquiry board that studied the failure of the inaugural flight of the Ariane 5 launcher. He played a crucial role in identifying the cause of the failure as a software bug. He saw this as an opportunity to promote the systematic analysis of software by automatic tools and made sure the best researchers of INRIA had an opportunity to apply their skills on this example of a complex system involving mission-critical software components.

The scientific themes that Gilles Kahn studied during his career exhibit both a wide variety of styles, from the most practical concerns of developing program editors to the most theoretical ones of devising a theory of domains for infinite data-structures; from programming language semantics to proof theory. However, all these themes contribute to the same focal motivation: making programmers more efficient and reliable in their work. The question of proving that programs satisfied some specification was already present in the early work on Kahn Networks; networks of communicating processes and co-routines were also used in the architecture of the Centaur system; questions of semantics are still among the most active application areas of theorem proving tools.

### Gilles Kahn's influence on computer science in France

Beyond his own scientific achievements, Gilles Kahn helped in shaping modern computer science, mostly through his influence on INRIA's evolution. He was widely recognized for his scientific culture, thanks to which he was able to understand the key aspects of many scientific fields. He was good at sensing important new trends and at supporting the colleagues who explored them.

J.-L. Lions, the director of IRIA in the early 1970s, was quick to recognize Gilles' ability to understand the challenges of new research domains and gave Gilles the responsibility to preside over IRIA's project evaluation committee. Thus, Gilles was a researcher at INRIA, but with this specific task, he was already in a position to follow and appreciate the activity of colleagues in a much wider area than his own domain of expertise.

When INRIA decided to create a center in Sophia Antipolis, Gilles Kahn was again appointed as the head of the project committee, in effect being recognized by P. Bernhard as his main scientific adviser. At INRIA Sophia Antipolis, Gilles was a key actor in supporting the evolution of computer science in several domains. Gilles had a clear vision of the growth of personal computers linked together by a wide-area network and he insisted that INRIA should attract a high-level research team on networking. Later on, he supported the collaboration of INRIA in the World-Wide-Web consortium, an effort that helped shape the Internet as we know it today. In a similar vein, Gilles Kahn was an early supporter of interactions between computer science, biology, and medicine and he made sure that INRIA Sophia Antipolis would provide the right conditions for the growth of teams interested in artificial vision, computer-based analysis of medical imagery, or neuroscience.

Gilles Kahn also had an acute understanding of computer science's industrial impact. He was an early advocate of INRIA's approach to spawning start-up companies. For instance, he took in the early 1980s the initiative of meeting the top management of the Bull company to incite them to enter the market for personal workstations; he had noticed the SM-90 experimental architecture, then developed in the French telecommunication company's research laboratory (CNET). He convinced J.-F. Abramatic to start the Gipsy company on this market. Later, Gilles also convinced Pierre Haren to start the Ilog adventure with a team of bright young researchers. Ilog is now a much-cited success. Of course, the start-up movement around INRIA also had a few failures, some of them on topics that were directly derived from Gilles and his team's research, but Gilles made sure the dynamics would be maintained.

Gilles Kahn's vision on the future of computer science and control theory was more and more appreciated, this led to A. Bensoussan, the head of INRIA in 1994, appointing him as scientific director. When B. Larrouturou took over as Chief Executive Officer, he confirmed Gilles Kahn in this position. At the time, Gilles supervised the evolution of around 80 research teams and promoted new domains, in particular, the theme of bio-informatics was brought to the forefront.

In 2004, Gilles Kahn was appointed as INRIA's Chief Executive Officer. His influence at this position continued his work as scientific director. However, new aspects appeared, especially in his will to establish collaborations with the best research centers in the world, including research centers affiliated with major software companies.

## The selection of articles in this book

In this book, we collected articles to contribute to Gilles Kahn's memory. Some of the contributions are directly related to one of Gilles' scientific interests. The articles by R. Amadio and M. Dogguy, J.-P. Banâtre, P. Fradet, and Y. Radenac, P.-L. Curien, E. Lee and E. Matsikoudis, and D. MacQueen show the continued influence of Kahn Networks in computer science. In particular, D. MacQueen's article gives a comprehensive historical perspective of the influence of this work. The articles of T. Coquand, Y. Kinoshita, B. Nordström, and M. Takeyama, B. Courcelle, G. Kahn, and J. Vuillemin, J.-J. Lévy, and G. Ramalingam and T. Reps are related to the larger field of programming language semantics. In particular, the article by Courcelle, Kahn, and Vuillemin is an English translation of an article on recursive equations that was published in French and is reproduced here with the kind permission of Springer Science and Business Media. The articles by P. Klint, B. Nordström, A. Ranta, and E. Sandewall are more closely related to Gilles' interest in developing programming tools and environments. The articles of Y. Bertot, C. Urban and T. Nipkow, and C. Paulin-Mohring are more closely related to computer-based proofs of programming language properties. In particular, C. Paulin-Mohring's contribution demonstrates how computer-based proof can be applied to Kahn Networks.

Other contributions attempt to give a wider view of Gilles Kahn's influence on computer science. Because of his positions as scientific director, Gilles Kahn had supported many domains of computer science and control theory. We thus have a wide variety of themes being considered: N. Ayache and his team describe the interaction of computer science and medical imaging; F. Baccelli, G. Carofiglio, and S. Foss provide an article on the efficiency analysis of some networking techniques; P. Bernhard and F. Hamelin contribute a study of game theory applied to biological systems; and L. Hascoët describes advanced techniques in automatic differentiation.

In a third group of contributions, authors have attempted the difficult exercise of paying tribute to the visionary qualities of Gilles Kahn. A. Bensoussan's article evokes Gilles' influence on the evolution of INRIA; L. Cardelli studies the ways in which computer science can help biologists address the daunting task of understanding living objects; A. Herbert investigates the impact of computers on science at large; E. Ledinot evokes the influence of Gilles' work on programming environments; and R. Milner questions the links and difference between software science and software engineering.

**Acknowledgments**

We wish to thank D. Tranah, C. Dennison, and C. Appleton from Cambridge University Press for their help in elaborating this book. Other people who helped in preparing the articles are A. Appel, K. Avratchenko, G. Berry, J. Bertot, J.-D. Boissonnat, G. Boudol, F. Boussinot, M. Brady, R. Bryant, D. Caucal, O. Corby, Y. Coscoy, S. Coupet-Grimal, T. Despeyroux, D. Doligez, C. Dubois, F. Fages, S. Gaubert, T. Jensen, J.-M. Jezequel, C. Kirchner, J.-W. Klop, V. Komendantsky, B. Larrouturou, J. Laird, X. Leroy, Z. Luo, R. Marlet, C. Munoz, F. Pottier, L. Pottier, M. Pouzet, L. Rideau, M. Sorine, T. Streicher, L. Théry, B. Werner.