# Dansk Datamatik Center

Dines Bjørner[1], Christian Gram[2], Ole N. Oest[3] and Leif Rystrøm[4]

[1] bjorner@gmail.com, prof. emeritus DTU, Denmark
[2] chr.gram@ddf.dk, prof. emeritus DTU, Denmark
[3] ooest@attglobal.net, DDC-I Inc., Phoenix, AZ 85020, USA
[4] rystroem@mail.dk, Danish Road Directorate, Copenhagen, Denmark

**Abstract.** In 1979 a software research and development centre was created to demonstrate the power of systematic and formal methods in software development. One of the first and biggest projects at Dansk Datamatik Center (DDC) was to develop an Ada compiler and run-time system. DDC made the first DoD validated Ada compiler in Europe, and the Ada project was carried on in a subsidiary called DDC–I, Inc. The paper describes the background and start of DDC and some aspects of the formal development method RAISE, Rigorous Approach to Industrial Software Engineering, as well as other DDC activities.

**Keywords:** software development, formal methods, technology transfer.

## 1 The History of DDC

The idea of Dansk Datamatik Center [1] (DDC) was first aired by Christian Gram of the Department of Computer Science at the Technical University of Denmark (DTU) and discussed with his colleague Dines Bjørner during spring 1979. Software development in business, administration, and industry was then still very unsatisfactory. Already in 1968 the NATO Science Committee arranged a conference on software engineering [18] on *"a problem crucial to the use of computers, viz. the so-called software, or program, developed to control their action."* Many projects were late, more expensive than estimated, and full of errors. *"The phrase 'software engineering' was deliberately chosen as being provocative, in implying the need for software manufacture to be based on the types of theoretical foundations and practical disciplines, that are traditional in the established branches of engineering."* Even after a second conference on the same problem area, problems with software development in practice continued. We felt in 1979 that computer scientists had developed foundations and theories that – if properly implemented – could make programming a more professional, engineer-like profession and allow developing large, reliable programs on schedule.

We contacted ATV, the Danish Academy for Technical Sciences, an umbrella for a significant number of 'Science-Engineering-Technology' institutes working

[1] The DDC 'cube' was designed by Danish Design award winning Ole Friis.

to help industry applying the newest technologies in their field. ATV responded positively, and 10 of the largest users and/or producers of IT in Denmark agreed to become members of the institute, each paying 100.000 DKK per year. DDC was created in September 1979 as an ATV society for advanced software development. The members of DDC were Christian Rovsing A/S, Crone & Koch edb, Danish Defense Research Establishment, Datacentralen af 1959, Jydsk Telefon A/S, Kommunedata, Regnecentralen af 1979 (RC), Sparekassernes Datacenter (SDC), Teleteknisk Forskningslaboratorium (TFL) and ØK Data.

An attempt to make Danish financial institutes interested in DDC failed.

From 1979 to 1989, a number of projects on software tools were carried out, with Dines Bjørner as the scientific leader, in cooperation with one or more of the members. Typically the projects dealt with aspects of software development as, e.g., the cooperation with SDC on methods and tools for program construction ending up with a 230 pages report discussing how SDC's software development could be improved [17]. But two larger projects dominated the activities for several years, i.e., the Ada compiler development, sponsored by the Commission of the European Communities, CEC, and the CHILL compiler development in cooperation with the TFL.

Most projects used formal specification methods – VDM, the Vienna Development Method [2, 4] developed at IBM, and later RAISE, Rigorous Approach to Industrial Software Engineering [14], a method co-developed at DDC.

A main goal was to act as a link between Danish IT companies and Danish/European IT research, in order to make DDC capable of using the most advanced software design and development tools.

DDC had very little cooperation with the IT research in Nordic countries. We felt that there was not the same interest for and emphasis on the use of formal methods for software development. Because Denmark was a member of EU (and the only Nordic member at that time) it was natural for us to seek partners and financial support among EU members. But from 1984 we cooperated with the Swedish Försvarets Materielverk on the Ada project, and later also with Ericsson and Bofors.

The Ada compiler project was so successful that a separate company, DDC International A/S, was created in 1985 to market, sell and further develop the Ada system. DDC International later created the limited company, DDC–I Inc., which is still in the market, and the Ada system and other systems developed at DDC–I have been sold in USA, China, and other countries. Some years ago DDC–I moved the headquarter to Arizona, USA.

Over the years some of the original member companies lost interest in DDC and its services and it became difficult to find funding for the kind of projects, DDC wanted to further. After almost 10 years it was decided to close down DDC, but some larger projects were carried on under new hats. The development of Ada systems continued in DDC-I. The RAISE project and the people working on it were transferred to the Danish software house CRI Inc. (CRI), who completed the project and used it as a base for the LaCoS project (Large-scale Correct Systems using formal methods).

## 2 The Use of Formal Techniques at DDC

Both the CHILL and the Ada compilers were developed roughly as follows: The static and dynamic semantics of the languages were described in denotational form. Then, in several stages of refinement, more and more concrete prescriptions were systematically developed, covering both static and dynamic semantics for sequential and parallel language features [3]. These prescriptions (in the VDM language) included the expression of a number of ”standard” compiler requirements for micro-processor-based compilation as well as execution platforms. From the concrete requirements prescriptions multi-pass compilation administrators were developed and, subsequently, the $N$ passes of the compiler were coded (for Ada $N$ was 9) [4, 5].

In both projects the VDM approach was one of systematic development: formal specifications of all phases, stages and steps, but no formal proofs of correctness. See Sect. 3.2 for more details.

## 3 DDC Activities 1979–89

The 30-35 member professional staff worked with *advanced software development projects:* the CHILL and Ada compilers, etc.; *research:* the Chill and Ada Formal Definitions, etc.; *explorative studies:* Formal Methods Appraisal, Office Automation, etc.; *seminars; courses*; and other *dissemination*.

### 3.1 The CHILL Projects

**The Formal Definition of CHILL** In 1978 Prof. A. Kjerbye Nielsen, director of TFL, asked Dines Bjørner to follow the development of the CHILL programming language by an international group of the C.C.I.T.T.[2] (now ITU[3]). It was the intention to attempt a formal description of CHILL. Through the work of Dines Bjørner, his colleague, Hans Bruun, and some MSc students formal descriptions were researched and experimentally developed at DTU. Once DDC was established, that work was completed at DDC [16, 15].

The benefit of making formal descriptions may be illustrated by this: Hans Bruun found – during his painstaking analysis and in discussions with members of the C.C.I.T.T. group – a tiny identifier scope issue which, if simply removed, would shorten the definition by some 20%. The group ended up nullifying that scope issue – significantly simplifying any CHILL compiler.

**The CHILL Compiler Development** In 1979, before the formation of DDC, work started on the development of a compiler for the full CHILL language (CCITT High Level Language). The work was done by Peter Haff and Søren Prehn partially funded by TFL. The CHILL compiler was for the full CHILL

[2] Comité Consultatif International Téléphonique et Télégraphique

[3] International Telecommunication Union

programming language – including, for example, its three "independent" sets of parallel programming constructs. When completed the compiler was made public property by TFL and DDC and played a significant rôle in the teaching of CHILL worldwide [15].

### 3.2 The Ada Projects

#### The Ada Compiler Project

*The Danish/Italian Collaboration and the Initial Contract:* The Ada programming language had been designed on behalf of the US Department of Defense as a new *cure-it-all* standard programming language targeting development of embedded software. At the time more than 200 languages were in use within the US DoD and significant savings would be possible if this number could be reduced to just one, Ada.

The Commission of the European Communities (CEC) had set aside significant funding for the development of a European Ada compiler system because the CEC thought that similar savings might be possible within European software development. A French/German consortium was slated to receive the funding for this European Ada Compiler System.

Shortly before the call for proposal ended, DDC formed a consortium with Olivetti (Italy) and Christian Rovsing (CR) (Denmark) and in record time developed a bid for the funding.

Now followed several months of hectic technical evaluations of the DDC/Olivetti/CR proposal. It was very inconvenient, for certain people in the CEC, that this Danish/Italian consortium competed for money intended for the French/German consortium. Each time we went to Brussels for another technical evaluation, the experts found some 'show-stoppers' not discussed at the previous meeting. But it ended up, in early 1981, with a contract where 50% funding came from the CEC and 50% from Danish sources. DDC's part of the project was to develop a portable Ada compiler.

Some years later, after the Ada standard had been finalized, DDC won a contract for developing a formal specification of the language. We considered that a significant acknowledgement of VDM and of DDC's high technical level.

*Technical Approach:* The driving method behind the development of the DDC Ada Compiler System was the Vienna Development Method (VDM) [2, 4] and the demonstration of the viability of using VDM was initially perhaps considered more important to DDC and its members than the end product itself.

In 1980 a team of MSc students from DTU had developed a formal definition of Ada and of the underlying execution model [6]. This specification became the foundation for the actual compiler development. The compiler was developed as a number of *refinements* of this specification.

It was a requirement on the Danish/Italian consortium that the Ada compiler should be suitable for mini computers with limited memory resources. It was therefore designed as a multi-pass compiler, and the systematic development of the Ada compiler was basically refined into *buckets*, where each bucket would correspond to a compiler pass.

Even though the Ada language and the host computer platform underwent changes in this period, the project became a success and resulted in a commercial viable product. The DDC team even managed to achieve formal US DoD approval (*validation*) of the compiler ahead of the French/German project [9, 19].

The project – part of which was research-oriented – exceeded the original budget with less than 20%, but was on time, which is significantly better than most software projects.

*The First Commercial Sales:* DDC presented the DDC Ada compiler project and the use of VDM at several conferences [9, 19]. These presentations showed not only the high level of correctness of the compiler but also provided productivity numbers which showed that VDM was a cost-effective development method even when used without the support of computerized tools which only became available much later [14, 13].

Those technical presentations persuaded Nokia to license the DDC compiler technology for the purpose of developing an Ada compiler for a proprietary minicomputer. Without any sales and marketing organization DDC made its first commercial sale of what became known as the DDC *OEM Compiler Kit*.

Honeywell heard through Nokia of the DDC technology. Honeywell had two projects needing an Ada compiler, a mini computer and a main frame system. They talked with the French/German consortium but ended up placing both orders with DDC. Significant sales were also made to the COSTIND, China and NEC, Japan.

*The Formation of DDC International (DDC-I):* Several other OEM contracts followed and, in 1985, DDC created a subsidiary, DDC-I, to commercialize the DDC Ada Compiler System, which came to be used on many projects (aircrafts like Boeing 777, MD-80 and MD-90, satellites and various military programs) and is still in 2010 generating revenue for DDC-I.

Commercializing the other outcome of the project, the successful use of VDM, turned out to be much more troublesome. The industry was not ready to adopt formal methods until some 20 years later, where formal methods become mandatory in validation of, for example[etc.], MILS (Multiple Independent Levels of Security) operating systems.

**The Formal Definition of Ada** During the early 1980s a lively discussion forum, supported by CEC, resulted in the CEC supporting the R&D of a formal definition of Ada. A group was formed, again anchored in Denmark (DDC) and again with Italian partners (University of Pisa, University of Genoa, IEI (Istituto di Elaborazione della Informazione, Pisa) and CRAI [Cosenza]). The Ada formal definition project (running 1984-87) was truly a research project. It was not known exactly, from the outset, how the result, [24, 1, 7, 22, 10], would look: which specific abstraction and modelling techniques to deploy, which abstractions to use, etc. During the project a number of exciting research problems were solved and many scientific papers were published .

### 3.3 The RAISE Project

The RAISE project had two parts: a precursor project, FMA, and the main project, the RAISE development project.

**The Formal Methods Appraisal Project (FMA)** During the developments of the CHILL and Ada compilers the need for a revision of VDM was identified. Other formal technique-approaches to software development and, especially additional formal specification constructs had been introduced in the late 1970s and early 1980s. DDC obtained in 1983 funds for a Formal Methods Appraisal study [26, 23]. It ended with a number of requirements that formal specification languages should satisfy when applied to problems involving distributed, real-time and concurrent systems.

**The RAISE Development** We make a distinction between the RAISE project and the RAISE product.

*The RAISE Project:* After the FMA project a Danish/British consortium was formed and eventually DDC, ABB (Asea Brown Bovery, DK), ICL (UK) and STL (UK), obtained a contract with the CEC for researching and developing a successor to VDM.

RAISE was thus initially developed under that contract, from 1985 to 1990, with the aim of providing a unifying improvement over formal methods such as VDM, Z, CSP, Larch and OBJ. RAISE was later further developed (after DDC) in the Project LaCoS (Large-scale Correct Systems using formal methods), from 1990 to 1995.

RAISE stands for **R**igorous **A**pproach to **I**ndustrial **S**oftware **E**ngineering and introduce the use of formal (mathematical) techniques in the development of software: in requirements analysis and formulation, specification, design and development.

*The RAISE Product:* There are three facets to the RAISE product. (i) The RAISE Specification Language (RSL) which provides a rich, mathematically based notation in which requirements, specifications and steps of design of software may be formulated and reasoned about. RSL is a wide-spectrum language: it facilitates abstract, axiomatic styles of description as well as concrete, operational styles. It may be used from initial domain and requirements analysis through design to a level at which the specification may be translated into code. (ii) The RAISE Method which provides a set of techniques and recommendations for how to use RSL in the various life-cycle phases of software development, as well as techniques for verifying properties of specifications, implementations and their relationships, formally, rigorously or informally. (iii) The RAISE Tool Set which supports the use of RSL and the RAISE method.

RAISE comes with comprehensive documentation. Books on the language RSL [13] and the method [14] are provided, and further information is found on Internet [11, 12, 20, 21].

### 3.4 Office Automation Projects

DDC concluded several office automation projects during its existence.

In 1981 a study was performed describing office automation systems. An office automation system was understood as a computer based system assisting the office staff in their daily tasks. At the time there was a growing amount of literature and commercial products concerning office automation.

The two main purposes of the project were:

- to contribute to the understanding of the office automation area by establishing a taxonomy and a terminology for the area;
- to contribute to the further development of the area by specifying a generic office automation system (a system with special features for adaption to the tasks and working methods of a specific office).

The project also helped technology transfer, as the project was carried out by persons from DDC together with persons from DDC members.

In the taxonomy part of the project, [8], the concepts of the office automation area were identified, analysed and classified. In the terminology part general terms, concepts and abbreviations used within office automation were explained and listed alphabetically. More than 70 items were explained.

To overcome the problems with a general office automation system: not fulfilling the requirements of a specific office, tailored system; being expensive to update as the needs of an office changes – a generic office automation system was specified and characterised. The generic office automation system was modelled using the VDM formal specification language. The model was also described informally and could thus be used by persons without knowledge of the VDM language.

Another office automation project was FAOR, Functional Analysis of Office Requirements. The project was supported by the CEC. Organisations from UK, Germany and Denmark participated in the project 1983–87. The Danish participant was a DDC member, ØK Data with DDC as subcontractor.

The purpose of the project was to develop a method, which could help systems analysts to analyse and specify the office automation requirements to IT-systems. The work processes in an office for which IT-support was needed in the 1980s became more and more complex and were often unstructured. The FAOR project addressed this challenge of analysing the complex office work and specifying functional requirements to IT-solutions. The method gives the analyst detailed instructions for the analysis and refers to a set of techniques and tools to use. The method was tried in field studies within the FAOR project with good results. The project is described in [25].

### 3.5 Technology Transfer

***Cubus:*** In 1987 DDC expanded its technology transfer activities, bringing its know-how out to Danish IT companies, by introducing a quarterly magazine Cubus[4].

The main content in Cubus was technical and scientific articles. They covered subjects, in which DDC had deep insight from actual project work. The articles were mainly addressed to an IT-knowledgeable audience, but they did not require special knowledge in the subject covered. Cubus also had articles for a wider audience and overviews/summaries of conferences, and it listed new DDC project reports.

*Seminars and Courses:* DDC also arranged seminars and courses covering subjects from DDC projects and topics of general interest, but within the area of software design and development. Examples of seminars subjects are: *Object oriented programming*; *Why and how to change Ada*; *Local area data net*; and *Selling Danish software worldwide*.

*Reports:* To a large extent the results of DDC projects were publicly available through reports prepared during the projects. The reports were listed in the magazine Cubus and sold for the reproduction costs.

## 4 Appraisal of DDC

In the mid 80s some 40 people worked at DDC. Half of them had a master degree in computing. The two large compiler projects Ada and CHILL lead to usable and useful products. But in the long run it was difficult to derive advantage and profit from these products. Ada did not gain the expected widespread use in spite of US DoD and CEC support. The knowledge and the rights for the Ada system were transferred to the subsidiary DDC–I Inc., which earned money but not sufficient to give a surplus to DDC. CHILL was developed for Danish tele-authorities as a new standard language for the international tele-community CCITT, but after completion CHILL didn't generate income for DDC. At the same time, these products had very little interest for several of the DDC members.

A third large project was the RAISE software engineering method. Its scope was too ambitious to be of immediate interest to DDC's members. The project was transferred to the software company CRI Inc. where it was completed, used, and marketed with some success.

Besides the above compiler-oriented projects DDC had a number of smaller software development projects, including user-friendly software and administrative database projects.

To assess what benefit DDC achieved for the Danish software milieu is difficult, but in our opinion the major effects of DDC were:

[4] Please note the DDC 'cube' adorning the 'abstract' of this paper.

⋆ Some of the larger Danish IT users and producers became aware of modern software development techniques.

⋆ A reliable, DoD-verified Ada system was produced and became the basis for the incorporated company DDC-I Inc. still existing.

⋆ RAISE and LaCoS – methods and tools for large-scale reliable software development – were developed and brought to market by CRI Inc., which took over leading staff and all project rights from DDC.

⋆ Between 50 and 100 young masters in computer science got experience using advanced software technology and carried it with them into other companies in Denmark and internationally.

⋆ DDC completed a number of large projects with better performance and higher product quality than was common in the 1980s.

Where DDC failed was to convince major Danish companies of the benefits of using reliable software development based on formal methods. (But then DDC did not try very much.) But some of DDC's software engineers went to work for CRI Inc. (later sold to Terma), where they proceeded using formal methods "lite" in major European Space Agency and US defense industry projects.

**Acknowledgement** This is not the time and place for thanking those many people and institutions who made DDC possible. But we do wish to acknowledge the help of Assoc. Prof., Dr. Hans Bruun (emeritus). Without his diligent and painstaking investigations of how to formalise the static semantics of CHILL and Ada, DDC could not have gotten off the ground. In writing this paper we got help from Messrs Peter L. Haff, Klaus Havelund and Jan Storbank Pedersen, and acknowledge with gratitude their help.

## References

1. E. Astesiano, C. Bendix Nielsen, A. Fantechi, A. Giovani, E.W. Karlsen, F. Mazzanti, G. Reggio, and E Zucca. The Draft Formal Definition of Ada, The Dynamic Semantics Definition. Technical report, Dansk Datamatik Center/CRAI/IEI/University of Genoa, January 1987.
2. H. Bekič, D. Bjørner, W. Henhapl, C. B. Jones, and P. Lucas. A Formal Definition of a PL/I Subset. Technical Report 25.139, IBM Laboratory, Vienna, December 1974.
3. D. Bjørner. Programming Languages: Linguistics and Semantics. In *International Computing Symposium 77*, pages 511–536. European ACM, North-Holland Publ.Co., Amsterdam, 1977.
4. D. Bjørner and C.B. Jones, editors. *The Vienna Development Method: The Meta-Language*, volume 61 of *LNCS*. Springer, 1978.
5. D. Bjørner and C.B. Jones, editors. *Formal Specification and Software Development*. Prentice-Hall, 1982.
6. D. Bjørner and O.N. Oest, editors. *Towards a Formal Description of Ada*, volume 98 of *LNCS*. Springer, 1980.
7. N. Botta and J. Storbank Pedersen. The Draft Formal Definition of Ada, The Static Semantics Definition. Technical report, Dansk Datamatik Center, January 1987.

8. J. Bundgaard, G. Schmeltz Pedersen, J. Storbank Pedersen, K. Hansen, P. Kvorning, and B. Nilsson. Kontor-Automations-Systemer (KAS) – Et studieprojekt. DDC document: DDC 04/1981-04-30.
9. G.B. Clemmensen and O.N. Oest. Formal specification and development of an Ada compiler – a VDM case study. In *Proc. 7th International Conf. on Software Engineering, 26.-29. March 1984, Orlando, Florida*, pages 430–440. IEEE, 1984.
10. DDC, Univ. of Pisa, CRAI, CNRS IEI Pisa and Univ. of Genoa, editor. *The Draft Formal Definition of Ada*. 1987. 3 parts. Published by Dansk Datamatik Center.
11. C.W. George. Download for the RAISE Tool Set. ftp://ftp.iist.unu.edu/pub/RAISE/method_book/. United Nations University's International Institute for Software Technology, P.O.Box 3058, Macao SAR, China.
12. C.W. George. UNU-IIST's RAISE Web Pages. http://www.iist.unu.edu/raise/. United Nations University's International Institute for Software Technology, P.O.Box 3058, Macao SAR, China.
13. C.WG. eorge, P. Haff, K. Havelund, A.EH. axthausen, R. Milne, C. Bendix Nielsen, S. Prehn, and K. Ritter Wagner. *The RAISE Specification Language*. The BCS Practitioner Series. Prentice-Hall, Hemel Hampstead, England, 1992.
14. C.W. George, A.E. Haxthausen, S. Hughes, R. Milne, S. Prehn, and J. Storbank Pedersen. *The RAISE Development Method*. The BCS Practitioner Series. Prentice-Hall, Hemel Hampstead, England, 1995.
15. P. Haff and A.V. Olsen. Use of VDM within CCITT. In *VDM – A Formal Method at Work, eds. Dines Bjørner, Cliff B. Jones, Micheal Mac an Airchinnigh and Erich J. Neuhold*, pages 324–330. Springer, Lecture Notes in Computer Science, Vol. 252, March 1987. Proc. VDM-Europe Symposium 1987, Brussels, Belgium.
16. P.L. Haff, editor. *The Formal Definition of CHILL*. ITU (Intl. Telecmm. Union), Geneva, Switzerland, 1981.
17. H.H. Løvengreen. Metodikker og værktøjer til konstruktion af programmel (KOMET). Report DDC-05, 229 pages, Februar 1981.
18. P. Naur and B. Randall, editors. *Software Engineering: The Garmisch Conference*. NATO Science Committee, Brussels, 1969.
19. O.N. Oest. VDM From Research to Practice. In H.-J. Kugler, editor, *Information Processing '86*, pages 527–533. IFIP World Congress Proceedings, North-Holland Publ.Co., Amsterdam, 1986.
20. J. Storbank Pedersen. Information about industrial use of RAISE. http://spd-web.terma.com/Projects/RAISE/project.html. Terma Inc., Herlev, Denmark.
21. J. Storbank Pedersen. Terma Information about the RAISE Tool Set. http://spd-web.terma.com/Projects/RAISE. Terma Inc., Herlev, Denmark.
22. J. Storbank Pedersen. VDM in Three Generations of Ada Formal Descriptions. In *VDM87, VDM - A Formal Method at Work*, volume 252 of *Lecture Notes in Computer Science*. Springer, March 1987.
23. S. Prehn and I.Ø. Hansen. Formal Methods Appraisal. Technical report, Dansk Datamatik Center, 1983.
24. G. Reggio, P. Invarardi, E. Astesiano, A. Fantechi, A. Giovani, F. Mazzanti, and E Zucca. The Draft Formal Definition of Ada, The User Manual of the Meta-Language. Technical report, CRAI/IEI/University of Genoa, January 1986.
25. K. Schmidt. Kontorarbejde og kontoranalyse. in CUBUS, July 1987.

26. S. Prehn, I.Ø. Hansen, S.U. Palm, and P. Gøbel. Formal methods appraisal, first report. Technical Report DDC 86/1983-06-24, Dansk Datamatik Center, Lyngby, Denmark, 1983.

## About the Authors

**Dines Bjørner** (DB) received his MSc and PhD from the Technical University of Denmark (DTU) 1962 and 1969; was with IBM 1962–75 (Stockholm, San Jose CA, Menlo Park CA, ESRI Geneve and Vienna Austria); was guest prof. at Copenhagen Univ. 1975-76 and a full prof. at DTU 1976-2007. DB was the founding and first director of UNU-IIST, Macau 1991-97. DB has published 6 books, edited 11 books and more than a hundred papers.

**Christian Gram** received his MSc in Mathematics from Copenhagen University 1956. 1957-62 he was ass. prof. at DTU. 1960-73 he worked at Regnecentralen, Copenhagen, with numerical analysis, programming, education, and wrote books and manuals. 1973-2000 he was full prof. in Datalogy at DTU, teaching programming methods and HCI. Member of the board of Dansk Datahistorisk Forening.

**Ole N. Oest** received his MSc and PhD from DTU in 1974 and 1978, respectively. Ole was with the Danish Defense Research Establishment 1978-80, then project leader and department manager at DDC 1980-85. Assistant professor at Copenhagen University 1984-85. 1985-95 managing director DDC International A/S in Lyngby, Denmark. Since 1995 president and CTO at DDC-I, Inc., Phoenix Arizona.

**Leif Rystrøm** got his MSc from DTU in 1965. 1967-1980 he was with Christian Rovsing A/S. 1980-1988 he was managing director for DDC. 1989-1998 he was an IT consultant with Jens Bisballe Planlægning A/S. 1998-2001 he was with Cisco Cable Products and Solutions. Since 2001 working with IT projects at the Department for Traffic and Incident Management at the Danish Road Directorate.