Åbo Akademi
Department of Computer Science

Annual Report 1989, 1990

Editors

Ralph-Johan Back and Aimo Törn

Åbo Akademi
Reports on Computer Science & Mathematics, Ser. B, No 10.
March 1991

Rapporter från Åbo Akademi
Informationsbehandling & matematik, Ser. B, Nr 10.
Mars 1991

ISBN 951-649-860-4
ISSN 0358-3392

Åbo Akademis kopieringscentral
Åbo 1991

Contact information:

Åbo Akademi
Department of Computer Science
Lemminkäisenkatu 14
SF-20520 Turku, Finland

telephone: +358-21-654311
telefax: +358-21-654732
e- mail node: finabo.abo.fi

Contents

Chapter 1

Summary

1.1 Background

Åbo Akademi is the main Swedish-speaking university in Finland: teaching, research and administration are all in Swedish. Its purpose is to provide higher academic education for the Swedish-speaking population in Finland, but it is also open for students with Finnish as their native language. Academic education in Swedish is also offered by the Swedish School of Business Administration in Helsinki, the University of Helsinki and the Technical University of Helsinki, the latter two being bilingual (Finnish and Swedish).

Åbo Akademi has six faculties, Humanities, Mathematics and Natural Sciences, Economics and Political Science, Chemical Engineering, Theology and Education. The Department of Computer Science belongs to the Faculty of Mathematics and Natural Sciences (Figure 1.1). It is the only department in Finland that offers a full undergraduate and graduate programme for Computer Science in Swedish.

The department moved into new and modern facilites in February 1988, located in the new technology centre DataCity in Turku.

1.2 Personnel

At present, the department has 15 permanent positions, 13 of these belonging to the teaching faculty. In 1989 one of the instructor positions was changed to a lecturer posititon, an assistant professor of the Faculty of Chemical Engineering was placed in the department and a laboratory manager position was obtained. The permanent positions and the persons presently appointed to these are as follows:

Professor: Ralph-Johan Back, 1983 -
Professor: Aimo Törn, 1990 -
Associate professor (Administrative data-processing): Vacant
Lecturer: Ragnar Wikman, 1988 -
Lecturer: Patrik Eklund, 1989 -
Assistant professor [1] **(Computer science and mathematical statistics)**: Vacant
Assistant professor [1] [2] **(Systems Programming)**: Kaisa Sere, 1989-
Assistant professor [1]: Vacant

[1] "Överassistent" in Swedish
[2] Faculty of Chemical Engineering, position placed in our department

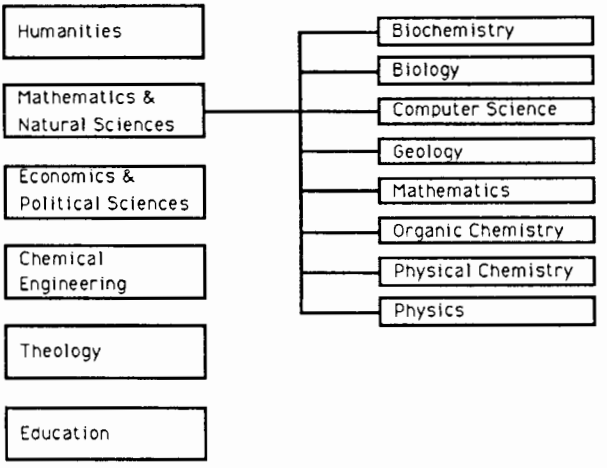

Figure 1.1: Position of the department within the university.

Assistant (Administrative data-processing) : Inger Eriksson, 1988 –
Assistant : Ulla Solin, 1983 –
Assistant : Hong Shen, 1990 –
Instructor: Paul Lindholm, 1985 –
Instructor: Vacant
Laboratory manager: Mats Aspnäs, 1989 –
Department secretary: Christel Engblom, 1988 –

In addition, the department has two docents, i.e., experienced researchers from other universities who may teach courses: Per-Åke Larson (Waterloo University, Canada) and Markku Nurminen (University of Turku).

The actual staff at the department during 1990 is shown in appendix A. Also people employed in the research projects are listed in the appendix. More detailed information about the personnel is given in Chapter 10.

1.3 Research

The main research areas of the department are the following:

1. *Programming methodology*: formal methods for program construction, construction of parallel and distributed systems, logic programming methodology, environments for program derivation and verification, mechanical theorem proving.

2. *Multiprocessor systems*: multiprocessor architecture and software, parallel programming methods, debugging and monitoring software for parallel systems, animation of parallel programs, parallel programming applications.

3. *Information systems*: human aspects in information system construction, end-user participation in system design, information system quality, software metrics, file organization, object-oriented modelling of information systems.

4. *Decision support*: global optimization methods, use of Petri nets in system simulation, theory of fuzzy sets.

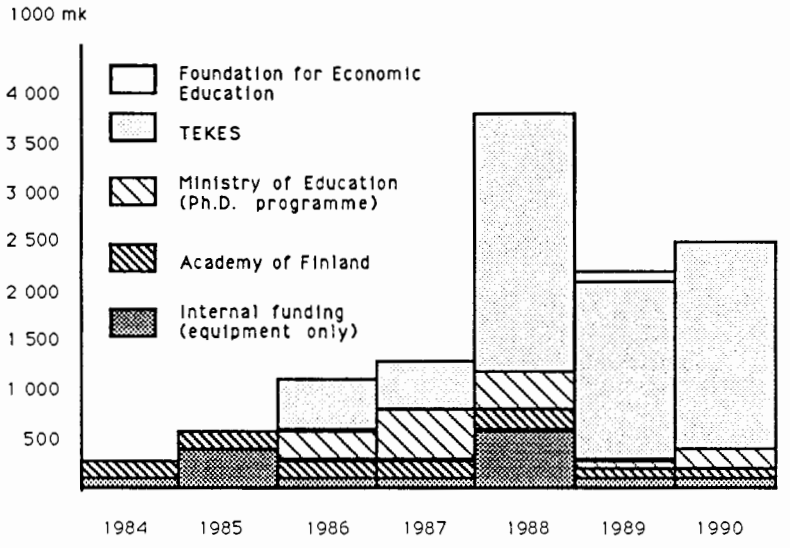

Figure 1.2: The structure of research funding

5. *Artificial intelligence*: knowledge based systems, medical expert systems, image analysis.

The main externally funded research projects in the department during the period 1989 and 1990 are:

A. *Millipede-project* (Ralph-Johan Back, 1988–1991), funded by TEKES.

B. *Centipede-project* (Ralph-Johan Back, 1988 – 1991), funded by TEKES.

C. *FINSOFT III co-ordination project* (Ralph-Johan Back, 1988–1991), funded by TEKES.

D. *SOLE-project* (A. Törn, 1988-1991), funded by the Academy of Finland and the Foundation for Economic Education.

E. *GeDeMeDeS-project* (Patrik Eklund, 1990 – 1992), funded by TEKES.

In addition, the department has co-operated actively with the *Knowledge and Work* – project led by Markku Nurminen at the University of Turku, 1986–1989, and funded by the Academy of Finland.

An important source for funding has also been the three national programmes for doctoral education in computer science, initiated by the Ministry of Education 1985 and ending 1990. A number of people from the department have been employed in this programme for shorter or longer periods. The programme has also funded some equipment.

The internal funding of Åbo Akademi has also been important, both for employing temporary personnel for the research projects and for aquiring equipment. The structure of research funding 1984–1990 is shown graphically in Figure 1.2 (for internal funding, only equipment is shown). Table 1.1 shows the main research projects and their funding 1988–1990.

The research in these areas is described in more detail in Chapters 3 – 7. The department's involvement in organizing scientific meetings is described in Chapter 8. Chapter 9 lists

	1988	1989	1990
A. Millipede (TEKES)	1 100	950	990
B. Centipede (TEKES)	800	750	790
C. Finsoft-III (TEKES)	250	250	250
D. Sole (Academy of Finland, Foundation for Economic Education)		220	150
E. GeDeMeDeS (TEKES)			150
Total (1000 mk)	2 860	2 170	2 330

Table 1.1: External funding of main research projects.

	1989	1990
1. Books	1	1
2. Journals	6	14
3. Collections	2	3
4. Proceedings	19	12
5. Technical Reports	20	11
6. Theses for Ph.D.	-	3
7. Theses for Ph.Lic.	2	1

Table 1.2: Publications in 1989 and 1990.

professional organizational activities. Chapter 10 describes the individual research interests and publications in 1989 and 1990 of the staff. In table 1.2 the number of publications in 1989 and 1990 are given. Chapter 11 describes scientific contacts in 1989 and 1990, both foreign visitors at the department and research visits and conference participation by staff and Chapter 12 contains accepted theses. Chapter 13 contains the publications, including abstracts, of department personnel in the period 1989 and 1990.

1.4 Teaching and degrees

The curriculum requires the students to take 160 credits for their M.Sc.degree. This is divided into *general studies* (15 credits), *subject studies* (105 credits) and *advanced studies* (40 credits, includes the M.Sc. thesis). Computer science accounts for at least 92 credits, mathematics and statistics for at least 30 credits and business administration for at least 15 credits.

There are five specialization areas currently, corresponding to the areas of research listed

	1989	1990
Intake	34	15
Degrees:		
Ph.D.	-	3
Ph.Lic.	2	1
M.Sc.	8	16

Table 1.3: Intake of students and degrees in 1989 and 1990.

above: programming methodology, multiprocessor systems, information systems, decision support and artificial intelligence.

The students are encouraged to choose the supporting disciplines to complement their specialization area in computer science. Several possibilities exist, ranging from mathematics, statistics and economics to psychology. As a standard mathematics (22 credits), business economics (15 credits) and statistics (8 credits) are recommended.

In table 1.3 the intake of students and awarded degrees during the years 1989 to 1990 is shown.

A more detailed description of the curriculum in computer science is given in Section 2.

1.5 Facilities

The department mainly uses Apple Macintosh computers and Sun workstations for research and teaching. Macintosh computers are used as personal workstations for text processing and terminal emulation. Most of the staff has a Macintosh computer for personal use. Sun workstations are mainly used for programming and advanced applications.

The departments Macintosh computers are connected to a LocalTalk network. They are also connected to Åbo Akademis Ethernet network via terminal servers, and can be used as terminals to the universitys central computers (Sun4/280, Solbourne 5/800, VAX 8800).

For students use, there are 6 Macintosh II computers (3 with 19" colour screens) and 2 Sun 3/50 workstations. There is also a laboratory with 12 Nokia Mikro Mikko 3 (IBM PC/AT compatible), which now are being upgraded to IBM PS/2 computers, and a laboratory with 16 Falco terminals connected to the university local area network.

For more advanced projects there is a laboratory with two Macintosh II computers and one Sun SparcStation 1+, all with 19" colour screens. These are mainly used in research projects. The department also has a Sun 3/160 workstation, which is connected to the Hathi-2 multiprocessor system.

The department has a large transputer-based multiprocessor system, Hathi-2, which was designed and built in the Hathi-project (finished 1988). It now serves as a central multiprocessor in the FINSOFT III research programme. The department's Sun 3 is used as a front end to Hathi-2, which enables users outside Åbo Akademi to access the multiprocessor system via the network.

The university is served by excellent network connections. A local area network connects

terminals, microcomputers, workstations and mainframes at all three university institutions in Turku. A 64Kbit line connects us to the Finnish University Network FUNET. The FUNET network is connected to NORDUNET, the network of Nordic universities and NORDUNET is connected via satellite to networks in Europe and North America.

The computing facilities available to the department are shown in Figure 1.3.

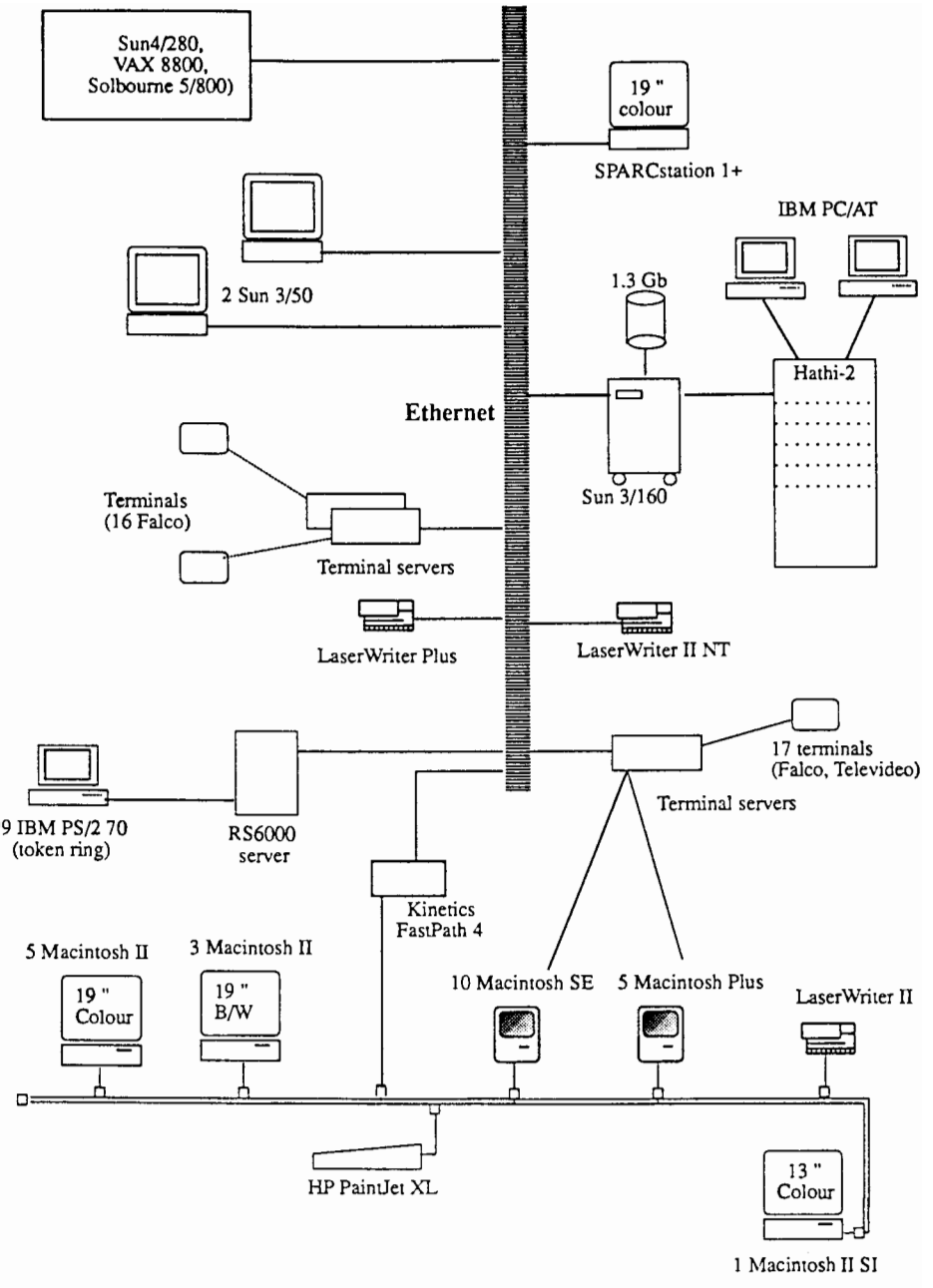

Figure 1.3: Facilities in the department

Chapter 2

Teaching

2.1 Undergraduate studies

The current curriculum allows students to choose their own area of interest and specialization in a much higher degree than earlier. Possible specialization areas are programming methodology, multiprocessor systems, information systems, decision support and artificial intelligence. The supporting disciplines are chosen to fit computer science studies, mathematics, statistics and business administration being the main supporting fields. The theses at the department during 1989 and 1990 are listed in Chapter 12.

The modules of the curriculum are listed below. The unit credit measures student work. One credit should ideally correspond to 1 week (40 h) of work averaged over the students. The total credits required for a Master of Science degree is 160 credits. This should be achievable in four to five years, although in many cases it does take longer.

A. **General studies** (17 credits)

1. Languages (Finnish, Swedish, English)
2. History of computer science and theory of science
3. Introduction to computer science
4. Programming techniques I
5. Digital electronics (Dept. of Physics)
6. Study orientation

B. **Subject studies** (106 credits)

1. **Computer science (43 credits)**
   1. Programming techniques II
   2. Data structures
   3. Programming methodology
   4. Computer architecture
   5. Logics
   6. Operating systems
   7. Data bases
   8. System development methods
   9. Seminar in computer science
2. **Mathematics (22 credits)**
   1. Introduction to calculus

2. Introduction to probability theory
3. Algebra A
4. Algebra B
5. Matrices
6. Applied Mathematics

3. Statistics (8 credits)
1. Statistics
2. Introduction to statistics II

4. Business administration (15 credits)
1. General business economics
2. Accounting and book-keeping
3. Marketing

5. Courses from one of the following:
1. Mathematics (5 credits)
2. Statistics (5 credits)
3. Business economics (5 credits)

6. Courses of the student's free choice (10 credits):

C. **Advanced studies** (40 credits)

1. Three advanced courses in computer science (15 credits)
2. Seminar (2 credits)
3. Special assignment work (7 credits)
4. M.Sc. thesis (16 credits)

A more detailed description of the regular courses of the department is given in Appendix B.

The courses at the advanced level are not necessarily given on a regular basis. The courses are both for undergraduate and graduate students. Below is a sample of advanced courses that have been given. Please note, that this classification is generalized, several areas are overlapping.

Programming Methodology
- Stepwise refinement methods
- Algorithm development
- Program specification methods
- Compiler theory

Multiprocessor Systems
- Data communication
- Computer architecture
- Real time systems
- Parallel and distributed processing
- Multiprocessor systems
- Theory of distributed systems

Information Systems
- EDP, the individual and society
- System development, tools and trends today
- Software administration and system development

Software quality
Object-oriented programming and computerised shared material
Computer systems in organizations

Decision Support
Simulation techniques
Petri nets

Artificial Intelligence
Artificial intelligence
Logic programming and expert systems
Pattern recognition
Knowledge based systems
Neural Networks

2.2 Graduate studies

Graduate studies aim at a Ph.D. in computer science, with a Ph.Lic. (Licentiate degree) as an intermediate degree. The Ph.Lic. degree requires an additional 100 credits after the M.Sc. degree, while the Ph.D. degree requires an additional 160 credits after the M.Sc. degree.

Ph.Lic. degree (100 credits)

1. M.Sc. degree
2. Advanced courses in computer science (15–25 credits)
3. Advanced courses in supporting disciplines (15–25 credits)
4. Ph.Lic. thesis (60 credits)

Ph.D. degree (160 credits)

1. M.Sc. degree
2. Advanced courses in computer science (15–25 credits)
3. Advanced courses in supporting disciplines (15–25 credits)
4. Ph.D. thesis (120 credits)

The difference between a Ph.D. and a Ph.Lic. is only the requirements put on the thesis. The Ph.D. thesis must have a more substantial and original academic contribution than the Ph.Lic. thesis. The results in the Ph.Lic. thesis may be reused in the Ph.D. thesis. Hence, in most cases graduate students take the intermediate Ph.Lic. degree. This is encouraged also by the fact that it is the formal requirement for assistant professors and lecturers.

The main part of the postgraduate studies are done within research projects. The students are encouraged to write their Ph.Lic. thesis as a collection of conference and journal papers and later upgrade these to meet the requirements for a Ph.D. thesis.

Chapter 3

Research on Programming Methodology

Formal methods in programming are being studied in the Centipede project, lead by Ralph-Johan Back. This project is funded by TEKES and is part of the FINSOFT III research program. It was started April 1988 and will end in March 1991. Total funding is approximately 2 300 000 mk (the actual funding is decided on an annual basis).

Aimo Törn has initiated research on the construction of software for system simulation. The technique derived, Simulation Nets, is based on Petri Nets. Another research direction has been the use of graphical methods to support program design and verification (PICA), pursued as a sub-project to the SOLE-project.

Jan Komorowski has continued his work on program transformations and programming environments. The main theme in this research is partial deduction and environments for program derivation.

The main research directions within this research area are listed below, roughly in the order in which they were initiated at Åbo Akademi.

3.1 Program Information Charts with Assertions

This project is a subproject of SOLE investigating the effects on quality of software by using formal program development methods. The research aims at developing a technique and a tool PICA for formal program development using flowcharts. The formal technique applied is that of proving theorems of the type $\{pre\text{-}condition\}$ *algorithm* $\{post\text{-}condition\}$. This technique is applied to structured program flowcharts by adding assertion nodes containing program variable names and assertions about their values. An assertion node is connected to or from a statement node depending on if it represents a pre-condition or a post-condition.

A tool for convenient use of the technique has been implemented as an Add-On to the Design software of Meta Software on a Macintosh II by Benita Heinonen. The tool aids in the design and updates to the design. It keeps track of what is and what is not proved and reminds the user of the points necessary to check in a proof step. The use of the tool is described in an article published in the journal Acta Cybernetica [Törn 90c].

3.2 Simulation Nets

A tool, SimNet for design of software for simulating discrete event systems is developed. The tool is an extension of the well known Petri Nets. The idea is to graphically describe the objective system using Simulation Nets. These are exact enough to permit simulation to be carried out directly without furher programming.

One prototype capable of reading a Simulation Net design and carrying out the implied simulation has been written in Simula by Antti Raunio. This work has been continued by Törn. The use of the tool was described in [Törn 90b].

Work on a book, Simulation Modelling for illustrating the use of Simulation Nets in modelling simulation designs for computer simulation is in progress.

3.3 Formal Construction of Distributed Systems

This project, which was a joint effort between Ralph-Johan Back and Reino Kurki-Suonio (Tampere University of Technology), was started in 1983 [Back and Kurki-Suonio: Decentralization of Centralized Process Nets, *ACM Confererence on Distributed Computing*, Montreal 1983]. The aim was to develop formal methods for constructing distributed computer systems. These systems present programming problems that are often very difficult, and where the pay-off of formal methods is therefore potentially large. The work has centered around a specific model for describing the behavior of distributed systems, called *action systems.* The behaviour of a parallel and distributed system is described in this model in terms of the possible actions that can take place during system execution. The advantage of this method is that it becomes relatively easy to prove properties of the whole system execution, and standard techniques based on temporal logic can be applied directly to the verification of distributed systems [Back and Kurki-Suonio 89].

The main emphasis in the theoretical study of action systems has been on fairness issues. The fairness properties that were desirable for program construction were identified. Another set of fairness properties that could be achieved in an efficient manner in a distributed implementation was also identified. The relationship between these fairness properties was studied, especially the conditions under which the implementable fairness notions implied those needed in program construction. Implementations that guaranteed the more basic fairness properties were described and proved correct.

A particular concern has been to find efficient distributed implementations of action systems. The simplification of program design that the action system formalism permits is partially offset by a more complicated implementation problem.

Joint work by Ralph-Johan Back and Kaisa Sere shows, that efficient distributed implementations of action systems for point-to-point networks are possible, provided that certain reasonable restrictions are accepted [Back and Sere 89]. These restrictions are of the same nature that forbids the use of output guards in CSP and Occam. This work was initiated in the Hathi-project, and is being continued in the Centipede project. There is already one implementation of (unrestricted) action systems based on a protocol by Bagrodia (U. of Texas), built by Lena Ståhl (M.Sc. thesis) [Ståhl and Back 89], but it is not very efficient. Work is in progress in Centipede on building an efficient implementation of action systems for the Hathi-2, by Peter Dahl (M.Sc. work) and Kaisa Sere.

3.4 Calculus of Program Refinement

A central role in the Centipede project is played by the *Refinement Calculus*, described by Ralph-Johan Back in his Ph.D. thesis 1978. This provides a general mathematical theory

for the stepwise refinement approach to program construction. This calculus has become quite intesively studied and further developed during the last three years, especially in the United Kingdom (Oxford, Glasgow), in the Netherlands (Groningen), USA (Digital Equipment Corporation), Australia (Queensland) and at Åbo Akademi. The research on formal methods in Centipede concentrates on developing theory and tools to support this approach for program construction.

The refinement calculus extends the weakest precondition technique of Dijkstra to procedural and data refinement, and can also be used for stepwise refinement of parallel and distributed programs. Algorithms are derived by a series of correctness preserving refinements and program transformations from very high-level specification. The derivation is carried on until a program that meets the stated criteria of efficiency and implementability has been constructed.

The mathematical basis for the refinement calculus has been further studied and extended to handle difficult program features such as unbounded non-determinism, full recursion and procedures with parameters. The original method for data refinement has also been extended to permit non-functional data refinements [Back 89a].

A lattice-theoretical basis for specification languages with predicate transformer semantics has recently been developed in co-operation with Joakim von Wright [Back and vWright 89a, 89b, 89c]. This has provided a unified theory of program refinement, covering both demonic and angelic non-determinism, miraculous statements, action systems, bounded and unbounded non-determinism, as well as data abstraction and procedures. All these features seem to fit quite well within the simple lattice-theoretical framework. As a side product, this framework also provided a considerable generalization of Dijkstra's original weakest precondition technique. Joakim von Wright got his Ph.D. degree for this project [vWright90c].

A mechanized formalization of the refinement calculus, using some existing proof assistant software such as Higher Order Logic (HOL) is also being done by Joakim von Wright. This work uses the lattice-theoretical framework described above [Back and vWright90b]. The aim is in the first phase to be able to mechanically verify the correctness of different kinds of program refinement and transformation rules. At a later stage, this mechanical axiomatization will be integrated with the Centipede program derivation environment.

3.5 Stepwise Refinement of Parallel and Distributed Systems

One of the main original incentives for developing the action system approach was that it made it possible to construct distributed systems by stepwise refinement, by a method now known as *superposition*. This method was developed in 1983 by Ralph-Johan Back and Reino Kurki-Suonio (see [Back and Kurki-Suonio 89] for a survey).

Interest in this method was renewed when it was realized that it could also be applied to proving total correctness of action systems within the refinement calculus. Joint work by Ralph-Johan Back and Kaisa Sere has then continued in the Centipede project on this approach, which has proved to be very fruitful [Back 89b, Back and Sere 89a] . A systematic method for stepwise refinement of parallel programs within the refinement calculus has been developed, and it has been applied to a number of quite difficult parallel and distributed algorithms, including matrix multiplication and solving linear systems of equations on a ring of processors, and the implementation of processor farms on different kinds of processor topologies. Kaisa Sere got her Ph.D. degree on this topic [Sere 90b].

A drawback of the approach has been that it could only be used to construct parallel algorithms, i.e. programs for which only their input–output behaviour was of interest. This kind of systems are very common in multiprocessor applications, where the main purpose is

just to parallellize sequential algorithms to gain speed. However, reactive systems, for which the behaviour of the system during execution is also important, tend to arise quite easily when breaking a single parallel program into modules. This kind of distributed programs could not be handled originally in the refinement calculus. However, recently Ralph-Johan Back also showed how to use the technique previously developed for data refinement to model the stepwise refinement of reactive action system [Back 89c]. The actual method generalizes and formalizes an earlier method proposed by Leslie Lamport. Hence, the weakest precondition technique of Dijkstra, on which the refinement calculus is based, becomes applicable to the stepwise refinement of both parallel and reactive systems [Back and Kurki-Suonio 90].

3.6 A Graphical Environment for Program Refinement

One of the main problems in constructing programs by stepwise refinement is to master the sheer number of details and administer the large number of derivation steps and successive program versions that arise in a derivation. In addition, one needs to show that the successive program refinements do preserve correctness, i.e. that no errors are introduced during the refinement. When doing refinements by hand, it is also very easy to make simple clerical errors which lead the derivation astray.

A central task in the Centipede project is therefore to construct a workstation–based programming environment to support the stepwise refinement method and mechanize as much as possible of the derivation process. The environment will support the formal derivation of sequential and parallel algorithms, as well as derivation of reactive programs. Verification aspect of the methodology will be supported by the HOL systems.

The formal derivation environment will be complemented by a simulation and animation environment where intermediate versions in the program derivation can be executed and their efficiency (speed, degree of parallellism, efficiency bottlenecks etc.) studied. A compiler which takes an action system and compiles it to Occam for execution in the Hathi–2 multiprocessor will also be implemented, as described above. The environment will thus provide an integrated toolset for the whole derivation process, from initial high-level specification via formal derivations and actual testing of intermediate versions to a multiprocessor application running on a transputer system.

A preliminary design and prototype of the program refinement environment was constructed by Ralph-Johan Back on the Mac II using a hierarchical graph editor (Design). A more advanced version has been constructed by Jan Komorowski and Patrick Waxlax (M.Sc. thesis) in Prolog on the Mac II, while Jukka-Pekka Hekanaho (M.Sc. thesis) has studied an alternative implementation of the environment, based on the program transformation system Refine on Sun-4 and is now working on the integration of the different environments. Dan-Johan Still (M.Sc. work) is working on implementing the simulated parallel execution environment for action systems. A collection of program refinement rules has been included in the system (integrating these rules with another collection of rules designed at Oxford University/PRG) [Sere 90a]. Sten Agerholm (M.Sc. work) has been working on the HOL-formalization together with Ralph-Johan Back and Joakim von Wright.

3.7 Partial Deduction

Partial deduction (formerly known as partial evaluation in logic programming) was introduced in Jan Komorowski's Ph.D. thesis from 1981. The importance of partial deduction as an optimization technique has been realized more and more lately, and there is now a substantial and growing interest in it.

Informally, partial deduction can be described in the following way. A deductive system is a triple consisting of a set of well-formed formulae (wff) over a language, a set of axioms and a set of deductive rules. Given a deductive system and a wff, partial deduction generates a new set of axioms and, possibly, a new set of deductive rules, both of which are simplified with respect to (attempting) proving the formula to be a theorem of the system. The resulting deductive system should have the same set of conclusions and possibly valid formulae as the original one with respect to the wff. The aim of partial deduction is that the new deductive system is more efficient for proving the formula and its instances than the original system.

Traditionally, partial deduction has been applied to program optimization and compilation of embedded languages implemented via meta-interpreters. There have been, however, several problems which hampered many other applications. The research has extended standard partial deduction to alleviate some of these problems, using one of two tactics. The first one is called opening and helps control the exponential explosion of generated code. The second one handles recursive definitions and is called abbreviating. Extended partial deduction turns out to be a powerful synthesis, not only optimization, method for obtaining efficient programs from possibly inefficient specifications. In particular, it can automatically formulate correct inductive schemata. Moreover, proofs of problems like McCarthy's verification problem, which in other systems may be lengthy and require human-supplied lemmata, becomes straightforward and entirely automatic.

In general, it is shown that extended partial deduction supports advanced programming techniques like data, procedural and metalinguistic abstraction mechanisms as well as replacement-in-context (a form of stepwise refinement) [Komorowski 89a, 89b] and is a rather universal concept with applications in program optimization and synthesis, machine learning, compiler generation, databases, etc.

The theoretical research has been complemented with a development of a practical environment for synthesizing programs in the framework of partial deductions. Johan Lahtivuori (M.Sc. work) has implemented the first version of the environment.

Chapter 4

Research on Multiprocessor Systems

The Department of Computer Science at Åbo Akademi is responsible for a research programme on parallel computation and neural networks. It is part of the FINSOFT-research programme funded by TEKES, constituting subprogram III (FINSOFT III: Parallel computation and neural network). The duration is April 1988 – March 1991. Ralph-Johan Back was asked by TEKES to plan this subprogramme, and is now the director and scientific leader of it. FINSOFT III consists of 12 different research projects (two projects each in Åbo Akademi, VTT/TKO, Tampere University of Technology and in University of Helsinki and one project in VTT/TIK, Technical University of Helsinki, University of Jyväskylä and in Lappeenranta University of Technology). The total budget for this subprogramme is approximately 18 millon mk, for the whole 3-year period. The co-ordination of FINSOFT III is done in a special project at Åbo Akademi, with a total budget of approximately 750 000 mk. Atte Kortekangas from VTT/TIK is the administrative co-ordinator of the programme.

The Millipede-project is one of the FINSOFT III projects. It was started in April 1988 and ends in March 1991. The total TEKES funding for the 3-year period is approximately 3 000 000 mk (the actual funding is decided on an annual basis). Project leader is Ralph-Johan Back. The main emphasis is on building a user-friendly interface to the multiprocessor system, which hides the details and machine dependencies of the underlying hardware.

4.1 Multicomputer Architecture

The work in Millipede centers on the Hathi-2 multicomputer. This is a reconfigurable general purpose multiprocessor system consisting of 100 32-bit Inmos floating point transputers (T800), 25 16-bit transputers (T212) and 25 Inmos crossbar switches (C004). The system can be characterized as a loosely coupled MIMD multiprocessor, with a reconfigurable distributed interconnection network and a modular design. It consists of 25 identical boards, with the crossbar switches connected to each other in a torus. The crossbar switches form a distributed switching network which can be reconfigured by software. The T212 transputers form a separate control system that controls the distributed switching network and monitors the activities in the system. The parallel computing power of the system is 1000 (RISC-) MIPS/ 150 MFLOPS. A general overview of the Hathi multiprocessor system is given in [Aspnäs et al 89b].

Hathi-2 is used as a back-end computing resource, with the department's Sun-3 serving as front end. The system can be shared between a number of simultaneous users, by partitioning

it into several smaller independent multiprocessor systems. The host is connected to the university local area network, and can be accessed via it from other places in Finland. It now serves as a central computing resource for the projects in FINSOFT III.

The Hathi-2 system has been developed further within the Millipede-project. The amount of memory has been increased to 1.25 Mb per processor, giving the system a total of 125 Mb of central memory. John Aspnäs, Mats Aspnäs and Tor-Erik Malén have been in charge of the further development of the Hathi-2 system, in co-operation with VTT/TKO in Oulu.

The hardware project has generated one Ph.D. thesis in engineering (Kari Pehkonen) at the University of Oulu. A patent application for the switching network used in Hathi-2 has also been made.

4.2 Monitoring and Debugging Multiprocessor Systems

The monitoring utility in Hathi-2 was built in the previous Hathi project and is being further developed in the Millipede project. It is used for monitoring the utilization of the resources in the multiprocessor system during program execution. It is needed for finding bottlenecks in parallel programs executing on the system, and to provide information about the load balance of the programs. Monitoring is done by observing the CPU and link activity in the transputer network. The monitoring software is based on the monitoring hardware built into the Hathi-2 architecture, which makes it possible to monitor the system without introducing any substantial overhead on the main computation.

Monitoring data, i.e., data about CPU and link utilization on the transputers executing the monitored program, is gathered by the transputers in the control system. This data is sent through the control system to a host computer, where it is stored in a file and presented later to the user. The user can browse through the monitoring data both backwards and forwards in time at different timesteps and also look at the average utilizations during the whole execution. The project is co-operating with Teemu Kerola (University of Helsinki) on the performance evaluation aspects of the monitoring subsystem.

The initial version of the monitoring system was implemented by Stefan Levander (M.Sc. thesis) in the Hathi-project. The systen has been analyzed and considerably improved by Mats Aspnäs, Tor-Erik Malén and Thomas Långbacka (M.Sc. thesis) in the Millipede project [Aspnäs and Kerola 90], [Aspnäs and Långbacka 90]. The graphical environment for presenting the results of the monitoring to the user has been implemented by Henrik Gullberg (M.Sc. work).

4.3 Configuration of Transputer Systems

The initial version of the configuaration software for Hathi-2 was constructed by Tor-Erik Malén (M.Sc. thesis), in co-operation with Patrik Eklund. This system enables the user to connect the processors in Hathi-2 into different topologies, like toruses, trees and different irregular structures. The configuration system has been developed further in the Millipede project.

Work on developing efficient mapping algorithms, wich automatically maps a parallel program onto a processor network, has been carried out within the Millipede project by Mikael Norrbo (M.Sc. thesis) and Hong Shen (Ph.D. thesis). The main problems are algorithmic in nature: the general mapping problem is known to be NP-complete, so the effort is focused on finding more efficient heuristic algorithms to solve the problem. Hong Shen has investigated two different approaches to this problem, Divide-and-Conquer Mapping and Self-adjusting Mapping [Shen 89b, 89c, 89d]. A mapping algorithm can be functionally divided into three parts: grouping, placement and routing, where the routing is a special kind of path-disjoint

routing that is also NP-hard in general. The heuristic criteria for solving the path-disjoint routing problem have been studied closely and a fast mapping algorithm has also been implemented.

The mapping utility developed for Hathi-2 automatically maps a parallel program structure on to the transputers in Hathi-2 and establishes the needed link connections between the transputers. The input from the user to the mapping utility consists of the process graph of the distributed program. As output, it generates the configuration statements needed by the Occam configurer to place this program structure on to the physical processor network. The mapping utility also generates the commands needed by the reconfiguration software to connect the transputers into the desired topology. The method guarantees that an arbitrary process graph can be mapped onto the hardware structure of Hathi-2. This is acheived by iterating the grouping, placement and routing phases and grouping processes together when necessary, until the mapping succeeds.

The mapping utility makes it possible to hide the physical structure of Hathi-2 from the user. This is a very useful feature when writing parallel programs, since the design of the configuration statements is considered to be difficult and very error-prone. The mapping algorithm has been tested and compared with the previously implemented mapping algorithm, which was based on Bokharis algorithm, and it has proved to give very good results.

4.4 Parallel Algorithm Animation

Algorithm animation means visualization of program execution. The aim is to show the dynamic working of an algorithm during execution, in graphical form. Animation is a powerful tool for program debugging and for identifying excecution bottlenecks. This is especially true for parallel programs, as animation can give an overall perspective of a parallel execution that can be very difficult to grasp in any other way.

The research has produced a non-interferent method for animating parallel algorithms. This method is based on uniformly delaying the execution of all processes in the program, so that execution runs at a speed that is convenient for the user. The delaying technique makes it possible to simulate the execution of a many-processor parallel program on a single processor.

The project aims at developing an animation environment for debugging, testing and analysis of Occam programs. A prototype system has already been built in the Hathi-project, and is now running on the Hathi-2 Sun-workstation host. The animation technique has been developed by Ulla Solin in her Ph.Lic. thesis. The prototype implementation has been done together with Lena Ståhl, Yngve Nyman (M.Sc. thesis) and Marie-Louise Lindström (M.Sc. thesis). Ulla Solin is expected to get her Ph.D. degree from this project within the next year.

4.5 Multiprocessor Programming and Applications

One of the main goals in the previous Hathi-project was to try out the value of parallel computations in practical applications. To this end, the department initiated a number of applications projects, some internal to the department and some in co-operation with other research institutes. Most of these applications turned out to be quite successful and deliver the efficiencies expected. The applications were initially carried out on a smaller 16-transputer system (Hathi-1), and then ported to the larger Hathi-2 when it became ready, in order to measure speedups and efficiencies when more masssive parallellism was available.

The more important application projects were the following:

- *Fluid dynamics in two and three dimeensions.* The application was developed in co-operation with the Department of Heat Engineering at Åbo Akademi, by Tom Björkholm

(M.Eng. thesis) , Pekka Kuusela (M.Sc. thesis), Tor-Erik Malén, and Göran Öhman.

- *Real-time transformation of satellite pictures.* The application was developed in co-operation with VTT/TIK, by Atte Kortekangas, Aarne Rantala, Antti Raunio and Dan-Johan Still [Rantala and al. 89].

- *Three-dimensional cluster indentification in nuclear accelerator data.* This application was done in co-operation with the Departments of Physics at the University of Jyväskylä, by Jorma Hattula, Jens Granlund, Tom Lönnroth and Patrick Waxlax.

- *A multiprocessor system for full-text retrieval*, by Marina Walldén (M.Sc. thesis) and Kaisa Sere [Wallden and Sere 89].

- *Parallel implementations of some global optimization algorithms*, by Aimo Törn and Antanas Žilinskas [Törn and Žilinskas 90], [Törn 90e] .

Multiprocessor applications play a lesser role in the Millipede project, mainly because many of the other projects in FINSOFT III are now engaged in this kind of applications gathering experience that can be shared through the FINSOFT III co-operation.

An overview of these applications is given by Aspnäs and Back [Aspnäs and Back 89a].

4.6 Graphical Interface for Multiprocessor Programming

An integrated programming environment for Hathi-2 is being developed within the Millipede-project. The purpose is to hide the hardware structure from the programmer so that he only has to work with a simple conceptual parallel programming model. This model is basically the Occam model of logical (possibly nested) processes connected by logical communication channels. This environment integrates the existing programming tools, such as editor, compiler, post-mortem debugger, reconfiguration tool, mapping tool and performance monitoring system under a common graphical user interface, allowing the programmer to specify the process structure of a distributed program as a graph. The programmer can group processes together into hierarchical structures, which can be automatically allocated to the processors in Hathi-2 by the mapping utility. The mapping utility also automatically configures the multiprocessor system to the desired topology. The monitoring utility will also be integrated into the programming environment. The results from monitoring are presented to the user in a graphical way, where the presentation is based on the processor structure of the distributed program. The programming environment has been designed by Mats Aspnäs and Ralph-Johan Back [Aspnäs and Back 89], and is implemented by Jens Granlund (M.Sc. work), Henrik Gullberg (M.Sc. work), Thomas Långbacka and Elena Trishina.

Chapter 5

Research on Information Systems

The department has co-operated actively with the Knowledge and Work project led by Markku Nurminen at the University of Turku. This project started in 1986 and is still continuing. The purpose is to study the use of information systems, especially how communication, co-ordination and co-operation are mediated or – sometimes – disrupted by the use of information systems. Inger Eriksson has been actively engaged in this project, and has got her Ph.D. thesis within it. The project is funded by the Academy of Finland and the Foundation for Economic Education.

A new project within the information system area is the SOLE-project led by Aimo Törn and Inger Eriksson. This project started in 1989. The purpose is to study empirically the use of software metric tools for measuring and surveying the quality of software used in commercial firms. The project is funded by the Academy of Finland.

5.1 Software Library Evolution

The project on software quality, SOLE, started 1988. Participants are researchers and students from the Department and from Turku Business School, but also from other universities. The project is lead by professor Aimo Törn and acting professor Inger Eriksson. The coordinator for Turku Business School is Econ. Lic. Liisa von Hellens. Funds have been obtained for the project from the Ministry of Education (UVM), Academy of Finland (FA) and the Foundation for Economic Education.

A paper relating quality and maintenance cost, the SOLE model, has been published in the Journal Systems and Software [Törn 90a], and an article describing SOLE reseach results in Tietojenkäsittelytiede [Eriksson and Törn 90a]. Results have been presented at the Dundee Software Quality Workshop [Eriksson and Törn 90b] and at NordDATA 90 [Eriksson and Törn 90c]. Technical reports on the use of the SOLE model [Törn 90d] and on a quality concept structure [Eriksson and Törn 89] has been published. Inger Eriksson finished her Ph.D. thesis, five Ph.Lic. works are in progress, seven M.Sc. theses have been completed and eight are in progress. Research seminars have been arranged each semester. Two questionnaires have been executed. The results of the first is reported in the M.Sc. thesis of Henrik Johansson and the results of the second will be reported by Kerstin Aller.

The effects on quality of using the technique is investigated by Håkan Sarén. An experiment during a programming course in the spring semester 1989 with second year students has been undertaken and results are now compiled by Åsa Backlund.

5.2 System Maintenance and Organisational Change

The project (SMOC) is planned to start in the autumn 1989 [Sørgaard et al 89]. Funds for further planning of the project have been obtained from the Academy of Finland.

The idea behind the SMOC project is that computer systems may become obstacles to change in the organisations. Such changes can be "flattening" of the organisation (i.e. a move towards cooperative work), decentralisation, or general modernisation and adaptation to changes in the organisation's environment. The latter point includes adaptation to changes in the market.

The working hypothesis is that there are too many assumptions about the organisation built into the system, often implicit assumptions about stability in the organisation. The point in studying maintenance is that this is a situation where the needs for change are confronted with the structures of the existing systems. The project does not focus on the technical quality of the system, i.e. on how the "same" system can be built at various levels of quality. Instead the focus is on what the systems do, i.e. their functional design.

The motivation for the project can be found in the high expenditure on software maintenance and in the inflexibility of many current systems. Users often get stress from computer use, customers are dissatisfied with inflexible service, etc.

The further realization of this project at Åbo Akademi is unprobable because Pål Sørgaard is now working at the Norwegian Computer Center in Oslo, Norway.

Chapter 6

Research on Decision Support

Research on decision systems has focused on global optimization methods and the use of Simulation Nets as a tool for simulating and analyzing system alternatives. The principal researcher is Aimo Törn. A major achievement in this area is a recent comprehensive survey of global optimization methods by Aimo Törn and Antanas Žilinskas that was completed in 1988 and published by Springer Verlag this year. The research on Simulation Nets is described in Chapter 3.

6.1 Global Optimization

The project has continued since 1968, when Aimo Törn was a visiting scholar at Stanford University. The aim of the project is to investigate and compare different approaches to global optimization, i.e., finding the smallest minimum of a mathematical function in a region where the function possibly has several local minima. One of the main approaches to to global optimization today, *clustering methods* has been developed by Aimo Törn.

There has been a lack of a monograph covering the field. As a result of several years of work by Aimo Törn and Antanas Žilinskas (Lithuanian Academy of Sciences) such a book was published 1989 as Lecture Notes in Computer Sciences 350 (Springer-Verlag) [Törn and Žilinskas 89].

Törn's new approach, *topographical global optimization* first realized by Christina Juselius has been further explored by Sami Viitanen, who has made experiments with several parallel versions of the algorithm on the Hathi-2 computer. In topographical global optimization sampled neighbouring points are connected so that topographical information about the function to be optimized is obtained. Results were presented by Törn at a seminar on global optimization at the Royal Institute of Technology, Stockholm in November 1990 (invited talk) and at the IIASA sponsored II Workshop on Global Optimization, Sopron, Hungary in December 1990 (invited talk) [Törn 90e].

The investigations on parallel global optimization by Törn and Žilinskas are continuing. Experiments with parallel Fortran on the Hathi-2 computer of the department were presented at the 14th IFIP Conference on Systems Modelling and Optimization in July 1989 [Törn and Žilinskas 90].

6.2 Simulation Nets

See Section 3.2.

Chapter 7

Research on Artificial Intelligence

The research is focused on expert and decision support systems involving uncertainty modelling, and is based on a theoretical foundation for general structures and categorical aspects of fuzzy logic. These foundations of fuzzy sets stand as a solid platform for the development of applications. The theoretical aspects have been developed at the Department of Mathematics at Åbo Akademi University since 1982 in co-operation with the *Linz-group* and the Mathematics Chapter of the International Fuzzy Systems Association (IFSA).

When oriented towards applications the research was carried over to the computer science department. Applications include medical and technical diagnostics, and a development of hypersystems integrating group decisions and social systems.

The research is strongly dependent on international contacts, including co-work with colleagues from Napels, Salerno and Trento in Italy, and from Berlin, Braunschweig, Leipzig and Wuppertal in Germany, and documented in more than 20 joint publications.

Nationally, we co-operate with the Turku University Central Hospital (TUCS) around medical diagnostics, and with industrial representatives from Imatran Voima, Kone Instruments, Wartsila Diesel International, and Wartsila Diesel Power Plants around technical diagnostics and management systems.

7.1 Fuzzy Logic and Neural Nets

The foundational work behind our applications include a relation between fuzzy logic programming and neural nets. Our ambition is the unlocking of neural nets, thus interpreting neural nets as corresponding fuzzy logic programs and also specifying uncertainty factors in rules by adaptations in related neural nets.

Learning techniques are designed for optimal diagnostics, but also, and in particular, for presentations of meaningful causal relations to be used by practitioners.

7.2 Medical and Technical Diagnostics

As a medical application of our diagnostic models we have designed a decision support system for diagnosing Nephropathia Epidemica, a type of haemorrhagic fever with renal syndrome. Diagnostic success rates are around 90 %, exceeding the performance e.g. of the Bayes scoring

method. The NE has turned out to be an ideal case study for evaluating strategies for improving diagnostic rates.

Another medical application is the development of an integrated system for diagnosing, specifying and treating polyneuropathies. The overall goal of this system is to reduce time, effort and cost for reaching a diagnosis, favourable for both doctors and patients. The system is designed to be used also to serve educational purposes.

In our technical applications we embed techniques for optimizing diagnostics into fault-avoidance knowledge systems.

Details of our research can be found in periodic reports and publications from the GeDeMeDeS project.

7.3 Group Decisions and Social Systems

In 1988 we designed a model for group decisions including a consensus reaching mechanism. Consensus is reached using a soft enforcement which contracts decision makers to a joint decision. The basic idea is to reach a system for electronic administration of meetings.

The model has been expanded to include several social aspects so as to present a precise and accurate picture of different classes of real life meetings. Corresponding software is being developed.

7.4 Knowledge Bases and Medical Informatics

In co-operation with Carnegie-Mellon University and Harvard Medical School Jan Komorowski has been developing a set of tools for authoring semantic networks. Among the first problems is the need for an informative display. The tools extract important details about a focus of interest and display the adjacent or the pertinent parts of the semantic network. We have found that the so-called "fish-eye" perspective helps identify the current position with respect to the global structure, facilitates browsing through adjacent related concepts, and is an appropriate representation of the network for editing. A fish-eye view is a specific distortion in which the neighbourhood of a region of interest is represented in great detail while further away only major landmarks are shown.

We have developed tools for browsing, navigating and authoring taxonomies of medical objects and relationsships in the MEDSORT/Unified Medical Language System. Our display methodology can be traced to Bush's memex: trails of nodes are grouped as entities. We also borrow some of the authoring strategies from advanced editors for structured objects, programming environments and natural language processing.

A browsing system has been implemented by Robin Rosenberg (M.Sc. work) and runs in Common Lisp on the Macintosh computer.

Chapter 8

Conference Activities

The FINSOFT III co-ordination project organizes a series of national seminars on the topi parallel computation and neural networks. The main purpose is to present and discuss resu achieved in the research projects belonging to FINSOFT III. There is usually one or m active researchers from abroad who also lecture at these seminars. There have been th seminars up till now, with the number of participants around 30 – 40.

The department was engaged in organizing a Nordic Workshop on Program Correctnes Uppsala in October 1989 and the First Nordic Transputer Seminar in Turku in October 19

A more detailed account of other conference activities is given below.

Mats Aspnäs:

1990 Member of the organizing committee for First Nordic Transputer Seminar. Turku, tober 1990.

Ralph-Johan Back:

1989 Workshop on Stepwise Refinement of Distributed Systems: Methods, Formalisms a Correctness, Nijmegen, the Netherlands, June 1989. Session chairman.

1990 Member of the programme committee for 2nd Nordic Workshop on Program Correctn Aalborg, Denmark, October 1990.

Member of the organizing committee for First Nordic Transputer Seminar. Turku, tober 1990.

Patrik Eklund:

1990 Member of Programme Committee of KIB90, Leipzig, Germany. June 1990.

Member of Organizing Committee of MEPP90, Ishcia, Italy. October 1990.

Henryk Jan Komorowski:

1989 2nd Scandinavian Conference on Artificial Intelligence, Tampere, Session Chairman.

1990 International Symposium on Methodologies for Intelligent Systems. Knoxville, U Session Chairman.

Pål Sørgaard:

1990 Member of the organizing committee of the 13th Information Systems Research Seminar in Scandinavia (IRIS). Turku, August 1990.

Aimo Törn:

1990 II Second Workshop on Global Optimization. IIASA, Sopron, Hungary, December 1990. Session Chairman.

Chapter 9

Professional organizational activities

9.1 University administration

The university administration is a hierarchical structure, where the essential elected bodies are the governing board of the university, the council of the faculty, and the council of the department. In addition there are various affiliated departments (such as the university library and the computer centre) whose governing boards consist of representatives from other departments.

The partipication by the department members in the various administrative bodies is listed below.

Governing Board of the University

- Ragnar Wikman (deputy)

Council of the Faculty

- Aimo Törn (Ralph-Johan Back, deputy)
- Inger Eriksson
- Joakim Waxlax, student representative

Council of the Department

- Patrik Eklund (chairman), head of the department 1.1. - 31.7.1990, Aimo Törn (chairman), head of the department 1.8.1990 -
- Christel Engblom (Mats Aspnäs, deputy)
- Annamari Soini (Inger Eriksson, deputy)
- Patrik Palm, student representative (Mikael Karv, deputy)
- Fredrick von Schoultz, student representative (Mårten Saarinen, deputy)

Board of the Computer Centre

- Aimo Törn (chairman)
- Tony Riissanen
- Ulla Solin

9.2 National and international organizations

The members of the department have been professionally active in a number of different organizations.

Ralph-Johan Back

- Member of the steering committee of the doctoral education programme in Computer Science (Ministry of Education)1.1.1986 –. Member of the steering committee of the doctoral education programme in Computer Technology (Ministry of Education) 1.1.1986 – .
- Director and scientific leader of subprogramme III (Parallel computation and neural networks) of the FINSOFT research programme of TEKES, 1.4.1988 – . Member of the Co-ordination Board 1.4.1988 –. Member of the steering committee of of subprogramme III 1.4.1988 – .
- Member of the Scientific Board of Center for Scientific Computing 1989 -.
- Member of the Scientific Advisory Council of the Finnish Information Processing Association.
- Member of the Academy for Technical Sciences in Finland.

Inger Eriksson

- Member of the Steering Committee for the doctoral education programme in Information Systems (Ministry of Education) 1990- .

Kaisa Sere

- Member of the Board of Finnish Society for Computer Science 1.1.1989 – 31.12.1990.

Aimo Törn

- Member of the IFIP WG 7.6 on Optimization-Based Computer-Aided Modelling and Design 1988 –.
- Member of the Steering Committee for the doctoral education programme in Information Systems (Ministry of Education) 1986- .
- Vice Chairman of the Scientific Board for Military Defence (Ministry of Defence) 1990 – .
- Member of the Board for Data Processing and Statistics of the City of Turku 1989, 1990.

Chapter 10

Personnel

AGERHOLM Sten, stud.phil., Research Assistant in the Centipede project.
E-mail: sagerhol@ra.abo.fi
Research interests: Theorem provers, program verification, semantics, functional languages parallelism,transputer.

AHLSKOG Peter, stud.phil., Acting Assistant Professor [1] .
E-mail: pahlskog@finabo.abo.fi
Research interests: Software metrics, software quality and cost.

ANDERSSON Thorbjörn, M.Sc., Researcher in the SOLE project.
E-mail: taanders@ra.abo.fi
Research interests: Software quality, systems quality and automation.
Publications 1989, 1990:

1. Andersson, T., *A Survey on Software Quality Metrics.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 120, 1990.

ASPNÄS John, M.Sc., Acting Assistant.
E-mail: aspnasj@finabo.abo.fi
Research interests: Distributed operating systems.

ASPNÄS Mats, M.Sc., Administrative leader of the Millipede project
E-mail: mats@ibsun.abo.fi
Research interests: Parallel programming, programming tools and environments for parallel systems, monitoring of parallel systems.
Publications 1989, 1990:

1. Aspnäs, M., Back, R.J.R., Kurki-Suonio, R., *Efficient Implementation of Multi-process Handshaking on Broadcasting Networks.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 75, 1989.

2. Aspnäs, M., Back, R.J.R., Malén, T-E., The Hathi-2 Multiprocessor System. *Microprocessors and Microsystems*, Vol 14, No 7, pp. 457–466. Butterworth – Heinemann Ltd. 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 80, 1989.

[1] "Överassistent" in Swedish

3. Aspnäs, M., Back, R.J.R., A Programming Environment for a Transputer-Based Multiprocessor System. In *Acta Cybernetica*, Tom. 9, Fasc. 3, pp. 291–301. Szeged 1990. Also in Gyimóthy, T. (ed.), Symposium on Programming Languages and Software Tools, pp. 94–103. Hungarian Academy of Sciences 1990. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 82, 1989.

4. Aspnäs, M., Malén, T-E., *Hathi-2 Users Guide, version 1.0.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. B, No 6, 1989.

5. Aspnäs, M., Back, R.J.R., Multiprocessor Applications in the Hathi-project. In Kankaala K., Nieminen, K. (eds), *Scientific Computing in Finland.* CSC Research Reports R1/89, pp. 5–21, 1989. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 96, 1989.

6. Aspnäs, M., Back, R.J.R., Sere, K., *The Hathi Project—A Research Project on Parallel Programming Technology 1986–88 (final Report).* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 95, 1989.

7. Aspnäs, M., Malen, T-E.,*Transputer Benchmark Tests.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 97, 1989.

8. Aspnäs, M., Kerola, T., *Modeling the Hathi-2 Monitoring System.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 110, 1990.

9. Boglaev, Y.P., Karpenko, A., Aspnäs, M., *An Implementation of the ODE Block Algorithm for Transputer System.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 113, 1990.

10. Aspnäs, M., Långbacka, T., *A Monitoring System for a Transputer-Based Multiprocessor.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 118, 1990. To be presented at Transputing'91, California, USA, April 1991.

BACK Ralph-Johan, Ph.D., Professor.
E-mail: backrj@finabo.abo.fi
Professional activities:

- Member of the steering committee of the doctoral education programme in Computer Science (Ministry of Education)1.1.1986 –. Member of the steering committee of the doctoral education programme in Computer Technology (Ministry of Education) 1.1.1986 – .

- Director and scientific leader of subprogramme III (Parallel computation and neural networks) of the FINSOFT research programme of TEKES, 1.4.1988 – . Member of the Co-ordination board 1.4.1988 –. Member of the steering committee of subprogramme III 1.4.1988 – .

- Member of the Scientific Board of Center for Scientific Computing 1989 -.

- Member of the Council of the Faculty of Mathematics and Natural Sciences at Åbo Akademi 1.1.1986 – .

- Editor of BIT.

- Member of the Scientific Advisory Council of the Finnish Information Processing Association.

- Member of the Academy for Technical Sciences in Finland.
- Reviewer for Acta Informatica, Science of Computer Programming, ACM Transactions of Programming Languages and Systems, International Journal of Parallel Computation, Formal Techniques in Programming, BIT.
- Member of thesis committee for Rassul Ayani, Royal Institute of Technology, Sweden 1989 and the Ph.D. candidacy examination committee for Limor Fix, Technion, Israel 1989.
- Opponent in Frank Stomps Ph.D. defence, Eindhoven University of Technology, The Netherlands 1989.
- Supervisor for the Ph.D studies of Kaisa Sere and Joakim von Wright.
- External supervisor for Ph.D Airi Salminen, University of Jyväskylä.

Research interests: Formal methods for program construction, parallel and distributed computing, programming methods, programming language semantics, program verification.

Publications 1989, 1990:

1. Back, R.J.R., Kurki-Suonio, R., Decentralization of Process Nets with Centralized Control. *Distributed Computing* Vol 3, No 2, May 1989, pp. 73-87. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 58.
2. Aspnäs, M., Back, R.J.R., Multiprocessor Applications in the Hathi-project. In Kankaala, K., Nieminen, K. (eds), *Scientific Computing in Finland.* CSC Research Reports R1/89, pp. 5–21, 1989. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 96, 1989.
3. Back, R.J.R., Changing Data Representation in the Refinement Calculus. *Proc. of Hawaii International Conference on System Sciences (HICSS-22)*, January, 1989, Kailua-Kona, Hawaii. (Best paper award on the Software Track).
4. Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems. In van de Snepscheut, J.L.A. (ed.), *Proc. of the Conference on Mathematics of Program Construction.* Lecture Notes in Computer Science No 375, pp. 115-138. Springer Verlag 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 78. .
5. Back, R.J.R., von Wright, J., A Lattice-theoretic basis for a specification language. In van de Snepscheut, J.L.A. (ed.), *Proc. of the Conference on Mathematics of Program Construction.* Lecture Notes in Computer Science No 375, pp. 139-156. Springer Verlag 1989.
6. Aspnäs, M., Back, R.J.R., A Programming Environment for a Transputer-Based Multiprocessor System. In *Acta Cybernetica*, Tom. 9, Fasc. 3, pp. 291–301. Szeged 1990. Also in Gyimóthy, T. (ed.), Symposium on Programming Languages and Software Tools, pp. 94–103. Hungarian Academy of Sciences 1990. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 82, 1989.
7. Aspnäs, M., Back, R.J.R., Kurki-Suonio, R., *Efficient Implementation of Multi-process Handshaking on Broadcasting Networks.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 75, 1989.

8. Back, R.J.R., Törn, A., *Åbo Akademi. Department of Computer Science. 5-year Report 1984 - 1989.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. B. No 7, 1989.

9. Back, R.J.R., von Wright, J., *Combining angels, demons and miracles in program specifications.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 86, 1989. To appear in Theoretical Computer Science.

10. Ståhl, L., Back, R.J.R., An *Implementation of Multiprocess Handshaking on Transputer networks.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 76, 1989.

11. Back,R.J.R., *On formal and informal methods in program construction*, in J.W. de Bakker, Liber Amoricum 1989.

12. Aspnäs, M., Back, R.J.R., Malén, T-E., The Hathi-2 Multiprocessor System. *Microprocessors and Microsystems*, Vol 14, No 7, pp. 457 - 466. Butterworth - Heinemann Ltd. 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 80, 1989.

13. Back, R.J.R., Sere, K., Stepwise Refinement of Parallel Algorithms. *Science of Computer Programming*, Vol 13, No 2-3, pp. 133-188. North-Holland 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 64, 1988.

14. Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems. To appear in *Structured programming*, January 1991. Also in van de Snepscheut, J.L.A. (ed.), Proc. Conference on Mathematics of Program Construction. Lecture Notes in Computer Science No 375, pp. 115-138. Springer Verlag 1989. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 78, 1989.

15. Back, R.J.R., von Wright, J., Duality in Specification Languages: A Latice Theoretic Approach. *Acta Informatica*, Vol 24, Fasc. 7, 1990, pp. 583-625. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 77, 1989.

16. Back, R.J.R., von Wright, J., Refinement Concepts Formalized in Higher Order Logic. *Formal Aspects of Computing*, No 2, pp. 247-272. BCS 1990. Also in Preprints of IFIP TC2 Working Conference on Programming Concepts and Methods, pp. 171-192. Israel 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 85, 1989.

17. Back, R.J.R., A Method for Refining Atomicity in Parallel Algorithms. In Odijk, E., Rem, M., Syre, J. (eds), *Proc. Conference on Parallel Architechtures and Languages Europe.* Lecture Notes in Computer Science No 366, Vol II, pp. 199-216. Springer Verlag 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 57, 1988.

18. Back, R.J.R., Refinement Calculus, Part II: Parallel and Reactive Programs. In *Proc. REX Workshop for Refinement of Distibuted Systems.* Lecture Notes in Computer Science, No 430, pp. 67-93. Springer Verlag 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 93, 1989.

19. Back, R.J.R., Sere, K., Deriving an Implementation of Action Systems. In *Proc. Third Refinement Workshop.* BCS FACS/IBM UK/Oxford University PRG 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 99, 1990.

20. Back, R.J.R., von Wright, J., Refinement Calculus, Part I: Sequential Nondeterministic Programs. In *REX Workshop for Refinement of Distibuted Systems.* Lecture Notes in Computer Science, No 430, pp. 42–66, 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 92, 1989.

21. Back, R.J.R., Kurki-Suonio, R., *Superposition and Fairness in Reactive System Refinement.* To appear in Proc. of 5th Jerusalem Conference on Information Technology (JCIT), 1990.

22. Back, R.J.R., von Wright, J., *Statement Inversion and Strongest Precondition.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 101, 1990.

23. Back, R.J.R., von Wright, J., *Command Lattices, Variable Environments and Data Refinement.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 102, 1990.

DAHL Peter, stud.phil., Research Assistant in the Centipede project.
E-mail: pdahl@finabo.abo.fi
Research interests: High-level methods for distributed programming.

EKLUND Patrik:, Ph.D., Lecturer
E-mail: peklund@finabo.abo.fi
Professional activities:

- Head of the department (1.1.–30.6.1990).
- Deputy member of the Society for work security of Åbo Akademi.

Research interests: category theory and computer programming, distributed systems, general structure theory, image analysis and pattern recognition, topology.
Publications 1989, 1990:

1. Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., Implementations of the C-calculus, *Connection Science, 1*, pp. 43–53, 1989.

2. Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., A Parallel Implementation of the C-calculus, *1st Italian Workshop on Parallel Architectures and Neural Nets* (ed. E. R. Caianiello), World Scientific, pp.46–62, 1989.

3. Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., Error Propagation in C-calculus, *2nd Italian Workshop on Parallel Architectures and Neural Nets*, Vietri sul Mare, Salerno, April, 1989.

4. Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., Parallel C-calculus: A Case Study, *Proc. 6th Scandinavian Conference on Image Analysis (6SCIA)*, Oulu, Finland, June, 1989.

5. Eklund, P., Optimal Mappings for Variable Architectures, *Proc. 14th IFIP Conference on System Modelling and Optimization*, pp. 78-81. Wissenschaftliche Berichte der Technischen Hochschule Leipzig, 5, 1989.

6. Eklund, P., Gähler, W., General Structures and Fuzzy Filters, *Proc. 3rd Congress of the International Fuzzy Systems Association (IFSA)*, (invited talk), Seattle, August, 1989.

7. Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., C-calculus and Uncertainty. *Proc. 3rd Congress of the International Fuzzy Systems Association (IFSA)*, Seattle, August, 1989.

8. Eklund, P., Gähler, W., Set Functors and General Spaces. To appear in *Communications of the IFSA Mathematics Chapter.* Also in Proc. 11th International Seminar on Fuzzy Set Theory, Applications of Category Theory to Fuzzy Subsets. Linz, 1989.

9. Eklund, P., Gähler, W., Generalized Cauchy Spaces, *Math. Nachr. 147*, pp. 201–215. 1990.

10. Eklund, P., Gähler, W., Fuzzy Filter Functors and Convergence. To appear in Höhle, U., Klement, E.P., Rodabaugh, S.E. (eds.), *Applications of category theory to fuzzy subsets.* Theory and Decision Library B. Kluwer, 1990.

11. Eklund, P., Lindholm, P., Merkuryeva, G., Group Decisions and Sociological Intelligence. To appear in *Proc. KIB90.* Betriebliche Informations- und Kommunikationssysteme, Erich Schmidt Verlag, Berlin/Bielefeld/München, 1990.

12. Eklund, P., J. Forsström, J., Palm, P., Virtanen, H., Waxlax, J., GEDEMEDES – A Generic System for Developing Medical Decision Support. To appear in *ECAI90, Workshop on Industrial Diagnostic Knowledge Based Systems.* Stockholm, 1990.

13. Eklund, P., Kaufmann, M., Hierarchical Wiring in Multigrids. *CONPAR90 - VAPP IV, Joint International COnference on Vector and Parallel Processing*, pp. 423–434. Zürich, 1990.

14. Eklund, P., J. Forsström, J., Diagnosis of Nephropathia Epidemica by Adaptation through Lukasiewicz Inference. To appear in Cercone, N., Gardin, F., Valle, G. (eds), *COMPUTATIONAL INTELLIGENCE, III – The International Conference on Computational Intelligence 90.* Elsevier Science Publishers B.V., North-Holland, 1991.

15. Ehrenberg, D., Eklund, P., Fedrizzi, M., Ventre, A.G.S., Dynamic Consensus in GDSS. In *AIRO '90, Models and Methods for Decision Support*, pp. 245–250. Operational Research Society of Italy, 1990.

16. Carlsson, C., Ehrenberg, D., Eklund, P., Fedrizzi, M., Lindholm, P., Merkuryeva, G., Ventre, A.G.S., A Case Study for Reaching Consensus in Group Decisions. To appear in *IFORS - SPC1.* Brussels 1991.

ENGBLOM, Christel, Department Secretary
E-mail: cengblom@finabo.abo.fi

ERIKSSON Inger, Ph.D., Assistant.
E-mail: ieriksson@finabo.abo.fi
Professional activities:

- Member of the Steering Committee for the doctoral education programme in Information Systems (ministry of Education) 1990–.
- Referee for Scandinavian Journal of Information Systems.
- Member of the organizing committee, chair of the programme committee and editor of the proceedings for the IFIP WG 9.1 conference Women, Work and Computerization to be held in Finland 1991.

Research Interests: End-user training, simulation, information system quality
Publications 1989, 1990:

1. Eriksson,I., Törn,A., *A Covering Structure of IS Quality Concepts.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 89, 1989.

2. Eriksson, I., Educating End-Users to Make More Effective Use of Information Systems. To appear in Gattiker, U.E., Larwood, L.(eds), *Technological Innovation and Human Resources: Enduser Training, Vol II.* Walter de Gruyter.

3. Eriksson, I., Learning Process in the Context of Using ISs. IFIP WG 8.2. In Klein, H.K., Kumar, K. (eds), *Systems Development for Human Progress*, pp. 267–275.. North-Holland 1989.

4. Eriksson, I., Simulation as a Learning Technique in Information Systems Development. In Bjerknes, G. et al. (eds), *Organizational Competence in System Development – A Scandinavian Contribution*, pp. 233–250. Studentlitteratur, Lund, Sweden, 1990. Also published in Organizational Competence in System Development. Proceedings of Nordic Seminar. Tranum Klitgaard, Denmark, Institute of Electronic Systems. Aalborg University Centre, 1988, pp. 57-68.

5. Eriksson, I., *Simulation for user training.* Ph.D. thesis, Åbo Akademi, Dept. of Computer Science, 1990. A summary published in Acta Academiae Aboensis, Ser. B, Mathematica et Physica, Vol. 50, No 3. Åbo Akademi, 1990.

6. Eriksson, I., Reijonen, P., Training Computer-Supported Work by Simulation. In *Education and Computing.* The International Journal, Vol. 6, Nos 1–2, pp. 129–136. 1990. Also published in Barta, B.Z., Fontell, L., Raymont, P., Lovis, F. (eds), Methodologies of Training Data Processing Professionals and Advances End-Users, pp. 129–136. Elsevier 1990.

7. Eriksson, I., Finnäs, A., Creating a Visual Simulation Model of an Inventory System. To appear in van den Besselaar, P., Clement, A., Järvinen, P., Fuchs-Kittowski, K., Lütterbeck, B. (eds), *Information System, Work and Organizational Design.*

8. Eriksson, I., Finnäs, A., A Visual Simulation Model of Information and Work Systems. In Bödker, S. (ed.), *Proc. of the 12th IRIS Conference.* DAIMI PB-296-I, Aarhus University, Part I, pp. 149–168. 1989. Revised version of No 7.

9. Eriksson, I., Törn, A.A., SOLE – Research on Information System Quality. In: *Conference Papers of the Software Quality Workshop*, Scotland, pp. 249-258. Dundee Institute of Technology, 1990.

10. Eriksson, I., Törn, A.A., The SOLE project: Efficient IS Management. In *Proc. NordDATA 90*, Part 1, pp. 211-217. Gothenburg 1990.

11. Eriksson, I., Törn, A.A., SOLE-IS Quality and Efficient IS Management. In *Tietojenkäsittelytiede*, Vol 1, No 1, pp. 29–37, 1990. A revised version of No 9.

12. Eriksson, I., *Reflections on the concept of knowledge.* To appear in the proc. of the seminar on the Fragmentatin of Knowledge, June 1989, Capri Italy.

13. Eriksson, I., Finnäs, A., Reijonen, P., Visual Simulation as an Aid for Understanding Computer Functions. To appear in *Interacting with Computer*, April 1991.

14. Eriksson, I., Törn, A.A., *A Model for IS Quality.* To appear in Software Engineering Journal.

FLÜCK Jan-Erik, M.Sc., Acting Laboratory Manager.
E-mail: jfluck@finabo.abo.fi
Research interests: Multiprocessor systems.

GUSTAFSSON Patrik, stud.phil., Acting Assistant.
E-mail: pgustafsson@finabo.abo.fi
Research interests: Group decision support systems.

HEKANAHO Jukkapekka, M.Sc., Research Assistant in the Centipede project.
E-mail: hekanaho@ibsun.abo.fi
Research interests: Program transformations, Program transformation systems.

KOMOROWSKI Henryk Jan, Ph.D., Acting Professor
E-mail: komorowski@finabo.abo.fi, jan@ra.abo.fi
Professional activities:

- Founding Member of the Editorial Board of the Journal of Logic Programming.
- The MIT Press: adviser to the chief editor on the development of new journal and book series; reviewer of book manuscripts submitted for publishing.
- Reviewer for National Science Foundation, Swedish Board for Technical Development (STU), Theoretical Computer Science, ACM Transactions on Programming Languages and Systems, Communications of the ACM, Computational Intelligence, professional conferences and publishing houses.
- Guest Editor, Special Issue of the Journal of Logic Programming on Partial Deductions, 1990.
- Member of a Ph.D. examination committee, Uppsala University (twice).
- Expert in associate professor appointments, Linköping University (three times).

Research Interests: Artificial intelligence and software systems, program transformation, knowledge management environments and medical informatics.
Publications 1989, 1990:

1. Komorowski, H.J., Barr, C.E., Greenes, R.A., Knowledge Modelling in the Unified Medical Language System, *Proc. of the Joint Scandinavian-Japanese Seminar on Information Modelling and Knowledge Bases*, Ellivuori, June 1989, Finland.
2. Komorowski, J., Towards Synthesis of Programs in the Partial Deduction Framework. *Proc. of the XIth International Joint Conference on Artificial Intelligence, Workshop on Automating Software Design*, Detroit, MI, USA, August, 1989.

LINDHOLM Paul, Ph.Lic., Lecturer.
E-mail: plindholm@finabo.abo.fi
Research interests: Pattern recognition, group decision support systems.
Publications 1989, 1990:

1. Eklund, P., Lindholm, P., Merkuryeva, G., Group Decisions and Sociological Intelligence. To appear in *Proc. KIB90.* Betriebliche Informations- und Kommunikationssysteme, Erich Schmidt Verlag, Berlin/Bielefeld/München, 1990.

2. Carlsson, C., Ehrenberg, D., Eklund, P., Fedrizzi, M., Lindholm, P., Merkuryeva, G., Ventre, A.G.S., A Case Study for Reaching Consensus in Group Decisions. To appear in *IFORS - SPC1.* Brussels 1991.

LÅNGBACKA Thomas, stud.phil., Research Assistant in the Millipede project.
E-mail: langback@ibsun.abo.fi
Research interests: Different aspects of Parallel and Distributed Computing.

1. Aspnäs, M., Långbacka, T., *A Monitoring System for a Transputer-Based Multiprocessor.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 118, 1990 To be presented at Transputing'91, Calofornia, USA, April 1991.

RIISSANEN Tony, stud.phil., Acting Laboratory Manager.
E-mail: triissanen@finabo.abo.fi
Professional activities:

- Member of the Board of the Computer Centre.

Research interests: Parallel programming, group decision support systems.

SERE Kaisa, Ph.D., Assistant Professor [1] [2]
e-mail: ksere@finabo.abo.fi
Professional activities:

- Member of Board of Finnish Society for Computer Science (1.1.1989 -)

Research interests: Formal methods in program construction, program verification, parallel and distributed programming.
Publications 1989, 1990:

1. Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems. In van de Snepscheut, J.L.A. (ed.), *Proc. Conference on Mathematics of Program Construction*, Groningen, the Netherlands, June, 1989. Lecture Notes in Computer Science No 375, pp. 115–138. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 78, 1989.

2. Back, R.J.R., Sere, K., Stepwise Refinement of Parallel Algorithms. *Science of Computer Programming*, Vol 13, No 2–3, pp. 133–188. North-Holland 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 64, 1988.

3. Back, R.J.R., Sere, K., Deriving an Implementation of Action Systems. In *Proc. Third Refinement Workshop.* BCS FACS/IBM UK/oxford University PRG 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 99, 1990.

4. Sere, K., *Laws of Action Systems Programming.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 100, 1990.

[1] "Överassistent" in Swedish
[2] Faculty of Chemical Engineering, position placed in our department

5. Sere, K., *Stepwise Derivation of Parallel Algorithms.* Åbo Akademi, Dept. of Computer Science. Ph.D. thesis 1990.

6. Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems. To appear in *Structured Programming 12*, January 1991.

SHEN Hong, Ph.Lic., Researcher in the Millipede project.
E-mail: shenh@finabo.abo.fi
Professional activities:

- Member of parallel computing society.
- Referee for BIT

Research interests: Parallel and distributed computing.
Publications 1989, 1990:

1. Shen, H., Mapping Parallel Programs onto Transputer Networks. In Hulskamp, J. (ed.), *Proc. of Australian Transputer and Occam User Group Conference.* RMIT 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 79, 1989.

2. Shen, H., Self-adjusting mapping: a heuristic mapping algorithm for mapping parallel programs onto transputer networks. In Wexler, J. (ed.), *Developing Transputer Applications* (OUG11), pp. 89–98. IOS Amsterdam 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 87,1989.

3. Shen, H., Fast Path-disjoint Routing in Transputer Networks. In Gyimóty, T. (ed.), *Symposium on Programming Languages and Software Tools*, pp. 157–167. Hungarian Academy of Sciences 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 83, 1989. To appear in Microprocessing and Microprogramming. North-Holland.

4. Shen, H., Occam Implementation of Path-disjoint Routing on the Hathi-2 Transputer System. *Microprocessing and Microprogramming*, No 30, pp. 93–100. North-Holland 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 98, 1989.

5. Shen, H., A Fast Parallel Algorithm for Integer Sorting. In Evans, D.J., Joubert, G.R., Peters, F.J. (eds), *Advances in Parallel Computing 89*, Vol II, pp. 331–336. North-Holland 1990. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 70, 1988.

6. Hagerup, T., Shen H., Improved Non-conservative Sequential and Parallel Integer Sorting. *Information Processing Letters*, Vol 36, No 2, pp. 1–7. North-Holland 1990. Also published in Technical report 10/1990, Univ. of Saarbrücken, Sonderforschungsbereich 124.

7. Shen, H., *Occam Implementation of Process-to-processor on the Hathi-2 Transputer System.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 114, 1990. To be presented at Transputing'91, California, USA, April 1991.

8. Shen H., *Contributions to Mapping, Routing and Sorting in Parallel Processing.* Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1990.

SOINI Annamari, M.Sc., Acting Instructor.
E-mail: soinia@finabo.abo.fi
Research Interests: Computer linguistics.

SOLIN Ulla, Ph.Lic., Assistant.
E-mail: usolin@finabo.abo.fi
Professional activities:

- Member of the Board of Computer Center
- Referee for BIT

Research Interests: Algorithm animation, parallel and distributed computing.
Publications 1989, 1990:

1. Solin, U., *Animering av parallella processer.* (Animation of Parallel Processes). Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1989.

STILL Dan-Johan stud.phil., Acting Assistant.
E-mail: dstill@finabo.abo.fi
Research Interests: Distributed programming, human-computer interfaces.

STOMP Frank, Ph.D., Acting Assistant Professor [3]
E-mail: fstomp@ra.abo.fi
Professional activities:

- Referee for the Journals IEEE Transactions on Software Engineering and Theoretical Computer Science.

Research interests: Formal methods for verification and design of parallel and distributed programs.
Publications 1990:

1. Stomp, F., A derivation of a broadcasting algorithm using sequently phased reasoning. In Logrippo, L., Probert, R.L., Ural, H., *Proc. Protocol, Specification, Testing and Verification X*, pp. 19–32. Canada, 1990.

SUOJANEN Marko, stud.phil., Research Assistant in the Centipede project.
E-mail: msuojanen@finabo.abo.fi
Research interests: Formal methods for program development.

SØRGAARD Pål, Ph.D., Assistant Professor [1]
E-mail: psorgaard@finabo.abo.fi
Professional activities:

- Co-editor of the Preceedings of the 13th IRIS. August 1990.

Research interests: System development, computer-supported cooperative work, computers in organisations, object-oriented programming.
Publications 1989, 1990:

[3] "Överassistent" in Swedish

1. Sørgaard, P., Transaction-supporting Systems and Organizational Change. *Office: Technology and People*, Vol 4, No. 3, pp. 229–243, June 1989.

2. Sørgaard, P., Report on the 1st eurpoean cscw conference. *Cosmos Information Exchange Network*, pp. 22–25. 1989.

3. Sørgaard, P., Nurminen, M., Forsman, U., System Maintenance and Organizational Change. In Bødker, S. (ed.), *Proc. 12th IRIS— Part II*, pp. 567–586. DAIMI PB—296-II, Aarhus University, Aarhus, 1989.

4. Sørgaard, P., Computer-supported cooperative work: a challenge to system developers. In *Proc. of Tietotekniikka 89*, Jyväskylä, 1989.

5. Sørgaard, P., *An Overview of Research in Maintenance.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 94, 1989.

6. Andersen, N., Kensing, F., Lundin, J., Mathiassen, L., Munk-Madsen, A., Rasbech, M., Sørgaard, P., *Professional Systems Development.* Prentice Hall, Business Information Technology Series. U.K. 1990.

7. Sørgaard, P., The case for destruction. In Hellaman, R., Ruohonen, M., S/orgaard, P. (eds), *Preceedings of the 13th IRIS* Part II , pp. 409–421. Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 108, 1990.

TRISHINA Elena, Ph.D., Visiting Researcher in the Millipede project.
E-mail: etrishin@ibsun.abo.fi
Professional activities:

- Referee for BIT, J. Computers and Artificial Intelligence, International Journal of Parallel Programming, Parallel Computing, IEEE Computers.

Research interests: Design and analysis of parallel algorithms, formal software development methods, functional programming.

TÖRN Aimo, Ph.D, Professor
E-mail: torn@finabo.abo.fi
Professional activities:

- Member of the Board for Data Processing and Statistics of the City of Turku.
- Vice Chairman of the Scientific Board for Military Defence (Ministry of Defence), 1990-.
- Member of the Steering Committee for the doctoral education programme in Information Systems (Ministry of Education) 1986- .
- Member of the IFIP TC-7 Working Group WG 7.6, Optimization-Based Computer-Aided Modelling and Design, 1988-.
- Head of the department 1.7. - 31.12.1990.
- Supervisor for the Ph.D. studies of Inger Eriksson 1990.

Research interests: Software engineering, simulation, Petri Nets, optimization, Monte Carlo methods.

Publications in 1989, 1990

1. Törn, A.A., Žilinskas, A., *Global optimization*, Lecture Notes in Computer Science 350. 255 pp. Springer–Verlag, Berlin, 1989.

2. Törn, A.A., An efficient procedure for determining the enabled set. *Petri Net Newsletter 31*, pp. 23–27, 1989.

3. Törn, A.A., PICA – A graphical program development tool. In *Acta Cybernetica*, Tom 9, Fasc. 3, pp. 303–321. Szeged 1990. Also in Gyimóthy, T (ed.), Symposium on Programming Languages and Software Tools, pp. 318–330. Hungarian Academy of Sciences 1989. Also in Proc. of the Joint Finnish-Soviet Software Symposium: Software Development Trends, pp. 65–78. Helsinki, 1989.

4. Eriksson, I., Törn, A.A., *A covering structure of IS quality concepts.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 89, 1989.

5. Back, R.J.R., Törn, A.A. (eds.), *5-year Report 1984–1989, Åbo Akademi, Department of Computer Science.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. B, No 7, 1989.

6. Törn, A.A., Žilinskas, A., Global optimization algorithms in optimal design. In Sebastian, H-J., Tammer, K. (eds), *System Modelling and Optimization.* Lecture Notes in Control and Information Sciences 143, pp. 951–960, 1990.

7. Eriksson, I., Törn, A.A., SOLE – IS quality and efficient IS management. *Tietojenkäsittelytiede*, Vol 1, No 1, pp. 29–37. 1990.

8. Törn, A.A., Models of software accumulation. *Journal of Systems & Software 12*, pp. 39–42, 1990.

9. Törn, A.A., Decision support by rapid simulation using Simulation Nets. *Decision Support Systems 6*, pp. 299–305, 1990.

10. Eriksson, I., Törn, A.A., SOLE – Research on Information System Quality. In: *Conference Papers of the Software Quality Workshop*, pp. 249–258. Dundee Institute of Technology, 1990.

11. Eriksson, I., Törn, A.A., The SOLE project: Efficient IS Management. In *Proc. NordDATA 90*, Part 1, pp. 211–217. Gothenburg 1990.

12. Törn, A.A., *Estimating software quality characteristics from maintenance costs.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 109, 1990.

13. Törn, A.A., *Topographical global Optimization.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 119, 1990.

14. Eriksson, I., Törn, A.A., *A Model for IS Quality.* To appear in Software Engineering Journal.

WAXLAX Joakim, stud.phil., Research Assistant in the GeDeMeDeS project.
e-mail: jwaxlax@finabo.abo.fi
Research interests: Neural networks and their industrial applications.

1. Eklund, P., J. Forsström, J., Palm, P., Virtanen, H., Waxlax, J., GEDEMEDES – A Generic System for Developing Medical Decision Support. To appear in *ECAI90, Workshop on Industrial Diagnostic Knowledge Based Systems.* Stockholm, 1990.

WESTERHOLM, Gundel, M.B.A., Project secretary in the Millipede and Centipede projects.
E-mail: gwesterholm@finabo.abo.fi

VIITANEN Sami, stud.phil., Acting Instructor.
E-mail: viitanens@finabo.abo.fi
Research Interests: Global optimization.

WIKMAN Ragnar, Ph.Lic., Lecturer
E-mail: rwikman@finabo.abo.fi
Research Interests: Data compression, artificial intelligence

VIRTANEN Harry, stud.phil., Research Assistant in the GeDeMeDeS project.
E-mail: hvirtanen@finabo.abo.fi
Research Interests: Logic programming, non-monotonic reasoning, neural nets.

1. Eklund, P., J. Forsström, J., Palm, P., Virtanen, H., Waxlax, J., GEDEMEDES – A Generic System for Developing Medical Decision Support. To appear in *ECAI90, Workshop on Industrial Diagnostic Knowledge Based Systems.* Stockholm, 1990.

Chapter 11

Scientific Contacts

The scientific contacts, both visitors to the department and visits, conference and advanced course participation by the personnel at the department in 1989 and 1990, are listed below.

11.1 Foreign visitors to the department

11.1.1 Long term visitors

1989 Joost Kok, Ph.D., University of Utrecht, The Netherlands, 1.8. - 31.12.1989.

1990 Frank Stomp, Ph.D., University of Nijmegen, The Netherlands, 1.3. - 31.12.1990.

Elena Trishina, Dr., USSR Academy of Sciences, Novosibirsk, 2.5. - 31.12.1990.

11.1.2 Short term visitors

1989 Ben Thompson, Dr., University of Manchester, U.K., 11-18.3.1989

Wlodek Drabent, Dr., University of Linköping, Sweden, 9-11.5.1989

Greg Burns, M.Sc., Ohio State University, U.S.A., 6-18.6.1989.

Jaco De Bakker, Professor, CWI, The Netherlands, 14 - 19.8.1989.

Willem Paul De Roever, Professor, Eindhoven University of Technology, The Netherlands, 21 - 26.8.1989.

Shmuel Zaks, Professor, Technion, Israel, 12 - 13.9.1989.

Jan Rutten, Ph.D., CWI, The Netherlands, 25 - 29.9.1989.

Dieter Ehrenberg, Professor, TH Leipzig, GDR, 25.9 - 15.10.1989.

Tilo Brock, M.Sc., TH Leipzig, DDR, 8.10 - 21.11.1989.

Jan Zytkow, Professor, Kansas, U.S.A., 10.10.1989.

Frits Vaandrager, M.Sc., CWI, The Netherlands, 15 - 19.10.1989.

Erik De Vink, M.Sc., CWI, The Netherlands, 18 - 28.10.1989.

Anders Haraldsson, Professor, Linköping University, Sweden, 9.11.1989.

The working group for the scienctific co-operation between DDR and Finland, 9.11.1990.

Carl Gustaf Jansson, Professor, The Royal Institute of Technology , Stockholm, Sweden, 13 - 16.11.1989.

Peter Knijnenburg, M.Sc., University of Utrecht, The Netherlands, 23.11. - 7.12.1989.

Aldo Ventre, Ph.D., University of Neapel, Italy, 27.11 - 10.12.1989.

Torbjörn Näslund, M.Sc., Linköping University, Sweden, 12.12.1989.

Elena Trishina, Dr., USSR Academy of Sciences, Novosibirsk, 16.12. - 5.1.1990.

1990

Dan Sahlin, M.Sc., Linköping University, Sweden, 29.1.1990.

Oleg Moskalev, Dr., Scientific Computer Centre of the USSR Academy of Sciences, 13.2.1990.

Nikolai Fetisov, Dr., Scientific Computer Centre of the USSR Academy of Sciences, 13.2.1990.

Mikhail Marov, Dr., M.V. Keldysh Institute of Applied Mathematics, USSR, 20.2.1990.

Ulf Nilsson, Ph.Lic., Linköping University, Sweden, 7.5.1990.

Wim Hesselink, Ph.D., University of Groningen, The Netherlands, 4 - 8.5.1990.

Tibor Csendes, Dr. Josef Attila University, Szeged, Hungary, 21 - 25.5.1990.

Werner Gähler, Dr., Berlin, DDR, 28.5 - 10.6.1990.

Yuri Larionov, Dr., Moscow Physical Engineering Institute, Dept. of Computing and Measuring Systems, USSR, June 1990.

Leslie Lamport, Ph.D., Digital Systems Research Center, U.S.A., 31.7. - 3.8.1990.

Tatjana Elizarova, Dr., M.V. Keldysh Institute of Applied Mathematics, USSR, 1 - 15.8.1990.

Yuri Skorov, Dr., M.V. Keldysh Institute of Applied Mathematics, USSR, 1 - 31.8.1990

Joost Kok, Ph.D., University of Utrecht, The Netherlands, 18.8. - 5.9.1990.

Smuel Katz, Professor, Technion, Istrael, 24.8 - 2.9.1990.

Nissim Francez, Professor, Technion, Israel, 31.8. - 9.9.1990.

Anatoli Karpenko, Dr., USSR Academy of Sciences, Moscow, 10 - 22.9.1990.

Youri Boglaev, Dr., USSR Academy of Sciences, Moscow, 10 - 22.9.1990.

Jan van de Snepscheut, Ph.D., California Institute of Technology, U.S.A., 17 - 22.9.1990

Frank Klawonn, M.Sc., Braunnschweig, Germany, 17 - 30.9.1990.

Hans-Jürgen Sebastian, Professor, TH Leipzig, Germany, 20 - 29.9.1990

Swedish Parliament Group. 3.12.1990.

Antonia Sinachopoulos, Dr., Free University of Brussels, Belgium, 12.12.1990.

11.2 Visits and conference participation by staff

Sten Agerholm

1990 Third International HOL Users meeting, Aarhus, Denmark. October 1990.

2nd Nordic Workshop on Program Correctness, Aalborg, Denmark. October 1990.

First Nordic Transputer Seminar, Turku, Finland. October 1990.

Peter Ahlskog

Quality and Maintenance. Lecturers: Dr. Barbara Kitchenham, prof. Aimo Törn och dir. Iikka Ahonen, Turku, Finland. April 1990.

Thorbjörn Andersson

1990 The Kilpisjärvi Seminar of the Doctoral Programme in Information Systems. April 1990.

13th IRIS Seminar, Turku, Finland. August 1990.

John Aspnäs

1989 Visited SICS, Stockholm. December 1989.

Mats Aspnäs

1989 10th Occam User Group Technical Meeting, Twente University, Enschede, the Netherlands. April 1989.

Visit to University of Groningen, Groningen, the Netherlands. April 1989.

First Finnish-Hungarian Workshop on Programming Languages and Software Tools, Szeged, Hungary (paper). August 1989.

Course on Instroduction to Distributed Algorithms, Helsinki, Finland. Lecturer: Prof. Shmuel Zaks, Technion, Israel. September 1989.

Visited SICS, Kista, Sweden. December 1989.

1990 Finsoft III Parallel Processing Seminar, Sjökulla, Finland (talk).

Visited University of Tampere, invited lecture in seminar "Rinnakkaiset järjestelmät". February 1990.

Visited KTH/TDS, Stockholm, invited lecture in seminar. February 1990.

Visited IVTAN (Institute for Very High Temperatures), Moscow, USSR. April 1990.

Invited lecturer in summer school "Programming Transputer Based Parallel Computers", DIKU, Copenhagen, Denmark. June 1990.

Workshop on Programming Parallel Computers. DIKU, Copenhagen, Denmark (talk). June 1990.

Participated in organizing committee meeting for the First Nordic Transputer Seminar, KTH, Stockholm. August 1990.

Seminar on Design Automation of Computer Hardware and Software, Helsinki University of Technology, Laboratory of Signal Processing and Computer Technology (talk). October 1990.

First Nordic Transputer Seminar, Turku, Finland (paper). October 1990.

1989 Hawaiian International Conference on System Sciences (HICS-22), Kailua-Kona, Hawaii, U.S.A. (paper, best paper award in Software Track). January 1989.

Visit to SRI, Palo Alto, California, U.S.A. (guest lecture). January 1989.

Visit to the Kestrel institute, Palo Alto, California, U.S.A. January 1989.

Visit to University of Eindhoven, Eindhoven, the Netherlands (short course). April 1989.

Visit to CWI, Amsterdam, the Netherlands (guest lecture). April 1989.

Visit to Technion, Haifa, Israel (guest lecture, thesis committe). April 1989.

Visit to University of Groningen, Groningen, the Netherlands (guest lecture). May 1989.

Visit to the Swedish Insititute of Computer Science, Kista, Sweden. June 1989.

Workshop on Stepwise Refinement of Distributed Systems, Nijmegen, the Netherlands (invited paper, session chairman). June 1989.

International Conference on Parallel Architectures and Languages Europe (PARLE 89), Eindhoven, the Netherlands (paper). June 1989.

International Confernce on Mathematics of Program Construction, Enskede, the Netherlands (papers). June 1989.

Visited SICS, Kista, Sweden. December 1989.

1990 BCS/FACS Refinement Workshop, Oxfrod, U.K, (invited speaker). January 1990.

Visit at the Lithuanian Academy of Sciences, Vilnius, USSR. January 1990.

IFIP WG 2.3. Meeting, Munich, FRG. March 1990.

Programming Concepts and Methods IFIP TC 2 Working Conference, Sea of Gallilei, Israel (paper). April 1990.

ESPRIT BRA Workshop on Realtime System Specification, Crete, Greece. April 1990.

Visit at SICS, Kista, Sweden. June 1990.

Invited lecturer in summer school "Programming Transputer Based Parallel Computers", DIKU, Copenhagen, Denmark. June 1990.

ProCos Seminar, Technical University of Denmark, Lungby, Denmark. August 1990.

SICS Workshop, Kista, Sweden. August 1990.

SPEC – Workshop on Formal Methods and Tools for the Development of Distributed and Real-Time Systems. Aalborg, Denmark. October 1990.

2nd Nordic Workshop on Program Correctness, Aalborg, Denmark. October 1990.

First Nordic Transputer Seminar, Turku, Finland. October 1990.

Visit at SICS, Kista, Sweden. November 1990.

Visit at Californian Institute of Technology, Los Angeles, U.S.A. December 1990.

IFIP WG 2.3. 25th Meeting, Santa Catalina Island, U.S.A. December 1990.

Visit at Digital Equipment Research Center, San Fransisco, U.S.A. December 1990.

Patrik Eklund

1989 Visit to Universität des Saarlandes, Institut für Informatik, Saarbrücken, FRG (talk).

Visit to Chalmers Tekniska Högskola, Institutionen för informationsbehandling, Gothenburg, Sweden (talk).

SCIA89, Oulu, Finland (paper).

Visit to Akademie der Wissenschaften, Karl-Weierstrass -Institut für Mathematik, Berlin, GDR (talk).

14th IFIP Conference on System Modelling and Optimization, Leipzig, GDR (paper).

1990 KIB90 (paper, member of the programme committee), Leipzig, GDR. June 1990.

Visit to Karl-Weierstrass-Institut Für Matematik, Berlin, GDR. June 1990.

ECAI90 (paper), Stockholm, Sweden. July 1990.

CONPAR90 - VAPP IV (paper), Zürich, Switzerland. September 1990.

Computational Intelligence 90 (paper), Milano, Italy. September 1990.

Visit to Universita di Trento, Trento, Italy. September 1990.

MEPP90 (member of the organizing committee), Ischia, Italy. October 1990.

Inger Eriksson

1989 A lecture series: A Human Activity Approach to User Interfaces, with subtopics: Ecological approach on studying artifacts, Prototyping, and Simulation methods, Ph.D. Susanne Bødker, University of Turku, Finland.

Information Systems Doctoral Research Course Syllabus, Prof. Heinz K. Klein, Kangasala, Finland.

IS Research Methodology, Prof. Rudy Hirschheim, Prof. Richard O. Mason, Dr. Lucy A. Suchman, Prof. Heinz K. Klein, Leivonmäki, Finland.

IFIP WG 3.4 Working Conference, Helsinki, Finland (paper).

Seminar on Fragmentation of Knowledge, Capri, Italy (paper).

Summer School on User Interfaces, Dr. Richard A. Bolt, Dr. Marc H. Brown, Prof. James D. Foley, Dr. Brian W. Kernighan, Dr. Ben Shneiderman, Tampere.

IFIP-HUB Conference, TC9, WG 9.1, "Information System, Work and Organization Design", Berlin, GDR (paper).

12th IRIS Seminar, Skagen, Denmark (paper).

Seminar on the Fragmentation of Knowledge, Capri, Italy. June 1989.

Summer School on User Interfaces. University of Tampere and Finnish Society for Computer Science, Finland. June 1989.

IFIP 9.1. Information Systems, Work and Organizational Design, Berlin, GDR. July 1989.

IFIP 3.2. Methodologies of Training Data Processing Professionals and Advanced End-Users, Helsinki. August 1989.

Visit to University of Aarhus, Denmark (invited lecturer). August 1989.

12th IRIS Conference (paper). Skagen, Danmark. August 1989.

1990 Seminar on Quality and Maintenance. Lecturer: Dr. Barbara Kitchenham, prof. Aimo Törn och dir. Iikka Ahonen, Turku, Finland. April 1990.

The Conference NordDATA 90, Gothenburg, Sweden. June 1990 (paper).

Research Seminar on Information Systems Strategy. Lecturer: Prof. Robert Galliers, University of Warwik, Helsingfors, Finland. September 1990.

Jukkapekka Hekanaho

1989 SCAI-89, Second Scandinavian Conferens on Artificial Intelligence, Tampere, Finland.

1st Nordic Workshop on Program Correctness, Uppsala, Sweden. October 1990.

1990 Visit to Swedish Institute of Computer Science, Kista, Sweden. May 1990.

SPEC – Workshop on Formal Methods and Tools for the Development of Distributed and Real-Time Systems. Aalborg, Denmark. October 1990.

2nd Nordic Workshop on Program Correctness, Aalborg, Denmark. October 1990.

ECHT'90, 1st European Conference on Hypertext, Paris, France. November 1990.

Jan Komorowski

1989 Second Scandinavian Conference on Artificial Intelligence, Tampere, Finland (invited talk).

International Workshop on Alternatives to Logic Programming, Kuopio, Finland (invited talk).

Joint Scandinavian-Japanese Seminar on Information Modelling and Knowledge Bases, Tampere, Finland (paper).

Visit to University of Turku, Turku, Finland (guest lecture).

Visit to University of Umeå, Umeå, Sweden (guest lecture).

Visit to University of Luleå, Luleå, Sweden (guest lecture).

Visit to University of Helsinki, Helsinki, Finland (guest lecture).

International Joint Conference on Artificial Intelligence. Workshop on Automating Software Design, Detroit, U.S.A. (invited presentation). August 1989.

Scandinavian Workshop on Program Verification, Uppsala, Sweden (paper).

1990 European Conference on Artificial Intelligence, Stockholm, Sweden (paper). August 1990.

Programming Languages and Logic Programming Conference, Linköping, Sweden. August 1990.

Visit at Bergen University, Norway.

Visit at the Norwegian Institute of Technology in Trondheim, Norway.

Visit at Imperial College, London, UK.

Visit at Centre for Research in Computer Science, Amsterdam, the Netherlands.

Visit at European Computer-Industry Research Center, Munich, Germany.

Visit at University of Namur, Belgium.

Paul Lindholm

MEPP90, Ischia, Italy. October 1990.

Thomas Långbacka

1989 Summer School on Alternatives of Logic Programming, Kuopio

1990 Seminar on Design Automation of Computer Hardware and Software, Helsinki, Finland. September 1990.

SPEC – Workshop on Formal Methods and Tools for the Development of Distributed and Real-Time Systems. Aalborg, Denmark. October 1990.

2nd Nordic Workshop on Program Correctness, Aalborg, Denmark. October 1990.

First Nordic Transputer Seminar, Turku, Finland. October 1990.

Kaisa Sere

1989 Course on Synchronous Concurrent Algorithms, Turku, Finland, lecturer: Ben Thompson, Ph.D., Univ. of Manchester, United Kingdom.

International Conference on Mathematics of Program Construction, Enschede, the Nether lands (paper).

Course on Concurrency Semantics, Turku, Finland. Lecturer: Prof. J. de Bakker, CWI, the Netherlands. August 1989.

Course on Compositionality and Modularity for Distributed and Real-time Systems, Turku, Finaldn. Lecturer: Prof. Willem Paul de Roever, Technical University of Eindhoven, the Netherlands, August 1989.

Course on Instroduction to Distributed Algorithms, Helsinki, Finland. Lecturer: Prof. Scmuel Zaks, Technion, Israel. September 1989.

Nordic Workshop on Program Correctness, Uppsala, Sweden. October 1989.

1990 CAAP90/ESOP90, Copenhagen, Denmark. May 1990.

FCS - FACS Workshop on Semantics for Concurrency, Leicester, UK, July 1990.

Workshop on Neural Networks, Methodologies and Tools for Design of Distributed Systemd and High-Speed Networks, SICS, Nässlingen, Sweden. Augusti 1990.

SPEC – Workshop on Formal Methods and Tools for the Development of Distributed and Real-Time Systems. Aalborg, Denmark. October 1990.

2nd Nordic Workshop on Program Correctness, Aalborg, Denmark (paper). October 1990.

Hong Shen

1989 Visit to Linköping University, Sweden.

Second Nordic Summer School on Neural Modelling and Computation, Uppsala, Sweden. August 1989.

First Finnish–Hungarian Workshop on Programming Languages and Software Tools, Szeged, Hungary (paper). August 1989.

International Conference on Parallel Computing'89, Leiden, the Netherlands (paper). August 1989

11th Occam User Group Technical Meeting, Edinburgh, UK (paper). September 1989.

1990 EUROMICRO'90, Amsterdam, The Netherlands (paper). August 1990.

Seminar on Design Automation or Computer Hardware and Software, Espoo, Finland (paper). September 1990.

1st Nordic Transputer Seminar, Turku, Finland. October 1990.

Dan-Johan Still

1989 6th Scandinavian Conference on Image Analysis, Oulu, Finland (paper)

Summer School on User Interfaces, Tampere, Finland

Frank Stomp

1990 10th IFIP WG6.1. International Symposium on Protocol Specification, Testing and Verification. Ottawa, Canada (paper). June 1990.

Visit to Swedish Institute of Computer Science, Kista, Sweden (talk). May 1990.

Pål Sørgaard

1989 Two talks at University of Umeå , Institute of Information Processing . February 1989.

Support, Society and Culture: Mutual uses of Cybernetics and Science, Amsterdam (invited talk). March 1989.

Talk at the University of Amsterdam, OOC programme. May 1989.

Tietotekniikka 89, Jyväskylä (paper). May 1989.

First European Conference on Computer Supported Cooperative Work, Gatwick, London, (talk). September 1989.

1990 13th IRIS Seminar, Turku (paper). August 1990.

Elena Trishina

1990 Visit to Swedish Institute of Computer Science, Kista, Sweden (talk). May 1990.

SPEC – Workshop on Formal Methods and Tools for the Development of Distributed and Real-Time Systems. Aalborg, Denmark. October 1990.

2nd Nordic Workshop on Program Correctness, Aalborg, Denmark (paper). October 1990.

1st Nordic Transputer Seminar, Turku, Finland. October 1990.

Visit to Joensuu University (invited lecturer). November 1990.

Aimo Törn

1989 14th IFIP Conference on Systems Modelling and Optimization, Leipzig, GDR (paper).

First Finnish-Hungarian Workship: Symposium on Programming Languages and Tools, Szeged, Hungary. August 1989.

1990 Seminar on Quality and Maintenance. Lecturer: Dr. Barbara Kitchenham, prof. Aimo Törn och dir. Iikka Ahonen, Turku, (organizer, speaker). April 1990.

The Kilpisjärvi Seminar of the Doctoral Programme in Information Systems (invited speaker). April 1990.

Software Quality Work Shop, Dundee, Scotland (paper). June 1990.

Visit to The Royal Institute of Technology, Dept. of Mathematics, Division of Optimization and Systems Theory, Stockholm (invited speaker). November 1990.

II Workshop on Global Optimization, IIASA, Sopron, Hungary (paper). December 1990.

Visit to Leipzig Technische Hochschule, Germany (talk). December 1990.

Harry Virtanen

1990 ECAI 90 European Conference on Artificial Intelligence, Stockholm, Sweden. August 1990.

Neuro-Nimes 90, 3rd Onternational Workshop Neural Networks & Their Applications, Nimes, France. November 1990.

Joakim Waxlax

1990 ECAI 90 European Conference on Artificial Intelligence, Stockholm, Sweden. August 1990.

Sheffield University, Dept. of Computer Science, Sheffield, UK. October 1990.

University of London, Queen Mary and Westfield College, Dept. of Computer Science, London, UK. October 1990.

Neuro-Nimes 90, 3rd Onternational Workshop Neural Networks & Their Applications, Nimes, France. November 1990.

Chapter 12

Accepted Theses

The theses accepted at the department during 1989 and 1990. The titles have been translated from Swedish to English.

12.1 Theses for Doctor of Philosophy

1990 Eriksson, I., *Simulation for user training*

Sere, K., *Stepwise Derivation of Parallel Algorithms*

von Wright, J., *A Lattice-theoretical Basis for Program Refinement*

12.2 Theses for Licentiate of Philosophy

1989 Solin, U., *Animation of Parallel Processes*

von Wright, J., *A Lattice-theoretical Basis for Stepwise refinement of Programs*

1990 Shen, H., *Contributions to Mapping, Routing and Sorting in Parallel Processing*

12.3 Master of Science

1989 Björkstrand, K., *Computer for purchase*

Bäckman, K., *Choice of CASE systems*

Heinonen, B., *A graphical tool for rigorous program development*

Juselius, C., *A topological global optimization method*

Levander, S., *Mossu: a monitoring tool for a multiprocessor system*

Lindström, M-L., *APA - A Tool for Animating Parallel Algorithms - An Implementation of Animation Processes*

Mörk, T., *The need of ADP in Finnish sport organizations*

Nissén, A-S., *User participation in system development*

Norrbo, M., *Realization of process graphs on processor graphs*

Nyman, Y., *APA - animation of parallel algorithms*

Sjöholm, G., *A Critical Survey of Decision Support Systems*

Ståhl, L., *An implementation of multi-process handshaking on transputer networks*

1990 Ahlskog, P., *Case Studies about Data Collection for Follow-up of Software Costs*

Aspnäs, J., *Operating Systems for Distributed Architechtures*

Finnäs, A., *Visualitation of Information Systems*

Hekanaho, J., *TRIAS, A Transformation System for Action Systems*

Helkiö, T., *Integration of CAD and DBMS*

Johansson, H., *Software Quality in Finnish Companies*

Lillqvist, J-A., *S.W.I.F.T.*

Lindroos, N., *Software Maintenance - A Quality-retaining Activity*

Lundström, R., *Simulation by Simscript II.5*

Långbacka, T., *Hathi-2 Monitoring*

Lönnqvist, T., *OMT - Diagrams - A Graphical Tool in Design of Relation Databases*

Palm, C., *Distributed Termination*

Pundars, B., *Strategic Planning by Expert Systems*

Ramstedt, M., *Documentation of Software Systems by Hypertext*

Sarén, H., *Experimentation in Software Engineering: Development Techniques - Quality*

Ståhls, M., *CSAC Analysis of Digit Signals in Convertion Chronograph Analysis Systems*

Söderman, J., *From a Hierachical Database System to a Relation Database System*

Waxlax, P., *A Graphical User Interface for a Program Transformation Environment*

Virtanen, H., *Fuzzy Prolog*

Chapter 13

Publications 1989, 1990

1989

Books

[Törn and Žilinskas 89] Törn, A., Žilinskas, A., *Global Optimization*. Lecture Notes in Computer Science, No 350, 255 pp. Springer Verlag, Heidelberg,1989.

Abstract. Global optimization is concerned with finding the global extremum (maximum or minimum) of a mathematically defined function (the objective function) in some region of interest.

In many practical problems it is not known whether the objective function is unimodal in this region. In many cases the objective function has proved to be multimodal. Unsophisticated use of local optimization techniques is normally inefficient for solving such problems. Therefore, more sophisticated methods designed for global optimization, i.e., global optimization methods, are important from a practical point of view.

Most methods discussed here assume that the extremum is attained in the interior of the region of interest, i.e., that the problem is essentially unconstrained. Some methods address the general constrained problem. What is excluded is the treatment of methods designed for problems with a special structure, such as quadratic programming with negatively quadratic forms.

This book is the first broad treatment of global optimization with an extensive bibliography (35 pages), covering research done both in east and west. The entries in the bibliography give the chapter for which they have relevance. Different ideas and methods proposed for global optimization are classified, described and discussed. The efficiency of algorithms is compared by using both artificial test problems and some practical problems. The solutions of two practical design problems are demonstrated and several other applications are referenced.

The book aims at aiding in the education and at stimulating the research in the field, and at advising practitioners in using global optimization methods for solving practical problems.

Journals

[Back and Kurki-Suonio 89] Back, R.J.R., Kurki-Suonio, R., Decentralization of Process Nets with Centralized Control. *Distributed Computing*, Vol 3, No 2, May 1989, pp. 73–87. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 58, 1988.

Abstract. The behavior of a net of interconnected, communicating processes is described in terms of the joint actions in which the processes can participate. A distinction is made between

centralized and decentralized action systems.In the former, a central agent with complete information about the state of the system controls the execution of the actions; in the latter no such agent is needed.Properties of joint action systems are expressed in temporal logic. Centralized action systems allow for simple description of system behavior. Decentralized (two-process) action systems again can be mechanically compiled into a collection of CSP processes. A method for transforming centralized action systems into decentralized ones is described. The correctness of this method is proved, and its use is illustrated by deriving a process net that distributedly sorts successive lists of integers.

[Caianiello et al 89b] Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., Implementations of the C-calculus, *Connection Science*, 1 (1989), pp. 41–51.

[Sørgaard 89b] Sørgaard, P., Transaction-supporting Systems and Organizational Change. *Office: Technology and People*, Vol 4, No 3, pp. 229–243, June 1989.

Abstract. Different types of organisations exploit different kinds of computer systems. The transaction cost theory has been used to explain the role of computer systems in different types of organisations. In the transaction cost theory a distinction is made between three types of organisation: the market, the bureaucracy, and the group. Two key factors are uncertainty and the tolerance of opportunistic behaviour. Markets require low uncertainty, but tolerate opportunistic behaviour. Groups can handle exchanges with a high degree of uncertainty because oppotunistic behaviour is absent. Bureaucracies have characteristics between markets and groups. In reality organisations exhibit a mix of these organisational forms. Computer supported cooperative work currently receives much attention. The group concept from the transaction cost theory can be used as a partial characterisation of cooperative work. Transaction supporting systems are computer systems which support the constituent transactions of an organisation. There is an ongoing discussion about possible shifts on the scale from market to group induced by transaction supporting systems. Transaction supporting systems supporting market organisation can remove some of the reasons for shifts from market to more complex organisations. The set of transaction supporting systems at the disposal for bureaucracies and groups is, however, much larger than the set available to market organisation. Computer systems supporting cooperative work can only be fully exploited by a group. Therefore any prediction of a shift will depend on the kinds of computer systems taken into consideration. The conclusion is that although the transaction cost theory is useful in characterising the role of computer systems in organisations, we cannot use the theory to make general assertions about the impact of computer systems on the relative benefits of different types of organisation.

[Sørgaard 89d] Sørgaard, P., Report on the 1st eurpoean cscw conference. *Cosmos Information Exchange Network*, pp. 22–25. 1989.

[Törn 89a] Törn, A., An efficient procedure for determining the enabled set. *Petri Net Newsletter* 31, pp. 23–27. Gesellschaft für Informatik, Bonn, 1989.

Abstract. Executing pure or extended Petri Nets means firing transitions belonging to the enabled set. The firing of a transition may change the enabled set, which therefore has to be redetermined after each firing. We here investigate a procedure by which this can be done without checking all transitions. The idea is to use a precomputed (transition × transition) matrix whose elements show how the firing of a transition changes the "enabling distance" of transitions. The "enabling distance" of a transition can be regarded as a distance from the current marking to a marking in which the transition is enabled.

[Walldén and Sere 89] Walldén, M., Sere, K., Free-text Retrieval on Transputer Networks. *Microprocessors and Microsystems*, Vol 13, No 3, April 1989, pp. 179–187.

Abstract. An implementation of a document retrieval system developed on Hathi-2 is described. Hathi-2 is a multi-processor system built of IMS T800 transputers. An efficient distributed search strategy in large free-text databases is derived following the processor farm paradigm for programming parallel and distributed systems. The processor farm approach is considered for several different transputer network configurations. The main purpose of this paper is to study the applicability of this approach on transputer networks and to report on the observed performance figures for the document retrieval system.

Collections

[**Caianiello et al 89a**] Caianiello, E.R., Eklund, P., Squillante, M., Ventre, A.G.S., Formalism and Implementations of C-calculus. In Martelli, I. A., Valle, G. (eds), *Computional Intelligence I*, pp. 15–26. North-Holland, 1989.

[**Eklund and Malén 89**] Eklund, P., Malén, T-E., Block Placement in Switching Networks. In Jesshope, C.R., Reinartz, K.D. (eds), *CONPAR 88*, pp. 299–305. Cambridge University Press, 1989.

Proceedings

[**Aspnäs and Back 89a**] Aspnäs, M., Back, R.J.R., Multiprocessor Applications in the Hathi project. *In Kankaala, K., Nieminen, K. (eds), Scientific Computing in Finland.* CSC Research Reports R1/89, pp. 5–21, 1989. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 96, 1989.

Abstract. This report describes some of the more important applications of parallel processing in the Hathi project. The aim of the applications was to evaluate the use of transputer-based parallel processors for solving large computational problems. The report presents the problems solved in the application programs and the experiences gained from this work.

[**Aspnäs and Back 89b**] Aspnäs, M., Back, R.J.R., A Programming Environment for a Transputer-Based Multiprocessor System. In *Gyimóthy, T. (ed.), Symposium on Programming Languages and Software Tools*, pp. 94 – 103. Hungarian Academy of Sciences 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 82, 1989.

Abstract. This paper presents a transputer-based multiprocessor system, Hathi-2, and the programming environment being developed for this system. Hathi-2 is mainly programmed in the language Occam, and thus the programming environment is based on the Occam model of parallelism and communication. The programming environment gives the user an abstraction of the physical structure of the multiprocessor system. The user sees the multiprocessor system as a pool of resources (processors and communication links), which are allocated to the the users program and connected to the topoloys described by the program structure. The environment is implemented on a Sun 3 graphical workstation.

[Back 89a] Back, R.J.R., Changing Data Representation in the Refinement Calculus. *Proc. Hawaii International Conference on System Sciences (HICSS-22)*, January 1989, Hawaii.

Abstract. The refinement calculus gives a formal basis for the stepwise refinement method of program construction. The caculus is an extension of the weakest precondition calculus of Dijkstra. In this paper we show how to systematically change the data representation in programs (*data refinement*)within this calculus. The original method for data refinement in the refinement calculus [Ba80] was only defined for functional data abstractions. We show here that this method can be extended to non-functional data abstraction, by extending the

weakest precondition calculus with conjugate statements. These model the total correctness of statements with don't know (or angelic) nondeterminism rather than the usual don't care (or demonic) nondeterminism assumed by Dijkstra. The data refinement method described here is very flexible, permitting different data abstractions to be combined in the same statement, as well as refinements of the data abstractions themeselves. An example of using this flexibility in program derivation is given in the paper.

[Back and Sere 89] Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems. In van de Snepscheut, J.L.A. (ed.), *Proc. Conference on Mathematics of Program Construction*, Groningen, the Netherlands, June, 1989. Lecture Notes in Computer Science No 375, pp. 115–138. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 78, 1989.

Abstract. A stepwise refinement method for the formal development of provably correct parallel algorithms is presented. Following our systematic method the entire derivation procedure is carried out in the context of purely sequential programs. The resulting parallel algorithm can be effectively executed in different architectures. The methodology is illustrated by showing the leading derivation steps in a construction of a parallel algorithm for square matrix multiplication.

[Back and vWright 89a] Back, R.J.R., von Wright, J., A Lattice-theoretic basis for a specification language. In van de Snepscheut, J.L.A. (ed.), *Proc. Conference on Mathematics of Program Construction*, Groningen, the Netherlands, June 1989. Lecture Notes in Computer Science No 375, pp. 139–156. (Short version of [Back and von Wright 89b].)

[Caianiello et al 89c] Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., A Parallel Implementation of the C-calculus. In Caianiello, E. R. (ed.), *1st Italian Workshop on Parallel Architectures and Neural Nets*, pp. 46–62. World Scientific, 1989.

[Caianiello et al 89d] Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., In Pietikäinen, M., Röning, J. (eds), Parallel C-calculus: A Case Study. *Proc. 6th Scandinavian Conference on Image Analysis (6SCIA)*, Vol II pp. 705–708. Oulu, June 1989.

[Caianiello et al 89e] Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., C-calculus and Uncertainty. *Proc. 3rd Congress of the International Fuzzy Systems Association (IFSA)*, pp. 678–680. Seattle, August 1989.

[Eklund 89] Eklund, P., Optimal Mappings for Variable Architectures. *Proc. 14th IFIP Conference on System Modelling and Optimization*, Leipzig, July, 1989, Wissenschaftliche Berichte der Technischen Hochschule Leipzig, 5, pp. 78–81, 1989.

[Eklund and Gähler 89] Eklund, P., Gähler, W., General Structures and Fuzzy Filters. *Proc. 3rd Congress of the International Fuzzy Systems Association (IFSA)*, (invited talk), pp. 266–268. Seattle, August, 1989.

[Eriksson 89] Eriksson, I., Learning Process in the Context of Using ISs. IFIP WG 8.2. In Klein, H.K., Kumar, K. (eds), *Systems Development for Human Progress* (panel presentation), pp. 267–275. Atlanta, Georgia, May 1987. North-Holland, 1989.

[Eriksson and Finnäs 89] Eriksson, I., Finnäs, A., A Visual Simulation Model of Information and Work Systems. In Bødker, S. (ed.), *Proc. 12th IRIS - Part I*, pp. 149–168. DAIMI PB—296-II, Aarhus University, Aarhus, 1989.

Abstract. To facilitate the use situation an interface that supports description of the information system, the structure of the organization, and division of labour is proposed. A Help-system

with a visual simulation model is an important part of this interface. The simulation model allows the user to follow the transactions on different hierachical levels of detail; transaction flows between departments and units, detailed manipulation on the section level, and step-by-step progress of computerized functions. The time dimension is taken into consideration and the transactions can be followed forward but also be traced backward. The backward trace is accomplished by two history files concerning data and actions taken. The model is designed to be used in two modes: simulation controlled by the system and simulation controlled by the user, controlled and interactive simulation, respectively. A prototype version of controlled simulation is presented as an example. It is implemented on Macintosh using Hypercard.

[**Rantala et al 89**] Rantala, A., Raunio, A., Still, D-J., A Multiprocessor System for Fast Geometric Image Transformation. In Pietikäinen, M., Röning, J. (eds), *Proc. 6th Scandinavian Conference on Image Analysis (6SCIA)*, Vol II, pp. 1182–1189. Oulu, June 1989. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 66, 1988.

Abstract. A prototype for a transputer based image processing system is described. The system performs a user-definable geometric (space domain) transformation that ca be used to correct distortions in image data received from a satellite and to produce images using a predefined projection and orientation. In order to shorten the processing time the problem has been partitioned for a processor farm. The results from the prototype suggests that a transputer based solution is efficient for this kind of real time applications.

[**Shen 89a**] Shen, H., Fast Path-disjoint Routing in Transputer Networks. In Gyimóthy, T., (ed.), *Symposium on Programming Languages and Software Tools*, pp. 157 –167. Hungarian Academy of Sciences 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 83, 1989. To appear in Microprocessing and Microprogramming. North-Holland.

Abstract. This paper addresses the problem of path-disjoint routing in transputer networks. In this paper, we first study the criteria for path-disjoint routing, then give heuristic approaches to the criteria, and finally present a fast heuristic algorithm to solve this problem on transputer networks. For routing k disjoint paths, our algorithm works on a $m \times n$ mesh, multigrid or torus structure in $O(km^2n^2)$ time. This algorithm has also been implemented in Occam on the Hathi-2 transputer network. The implementation result shows a layout with minimum path length and least path bends for all of the produced paths.

[**Shen 89b**] Shen, H., Self-adjusting mapping:a heuristic mapping algorithm for mapping parallel programs onto transputer networks. In Wexler, J. (ed.), *Developing Transputer Applications (OUG11)*, pp. 89–98. IOS Amsterdam 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 87, 1989.

Abstract. The problem of mapping parallel programs onto multiprocessor system is a fundmental problem of great significance in parallel processing, but it is NP-hard in general. In this paper we propose a fast heuristic algorithm to solve this problem on transputer networks. Our mapping algorithm consists of three modules: grouping, placement and routing, where grouping groups processes in the program into tasks which can be placed onto processors in the transputer network in a way of one-to-one, placement places the grouped tasks onto the processors and routing produces physical communication paths for logical communication requirements. The three modules work co-operatively in a way of progressive self-adjusting, and finally produce a satisfactory solution for the mapping problem.

[**Shen 89c**] Shen, H., Mapping Parallel Programs onto Transputer Networks. In *Proc. of Australian Transputer and Occam User Group Conference 90*, pp. 85–94, RMIT Aus-

tralia 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 79, 1989.

Abstract. A fundmental problem of great significance in parallel processing is to map parallel programs onto parallel computers. In this paper we begin with studying the mathematical model for the mapping problem, then discuss different strategies to solve this problem, and finally present a fast heuristic algorithm to solve this problem on transputer networks. Our mapping algorithm functionally consists of partitioning (grouping), placement and routing that work co-operatively in a divide-and-conquer way. It finally produces a satisfactory result for the mapping problem. The proposed algorithm can also be easily realized in a parallel way.

[Sørgaard 89c] Sørgaard, P., Computer-supported cooperative work: a challenge to system developers. *Proc. of Tietotekniikka 89*, University of Jyväskylä, May, 1989.

Abstract. Computer-supported cooperative work (cscw) is a relatively new research field. One of the outcomes of this research is a number of so-called cscw-applications. Human cooperation, however, is not new, and it is only some specific kinds of cooperation which is addressed by the typical cscw-applications. Every system developer should therefore ask: How can we support cooperation with the systems we develop?

The way people cooperate is often specific to the task in question. Good systems development must therefore be based on knowledge about the task and the way people cooperate. This is a necessary conservative point of view. We must also take care so that the systems do not freeze the current patterns of cooperation and we must use the opportunities for improved support for cooperation offered by the introduction of computers. This is a necessary radical point of view. It is in the systems development process a balance has to be found between these conservative and radical points of view.

[Sørgaard at al 89] Sørgaard, P., Nurminen, M., Forsman, U., System Maintenance and Organizational Change. In Bødker, S. (ed.), *Proc. 12th IRIS - Part II*, pp. 567–586. DAIMI PB—296-II, Aarhus University, Aarhus, 1989.

Abstract. There are many indications that maintenance of computer-based information system is critical. It consumes many resources and it is highly necessary in organisations which have made their information systems a central part of their operations.

We can also observe that organisations constantly change. Besides the permanent need to adjust to changes in the organisation's environment, there are tendencies towards decentralisation and to move away from classical hierarchy towards flatter organisations (e.g., cooperative work).

This raises the question: how and to what extent can the /cbiss¡¿ become obstacles to organisational change? How is this phenomenon related to the difficulties in maintaining and changing the current systems?

We propose that research is made to investigate this issue. Such a research effort can be justified by existing quantitative results showing that most of the maintenance work is done due to changed user requirements. We want to study this problem in detail, and we therefore propose a research project with four phases: preliminary investigations (interviews and literature study), in-depth analysis of the situation in one or two case-organisations, formulation of change strategies, and implementation of one or more of the change strategies. We do in other words propose case-based, qualitative research in an area which primarily has been researched by quantitative techniques.

[Törn 89b] Törn, A.A., PICA – A graphical program development tool. In Gyimóthy, T. (ed.), *Symposium on Programming Languages and Software Tools*, pp. 318–330. Hungary Academy of Sciences 1989. Also in Proc. of the Joint Finnish-Soviet Software Symposium: Software Development Trends, pp. 65–78. Helsinki, 1989. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 60, 1988.

Abstract. A technique and a tool PICA for formal program development using flowcharts is presented. The formal technique applied is that of proving theorems of the type {*pre-condition*}, *statement*, {*post-condition*}. This technique is applied to stuctured program flowcharts by adding assertion nodes containing program variable names and assertions about their values. An assertion node is connected to or from a statement node depending on if it represents a pre-condition or a post-condition. A tool for convenient use of the technique has been implemented as an Add-On to the Design software of Meta Software on a Macintosh II. The feasibility of using PICA is demonstrated by developing an algorithm for a small non-trivial programming task. The incentive for presenting the PICA technique is to create broader interest for formal programming methods by presenting one formal technique applicable to program development using flowcharts.

Technical Reports

[**Aspnäs and Malén 89a**] Aspnäs, M., Malén, T-E., *Hathi-2 Users Guide, version 1.0.* Åbo Akademi. Reports on Computer Science & Mathematics, Ser. B. No 6, 1989.

Abstract. Hathi-2 is a reconfigurable general purpose loosely coupled MIMD multiprocessor system consisting of 100 32-bit IMS T800 transputers and 25 16-bit IMS T212 transputers. Hathi-2 has a distributed switching network built of IMS C004 crossbar switches and a distributed control system. The system can be used by several independent users simultaneously. Each user is assigned a separate subsystem of Hathi-2. This report describes the Hathi-2 system and how it can be used.

[**Aspnäs and Malén 89b**] Aspnäs, M., Malén, T-E., *Transputer benchmark tests.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 97, 1989.

Abstract. The transputer is a powerful processor designed to be used as a component in multiprocessor systems. In this paper, the transputers processing power is compared to some other commonly available computer systems by means of a number of synthethic benchmark programs. The benchmarks programs mainly test how well a processor performs scientific computation. The test programs consist of short sequential Fortran and Occam programs, which are executed on a number of different computer systems. The transputers communication capacity is also tested and the factors that affect the communication capacity over transputer links are investigated.

[**Aspnäs et al 89a**] Aspnäs, M., Back, R.J.R., Kurki-Suonio, R., *Efficient Implementation of Multi-process Handshaking on Broadcasting Networks.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 75, 1989.

Abstract. Multi-process handshaking is a generalization of ordinary handshake communication between two processes. It allows an arbitrary number of processes to be synchronized in a common handshake, for the purpose of carrying out a joint action, consisting of a sequential statement that is executed in the combined state space of the processes participating in the handshake. The statement updates the local variables of the processes in a manner that depends on the values of the local variables of the other processes. The paper is concerned with implementing this communication mechanism on networks with a broadcasting facility. The implementation is described and its correctness is proved. The efficiency is calculated analytically and verified experimentally by simulation studies.

[**Aspnäs et al 89b**] Aspnäs, M., Back, R.J.R., Malén, T-E., *Hathi-2 Users Guide, version 1.0.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. B, No 6, 1989.

Abstract. Hathi-2 is a reconfigurable general purpose loosely coupled MIMD multiprocessor system consisting of 100 32-bit IMS T800 transputers and 25 16-bit IMS T212 transputers.

Hathi-2 has a distributed switching network built of IMS C004 crossbar switches and a distributed control system. The system can be used by several independent users simultaneously. Each user is assigned a separate subsystem of Hathi-2. This report describes the Hathi-2 system and how it can be used.

[Aspnäs et al 89c] Aspnäs, M., Back, R.J.R., Sere, K., *The Hathi Project—A Research Project on Parallel Programming Technology 1986–88 (final Report).* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 95, 1989.

[Back 89b] Back, R.J.R., *Refinement Calculus, Part II: Parallel and Reactive Programs.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 93, 1989. To appear in Proc. REX Workshop for Refinement of Distibuted Systems. Nijmegen, July 1989.

Abstract. It is shown how to apply the refinemtn calculus to stepwise refinement of both parallel programs and reactive programs. The approach is based on using the action systems model to describe parallel and reactive systems. Action systems are sequential progrmas which can be implemented in a parallel programs expressed in this framework. Refinement of reactive programs can be expressed and proved in the refinement calculus by using the methods of data refinement from the sequential refinement calculus.

[Back and Törn 89] Back, R.J.R., Törn, A., *Åbo Akademi. Department of Computer Science. 5-year Report 1984–1989.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. B. No 7, 1989.

[Back and vWright 89b] Back, R.J.R., von Wright, J., *Duality in Specification Languages: A Latice Theoretic Approach.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 77, 1989. To appear in Acta Informatica.

Abstract. A very general lattice-based language of commands, based in the primitive operations of substitution and test for equality, is constructed. This base language permits unbounded nondeterminism, demonic and angelic nondeterminism. A dual language permitting miracles is constructed. Combining these two languages yields an extended base language which is complete, in the sense that all unifying framework for various specification languages; we show how two Dijkstra-style specification languages can be embedded in it.

[Back and vWright 89c] Back, R.J.R., von Wright, J., *Refinement Concepts Formalized in Higher Order Logic.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 85, 1989. To be presented at IFIP TC2 Working Conference. April 1990, Sea of Galilei, Israel.

Abstract. A theory of commands with weakest precondition semantics is formalized using the HOL proof assistant system. The concept of refinement between commands is formalized, a number of refinement tules are proved and it is shown how the formalization can be used for proving refinements of actual program texts correct.

[Back and vWright 89d] Back, R.J.R., von Wright, J., *Combining angels, demons and miracles in program specifications.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 86, 1989. To appear in Theoretical Computer Science.

Abstract. The complete lattice of monotonic predicate transformers is interpretated as a command language with a weakest precondition semantics. This command lattice contains Dijkstra's guarded commands as well as miracles. It also permits unbounded nondeterminism and angelic nondeterminism. The language is divided into sublanguages using criteria of demonic and angelic nondeterminism, termination and absence of miracles. We investigate dualities between the sublanguages and how they can be generated from simple primitive commands. The notions of total correctness and refinement are generalized to the command lattice.

[**Back and vWright 89e**] Back, R.J.R., von Wright, J., *Refinement Calculus, Part I: Sequential Nondeterministic Programs.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 92, 1989. To appear in Proc. REX Workshop for Refinement of Distibuted Systems, Nijmegen July 1989.

Abstract. A lattice theoretic framework for the calculus of program refinement is presented. Specifications and program statements are combined into a single (infiniary) language of commands which permits miraculous, angelic and demonic statements to be used in to cover this larger class of statements and a game-theoretic interpretation is given for these constructs. The language is complete, in the sense that every monotonic predicate transformer can be expressed in it. The usual program constructs can be defined as derived notions in this language. The notion of inverse statements is defined and its use in formalizing the notion of data refinement is shown.

[**Ehrenberg et al 89**] Ehrenberg, D., Eklund, P., Fedrizzi, M., Ventre, A., *Consensus in Distributed Soft Environments.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 88, 1989.

[**Eriksson and Törn 89**] Eriksson, I., Törn, A., *A Covering Structure of IS Quality Concepts.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 89, 1989.

Abstract. Different concepts of software quality are reviewed and discussed. Based on this a hierarchical model of information system (IS) quality concepts developed in the research project SOLE is presented. This model aims at a division of quality concepts consistent with the different decision makers and decisions made during the software life-cycle. The main division is: *IS Cost Effectiveness, IS Use Quality* and *IS Work Quality.* The last two are further divided into Requirement Quality, Interface Quality, and Efficient IS Management, IS Design and Implementation Quality respectively.

[**Kok 89a**] Kok, J., *Specialization in Logic Programming: From Horn Clause Logic to Prolog and Concurrent Prolog.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 90, 1989.

Abstract. A Prolog or a Concurrent Prolog program can be seen as a specialization or refinement of a program in Horn Clause Logic: In addition to the logic component a Porolog or Concurrent Prolog program contains information about the flow of control. In Prolog we have the cut statement and a leftmost depth first search strategy, in Concurrent Prolog we have read-only variables and commits. In this paper we study the flow of control of these languages by giving transition systems for abstract versions of Prolog, Horn Clause Logic and Concurrent Prolog. On the basis of these transition systems we define operational semantics for all three languages. Three basic sets (success set, finite failure set and the infinite failure set or divergence set) can be derived from the opperational semantics. A comparison is made between the different sets: for Horn Clause Logic and Prolog we show that the success set and the finite failure sets of a Prolog program are smaller than the corresponding sets of a Horn Clause Logig program. The infinite failure sets are incomparable. A similar comparison is made between the success set and the finite failure sets fo Horn Clause Logic and Concurrent Prolog. These comparisons give some feeling what happens if put extra logical information in Horn Clause Logic programs.

Remark: part of the work was carried out in the ESPRIT Basic Research Action Integration.

[**Kok 89b**] Kok, J., *Traces, Histories and Streams in the Semantics of Nondeterministic Data Flows.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 91, 1989.

Abstract. We study several models for (possibly nondeterministic) dataflow. These models are based on histories, streams and traces. In the literature there are several proposals about how to build these models (for example from specifications of the nodes in a network). Important issues are compositionality and full abstraction.

[Komorowski 89] Komorowski, J., *Towards Synthesis of Programs in the Partial Deduction Framework.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 81, 1989. Shorter version to appear in Proc. XIth International Joint Conference on Artificial Intelligence, Workshop on Automating Software Design, Detroit, MI, USA, August 1989.

Abstract. We are developing principles for synthesis of programs based on the principle of partial deduction and a generic interactive proof-support environment. Partial deduction in logic programming (previously called partial evaluation in logic programming) can be described in the following way. Given a program P and a goal G, partial deduction produces a new program P' which is simplified with respect to the goal G. The resulting P' should have the same semantics as P with respect to G, that is, the computed and the correct answers for G in P' should be equal to answers for G in P. It is also expected that G should be executed more efficiently in P' than in P. Standard partial deduction is a powerful synthesis method, but it still has a number of problems. We propose to alleviate some of them by extending it with two tactics: opening, a tactic that controls exponential growth of generated code and abbreviating, a tactic that handles recursion. When compared with some other synthesis methods proofs in the extended partial deduction framework are shorter and do not require user-supplied lemmata. In particular, partial deduction can automatically formulate inductive schemata. The partial deduction engine is then incorporated into a generic proof-support environment with a two-dimensional display of derivations represented as trees. Each branch of the derivation tree is a sequential representation of successive refinements. Since the entire derivation state is available to the user, it is possible to examine past refinements, make modifications, and rederive the programs in the nodes below the modification. This approach is different from, for example, the approach taken in KBEmacs. In KBEmacs, there is only a current state of the derivation available to the user. Without retracting derivation steps it seems impossible to make changes in the past states and to rederive a slightly modified program.

[Shen 89d] Shen, H., *Occam Implementation of Path-disjoint Routing on the Hathi-2 Transputer System.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 98, 1989.

Abstract. This paper shows how to develop an Occam program to heuristically solve the path-disjoint routing problem in transputer networks. In this paper, we first introduce the heuristic criteria for path-disjoint routing and a fast routing algorithm based on the criteria, then describe the design of the Occam program for path-disjoint routing based on the algorithm, and finally show the implementation result of the Occam program on the Hathi-2 transputer system. With the main feature of no batched data-swapping during program execution, our program can be efficiently implemented. For the problem to find k edge-disjoint paths in a $m \times n$ mesh, multigrid or torus, the program runs in time $O(k^2 + km^2n^2)$ on one transputer. All paths in a successful solution produced by the program are with possibly minimum length and least bends, which provides an optimal embedded layout.

[Sørgaard 89a] Sørgaard, P., *An Overview of Research in Maintenance.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 94, 1989.

Abstract. Research in maintenance is surveyed. Software maintenance can be adressed in many ways. This survey focuses on the possible interaction between software maintenance and the organisations using the software. Maintenance is seen to cover all activities performed on a

system after it has been put in production. Three broad classes of maintenance are identified: *repair maintenance, functional modification,* and *miscellaneous maintenance.*

Estimates of the cost of maintenance vary considerably. Due to differences in terminology and units of measurement the different reports are hard to compare.

The importance of maintenance is a topic of discussion. The high costs are in themselves a reason for considering maintenance important. Empirical studies of attitudes to maintenance and related issues indicate that maintenance is not considered a top priority issue.

Large efforts have been made at developing and testing software metrics. Many attempts have been made to apply these to predict the maintainability and error frequencies of software. The results are not very convincing.

There are many relevant studies of maintenance and other aspects of computing. It is generally agreed that *functional modification* is the major kind of maintenance. It has also been demonstrated that many computer-based information systems do not work well in practice, that their designs are disputed, and that the development processes are conflict-ridden. This can explain the high volume of *functional modification.*

[Ståhl and Back 89] Ståhl, L., Back, R.J.R., *An Implementation of Multiprocess Handshaking on Transputer networks.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 76, 1989.

Abstract. Multiprocess handshaking is a generalization of ordinary handshake communication between two processes. It allows an arbitrary , though pre-defined number of asynchronous processes to be synchronized in a common handshake, for the purpose of carrying out a communication event. This is a sequential statement that is executed in the combined state space of the processes participating in the handshake. The statement updates the local variables of the processes, in a manner that depends on the values of the local variables of the other processes. The paper is concerned with implementing this communication mechanism on transputer networks. Bagrodias Event-manager algorithm is used for synchronizating the processes.

[vWright 89a] von Wright, J., *Stepwise Derivation of a Parallel Matrix Multiplication Algorithm.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 84, 1989.

Abstract. A standard multiplication algorithm for square matrices is transformed into a distributed algorithm. Every transformation step is verified within the refinement calculus. Some high-level rules of refinement that are used in the derivation are proved.

Theses for Ph.Lic.

[Solin 89] Solin, U., *Animation of Parallel Processes.* Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1989.

[vWright 89b] von Wright, J., *En Lattisteoretisk bas för stegvis programutveckling.* (A Lattice-theoretic Basis for Stepwise Refinement of Programs). Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1989.

1990

Books

[Andersen et al 90] Andersen, N., Kensing, F., Lundin, J., Mathiassen, L., Munk-Madsen, A., Rasbech, M., Sørgaard, P., *Professional Systems Development.* Prentice Hall, Business Information Technology Series (U.K.) 1990.

Journals

[Aspnäs and Back 90] Aspnäs, M., Back, R.J.R., A Programming Environment for a Transputer-Based Multiprocessor System. *Acta Cybernetica*, Tom 9, Fasc. 3, pp. 291–301. Szeged 1990. Also published in Gyimóthy, T. (ed.), Symposium on Programming Languages and Software Tools, pp. 94–103. Hungarian Academy of Sciences 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 82, 1989.

Abstract. For abstract, see [Aspnäs and Back 89b].

[Aspnäs et al 90] Aspnäs, M., Back, R.J.R., Malén, T-E., The Hathi-2 Multiprocessor System. *Microprocessors and Microsystems*, Vol 14, No 7, pp. 457–466. Butterworth-Heinemann Ltd, 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 80, 1989.

Abstract. Hathi-2 is a reconfigurable general purpose loosely coupled MIMD multiprocessor system consisting of 100 32-bi IMS T800 transputers and 25 IMS T212 transputers. Hathi-2 has a distributed switching network and a distributed control system which contains an interrupt system and hardware support for distributed monitoring. This report describes the architecture of the Hathi-2 system and the design goals of the system. The system software that has been developed for the system is also presented.

[Back and Sere 90a] Back, R.J.R., Sere, K., Stepwise Refinement of Parallel Algorithms. *Science of Computer Programming*, Vol 13, No 2–3, pp. 133–188. North-Holland 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 64, 1988.

Abstract. The refinement calculus and the action system formalism are combined to provide a uniform method for constructing parallel and distributed algorithms by stepwise refinement. It is shown that the sequential refinement calculus can be used as such for most of the derivation steps. Parallelism is introduced during the derivation by refinement of atomicity. The approach is applied to the derivation of a parallel version of the Gaussian elimination method for solving simultaneous linear equation systems.

[Back and Sere 90b] Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems. To appear in *Structured programming*, January 1991. Also in van de Snepscheut, J.L.A. (ed.), Proc. Conference on Mathematics of Program Construction. Lecture Notes in Computer Science No 375, pp. 115–138. Springer Verlag 1989. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 78, 1989.

Abstract. For abstract, see [Back and Sere 89a]

[Back and vWright 90a] Back, R.J.R., von Wright, J., Duality in Specification Languages: A Latice Theoretic Approach. *Acta Informatica*, Vol 24, Fasc. 7, 1990, pp. 583–625. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 77, 1989.

Abstract. For abstract, see [Back and vWright 89b]

[Back and vWright 90b] Back, R.J.R., von Wright, J., Refinement Concepts Formalized in Higher Order Logic. *Formal Aspects of Computing*, No 2, pp. 247–277. BCS 1990. Also in preprints of IFIP TC2 Working Conference on Programming Concepts and Methods, pp. 171–192. Israel 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 85, 1989.

Abstract. For abstract, see [Back and vWright 89c]

[Eklund and Gähler 90a] Eklund, P., Gähler, W., Generalized Cauchy Spaces. *Math. Nachr. 147*, pp. 201–215. 1990.

[Eriksson and Reijonen 90] Eriksson, I., Reijonen, P., Training Computer-Supported Worl by Simulation. In *Education and Computing.* The International Journal, Vol. 6, Nos 1–2, pp. 129–136. 1990. Also published in Barta, B.Z., Fontell, L., Raymont, P., Lovis, F. (eds), Methodologies of Training Data Processing Professionals and Advances End-Users, pp. 129–136. Elsevier 1990.

Abstract. Simulation has traditionally been used to describe complicated systems, to plan expensive or oteherwise critical systems, and for education. A rather new domain for simulation is information systems design and use. We claim that simulation with its various techniques can profitably be used to improve end-users' competence to use a computer-supported information system in work situations and to participate in systems development. The claim is based on the following facts: The main difference between information systems and other systems, e.g. a manufacturing system, is the level of abstraction. Simulation is one of the best ways to conretize abstract concepts and relations between them and therefore it offers good opportunities to facilitate the use of simulation learning processes. Economy is another good reason for using simulation in training work situations. Technological development will facilitate the use of simulation, animation and graphical representation so we expect more transparent, self-explaining, and easy-to-understand systems for the future.

[Eriksson and Törn 90a] Eriksson, I., Törn, A.A., SOLE – IS quality and efficient IS management. *Tietojenkäsittelytiede*, Vol 1, No 1, pp. 29–37. 1990. Revised version of [Eriksson and Törn 90c].

Abstract. The research project SOLE aiming at supporting efficient IS management is presented. This includes improvement of used IS procedures and efficient management of the growing and aging software library. The means to achieve the aim is to focus on IS quality questions and to find relations between IS quality and IS costs. Models describing software library accumulation at different quality levels and an IS quality model are developed. A software tool facilitating the collection of quality and cost data from daily IS activities is proposed. It is also planned to develop a simulation model for evaluating the results from using different development and maintenance strategies. Other topics are quantitative quality models, quality improving techniques and software maintenance.

[Hagerup and Shen 90] Hagerup, T., Shen H., Improved non-conservative sequential and parallel integer sorting. *Information Processing Letters*, Vol 36, No 2, pp. 1–7. North-Holland 1990. Also published in Technical report 10/1990, Univ. of Saarbrücken, Sonderforschungsbereich 124.

Abstract. We consider the problem of deterministic integer sorting on unit-cost sequential and parallel machines with a large word length and show that n integers drawn from $\{0, \ldots, m-1\}$ can be sorted using a word length of $O(m \log n)$ bits either in $O(n)$ time on a unit-cost RAM or in $O(\log n)$ time on a unit-cost EREW PRAM with $O(n/\log n)$ processors. Spending $O(\log\log\log m)$ additional sequential or parallel time, we can reduce the necessary word length to $O(\min\{n \log n \log m + (\log m)^{1+\epsilon}, m^{\epsilon} + \log n\})$ bits, for any fixed $\epsilon > 0$. Previous algorithms with a linear time-processor count either cannot sort arbitrary integers or require a much larger word length.

[Shen 90a] Shen, H., Occam Implementation of Path-disjoint Routing on the Hathi-2 Transputer System. *Microprocessing and Microprogramming*, No 30, pp. 93–100. North-Holland 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 98, 1989.

Abstract. For abstract, see [Shen 89d]

[Törn 90a] Törn, A.A., Models of software accumulation. *Journal of Systems & Software* 12, pp. 39–42. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 59, 1988.

Abstract. The administration of a software unit in an organization aims at producing and maintaining needed software as efficiently as possible. High efficiency means low total cost for existing software. The total cost consists of the cost for developing the software, including initial development and later additions of new features and the cost for maintaining it, including correction of errors and bad design, and modifications needed because of environmental changes. Software will accumulate over time depending on available resources. For a fixed budget the accumulation of software will be the fastest in the early years of computerization when only little software has to be maintained, and will decline over time as maintenance will need a growing share of the budget. Simple models describing this accumulation process are developed and implications of strategic decisions for software development are illustrated based on these models.

[Törn 90b] Törn, A.A., Decision support by rapid simulation using Simulation Nets. *Decision Support Systems* 6, pp. 299–305.

Abstract. The working of a tool, *Simulation Nets*, for designing and executing models for simulation of systems is presented. The tool is an extension of the theoretically attractive Petri Nets, whose suitability in general simulation modeling has largely been overlooked in the simulation community. Simulation Nets helps in obtaining a correct simulation model because of their good graphical properties, their strength in describing concurrent processes, and because of the possibility in proving correctness for some parts of the model by applying the well known *reachability tree* technique. The resulting graphical model consists of a number of submodels. The submodels are exact enough to permit simulation experiments to be performed without the need of programming. This permits easy incremental validation of the model, i.e., validation of the submodels and a hierarchy of coupled submodels. Simulation nets thus facilitates rapid modeling and experimentation and thus supports the decision maker in obtaining the data needed for him to make his decision. Experiences with a working prototype are presented.

[Törn 90c] Törn, A.A., PICA - A graphical program development tool. *Acta Cybernetica*, Tom. 9, Fasc. 3, pp. 303–321. Szeged 1990. Also in Gyimóthy, T. (ed.), Symposium on Programming Languages and Software Tools, pp. 318–330. Hungarian Academy of Sciences 1989. Also in Proc. of the Joint Finnish-Soviet Software Symposium: Software Development Trends, pp. 65–78. Helsinki, 1989. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 60, 1988.

Abstract. For abstract, see [Törn 89b]

Collections

[Eriksson 90a] Eriksson, I., Simulation as a Learning Technique in Information Systems Development. In Bjerknes, G. et al. (eds), *Organizational Competence in System Development – A Scandinavian Contribution*, pp. 233–250. Studentlitteratur, Lund, Sweden, 1990. Also published in Organizational Competence in System Development. Proceedings of Nordic Seminar, pp. 57-68. Tranum Klitgaard, Institute of Electronic Systems. Aalborg University Centre, Denmark, 1988.

Abstract. End-users' organizational competence is required in developing computer-based information systems if these are to meet the needs of the users. Simulation is proposed as a means of making it easier for users to participate. Different ways of using simulation techniques

are presented, such as simulation of changes in information systems or in work organization, role-play and manual simulation of the functions of the computer. Although evolutionary systems design environments provide the best conditions for simulation, the system life-cycle philosophy can also be applied.

[Eriksson 90d] Eriksson, I., Educating End-Users to Make More Effective Use of Information Systems. In Gattiker, U.E., Larwood, L.(eds), *Technological Innovation and Human Resources 2.* Walter de Gruyter, 1990.

Abstract. We suggest that end-users need comprehensive system knowledge to use their information systems efficiently. A programme to give this competence is presented. It is composed of the following five phases: individual interviews, group discussions, lectures, simulation, and personal guidance. Such improved end-user competence might address the problems of the current use situation, for example, the increasing maintenance costs for software and the shortage of qualified EDP-personnel.. For the future, it is recommended that training be integrated into information systems design. Facilities such as advanced user interfaces and system simulators are also proposed.

[Törn and A. Žilinskas 90] Törn, A., Žilinskas, A., Parallel global optimization algorithms in optimal design. In Sebastian, H-J., Tammer, K. (eds.), *Modelling and Optimization.* Lecture Notes in Control and Information Sciences 143, pp. 951–960. DDR 1990.

Abstract. The procedure in solving two real life optimal design problems suggest that explicite global optimization methods rather than some ad hock combination of local optimization techniques should be used. It is argumented that the large computing time needed in applying global optimization techniques and the suitability of some of these algorithms to parallelization makes them ideal candidates for execution on parallel computers. Results obtained with parallel Fortran on a 100 processor parallel computer using the processor farm configuration show that good speedup is achievable.

Proceedings

[Back 90a] Back, R.J.R., A Method for Refining Atomicity in Parallel Algorithms. In Odijk, E., Rem, M., Syre, J. (eds), *Proc. Conference on Parallel Architechtures and Languages Europe.* Lecture Notes in Computer Science No 366, Vol II, pp. 199–216. Springer Verlag 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 57, 1988.

Abstract. The paper extends the refinement calculus of sequential programs to handle stepwise refinement of parallel and distributed programs. The refinement steps preserve the total correctness of the programs. The action system approach is used to model the behaviour of parallel programs. The central topic is refinement of atomicity in such programs. It is shown how the refinement calculus can be used as a foundation for the derivation of parallel and distributed program by systematically decreasing the grain of interleaving in these and thereby increasing the amount of potential parallelism in the programs.

[Back 90b] Back, R.J.R., Refinement Calculus, Part II: Parallel and Reactive Programs. In *Proc. REX Workshop for Refinement of Distibuted Systems.* Lecture Notes in Computer Science, No 430, pp. 67–93. Springer Verlag 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 93, 1989.

Abstract. For abstract, see [Back 89b]

[Back and Sere 90c] Back, R.J.R., Sere, K., Deriving an Implementation of Action Systems. In *Proc. Third Refinement Workshop.* BCS FACS/IBM UK/oxford Univer-

sity PRG 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 99, 1990.

Abstract. The design of parallel and distributed systems in the event-based action system formalism allows the separation of concerns between the problem to be solved and the target language and machine architecture. The problems involved in executing action systems on distributed architectures is studied in this paper. Our main contribution is the definition of a class of action systems that can be mechanically compiled into occam. The occam-implementation is formally derived using a combination of the refinement calculus and the action systems methodology.

[Back and vWright 90c] Back, R.J.R., von Wright, J., Refinement Calculus, Part I: Sequential Nondeterministic Programs. In *REX Workshop for Refinement of Distibuted Systems*. Lecture Notes in Computer Science, No 430, pp. 42–66, 1990. Also published in reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 92, 1989.

Abstract. For abstract, see [Back and vWright 89c]

[Ehrenberg et al 90] Ehrenberg, D., Eklund, P., Fedrizzi, M., Ventre, A.G.S., Dynamic Consensus in GDSS. In *AIRO '90, Models and Methods for Decision Support*, pp. 245–250. Operational Research Society of Italy, 1990.

[Eklund and Gähler 90b] Eklund, P., Gähler, W., Set Functors and General Spaces. To appear in *Communications of the IFSA Mathematics Chapter*. Also in Proc. 11th International Seminar on Fuzzy Set Theory, Applications of Category Theory to Fuzzy Subsets. Linz, 1989.

[Eklund and Kaufmann 90] Eklund, P., Kaufmann, M., Hierarchical Wiring in Multigrids. *CONPAR90 - VAPP IV, Joint International Conference on Vector and Parallel Processing*, pp. 423–434. Zürich, 1990.

[Eriksson and Törn 90b] Eriksson, I., Törn, A.A., SOLE – Research on Information System Quality. *In: Conference Papers of the Software Quality Workshop*, pp. 249–258. Dundee Institute of Technology, UK 1990.

Abstract. For abstract, see [Eriksson and Törn 90a]

[Eriksson and Törn 90c] Eriksson, I., Törn, A.A., The SOLE project: Efficient IS Management. *NordDATA-90*, Part 1, pp. 211–217. Gothenburg 1990.

Abstract. A research project SOLE, aiming at supporting efficient IS management, is presented. The means to achieve the aim is to focus on IS quality questions and to find relations between IS quality and IS costs. Models describing software library accumulation at different quality levels and an IS quality model are developed. A computer implementation of the models facilitate obtaining data describing the quality of existing software units.

[Shen 90b] Shen, H., A Fast Parallel Algorithm for Integer Sorting. In Evans, D.J., Joubert, G.R., Peters, F.J. (eds), *Advances in Parallel Computing*, Vol II, pp. 331 – 336. North-Holland 1990. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 70, 1988.

Abstract. This paper presents a deterministic and optimal parallel algorithm to solve the integer sorting problem. The first version of the algorithm is to sort n distinct integers and the second version is to sort n random integers. The algorithm runs in $O(\log n)$ time on $n/\log n$ processors in the EREW PRAM model.

[Sørgaard 90] Sørgaard, P., The case for destruction. In Hellman, R., Ruohonen, M., Sørgaard, P. (eds), *Preceedings of the 13th IRIS* Part II , pp. 409–421. Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 108, 1990.

[Stomp 90] Stomp, F., A derivation of a broadcasting algorithm using sequently phased reasoning. In Logrippo, L., Probert, R.L., Ural, H., *Protocol, Specification, Testing and Verification X*, pp. 19–32. Canada, 1990.

Technical Reports

[Aspnäs and Kerola 90] Aspnäs, M., Kerola, T., *Modeling the Hathi-2 Monitoring System.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 110, 1990.

Abstract. The performance of the monitoring system in a transputer-based multiprocessor is investigated using queuing network models. The goal is to investigate the efficiency of the implemented monitoring system and to provide tools which can be used for evaluating different implementation strategies before they are implemented in full scale.

[Aspnäs and Långbacka 90] Aspnäs, M., Långbacka, T., *A Monitoring System for a Transputer-Based Multiprocessor.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 118, 1990.

Abstract. This paper describes a distributed performance monitoring system for a transputer-based multiprocessor, Hathi-2. The monitoring system allows the programmer to measure how well a parallel program utilizes the hardware resources of the multiprocessor system, i.e., the processor CPU's and communication links, during program execution. It can be used for identifying performance bottlenecks in parallel programs. We present the different components of the system, together with a case study where we monitor an existing application program.

[Back and vWright 90d] Back, R.J.R., von Wright, J., *Statement Inversion and Strongest Precondition.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 101, 1990.

Abstract. A notion of inverse commands is defined for a language with a weakest precondition semantics that permits both demonic and angelic nondeterminism as well as miracles and nontermination. Every conjunctive and terminating command is invertible, the inverse being non-miraculous and disjunctive. A simulation relation between commands is described using inverse commands. A generalized form of inverse is defined for arbitrary conjunctive commands. The generalized inverses are shown to be closely related to strongest postconditions.

[Back and vWright 90e] Back, R.J.R., von Wright, J., *Command Lattices, Variable Environments and Data Refinement.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 102, 1990.

Abstract. We introduce a lattice-based specification language with commands that introduce and remove variables, in addition to miracles and angelic nondeterminism. A notion of inverse commands is defined. Data refinement is possible within the same framework as ordinary refinement, using encoding commands and their inverses as bridges between different state spaces. This yields a calculational theory of data refinement.

[Boglaev et al 90] Boglaev, Yu.P., Karpenko, A.P., Aspnäs, M., *An implementation of the ODE block algorithms for transputer system.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 113, 1990.

Abstract. In this paper we consider implementation of the block representations of operators in parallel computations in linear second order ODE. We suggest an approach for constructing

selfadapted algorithms for MIMD architecture systems. It reduces to adaptor code which is mapping the parallel ODE solver on the concrete architecture. We describe the implementation of this approach on the transputer-based HATHI-2 system.

[Sere 90a] Sere, K., *Laws of Action Systems Programming.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 100, 1990.

Abstract. We present a collection of transformation rules which take a program specification into a parallel algorithm suitable for execution in some multiprocessor environment. The action system formalism to model parallel activity is used.

[Shen 90c] Shen, H., *Occam Implementation of Process-to-processor on the Hathi-2 Transputer System.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 114, 1990. To be presented at Transputing'91, California, USA, April 1991.

Abstract. This paper presents a polynomial-time Occam program for automatically mapping parallel programs onto multiprocessor systems that is a fundamental problem in parallel processing but NP-hard in general. Based on the heuristic strategy of *self-adjusting mapping*, our program consists of grouping, placement, routing and self-adjusting procedures, where grouping groups the user-defined processes in a parallel program into target tasks with a possible load-balancing, placement places the target tasks onto the processors in a transputer network, routing produces edge-disjoint physical communication paths for logical communication requirements among the placed tasks in the network, and self-adjusting adjusts first the placement scheme when the routing fails and then the grouping scheme when the necessary adjusting for placement is unable to make the routing succeed. These four procedures of the program work co-operatively until a successful process-to-processor mapping has been finally achieved after a series of progressive self-adjustments. For the problem of mapping n processes in an arbitrary task graph onto m processors in a transputer network configured in a torus, the program runs in time $O(\max\{n^2, m^5\})$ on one processor under full adjusting. To the same problem, by degrading the adjusting heuristic into semi-adjusting the program can run in time $O(\max\{n^2, m^4\})$. By eliminating the adjusting heuristic, our program can also run in time $O(\max\{n^2, m^2\})$ for transputer networks providing message routing and multiplexing. The implementation of the program shows that for both regular and irregular task graphs the program works very well and produces satisfactory results.

[Törn 90d] Törn, A.A., *Estimating software quality characteristics from maintenance costs.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 109, 1990.

Abstract. The maintenance cost of software is dependent on the quality of software. On the other hand the maintenance cost per size unit may be used as a measure of the quality of software. We show how a theoretical model SOLE based on general laws of software evolution can be used to calculate quality characteristics of software from realized maintenance cost data. Based on these characteristics it is shown how to pinpoint problem software and deficiences of software production process procedures.

[Törn 90e] Törn, A.A., *Topographical global optimization.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 119, 1990.

Abstract. A method for global optimization of a function $f(x)$, $x \in A \subset R^n$ based on the use of topographical information of f in A is presented. The topographical information is contained in direced graph constructed by connecting uniformly distributed points in A on a k nearest neighbour bases. Some experiments with a sequential algorithm for finding the global minima for some often used test function are undertaken. The results are then compared to those obtained with parallel versions of the algorithm run on a transputer based multiprocessor system.

[vWright 90a] von Wright, J., *Program Inversion in the Refinement Calculus.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 103, 1990. To appear in Information Processing Letters.

Abstract. We present a calculational method for inverting programs by using angelic nondeterminism in the calculations. A simple example illustrates the method.

[vWright 90b] von Wright, J., *A Mechanization of a command Lattice with Recursion.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 104, 1990.

Abstract. A command language with a weakest precondition semantics is formalized using the proof assistant HOL. The language is lattice-based with a strong sense of duality. It permits demonic and angelic nondeterminism as well as recursion. The notion of refinement is defined for the language and the formalization is shown to support interactive program development within the framework of the refinement calculus.

Theses for Ph.D.

[Eriksson 90b] Eriksson, I., *Simulation for user training.* Åbo Akademi, Dept. of Computer Science. Ph.D. thesis 1990. A summary published in Acta Academiae Aboensis, Ser. B, Mathematica et Physica, Vol. 50, No 3. Åbo Akademi 1990.

[Sere 90b] Sere, K., *Stepwise Derivation of Parallel Algorithms.* Åbo Akademi, Dept. of Computer Science. Ph.D. thesis 1990.

[vWright 90c] von Wright, J., *A Lattice-theoretic Basis for Program Refinement.* Åbo Akademi, Dept. of Computer Science. Ph.D. thesis 1990.

Theses for Ph.Lic.

[Shen 90d] Shen H., *Contributions to Mapping, Routing and Sorting in Parallel Processing.* Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1990.

Working papers

[Back and Kurki-Suonio 90] Back, R.J.R., Kurki-Suonio, R., *Superposition and Fairness in Reactive System Refinement.* To appear in Proc. of 5th Jerusalem Conference on Information Technology (JCIT), 1991.

[Carlsson et al 90] Carlsson, C., Ehrenberg, D., Eklund, P., Fedrizzi, M., Lindholm, P., Merkuryeva, G., Ventre, A.G.S., *A Case Study for Reaching Consensus in Group Decisions.* To Appear in IFORS - SPC1. Brussels 1991.

[Eklund and Forsström 90] Eklund, P., Forsström, J., *Diagnosis of Nephropathia Epidemica by Adaptation through Lukasiewicz Inference.* To appear in Cercone, N., Gardin, F., Valle, G. (eds), COMPUTATIONAL INTELLIGENCE, III – The International Conference on Computational Intelligence 90. Elsevier Science Publishers B.V., North-Holland, 1991.

[Eklund and Gähler 90c] Eklund, P., Gähler, W., *Fuzzy Filter Functors and Convergence.* To appear in Höhle, U., Klement, E.P., Rodabaugh, S.E. (eds.), Applications of category theory to fuzzy subsets. Theory and Decision Library B. Kluwer, 1990.

[Eklund et al 90a] Eklund, P., Lindholm, P., Merkuryeva, G., *Group Decisions and Sociological Intelligence.* To appear in Proc. KIB90. Betriebliche Informations- und Kommunikationssysteme, Erich Schmidt Verlag, Berlin/Bielefeld/München, 1990.

[**Eklund et al 90b**] Eklund, P., J. Forsström, J., Palm, P., Virtanen, H., Waxlax, J., *GEDE-MEDES - A Generic System for Developing Medical Decision Support.* To appear in ECAI90, Workshop on Industrial Diagnostic Knowledge Based Systems. Stockholm, 1990.

[**Eriksson 90c**] Eriksson, I., *Reflections on the concept of knowledge.* To appear in Proc. of the seminar on the Fragmentatin of Knowledge, June 1989, Capri Italy.

[**Eriksson and Finnäs 90**] Eriksson, I., Finnäs, A., *Creating a Visual Simulation Model of an Inventory System.* In van den Besselaar, P. et al (eds), Information System, Work and Organization Design. In print.

Abstract. Organizations are heavily dependent on their information systems but there are problems in using them. One reason might be that the interaction between work and information systems is covert. Description techniques do not support users in interpreting the information systems in their organizational context.

We propose that information systems should be designed in such a way that they make the actors visible and improve mastering of the work situation as a whole. As an example we implement a context-sensitive and visualized simulation model to be used as a Help-facility, based on a simplified version of a real inventory system. Simulation is controlled by the system and one transaction at a time is treated. Transactions are described on different hierarchical levels: transaction flows between departments and units, manipulation on the detailed level of each section, and step-by-step presentation of computerized functions. The model is also described in the time dimension. The user can put what-if questions (or rather make experiments) about the future and how-did-we-arrive-at-this-position questions about the past. The backward trace is accomplished by two history files concerning the data and the actions taken. The simulation model is implemented on Macintosh using HyperCard.

This kind of simulation is primarily aimed for use in situations where new material is to be learned. This may mean that a new system is implemented or new workers are employed. It might also support every-day work situations; to assist in fault-finding and error recovery, and to master unusual situations. Further development and generalization of the model is discussed. Broad understanding offered by our Help-facility is beneficial on both the individual and organizational level and can be expected to improve both the quality and efficiency of the work situation

[**Eriksson and Törn 90d**] Eriksson, I., Törn, A.A., *A model for IS quality.* To appear in Software Engineering Journal. Revised version of [Eriksson and Törn 89].

Abstract. Different concepts of software quality are reviewed and discussed. Based on this, a hierarchical model of information system (IS) quality concepts developed in the research project SOLE is presented. This model aims at a division of quality concepts consistent with the different decision makers and decisions made during the software life-cycle. The main division is: *IS Cost Effectiveness, IS Use Quality* and *IS Work Quality.* The last two are further divided into *Requirement Quality, Interface Quality,* and *Efficient IS Management, Evolution Quality* and *Operation Quality* respectively.

[**Eriksson et al 90**] Eriksson, I., Finnäs, A., Reijonen, P., Visual Simulation as an Aid for Understanding Computer Functions. To appear in *Interacting with Computer*, April 1991.

Abstract. A help-system with a visual simulation model is presented and evaluated. The simulation model allows the user to follow the transactions in a storage department on different hierarchical levels of detail; transaction flows between departments and units, detailed manipulation on the section level, and a step-by-step progress of computer functions. The time dimentsion is taken into consideration and the transactions can be followed forwards but also

be traced backwards. The backward trace is accomplished by two history files concerning data and actions taken. The model is designed to be used in two modes: simulation conrolled by the system and simulation controlled by the user, conrolled and interactive simulation, respectively. A prototype version of controlled simulation is presented as an example. Performed experiments show that the simulation model is useful as an aid in learning to understand computer functions.

[Sere 90c] Sere, K., *Communications in Processor Farms.* Dept. of Computer Science, Åbo Akademi, 1990.

[Törn 90f] Törn, A.A., *Simulation modelling.* Book, 80 pp.

Abstract. Textbooks on simulation are of course written with the intent to cover all important aspects of simulation. Like the situation with textbooks on programming details of programming are rather well covered. Another aspect that is rather well covered is the statistical problems. Methodology is normally puerly treated. An important part of the methodology is modelling, i e., the work in constructing the logical model used as a base for programming. The evolving model is normally represented by using some graphical tool. Such graphical tools are normally tied to a specific simulation language and they are all *ad hock* constructions without theoretical properties consisting of a host of different graphical elements. This book covers simulation modelling. Some graphical tools used are reviewed and discussed. A design tool with theoretical properties, *Simulation Nets*, based on the well known Petri Nets is presented and its use is illustrated and discussed. A computer tool *SimNet* capable of performing the implied simulation given a simulation net is demonstrated. With such a tool available simulation experiments can be made directly with the logical simulation model without further computer programming.

Appendix A

Staff

Permanent positions and their deputies

Professor: Ralph-Johan Back

Professor: Aimo Törn (1.7.-)
Deputy: Jan Komorowski (1.1.-30.6)

Associate professor (Administrative data-processing): Aimo Törn, vacant (1.7.-)
Deputy: Patrik Eklund (1.7.-31.12)

Lecturer: Ragnar Wikman
Deputy: Ulla Solin (1.8.-31.12)

Lecturer: Patrik Eklund
Deputy: Paul Lindholm (1.7.-31.12)

Assistant professor [1]: Pål Sørgaard
Deputies: Inger Eriksson (1.1.-30.3), Ulla Solin (1.8.-31.12)

Assistant professor [1] (Computer science and mathematical statistics): Vacant
Deputies: Ulla Solin (1.1.-31.7), Frank Stomp (1.8.-31.12)

Assistant professor (Systems programming) (from 1.8.1989)[1] [2]: Kaisa Sere

Assistant: Ulla Solin
Deputies: Marko Suojanen (1.1.-31.3), John Aspnäs (1.4.-31.7), Patrik Gustafsson (1.8.-31.12)

Assistant: Vacant (1.7.-31.7), Hong Shen (1.8.-)
Deputies: Dan-Johan Still (1.1.-28.2), Frank Stomp (1.3-31.7)

Assistant (Administrative data-processing): Inger Eriksson
Deputies: Jukkapekka Hekanaho (1.1.-31.7.), Dan-Johan Still (1.8.-31.12)

Instructor: Paul Lindholm
Deputy: Sami Viitanen (1.1.-31.12)

Instructor: Vacant
Deputy: Annamari Soini (1.1.-31.12)

Laboratory Manager (from 1.1.1989): Mats Aspnäs
Deputy: John Aspnäs (1.1.-31.3), Jan-Erik Flück (1.4.-31.7.), Tony Riissanen (1.8.-31.12)

Department secretary: Christel Engblom

[1] "Överassistent" in Swedish
[2] Faculty of Chemical Engineering, position placed in our department

Externally funded project staff

Thorbjörn Andersson (Ph.D. programme)

Mats Aspnäs (administrative leader of Millipede)

Inger Eriksson (SOLE, Foundation for Economic Education, Swedish School of Economics and Business Administration in Helsinki)

Jukkapekka Hekanaho (Centipede, Ph.D. Programme)

Petri Luostarinen (Ph.D. Programme)

Hannu Salmela (SOLE, Academy of Finland)

Kaisa Sere (administrative leader of Centipede)

Hong Shen (Millipede)

Elena Trishina (Millipede)

Gundel Westerholm (project secretary, Millipede, Centipede, Finsoft III co-ordination project)

Joakim von Wright (Swedish School of Economics and Business Administration in Vaasa, Ph.D. programme)

Undergraduates employed in projects

Sten Agerholm (Centipede)

Peter Ahlskog (SOLE)

Kerstin Aller (SOLE)

Åsa Backlund (SOLE)

Peter Dahl (Centipede)

Thomas Långbacka (Millipede)

Mikael Norrbo (Millipede)

Kaj Nylund (SOLE)

Håkan Sarén (SOLE)

Dan-Johan Still (Centipede)

Marko Suojanen (Centipede)

Roger Wallin (SOLE)

Joakim Waxlax (GeDeMeDeS)

Patrick Waxlax (Centipede)

Harry Virtanen (GeDeMeDeS)

Series A

1990

105. *Patrik Eklund, Michael Kaufmann*, Hierarchical Wiring in Multigrids.
106. *Patrik Eklund, Werner Gähler*, Fuzzy Filter Functors and Convergence.
107. *Riitta Hellman, Mikko Ruohonen, Pål Sørgaard*, Precedings of the 13th IRIS - Part I.
108. *Riitta Hellman, Mikko Ruohonen, Pål Sørgaard*, Precedings of the 13th IRIS - Part II.
109. *Aimo A. Törn*, Estimating Software Quality Characteristics from Maintenance Costs.
110. *Mats Aspnäs, Teemu Kerola*, Modeling the Hathi-2 Monitoring System.
111. *Patrik Eklund, Jari Forsström*, Diagnosis of Nephropathia Epidemica by Adaptation through Lukasiewicz Inference.
112. *Göran Högnäs*, Sequences of Random Transformations.
113. *Yu. P. Boglaev, A.P. Karpenko, M. Aspnäs*, An implementation of the ODE block algorithms for transputer system.
114. *Hong Shen*, Occam Implementation of Process-to-processor Mapping on the Hathi-2 Transputer System.
115. *Paavo Salminen*, On a First Passage Problem of Neveu for Critical and Subcritical Branching Brownian Motions.
116. *Paavo Salminen*, A Ratio Limit Theorem for Erased Branching Brownian Motion.
117. *Paavo Salminen*, Cutting Markovian Trees.
118. *Mats Aspnäs, Thomas Långbacka*, A Monitoring System for a Transputer-Based Multiprocessor.
119. *Aimo Törn*, Topographical Global Optimization.
120. *Thorbjörn Andersson*, A Survey on Software Quality Metrics.

1991

121. *R.J.R. Back, M. Aspnäs, J. Granslund, J. Hattula, R. Julin, A. Lampinen, T. Lönnroth, P. Waxlax*, Analysis of Three-Dimensional Nuclear Data on a Transputer-Based Multiprocessor System.
122. *Kaisa Sere*, Communication in Processor Farms: A Case Study in Reactive Systems Refinement.

Series B

1990

9. *Pål Sørgaard, Gundel Westerholm (eds)*, Åbo Akademi, Department of Computer Science. List of Publications 1984–1989 (incl. Abstracts).

1991

10. *R.J.R. Back, A. Törn (eds)*, Åbo Akademi, Department of Computer Science. Annual Report 1989, 1990.