Åbo Akademi
Department of Computer Science

5-year Report 1984-1989

Editors

Ralph-Johan Back and Aimo Törn

Åbo Akademi
Reports on Computer Science & Mathematics, Ser. B, No 7.
October 1989

ISBN 451-649-647-4
ISSN 0358-3392

Åbo Akademis kopieringscentral
Åbo 1989

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

Overview

1.1 General

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 first positions in Computer Science at Åbo Akademi were a part time associate professorship in administrative data-processing at the Business School at Åbo Akademi (1970 – 75, full time 1975 –) an assistant (1971), an assistant professor (research) (1972) at the Faculty of Mathematics and Natural Sciences. The first full professorship in computer science was obtained in 1980. At the same time all the staff positions in computer science were placed with the Faculty of Mathematics and Natural Sciences. The department remained very small during the first half of the 1980's, with only six positions in 1984 (professor, associate professor, assistant professor (research), two assistants and an instructor) and no research personnel.

In the latter half of the 1980's the department has grown considerably. The university initiated a development plan for computer science which has brought a number of new positions to the department, so that there is now altogether 15 permanent positions. The evolution of the department is shown in Figure 1.2.

A large research project on parallel programming technology (the Hathi-project) funded by the Technology Development Centre (TEKES) and a research project on formal methods in program construction funded by the Academy of Finland initiated together a substantial increase in research activities in 1986. At the same time the department became quite heavily involved in the Ministry of Education's doctoral programmes in computer science, filling one position out of three co-ordinating professors (computing science). The number of persons employed in research projects at the moment is approximately equal to the number of permanent positions, making the total number of people working in the department now around 30. The research personnel is placed either in one of the three current larger research projects in the department (Millipede, Centipede and Sole) or employed directly by the doctoral education programme. The department is now also responsible for a large technology programme in par-

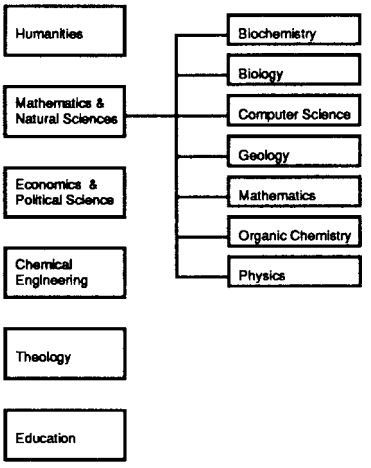

Figure 1.1: Position of the department within the university.

allel computing technology, FINSOFT III, financed by TEKES and consisting of 12 different research projects located all over Finland.

The official quota for the annual intake of students to the department is 25. The department has tried to increase the number of students and the actual figure now is around 30 new students each year. The number of M. Sc. theses accepted was very small until 1987, on average 4 – 5 a year, and has been the weak spot of the department for a long time. The good employment opportunities made the drop-out rate very high for computer science students. The situation has recently changed quite drastically, with 19 M.Sc. theses accepted in 1988 and 9 M.Sc. theses accepted in the first half of 1989. Some of the increase in 1988 can be explained by administrative changes that required old students to finish their degrees that year. However, the more important reason is the way in which the supervision of M.Sc. theses has improved. The thesis work has to be better planned and the progress of the thesis is followed up in seminars. Most theses are now written within some research project, where the supervision is continuous and the time limits are more strictly enforced.

The department moved into new and modern facilites in February 1988, located in the new technology centre DataCity in Turku. The floorspace was increased considerably, and the present environment can be considered fully adequate. Also the available computer equipment park has been considerably expanded, partly by research grants and partly by internal funding of the university.

1970 Associate professor (Administrative data processing)
1971 Assistant
1972 Assistant professor [1]
1980 Professor
1981 Assistant (Administrative data-processing)
1983 Instructor
1985 Assistant professor [1] (Computer science and mathematical statistics)
Instructor
1986 Assistant
1987 Lecturer
1988 Professor
Department secretary
1989 Laboratory manager
Lecturer
Assistant professor [1] [2] (Systems programming)

Figure 1.2: Growth of the department 1970 – 1989

1.2 Personnel

At present, the department has 15 permanent positions, 13 of these belonging to the teaching faculty. The permanent positions and the persons presently appointed to these are as follows:

Professor: Ralph-Johan Back, 1983 –
Professor: Vacant
Associate professor (Administrative data-processing): Aimo Törn, 1972 –
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]: Pål Sørgaard, 1989 –
Assistant (Administrative data-processing) : Inger Eriksson, 1988 –
Assistant : Ulla Solin, 1983 –
Assistant : Vacant
Instructor: Paul Lindholm, 1985 –
Instructor: Vacant
Laboratory manager: Mats Aspnäs, 1989 –
Department secretary: Christel Engblom, 1988 –

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

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

The actual staff at the department during the academic year 1988–89 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.
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 maintenance, 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.
5. *Artificial intelligence*: knowledge based systems, medical expert systems, image analysis.

The main externally funded research projects in the department during the period 1984 – 1991 are:

A. *Program Modularization Methods* (Ralph-Johan Back, 1982 – 1985), funded by the Academy of Finland.

B. *Construction of Distributed Systems* (Ralph-Johan Back, 1986 – 1988), funded by the Academy of Finland.

C. *Hathi-project* (Ralph-Johan Back, 1986–88), funded by TEKES.

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

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

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

G. *SOLE-project* (A. Törn, 1988-1990), funded by the Academy of Finland and the Foundation for Economic Education.

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.

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 professional organizational activities. Chapter 10 describes the individual research interests, main publications and publications in 1988 – 89 of the staff. Chapter 11 describes scientific contacts in 1988 – 89, both foreign visitors at the departments and research visits and conference participation by staff and Chapter 12 contains accepted theses. Chapter 13 lists the publications of department personnel in the period 1984 – 89 on an annual basis.

1.4 Teaching

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.

Our curriculum was thoroughly restructured in 1986. The aim was twofold; to make the studies more streamlined and thus make it possible for the students to graduate within four years rather than five and to tie advanced studies more closely to the research activities of the department.

The first aim has been achieved by restructuring the course schedule plan as well as some of the courses themselves, while the second aim has been achieved by completely reforming advanced studies.

Whereas earlier the students could choose to specialize either in administrative data processing or in systems programming, and this specialization affected the courses that had to be taken also at the intermediate level (subject studies), the structure is now more flexible. The courses up to and including intermediate level are the same for all students, thus providing everyone with a certain basic platform of knowledge needed for further studies. Specialization takes place on the advanced level, where students can choose to specialize in one of the research areas which are actively pursued at the institution. This brings the students into the different research areas at an early stage and also guarantees that the advanced level courses are given by competent and highly motivated personnel.

There are five specialization areas currently, corresponding to the areas of research listed above: programming methodology, multiprocessor systems, information systems, decision support and artificial intelligence. The selection of specialization areas is flexible and follows the research interests of the department staff, but so that a certain degree of continuity is retained.

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.

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

1.5 Funding

The traditional source of funding for basic research is the Academy of Finland. It has funded three major projects in the department, Program Modularization Methods, Construction of Distributed Systems and Sole, as well as the Knowledge and Work project at the University of Turku.

The Technology Development Centre (TEKES) has also played a very important role in funding reserach in the department. TEKES has traditionally provided funding mainly for applied research, done in co-operation with industrial partners. It has, however, also funded the more basic research-oriented Hathi-project, as well as the Millipede, Centipede and FINSOFT III Co-ordination projects at the department.

The third source for funding has been the three national programmes for doctoral education in computer science, initiated by the Ministry of Education 1985 and still continuing. A number of people from the department have been employed in this programme for shorter or longer periods (Inger Eriksson, Kaisa Sere, Ulla Solin, Joakim von Wright) and Ralph-Johan Back has been employed as the co-ordinating professor for the programme in Computing Science (January 1986 – July 1988). The programme has also funded some equipment.

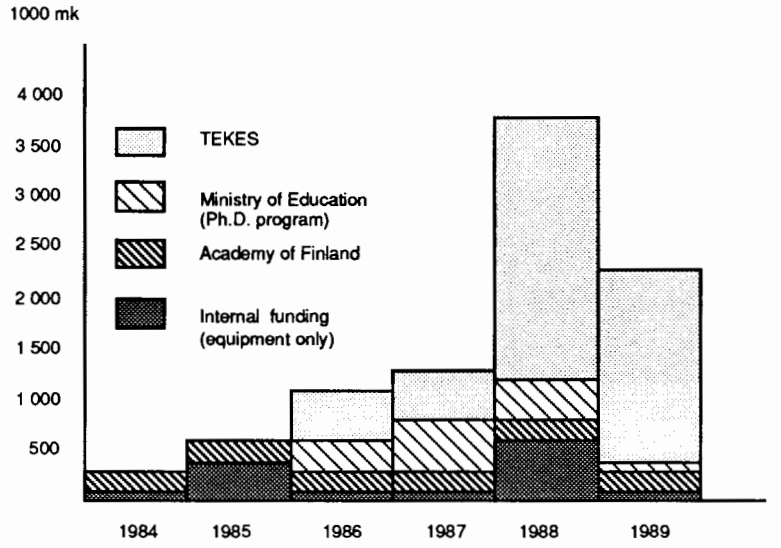

Figure 1.3: The structure of research funding

The internal funding of Åbo Akademi has also been important, both for employing temporary personel for the research projects and for aquiring equipment. The structure of research funding 1984–1989 is shown graphically in Figure 1.3 (for internal funding, only equipment is shown). Table 1.1 shows the main research projects and their funding annually 1984 – 1989.

1.6 Facilities

The equipment for research and teaching has changed considerably during the last three years. Traditionally IBM PC:s and centralized VAX/VMS computers have been used. The university computing center now has a Vax 8800 and a Sun-4. These are mostly used for teaching, while research and advanced courses use Sun/UNIX-workstations and Apple Macintosh computers.

The department has a large multiprocessor system, Hathi-2, that was designed and built in the Hathi-project. It now serves as the central multiprocessor in the FINSOFT III research programme. The department's Sun 3/160 is used as a front end to Hathi-2. Hathi-2 is connected to the university Ethernet, and through that accessible also to research projects in other parts of Finland.

Workstations and microcomputers are used both for textprocessing and for program development. The most common microcomputer is Apple Macintosh. These are connected to a LocalTalk network and to terminal servers by the aid of Ethernet. The users can use the facilities on the LocalTalk (laser writers, hard disks etc.) as well as terminal emulation to log into central computers.

The students can choose between different computers for their projects. There is a laboratory with 5 Macintosh II, 2 Macintosh SE and 2 Sun 3/50. Another laboratory (also serving as a microcomputer class) has 12 Nokia Mikro Mikko 3 (IBM PC/AT compatible). There is also a laboratory with 16 Falco terminals connected to the university local area network.

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

	1984	1985	1986	1987	1988	1989
A. Modularization (Academy)	167	169				
B. Construction (Academy)			243	243	243	
C. Hathi (TEKES)			467	467	467	
D. Millipede (TEKES)					1 100	950
E. Centipede (TEKES)					800	750
F. Finsoft-III (TEKES)					250	250
G. Sole (Academy)						300
Total (1000 mk)	167	167	710	710	2 860	2 250

Table 1.1: External funding of main research projects.

terminals, microcomputers, workstations and mainframes at all three university institutions in Turku. A 64Kbit line connects us to the Finnish University Network FUNET. Thus logins from any other FUNET site are possible and have proved to be very useful, especially when outside users want to use Hathi-2. 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 1.4.

1.7 Problems and difficulties

The rapid expansion of the department has also brought a number of problems with it. The main problem has been in obtaining competent personnel to fill the permanent positions. This has been a problem for most departments of computer science in Finland in the 1980's, but for Åbo Akademi it is compounded by the fact that the main recruitment area is the Swedish-speaking population of Finland. Even if the positions are not restricted to these, mastering Swedish well is a formal requirement for the position, making the positions less attractive for Finnish-speaking people. To solve this problem, the department has emphasized postgraduate education and research at the cost of teaching for the last three years in order to educate a new generation of researchers competent to fill the open positions. This has also been fruitful, and presently most of the positions are filled or could be filled with formally competent people. The department has also been quite active in recruiting people from abroad and especially from the other Nordic countries for which the language barrier for working at Åbo Akademi is low. Presently four researchers from abroad are employed in the department, coming from Sweden, Norway, the Netherlands and the Peoples' Republic of China.

Due to the peculiarities of the Finnish academic system, the fact that there are competent people appointed to a position does not mean that they in fact also work in these positions. Presently most people are on leave of absence from their ordinary positions, either in order to fill a higher position in the department or in order to do full-time research. As a consequence, most positions are temporarily filled with people that are not formally qualified for the positions. The stiffness in the employement structure at the Finnish universities makes it difficult

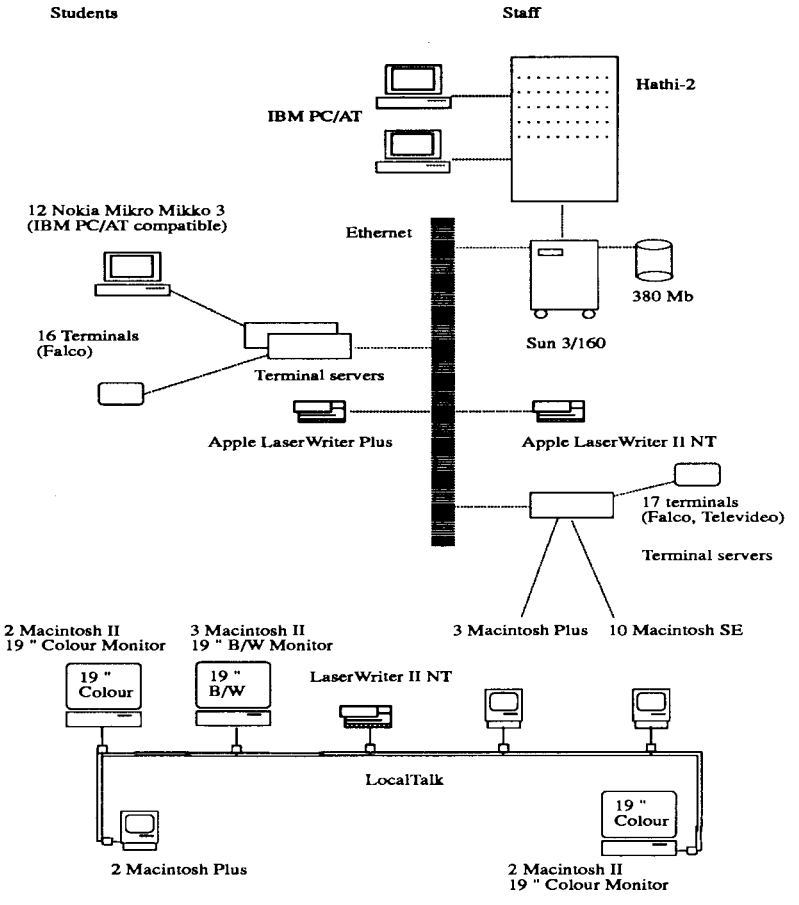

Figure 1.4: Facilities in the department

to share the load of teaching and research in a more equal way between the people working in the department: the options are in practice either to do full-time teaching and administration, or to do full-time research. The situation has a negative effect on teaching, where many courses are given by inexperienced teachers, even when there are actually qualified people in the research projects available. Long-term planning of courses is also made more difficult by this arrangement. However, it is expected that the situation will improve considerably within the next year or two, when the number of qualified people that are interested in working and doing research in the department has increased.

Another difficult problem, which the department shares with the other computer science departments in this country, is to get gifted students to continue for postgraduate degrees. The salaries and career opportunities in industry and commerce are very tempting, and we have lost quite a few good potential researchers in this way. The problem is to make an academic career more attractive: considerably higher salaries would be needed and more flexibility in the balance between research, teaching and administration is also necessary.

Chapter 2

Teaching

2.1 Undergraduate studies

A major revision of our curriculum was carried out in 1986, so that the curriculum now better reflects the research areas of the department. 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 1984 – 1989 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** (15 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. Introduction to system development

B. **Subject studies** (105 credits)

1. Computer science (42 credits)

1. Programming techniques II
2. Data structures
3. Programming methodology
4. Computer architecture
5. Operating systems
6. Data bases
7. System development methods
8. 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 (8 credits)
2. Statistics (8 credits)
3. Business economics (8 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. A sample of advanced courses that have been given is as follows:

Programming methodology
Stepwise refinement methods
Algorithm development
Theory of distributed systems
Program specification methods
Logic programming

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

Information Systems
Software engineering
EDP, the individual and society
System development, tools and trends today
System development theory and practice

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

Decision Support
Simulation techniques
Petri nets
Strategic planning

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

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

There have been three major externally funded research projects on programming methodology at the department. The first one, Program Modularization Methods, was started in January 1982 at the University of Helsinki and continued in Åbo Akademi when the project leader Ralph-Johan Back moved there. The project ended in December 1985. It was funded by the Academy of Finland and carried out in co-operation with the University of Helsinki, the University of Jyväskylä and of Tampere and the Tampere University of Technology. Total funding was 676 000 mk. The topic was to study the theory and methods for constructing correct sequential and distributed programs. [Back 86b] is the final report of this project.

A new project, Construction of Distributed Systems , was started in January 1986, again funded by the Academy of Finland. This project was in practice merged with the Hathi-project (multiprocessor technology) funded by TEKES which started half a year later, and focused in particular on the formal construction of parallel multiprocessor programs. The project ended in December 1988. Total funding for this project was 730 000 mk. The main co-operation was with the Tampere University of Technology.

Presently formal methods are being studied in the Centipede project, also 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. The project continues the work on formal methods for constructing parallel programs from the two earlier projects and also tries to construct a graphical environment for formal derivations of programs. Total funding is approximately 2 200 000 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, pursued as a sub-project to the SOLE-project.

Since moving to Åbo Akademi last year, Jan Komorowski has established a small research team to continue his work on program transformations and programming environments. This research complements the profile of the Centipede group. The two main themes in this research are 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 now a subproject of the SOLE project, investigating the effects of quality on software by using formal program development methods. The research aims at developing a technique and a tool (PICA) for formal program development using flowcharts [Törn 88e]. The formal technique applied is that of proving theorems of the type *{pre-condition} statement {post-condition}*. The 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 whether 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 graphical editor on a Macintosh II by Benita Heinonen (M.Sc. thesis). The tool aids both in the design and in 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 effects on quality of using the technique are being investigated by Håkan Sarén (M.Sc. work). An experiment during a programming course in the spring semester 1989 with second-year students has been undertaken.

3.2 Simulation nets

A tool for designing simulation software for discrete event systems is being developed. The tool is an extension of the well known Petri-net formalism. The idea is to graphically describe the object system using Simulation Nets. These are exact enough to permit simulation to be carried out directly without further programming [Törn 85, 88a, 88b].

A first prototype capable of reading a simulation net design and carrying out the implied simulation has been written in Simula by Antti Raunio. A second prototype, facilitating the graphical design, has been written for the Machintosh computer by Tom Hägg (M Sc. thesis).

Work on the efficient choice of transitions to fire when executing simulation nets has been carried out by Aimo Törn, who is also the leader of the project [Törn 89].

Future plans include an implementation of a third prototype on the Machintosh II, using Design and the Petri Net simulators already written in Design.

3.3 Semantics of non-determinism, concurrency and modularity

Work in this area was done by Ralph-Johan Back and Heikki Mannila in 1982–1984 at the University of Helsinki [Back and Mannila 84, 85]. The emphasis was on studying the semantics of certain important language mechanisms that in one way or the other are problematic: non-determinism, concurrency, communication and modularity. The work was pure basic research, aiming at designing and studying different semantic models of these mechanisms. Semantic studies of this kind become important when designing new programming languages and when building proof systems and developing programming methods for these languages.

3.4 An interactive program verification environment

This project started in 1982 and has resulted in a computer-supported environment for program verification [Back and Hielata 84, 86]. The system is called I3V (standing for Interactive, Incremental and Iterative program Verification). It has been designed by Ralph-Johan Back and Pentti Hietala (University of Tampere), and has been implemented by the latter. A working prototype has been running since 1984. The extent to which it supports program

verification has been evaluated in several empirical studies by Pentti Hietala. The purpose of the verification environment is to help the programmer in managing the proof construction process, in particular in keeping track of all the tedious details involved in the proof. The system has later been ported to a Sun-3 workstation environment after being reprogrammed in Prolog. The intention is to extend the system to support the verification of parallel programs. Pentti Hietala got his Ph.Lic. degree in this project and is now working on his Ph.D on the same topic in an independent project.

3.5 Derivation of graph-marking algorithms

A very specific programming problem, marking all nodes reachable from a given root node in a graph, was chosen as a case study in constructing provably correct programs. The aim was to study the power and applicability of formal program construction methods in solving really difficult programming problems. Actually two different case studies were carried out, one by Ralph-Johan Back, Heikki Mannila (University of Helsinki) and Kari-Jouko Räihä (University of Tampere), using program transformation methods [Back, Mannila and Räihä: Derivation of efficient DAG-marking algorithms, *ACM Conference on Principles of Programming Languages, Austin 1983*] , and another by Ralph-Johan Back and Anssi Rantanen, using more traditional methods based on identifying the program invariants involved [Back and Rantanen 84]. The results of these case studies are interesting in themselves: a new algorithm for marking acyclic graphs was invented, which considerably improved on the asymptotic complexity of previously known algorithms. The experiment thus showed the advantage of using formal methods (in a slightly informal manner) as a tool also in algorithmic research. This project was carried out in 1983–84.

3.6 Specification of document manipulation systems

The use of formal specification methods as a fast prototyping tool has been studied by Ralph-Johan Back and Airi Salminen (University of Jyväskylä) in a project that was started in 1984 and still continues, as an independent project by Airi Salminen [Salminen and Back 85, 86]. The specification of document manipulation systems was singled out as the particular application area to be focused upon. A general framework within which systems of this kind can be formally specified was constructed and used to model different features of concrete document manipulation systems. Airi Salminen has continued this work to a Ph.D., based on a novel approach of using Prolog to model abstract data types. The work has resulted in a very general model for document data bases and also lead to an executable prototyping environment in which to describe this kind of applications and study their look, feel and functionality, before embarking on a larger scale implementation of such a system.

3.7 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 84a, 84b, 85, 87, 88a, 88b, 89].

The main emphasis in the theoretical study of action systems was 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. However, it turned out that action systems can be quite efficiently implemented on local area networks based on a shared broadcasting channel. Mats Aspnäs (M.Sc thesis), Eeva Hartikainen (University of Helsinki) and Patrik Eklund joined in studying this problem [Back and al 84, 85, Eklund 84c]. Eeva Hartikainen got her Ph.Lic. degree for her work on modelling broadcast implementations of action systems.

For point-to-point networks, it is more difficult to find efficient distributed implementations of action systems. However, as shown in joint work by Ralph-Johan Back and Kaisa Sere, efficient implementations 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, and can thus be considered acceptable. This work was done 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, Peter Dalh (M.Sc. work) and Kaisa Sere.

3.8 Calculus of program refinement

A central role in the Centipede project is played by the *Refinement Calculus*, proposed by Ralph-Johan Back in his Ph.D. thesis 1978 (see Back, main publications [1 – 5]). 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 at Åbo Akademi, in the United Kingdom (Oxford, Glasgow) and in the Netherlands (Groningen). 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 was further studied and developed in the "Construction of distributed systems" project by Ralph-Johan Back, and extended to handle difficult program features such as unbounded non-determinism, full recursion and procedures with parameters [Back 87a, 87b, 88a]. The original method for data refinement

was also extended to permit non-functional data refinements [Back 88b, 89a].

A lattice-theoretical basis for specification languages with predicate transformer semantics has been recently developed in co-operation with Joakim von Wright [Back and von Wright 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. Joachim von Wright is expected to get his Ph.Lic. degree for this project within this year.

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. 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 program derivation environment being developed in a separate sub-project described below.

3.9 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 in the Construction of Distributed Systems project, 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 88a, 88b, 89a, Sere 87b, Sere 88a, 88b, 88c] . 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.Lic. degree for this project, and is now working on her Ph.D. on the same topic.

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.

3.10 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 proof assistance like the Cambridge LCF and HOL sysytems.

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 is now being constructed by Jan Komorowski and Patrick Waxlax (M.Sc. work) in Prolog on the Mac II, while Jukka-Pekka Hekanaho (M.Sc. work) is studying an alternative implementation of the environment, based on the program transformation system Refine on Sun-4. Dan-Johan Still (M.Sc. work) is working on implementing the simulated parallel execution environment for action systems. Kaisa Sere is working on a collection of program refinement rules that would be included in the system (integrating these rules with another collection of rules designed at Oxford University/PRG) [Sere 89] .

3.11 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. In our current research we have aimed to extend standard partial deduction so as to alleviate some of these problems. To

this end we have defined 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. The main new result of our research is that extended partial deduction is a powerful synthesis, not only optimization, method for obtaining efficient programs from possibly inefficient specifications. In particular, extended partial deduction 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, we have 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.

In addition, our results are applicable to pure Prolog and to "systems" languages in the Prolog family (i.e., Parlog, Concurrent Prolog, GHC, ANDORA Prolog, etc), but a full treatment of partial deduction in the systems languages requires a deeper analysis of the commit operators.

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.

Plans for thefuture include an investigation of the relationship between partial deduction, machine learning and non-monotonic reasoning. One of the problems which also needs to be addressed is the inability of the logic programming language to express higher-order properties such as the first argument of a given predicate is always a constant, or that a given variable is of type integer, etc. Expressing such constraints would improve many partial deductions. The ongoing research on higher-order logic programming languages or inductive assertions may bring an answer to such problems.

3.12 A generic environment for program derivation

There is a proliferation of theorem provers and their implementations. However, many of them are rather difficult to use due to a poor user interface. In order to overcome this problem we are designing a generic environment for program derivation which is a form of theorem proving. The lessons learnt from logic programming suggest that theorem provers should be equipped with advanced programming environments. Therefore we believe that a theorem prover should be built on top of a good programming environment. In order to avoid a repetition of tasks for different proof systems a theorem-proving environment should be parameterizable for syntactic units, that is, for the syntax of the underlying (formal) language and for the inference rules of the proof system. The components which need to be added are a presentation paradigm and tools for correct handling of the syntactic units.

A graphical environment for program refinement has been preliminarily designed by Ralph-Johan Back. The basic problem addressed in that design is a two-dimensional representation of the derivation structure in the process of refinement.

The handling of syntactic units has been designed by Jan Komorowski and is based on his work on syntax-directed editor, whose specification was used as a large example for illustrating rapid prototyping methodology in logic programming. The environment is generic in that it allows to define any syntax and any set of transformation rules. It then allows for syntax-directed manipulation of programs and transformations. The implementation follows the editor specified in definite clauses in [Komorowski and Maluszynski 87] and is done by Patrick Waxlax.

Chapter 4

Research on Multiprocessor Systems

Research on multiprocessor systems started in the department with the Hathi-project in August 1986. The project was funded by TEKES and the Ministry of Education (Ph.D program) and ended in December 1988. It was carried out jointly with the Technical Research Centre of Finland/Computer Technology Laboratory in Oulu (VTT/TKO). Ralph-Johan Back was the leader of the project. TEKES funding was 1 400 000 mk, mostly for hardware and equipment, while the Ministry of Education funded some research personnel. In practice, the project was merged with the Construction of Distributed Systems project funded by the Academy of Finland, as these were simultaneous in time and complemented each other well. The purpose was to construct hardware and software for a multiprocessor system based on the Inmos transputer, and to study methods for programming parallel applications, monitoring and debugging parallel systems and study the efficiency of parallellism in practical applications. The project co-operated quite extensively with other research institutes on multiprocessor applications, especially the Technical Research Centre of Finland/ Laboratory for Information Processing in Helsinki (VTT/TIK), the Faculty of Chemical Engineering at Åbo Akademi and the Department of Physics at the University of Jyväskylä.

The Hathi-project has been followed up by a whole research programme on parallel computation, now also including 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 a continuation of the Hathi-project, with a more focused aim, and 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.

The different research directions in the above research projects are described in more detail below.

4.1 Design and construction of a massively parallel computer

The design and construction of the Hathi-2 multiprocessor system was a central part of the Hathi project. The system was specified jointly by the Technical Research Centre of Finland (VTT/TKO) in Oulu and Åbo Akademi, while the detailed design and construction was done by the former, by a design team including Kari Leppälä, Kari Pehkonen, Kyösti Rautiala and Tapani Äijänen.

Hathi-2 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 and 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 is being developed further within the Millipede-project. The amount of memory is increased to 1.25 Mb per processor, giving the system a total of 125 Mb of central memory. A number of peripheral units, such as disk storage units and graphics controllers will also be connected to the system. John Aspnäs, Mats Aspnäs and Tor-Erik Malen are the principal people 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 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 the user's 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. The project is co-operating with Teemu Kerola (University of Helsinki) on the performance evaluation aspects of the monitoring subsystem.

The monitoring of resource utilization will be extended in the Millipede project to provide a general debugging environment for multiprocessor systems. The monitoring hardware and software will be integrated with the tools developed in the animation project. This will permit the users to describe the execution of their parallel programs in graphical form in any way they want. They can then choose to show only the essential aspects of the execution, thus making logical and performance errors easier to identify and locate.

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. work) in the Millipede project.

4.3 Configuration of transputer systems

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 task graph of the distributed program. As output, it generates the configuration statements needed by the Occam configurer to place this program structure on to a physical topology. The mapping utility also generates the commands needed by the reconfiguration software to connect the transputers into the topology described by the task graph.

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 of the processor graph on to the physical structure of Hathi-2 is done by a heuristic algorithm.

It is possible to find processor structures that cannot be mapped to the hardware structure of the multiprocessor system. First, not all graphs can be established on a transputer network, because a transputer has only four links. One example of this is a 5-dimensional hypercube, which requires a node degree of five. Second, the architecture of the distributed switching network in Hathi-2 imposes some limitations on which graphs can be established. The main limiting factor here is that there are only four links available between every pair of neighbouring boards in the static torus interboard connection. Finally, the heuristic algorithm used in the mapping utility does not guarantee that a mapping of a graph to the structure of Hathi-2 is found even if there is one. However, the mapping algorithm has proved to work well in practice for a large class of problems.

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 [Eklund 88, Eklund and Malén 88]. Work on developing more efficient mapping algorithms has been continued within the Millipede project by Mikael Norrbo (M.Sc. work) and Hong Shen. The main problems are algorithmic in nature: the general mapping problem is known to be NP-complete, so the effort is now 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 algorithm has also been implemented. The implementation shows quite good result. Hong Shen is expected to get his Ph.Lic. degree on this topic within this year.

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 [Solin 86, 88]. The 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. The prototype implementation has been done together with Lena Ståhl, Yngve Nyman (M. Sc. thesis) and Marie-Louise Lindström (M. Sc. work). Ulla Solin is expected to get her Ph.Lic. degree from this project within this year.

The animation research is being continued in the Millipede project to provide graphical monitoring and debugging of programs running on many processors in the Hathi-2 system, as explained above.

4.5 Multiprocessor programming and applications

One of the main goals in the 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 [Malén and Öhman 88].

- *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.

- *Analogue circuit simulation.* This was done in co-operation with Nokia Research Institute, by Tarmo Leinonen and Antti Raunio.

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

- *Implementation of a production system architecture on transputers*, by Johnny Boman (M.Sc. thesis).
- *Parallel implementation of the Rete-algorithm used in OPS5*, by Edward Eriksson (M. Sc. thesis).
- *Parallel implementations of some global optimization algorithms*, by Aimo Törn [Törn 88c] .

Multi-processor 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.

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 will integrate the existing programming tools under a common graphical user interface, allowing the programmer to specify the process structure of a distributed program in graphical form. The programmer can combine processes into tasks, which are then automatically allocated to the processors in the system by a mapping utility. The mapping utility also automatically configures the multiprocessor system to the desired topology. The monitoring utility and the animation utility will also be integrated into the programming environment. The results from monitoring and animation are presented to the user in a graphical way, where the presentation is based on the process structure of the distributed program. The programming environment is being designed by Mats Aspnäs and Ralph-Johan Back [Aspnäs and Back 89], and is implemented by Jens Granlund (M.Sc. work) and Henrik Gullberg (M.Sc. work).

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.Lic. 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.

Aimo Törn has done research on the efficient organization of files for one-probe access.

A new project, SMOC, on the interplay between organizational change and maintenance of information systems, is also planned.

5.1 Overflow indexing

The aim of the project, led by Aimo Törn and finished in 1984, was to investigate the feasibility of solving the overflow problem for a hash structure in file organization by using an index containing one entry for each overflow. For successive operations on the hash structure the theoretical minimum number of probes (one probe) for search, updating and deletion is proved for all items [Törn 84]. Insertion will need two or three probes. It is concluded that OVI may be a reasonable alternative for use with static to moderately dynamic files, as well as small to medium-sized and, in some cases, even large files, where fast operation is important or where worst-case performance is critical. An implementation for experimental testing of OVI was made by Barbro Sjöblom (M.Sc. thesis).

5.2 Software library evolution

The main goal in this project (SOLE) is to establish new results relating quality and costs of software. This requires quantitative data about the relative importance of different software costs, classified in some natural way to be available. It further requires that relations between

the most important cost factors and those quality factors that affect these are known. In order to be able to advise an organization about what measures to take in minimizing the total cost of its software, it is necessary that library information about the cost profile of software in the organization is available. This means that the costs related to software in the organization should continuously be collected and organized in a certain way.

The SOLE project, started 1988, is a research project in the Department of Computer Science of Åbo Akademi with participants also from Turku Business School. The project is lead by Professor Aimo Törn and Phil.Lic. Inger Eriksson. Funds have been obtained for the project from the Ministry of Education, the Academy of Finland and the Foundation for Economic Education. Total funding for the period 1988–1990 is 455 000 mk.

The project is very much in its beginning with one full-time researcher (Inger Eriksson) and six students. Aimo Törn will be a full-time researcher in the project from July, 1989 to June, 1990, on a senior research grant from the Academy of Finland. An analytical model for the optimal growth of the software library relative to a given yearly budget covering development and maintenance has been developed [Törn 88d]. The model is based on some generally accepted assumptions on soft-ware evolution and contains two characteristics relating cost and quality. Work is being done on for validating the model empirically. A case study for testing the feasibility of measuring these characteristics in practice will be performed. A questionnaire for deciding which organizations to include in a later investigation has also been completed.

5.3 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 aim of the project is not to measure the size of the problem described above, but instead to study its nature. Emprical research will be performed in two stages, first a preliminary study and later a deep intervention in one or two selected case-organisations. The aim is to achieve a deeper understanding of the problem, to find ways to reconstruct existing systems, and to identify ways of developing systems so that maintenance problems are reduced to a tolerable level.

The project is related to earlier research by the participants in the project. The idea of reconstructing an information system comes from the Knowledge and Work project (Markku Nurminen and others). There is also a relation to Pål Sørgaard's Ph.D. thesis on computer-supported cooperative work, in the sense that the SMOC project will try to find ways to remove computer-disrupted cooperative work, and in the sense that we will try to use the idea

of computerised shared material [Sørgaard 88b] as an idea for reconstruction of the systems. The research method in the SMOC project derives from the Knowledge and Work project and from the MARS project [Andersen et al. 89].

The project will need industrial partners. One such partner is already involved, since one of the participants in the project, Ulf Forsman, works in the City of Turku. It is our plan that Ulf Forsman will become a so-called industry-researcher co-financed by the Academy of Finland and the City of Turku. Ulf Forsman's work in the modernisation of the economic administration of the City of Turku is a part of the background of the project.

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 this year as Lecture Notes in Computer Sciences 350 (Springer-Verlag).

A new approach, *topological global optimization*, has recently been developed by Christina Juselius (M.Sc. thesis). The idea is to connect sampled neighbouring points so that a topology of the function to be optimized is obtained.

Currently Törn and Žilinskas are investigating the potential in using multiprocessor systems in global optimization. Experiments with parallel Fortran on the Hathi-2 computer of the department are undertaken [Törn 88c, Törn and Žilinskas 1989b].

Chapter 7

Research on Artificial Intelligence

Jan Komorowski has recently initiated some new research in the department in this field.

7.1 Knowledge bases and medical informatics

In co-operation with Carnegie-Mellon University and Harvard Medical School Jan Komorowski is 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.

In a longer range we plan to intensify our co-operation with Professor Niilo Saranummi's group at the Tampere University of Technology.

Chapter 8

Conference Activities

In August 1987 a two-week Nordic Summer School on Design of Distributed Systems was organized iat Nässlingen, Sweden, by Ralph-Johan Back, Dag Belsnes (Norway), Björn Pehrsson (Sweden) and Anders Ravn (Denmark). It was funded by the Nordic Research Courses organization. The lecturers were Eike Best from Germany, Ed Brinksma from the Netherlands, Ole-Johan Dahl from Norway, Mathew Hennessy and David May from U.K, Leslie Lamport from U.S.A and Amir Pnueli from Israel. The number of participants was around 60.

Ralph-Johan Back organized in co-operation with Reino Kurki-Suonio from Tampere University of Technology a one-week International Summer School on Formal Methods in Programming at Orivesi, Finland, July 1988. The summer school was held in connection with the ICALP conference in Tampere 1988. The lecturers were Rod Burstall, Michael Gordon, Tom Melham and John Tucker from U.K., Nissim Francez from Israel and Jean-Louis Lassez and Jay Misra from U.S.A and both organizers. The number of participants was approximately 60.

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

In 1988 Jan Komorowski was a member of the programme committee of the XIIth Annual Symposium on Computer Applications in Medical care in Washington, DC. It is a major medical informatics conference with several hundred papers and over two thousand participants. In addition he was a tutorial instructor and a member of the expert group on knowledge-based systems.

The department is presently engaged in organizing a Nordic Workshop on Program Correctness, to be held in Uppsala in October 1989.

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

Ralph-Johan Back:

1988 IEEE International Conference on Distributed Systems, Berlin, August 1987. Member of the Organizing Committee.

1988 IEEE International Symposium on Circuits and Systems, Helsinki University of Technology, Helsinki, June 1988. Session chairman.

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

Henryk Jan Komorowski:

1988 Symposium on Computer Applications in Medical Care, Washington, DC, Member of the Programme Committee.

AAAI Spring Symposium Series, CA, U.S.A. Panel chairman.

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

Aimo Törn:

1987 IMACS International Symposium on AI, Expert Systems and Languages in Modelling and Simulation, June 2-4, 1987, Barcelona, Session chairman.

4. Tagung Modellierung, Optimierung und Steuerung von Systemen, June 15-19, 1987, Leipzig, member of the program committee, Session chairman.

1988 3rd Finnish-Polish Symposium on Methodology and Applications of Decision Support Systems, September 26-29, 1988, Gdansk, 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

- Ralph-Johan Back (Aimo Törn,deputy)
- Inger Eriksson
- Tuija Pirttilä, student representative

Council of the Department

- Aimo Törn (chairman), head of the department
- Patrik Eklund (Kaisa Sere, deputy)
- Inger Eriksson (Ulla Solin, deputy)
- Thomas Lönnqvist, student representative (Tuija Pirttilä, deputy)
- Patrik Palm, student representative (Allan Nyman, deputy)

Board of the Computer Centre

- Aimo Törn (chairman)

9.2 National and international organizations

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

Ralph-Johan Back

- Co-ordinating professor for the doctoral education programme in Finland (Computer Science) 1.1.1986 – 31.7.1988. Member of the Co-ordination Board 1.1.1986 – 31.7.1988. Member of the steering committee for Computer Science 1.1.1986 –. Member of the steering committee for Computer Technology 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 Board of Natural Sciences of the Academy of Finland, 1.1.1986 – 31.12.1988
- Member of the Scientific Advisory Council of the Finnish Information Processing Association.
- Member of the Academy for Technical Sciences in Finland.

Kaisa Sere

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

Aimo Törn

- Member of the IFIP WG 7.6 on Optimization-Based Computer-Aided Modelling and Design 1988 –.
- Member of the Board for University Education in Natural Sciences (Ministry of Education) 1981 – . Member of the Board for Data-Processing 1986 – .
- Member of the Steering Committee for the Ph.D. programme in Information Systems (Ministry of Education) 1986- .
- Stand-by member of the Scientific Board for Military Defence (Ministry of Defence) 1985 – .
- Member of the Board for the Finnish Central Computer (Ministry of Education) 1980 –1985.
- Member of the Board for Data Processing and Statistics of the City of Turku 1985 – .

Chapter 10

Personnel

ASPNÄS John, stud.phil., Acting Laboratory Manager.
E-mail: aspnasj@finabo.abo.fi
Research interests: Distributed operating systems.

ASPNÄS Mats, M.Sc., Administrative leader of the Millipede project
E-mail: aspnasm@finabo.abo.fi
Research interests: parallel programming, monitoring of parallel systems.
Publications 1988, 1989:

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

3. Aspnäs, M., Back, R.J.R., Malén, T-E., *The Hathi-2 Multiprocessor System.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 80, 1989.

4. Aspnäs, M., Back, R.J.R., A Programming Environment for a Transputer-Based Multiprocessor System. To appear in *Proc. of the First Finnish-Hungarian Workshop on Programming Languages and Software Tools*, Szeged, Hungary, August 8-11, 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 82, 1989.

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

- Co-ordinating professor for the doctoral education programme in Finland (Computer Science) 1.1.1986 – 31.7.1988. Member of the Co-ordination Board 1.1.1986 – 31.7.1988. Member of the steering committee for Computer Science 1.1.1986 –. Member of the steering committee for Computer Technology 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 Board of Natural Sciences of the Academy of Finland, 1.1.1986 – 31.12.1988
- 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 Joachim Parrow, University of Uppsala, Sweden 1986 and Rassul Ayani, Royal Institute of Technology, Sweden 1989 and the Ph.D. candidacy examination committee for Limor Fix, Technion, Israel 1989.
- Opponent in Stein Gjessings Ph.D. defence, University of Oslo, Norway 1985 and Bengt Johnssons Ph.D. defence, University of Uppsala, Sweden 1987.
- External supervisor for the Ph.D. studies of Ph.Lic. Pentti Hietala, University of Tampere, Ph.Lic. Eeva Hartikainen, University of Helsinki and 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.

Main publications:

1. Back, R.J.R., *Correctness Preserving Program Refinements: Proof Theory and Applications.* Matematisch Centrum Tracts 131, Matematisch Centrum, Amsterdam 1980.
2. Back, R.J.R., Proving total correctness of nondeterministic programs in infinitary logic. *Acta Informatica* 15, pp. 233-249, 1981.
3. Back, R.J.R., On correct refinement of programs. *Journal of Computer and System Sciences*, vol. 23, 1, 1981.
4. Back, R.J.R., A continous semantics for unbounded nondeterminism. *Theoretical Computer Science* 23, 1983.
5. Back, R.J.R., A Calculus of Refinements for Program Derivations. *Acta Informatica* Vol 25, No 6, 1988, pp. 593-624.
6. Back, R.J.R., Kurki-Suonio, R., Distributed Co-operation with Action Systems. *TOPLAS, ACM Transactions on Programming Languages and Systems* Vol 10, No 4, 1988, pp. 513-554.
7. Back, R.J.R., Kurki-Suonio, R., Decentralization of process nets with a centralized control. *Distributed Computing* Vol 3, No 2, May 1989.

Publications 1988, 1989:

1. Back, R.J.R., A Calculus of Refinements for Program Derivations. *Acta Informatica* Vol 25, No 6, 1988, pp. 593-624.

2. Back, R.J.R., Kurki-Suonio, R., Distributed Co-operation with Action Systems. *TOPLAS, ACM Transactions on Programming Languages and Systems* Vol 10, No 4, 1988, pp. 513-554.

3. Back, R.J.R., Kurki-Suonio, R., Serializability in Distributed Systems with Handshaking. *Proc. of the 15th International Colloquium on Automata, Languages and Programming,*, Tampere, July 1988. Lecture Notes in Computer Science 317, pp. 52-66. Springer-Verlag. Heidelberg, 1988.

4. Back,R.J.R., Sere, K., *Stepwise Refinement of Parallel Algorithms.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 64, 1988. Submitted for publication.

5. Back,R.J.R., Sere, K., *An Exercise in Deriving Parallel Algorithms: Gaussian Elimination.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 65, 1988.

6. Back, R.J.R., *Data Refinement in the Refinement Calculus.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 68, 1988. Submitted for publication.

7. 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.

8. Back, R.J.R., Changing Data Representation in the Refinement Calculus. *Proce. of Hawaii International Conference on System Sciences (HICSS-22)*, 3-6.1.1989, Kailua-Kona, Hawaii. (Best paper award on the Software Track).

9. 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.

10. Aspnäs, M., Back, R.J.R., Malén, T-E., *The Hathi-2 Multiprocessor System.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 80, 1989.

11. Back, R.J.R., Refining Atomicity in Parallel Algorithms. *Proc. of the Conference on Parallel Architectures and Languages Europe*, Eindhoven, the Netherlands, June 12-16, 1989. Springer Lecture Notes in Computer Science. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 57.

12. 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.

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

14. Back, R.J.R., v. Wright, J., A Lattice-theoretic basis for a specification language. *Proc. of the Conference on Mathematics of Program Construction*, Groningen, the Netherlands, June 26-30,1989. Lecture Notes in Computer Science No 375, pp. 139-156. Also published in Reports on Computer Science & Mathematics,Åbo Akademi, Ser. A, No. 77.

15. Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems. *Proc. of the Conference on Mathematics of Program Construction*, Groningen, the Netherlands, June 26-30,1989. Lecture Notes in Computer Science No 375, pp. 115-138. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 78.

16. Aspnäs, M., Back, R.J.R., A Programming Environment for a Transputer-Based Multiprocessor System. To appear in *Proc. of the First Finnish-Hungarian Workshop on Programming Languages and Software Tools*, Szeged, Hungary, August 8-11, 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 82, 1989.

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

EKLUND Patrik:, Ph.D., Lecturer
E-mail: peklund@finabo.abo.fi
Research interests: category theory and computer programming, distributed systems, general structure theory, image analysis and pattern recognition, topology
Main publications:

1. Eklund, P., A Class of Completely Regular Spaces, *Internat. J. Math. & Math. Sci.* 7 (1984), 197–200.
2. Eklund, P., Category Theoretic Properties of Fuzzy Topological Spaces, *Fuzzy Sets and Systems*, 13 (1984), 303–310.
3. Eklund, P., W. Gähler, Basic Notions for Fuzzy Topology, I, *Fuzzy Sets and Systems*, 26 (1988), 333–356.
4. Eklund, P., W. Gähler, Basic Notions for Fuzzy Topology, II, *Fuzzy Sets and Systems*, 27 (1988), 171–195.
5. Caianiello, E.R., Eklund, P. E., Ventre, A.G.S., Implementations of the C-calculus, *Connection Science*, 1 (1989), to appear.
6. Eklund, P., Gähler, W., Generalized Cauchy Spaces, *Mathematische Nachrichten*, to appear.

Publications 1988, 1989:

1. Eklund, P., Modelling Configuration Expert Systems. In *Scandinavian Conference on Artificial Intelligence. Proc. of the 1st SCAI* (Danielsson, T. (ed.)). Tromsø, March 1988. International Organization Services. Amsterdam 1988.
2. Caianiello, E. R., Eklund, P. E., Squillante, M., Ventre, A. G. S., C-calculus: Sequential and parallel algorithms, SIAM meeting on Discrete Mathematics (poster), San Francisco, June 13-16, 1988.
3. Eklund, P., Gähler, W., Basic Notions for Fuzzy Topology, I. *Fuzzy Sets and Systems*, 26 (1988), pp. 333-356.
4. Eklund, P., Gähler, W., Basic Notions for Fuzzy Topology, II. *Fuzzy Sets and Systems*, 27 (1988), pp. 171-195.
5. Eklund, P., Malén,T-E., Block Placement in Switching Networks. *Proc. of CONPAR88*, Manchester, September 12-16,1988, pp. 289-295. Cambridge University Press. 1988.
6. Caianiello, E.R., Eklund, P., Squillante, M., Ventre, A.G.S., Formalism and Implementations of C-calculus. *Proc. of the International Conference on Computational Intelligence (CI88)*. Milano, September 26-30, 1988. North-Holland.

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

8. 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).

9. 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 27-28, 1989.

10. 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 19-22, 1989.

11. 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 (1989), pp. 78-81.

12. Eklund, P., Gähler, W., General Structures and Fuzzy Filters, *Proc. 3rd Congress of the International Fuzzy Systems Association (IFSA)*, (invited talk), Seattle, August 6-11, 1989.

13. 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 6-11, 1989.

14. Eklund, P., Gähler, W., Set Functors and General Spaces, *Proc. 11th International Seminar on Fuzzy Set Theory, Applications of Category Theory to Fuzzy Subsets*, (invited talk), Linz, September 11-15, 1989, to appear.

ERIKSSON Inger, Ph.Lic. Assistant.
E-mail: ieriksson@finabo.abo.fi
Research Interests: end-user training, simulation, information system quality
Main Publications:

1. Eriksson, I., *Användarens ADB-kunnande ett villkor för effektivt nyttjande av informationssystem.* (User's EDP-knowledge - A Necessity for Efficient Use of ISs). Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1988.

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.

Publications 1988, 1989:

1. Eriksson, I., Användarutbildning - erfarenheter och visioner. *Konferensdokumentation NordDATA 88*, part 2, pp. 188-192, Helsinki, 1988. Also published in *DATA Nordisk DataNytt* 5/1988, pp. 11-14. (In Swedish). Transl: End-user training - experiences and visions.

2. Eriksson, I., Simulation as a Learning Technique in Information Systems Development. In: "Organizational Competence in System Development". *Proceedings of Nordic Seminar*, November 16-18, 1988. Tranum Klitgaard, Denmark, Institute of Electronic Systems. Aalborg University Centre, 1988, pp. 57-68. To appear in Organizational Competence in System Development. Studentlitteratur, Lund, Sweden, 1989.

3. Eriksson, I., *Användarens ADB-kunnande ett villkor för effektivt nyttjande av informationssystem.* (User's EDP-knowledge - A Necessity for Efficient Use of ISs). Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1988.

4. 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.

5. Eriksson, I., Reijonen,P., Training Computer-Supported Work by Simulation. To appear in *Education and Computing.* Also to appear in *Proc. of the IFIP WG 3.4 Working Conference in Helsinki*, July 31-August 4, 1989

6. Eriksson, I., Finnäs, A., Creating a Visual Simulation Model of an Inventory System. To appear in *Proc. of the 12th IRIS Conference*, Skagen, Denmark, August 13-16,1989. Also presented at IFIP-HUB Conference, TC 9, WG 9.1., "Information System, Work and Organization Design", Berlin July 10-13, 1989.

7. Eriksson, I., Learning Process in the Context of Using ISs. Proc. IFIP WG 8.2. Working Conference on Information Systems Development for Human Progress in Organizations. North-Holland 1989.

GRANLUND Jens, stud.phil, Acting Assistant Professor [1]
E-mail: jgranlund@finabo.abo.fi
Research interests: Parallel Programming Environments, Distributed Operating systems, Object Oriented Programming.

HEKANAHO Jukkapekka, stud.phil, Research Assistant in Centipede.
E-mail: hekanaho@ibsun.abo.fi
Research interests: Knowledge-Based Systems, Program Transformations

HÖGLUND, Gundel, M.B.A., Project secretary for Millipede and Centipede.
E-mail: ghoglund@finabo.abo.fi

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. 1984 to present.
- 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.
- Member of the Experts Group, Symposium on Computer Applications in Medical Care, Washington, DC, 1988.
- Guest Editor, Special Issue of the Journal of Logic Programming on Partial Deductions, 1990.

[1] "Överassistent" in Swedish

Research Interests: Artificial Intelligence and Software Systems, Program Transformation, Knowledge Management Environments and Medical Informatics.
Main Publications:

1. Komorowski, H.J., *A specification of an Abstract Prolog Machine and its application to partial evaluation.* Linköping Studies in Science and Technology Dissertations, No. 69. Ph.D. thesis.

2. Komorowski, H.J., Partial evaluation as a means for inferring data structures in an applicative language: a theory and implementation in the case of Prolog, *Conference Record of the IXth Annual ACM Symposium on Principles of Programming Languages*, Albuquerque, NM, 1982.

3. Komorowski, H.J., Omori, S., A model and an implementation of a logic programming environment. *Proc. of the ACM SIGPLAN'85 Symposium on Language Issues in Programming Environments*, Seattle, WA, June 1985.

4. Maluszynski, J., Komorowski, H.J., Unification-free execution of Horn-clause programs. *Proceedings of the 2nd Logic Programming Symposium*, IEEE, Boston, MA, July 1985.

5. Komorowski, H.J., Greenes, R.A., Pattison, E., The Use of Fish-eye Views for Displaying Semantic Relationships in a Medical Taxonomy. *Proc. of the XIth Symposium on Computer Applications in Medical Care*, Washington, DC, November 1987, IEEE Computer Society.

6. Komorowski, H.J., On the mechanization of programmer's knowledge, *Proc. of the Second International Symposium on Methodologies for Intelligent Systems* (Z. Ras, Z., Zemankova, M. (eds)). North Holland. 1987.

7. Komorowski, H.J., Maluszynski, J., Logic Programming and Rapid Prototyping. *Science of Computer Programming 9* (1987), pp. 179-205, North-Holland.

8. Komorowski, H.J., Knowledge assimilation and extension of theory as a basis for Computer-Aided Software Engineering, *Proceedings of the 1st International Workshop on Computer-Aided Software Engineering*, Cambridge, MA, May 1987.

Publications 1988, 1989:

1. Komorowski, H.J., A Declarative Logic Programming Environment. *The Journal of Systems and Software 8*, pp. 77-89, 1988.

2. Barr, C.E., Komorowski, H.J., Pattison-Gordon, E., Greenes, R.A., Conceptual Modeling for the Unified Medical Language System. *Proc. of the XIIth Symposium on Computer Applications in Medical Care*, Washington, DC, November 1988, IEEE Computer Society.

3. Apple, R., Komorowski, H.J., Barr, C.E., Greenes, R.A., Intelligent Focusing in Knowledge Indexing and Retrieval - The Relatedness Tool. *Proc. of the XIIth Symposium on Computer Applications in Medical Care*, Washington, DC, November 1988, IEEE Computer Society.

4. Komorowski, H.J., Greenes, R.A., Barr, C.E.,Pattison, E., Browsing and Authoring Tools for a Unified Medical Language System, RIAO'88, *The Conference on User-Oriented Content-Based Text and Image Handling*, Cambridge, MA, March 1988.

5. 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.

6. Komorowski, J., *Synthesis of Programs in the Framework of Partial Deduction.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 81, 1988.

7. 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 20-26, 1989.

LINDHOLM Paul, Ph.Lic., Lecturer.
E-mail: plindholm@finabo.abo.fi

LÅNGBACKA Thomas, stud.phil., Acting Assistant
E-mail: langbackat@finabo.abo.fi, langback@ibsun.abo.fi

MALÉN Tor-Erik, stud.phil., Research Assistant in Millipede.
E-mail: tmalen@finabo.abo.fi
Research interests: programming tools and operating systems for multiprocessor systems.
Publications 1988, 1989:

1. Eklund, P., Malén,T-E., Block Placement in Switching Networks. *Proc. of CONPAR88*, Manchester, September 12-16,1988, pp. 289-295. Cambridge University Press. 1988.

2. Öhman, G., Mal'en, T–E., Kuusela, P., *Numerical Fluid Flow and Heat Transfer Calculations on Multiprocessor Systems.* Report 88-3. Dept. of Chemical Engineering, Heat Engineering Laboratory, Åbo Akademi. 1988.

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

4. Aspnäs, M., Back, R.J.R., Malén, T-E., *The Hathi-2 Multiprocessor System.*Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 80, 1989.

NYMAN Yngve, stud.phil., Research assistant in the Millipede Project
E-mail: nymany@finabo.abo.fi

RIISSANEN Tony, stud.phil., Acting Assistant
E-mail: triissanen@finabo.abo.fi

RAUNIO Antti, M.Sc., Project Researcher
E-mail: raunioa@finabo.abo.fi
Publications 1988, 1989:

1. Rantala, A., Raunio, A., Still, D-J., A Multiprocessor System for Fast Geometric Image Transformation. To appear in *Proc. of the 6th Scandinavian Conference on Image Analysis*, Oulu, June 19-22, 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 66, 1988.

ROSENBERG Robin, stud.phil., Acting Assistant
E-mail: rosenbergr@finabo.abo.fi

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

- Member of Council of Computing Centre at Åbo Akademi (- 31.12.1985)
- Deputy member of Council of Department of Computer Science
- Member of Board of Finnish Society for Computer Science (1.1.1989 -)
- Referee for the journals BIT and International Journal of Parallel Programming

Research interests: formal methods in program construction, program verification, parallel and distributed programming.
Main publications:

1. Sere, K., Stepwise removal of virtual channels in distributed algorithms. *Proc. Distributed Algorithms, 2nd International workshop*, Amsterdam, the Netherlands, July 1987. Lecture Notes in Computer Science 312, pp. 408-428.
2. Sere, K., *Stegvis utveckling av parallella algoritmer.* (Stepwise Refinement of Parallel Algorithms). Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1988.
3. Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems. *Conference on Mathematics of Program Construction*, Groningen, the Netherlands, June 26-30,1989. Lecture Notes in Computer Science No 375, pp. 115-138. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 78.

Publications 1988, 1989:

1. Sere, K., *Transforming Communication Topology in Distributed Algoritms.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 67, 1988. Submitted for publication.
2. Sere, K., *Deriving Action Systems for Processor Farms.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 69, 1988. Submitted for publication.
3. Walldén, M., Sere, K.,*Design and Implementation of Full-Text Retrieval on Transputer Networks.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 63, 1988.
4. Sere, K., *Stegvis utveckling av parallella algoritmer.* (Stepwise Refinement of Parallel Algorithms). Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1988.
5. Walldén, M., Sere, K., Free-text Retrieval on Transputer Networks. *Microprocessors and Microsystems*, Vol 13, April 1989, pp. 179-187.
6. Back, R.J.R., Sere, K., *Stepwise Refinement of Parallel Algorithms.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 64, 1988. Submitted for publication.

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

7. Back, R.J.R., Sere, K., *An Exercise in Deriving Parallel Algorithms: Gaussian Elimination.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 65, 1988.

8. Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems. *Conference on Mathematics of Program Construction*, Groningen, the Netherlands, June 26-30,1989. Lecture Notes in Computer Science No 375, pp. 115-138. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 78.

SHEN Hong, M.Sc., Research Assistant in Millipede.
E-mail: shenh@finabo.abo.fi
Professional activities:

- Member of parallel computing society.
- Referee for BIT

Research interests: parallel algorithms, parallel and distributed programming, process-to-processor mapping.
Main publications:

1. Chen, K.L., Shen, H., Bitonic selection algorithm on SIMD machines, *Proc. 2nd International Conference on Computers and Applications*, Beijing, China, 1987, 176-182. Also published in Computer Studies and Development (Chinese computer journal), Vol. 25, No. 1, 1988, pp. 1-14.

2. Chen, K.L., Shen, H., Bitonic selection network and bitonic selection algorithm on multiprocessors, *Computer Studies and Development* (Chinese computer journal), Vol. 24, No. 9, 1987, pp. 1-10.

3. Shen, H, Chen, K.L., An optimal parallel selection algorithm, *Acta University of Science and Technology of China*, 1988.

4. Shen, H., A fast parallel algorithm for integer sorting. To appear in *Proc. International Conference on Parallel Computing*, Leiden, Holland, August 29 - September 1, 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 70, 1988.

5. Shen, H, Fast path-disjoint routing for transputer networks. To appear in *Proc. of the First Finnish-Hungarian Workshop on Programming Languages and Software Tools*, Szeged, Hungary, August 7-11, 1989.

Publications 1988, 1989:

1. Shen, H., A Fast Parallel Algorithm for Integer Sorting. To appear in *Proc. of Parallel Computing 89*, Leiden, The Netherlands, August 29 - September 1, 1989. North-Holland. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 70, 1988.

2. Shen, H., Mapping Parallel Programs onto Transputer Networks. To appear in *Proc. of Australian Transputer and Occam User Group Conference*, Melbourne, Australia, July 5-6, 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 79.

3. Shen, H., Self-adjusting mapping: a heuristic mapping algorithm for mapping parallel programs onto transputer networks. To appear in *Proc. of the 11th Occam User Group Technical Meeting*, Edinburgh, U.K., September 25-26, 1989.

4. Shen, H., Fast Path-disjoint Routing in Transputer Networks. To appear in *Proc. of the First Finnish-Hungarian Workshop on Programming Languages and Software Tools*, Szeged, Hungary, August 7-11, 1989.

SOINI Annamari, M.A., Acting Instructor.
E-mail: soinia@finabo.abo.fi

SOLIN Ulla, M.Sc., Assistant.
E-mail: usolin@finabo.abo.fi
Research Interests: Algorithm animation
Main Publications:

1. Solin, U., Aspnäs, J., Levander, S., Raunio, A., A Distributed Election Algorithm Allowing Process Failures. *Proc. of the 2nd Finnish Summer School on Theoretical Computer Science*, Pargas, August 1-16, 1985. Also published in Reports on Computer Science & Matemathics, Åbo Akademi, Ser. A., No 43, 1985.

2. Solin, U., *Parallel Algorithm Animation.* Reports on Computer Science & Matemathics, Åbo Akademi, Ser. A, No 50, 1986.

Publications 1988, 1989:

1. Solin, U., Animation Techniques for Parallel Algorithms, *Proc. of the International Conference on Parallel Processing and Applications*, 437-445, North-Holland 1988.

STILL Dan-Johan stud.phil., Research Assistant in Centipede.
E-mail: stilldj@finabo.abo.fi
Research Interests: Formal Development of Parallel Programs
Publications 1988, 1989:

1. Rantala, A., Raunio, A., Still, D-J., A Multiprocessor System for Fast Geometric Image Transformation. To appear in *Proc. of the 6th Scandinavian Conference on Image Analysis*, Oulu, June 19-22, 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 66, 1988.

SÖDERHOLM Eva, stud.phil, Acting Assistant Professor [1]
E-mail: soderholme@finabo.abo.fi

SØRGAARD Pål, Ph.D., Assistant Professor [1]
E-mail: psorgaard@finabo.abo.fi
Research interests: System development, computer-supported cooperative work, computers in organisations, object-oriented programming.
Main publications

1. Sørgaard, P., Evaluating expert system prototypes. In Hans-Erik Nissen and Gunhild Sandström, editors, *Quality of work versus quality of information systems. Report of the 9th Scandinavian research seminar on systemeering*, pp.187–201, University of Lund, Department of Information and Computer Sciences, Lund, December 1986.

[1] "Överassistent" in Swedish

2. Andersen, N.E., Kensing, F., Lassen, M., Lundin, J., Mathiassen, L.,Munk-Madsen, A., Sørgaard, P., *Professionel systemudvikling.* Teknisk Forlag, København, 1986. The book will appear in English entitled *Professional Systems Development* on Prentice-Hall, Business Information Technology Series.

3. Nygaard, K., Sørgaard, P., The perspective concept in informatics. In Gro Bjerknes, Pelle Ehn, and Morten Kyng, editors, *Computers and Democracy - A Scandinavian Challenge*, pp. 371–393, Avebury, Gower Publishing Company, Aldershot, England, 1987.

4. Sørgaard, P., Object oriented programming and computerised shared material. In Stein Gjessing and Kristen Nygaard, editors, *ECOOP '88 European Conference on Object-Oriented Programming, Oslo, Norway, August 1988, Proceedings*, pp. 319–334, Lecture Notes in Computer Science 322, Springer Verlag, Heidelberg, 1988. This paper won the 1988 international award for the best paper on the theme of Interactive Interfaces between Collective Support Systems and their users. The award was offered by the University of Amsterdam, "Support, Survival & Culture" (OOC) Program.

5. Sørgaard, P., *Programming Environments and System Development Environments.* PB 252, Computer Science Department, Aarhus University, Århus, May 1988.

6. Sørgaard, P., Transaction supporting systems and organisational change. *Office: Technology and People*, Vol 4, No 3, pp. 229–243, June 1989. To appear.

Publications 1988, 1989:

1. Sørgaard, P., A Framework for Computer Supported Cooperative Work. In: Kaasböll, J. (ed.),*Report of the 11th IRIS (Information System Research Seminar in Scandinavia)*, August 10-12, 1988, pp. 620-640. Research Report No. 116, Dept. of Informatics, University of Oslo, 1988.

2. Sørgaard, P., Object Oriented Programming and Computerised Shared Material. Gjessing, S., Nygaard,m K. (eds), *ECOOP88 European Conference on Object Oriented Programming*, Oslo, August, 1988, pp. 319-334. Lecture Notes in Computer Science 322, Springer Verlag (Heidelberg) 1988.

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

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

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

- Member of the Board for the Finnish Central Computer (Ministry of Education) 1980-1985.

- Member of the Board for University Education in Natural Sciences (Ministry of Education) 1981- .

- Member of the Board for Data-Processing and Statistics of the Cit of Turku (Åbo) 1985-

- Deputy member of the Scientific Board for Military Defence (Ministry of Defence) 1985-.
- Member of the Board for University Education in Data-Processing (Ministry of Education) 1986- .
- Member of the Steering Committee for the Ph.D. 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 -

Research interests: Monte Carlo methods, global optimization, programming techniques, simulation, and software metrics.

Main publications:

1. Törn, A.A., Crude Monte Carlo quadrature in infinite variance case and the central limit theorem. *BIT 6*, pp. 339-346, 1966
2. Törn, A.A., Cluster analysis using seed points and density-determined hyperspheres with an application to global optimization. *IEEE trans. on Systems, Man and Cybernetics 7*, pp. 610-616, 1977.
3. Törn, A.A., *A sampling-search-clustering approach for exploring the feasible/efficient solutions of MCDM problems*, Comput. & Ops. Res. 7, pp. 67-79, 1980.
4. Törn, A.A., *Programming - from Problem to Documentation , with Introduction to Algol 60, Basic, Fortran IV and 77, Pascal and Simula.* (Studentlitteratur, Lund) 224 pp, 1981.
5. Törn, A.A., Hashing with overflow indexing. *BIT 24*, pp. 317-332, 1984.
6. Törn, A.A., *Models of software accumulation.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 59, 7 pp. 1988. To appear in *The Journal of Systems & Software*, 1988.
7. Törn, A.A., Systems modelling and analysis using simulation nets, In: C.A. Kulikowski, R.M. Huber and G.A. Ferraté (eds.). *Artificial Intelligence, Expert Systems and Languages in Modelling and Simulation*, Elsevier Science Publishers B.V. (North-Holland), pp. 283-288, 1988.
8. Törn, A.A., Žilinskas, A., *Global optimization*, Lecture Notes in Computer Science 350, (Springer–Verlag, Berlin), 255 pp., 1989.

Publications in 1988, 1989

1. Törn, A.A., Simulation modeling formalism: Extended Petri-net graphs. In: M.G. Singh (ed.), *Encyclopedia of Systems and Control* (Pergamon Press, Oxford), pp. 4348-4350, 1988.
2. Törn, A.A., *Models of software accumulation.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 59, 7 pp. 1988. To appear in *The Journal of Systems & Software*, 1988.
3. Törn, A.A., Systems modelling and analysis using simulation nets, In: C.A. Kulikowski, R.M. Huber and G.A. Ferraté (eds.). *Artificial Intelligence, Expert Systems and Languages in Modelling and Simulation*, Elsevier Science Publishers B.V. (North-Holland), pp. 283-288, 1988.

4. Törn, A.A., *PICA - A graphical program development tool.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 60, 13 pp., 1988.

5. Törn, A.A., Parallel Monte Carlo with application to global optimization, In: K. Juslin and E. Silvennoinen (eds.), *Numerical Simulation of Processes, SIMS-88 Symposium, The 30th Annual Meeting of the Scandinavian Simulation Society (SIMS)*, Espoo, Finland 21-22 April, 1988, VTT Symposium 84, (Technical Research Centre of Finland, Espoo), pp. 156-165, 1988.

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

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

8. Törn, A.A., Žilinskas, A., Global optimization algorithms in optimal design. Presented at the *14th IFIP Conference on System Modelling and Optimization*, Leipzig, July 3-7, 1989.

WAXLAX Patrick, stud.phil., Research Assistant in Centipede.
E-mail: waxlax@ibsun.abo.fi
Research interests: human-computer interface, hypertext, program transformation systems

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

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

1. Wikman, R., Data compression. *Some Notes on the Effects of Combining Two Compression Methods.* Reports on Computer Science & Mathematics, Åbo Akademi Ser. A, No 47, 1985.

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, are listed below for the period 1988–89.

11.1 Foreign visitors to the department

1988 Dieter Ehrenberg, Dr., Technische Hochschule Leipzig, GDR, 21.3- 2.4.1988.

Carroll Morgan, Dr., Oxford University, U.K., 11-15.4.1988

Bengt Jonsson,Dr., SICS, Sweden, 27.4.1988

Anatoli Karpenko, Senior Researcher, Moscow, 9-19.5.1988

Mario Finka, M.Sc., Slovak Academy of Sciences, Czechoslovakia, 13-17.6.1988.

John V. Tucker, Prof., University of Leeds, U.K., 25-26.7.1988.

Jay Misra, Prof., University of Texas at Austin, U.S.A., 15.7.1988.

Nissim Francez, Prof., Technion, Haifa, Israel. 28.6-31.8.1988

Hans-Jürgen Sebastian, Prof., Sektion Mathematik und Informatik, Technische Hochschule Leipzig, GDR, 5-14.10.1988.

Andrzej Straszak, Prof., Systems Research Institute, Polish Academy of Science, Poland, 7.11.1988.

Tilo Brock, Stud., Sektion Mathematik und Informatik, Technische Hochschule Leipzig, GDR, 17.11-17.12.1988.

Beata Konikowska, Dr., Polish Academy of Sciences, Poland, 7-8.12.1988.

Antanas Žilinskas, Dr., Institute of Mathematics and Cybernetics, Lithuanian Academy of Sciences, USSR, 15.11-15.12.1988.

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.

11.2 Visits and conference participation by staff

Mats Aspnäs

1988 8th Occam User Group Technical Meeting, University of Sheffield, Sheffield, Great Britain.

IEEE International Symposium on Circuits and Systems, Helsinki University of Technology, Helsinki.

ICALP Summer School on Formal Methods in Programming, Orivesi, Finland.

CONPAR-88, UMIST, Manchester, Great Britain.

Formal Techniques in Real-Time and Fault-Tolerant Systems, University of Warwick, Coventry, Great Britain.

9th Occam User Group Technical Meeting, University of Southampton, Southampton, Great Britain.

Edinburgh Concurrent Supercomputer First Annual Seminar, University of Edinburgh, Edinburgh, Great Britain.

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

Visit to University of Groningen, Groningen, Netherlands.

Ralph-Johan Back

1988 Visit to University of Texas at Austin, Austin, Texas, U.S.A (guest lectures).

Visit to Cornell University, Ithaca, New York, U.S.A.

Visit to MCC, Austin, Texas, U.S.A (guest lectures).

8th Occam User Group Technical Meeting, University of Sheffield, Sheffield, Great Britain.

IEEE International Symposium on Circuits and Systems, Helsinki University of Technology, Helsinki (session chairman).

15th International Conference on Automata, Languages and Programming (ICALP 88), Tampere, Finland (paper).

ICALP Summer School on Formal Methods in Programming, Orivesi, Finland (co-organizer, lectures).

Formal Techniques in Real-Time and Fault-Tolerant Systems, University of Warwick, Coventry, Great Britain.

9th Occam User Group Technical Meeting, University of Southampton, Southampton, Great Britain.

Edinburgh Concurrent Supercomputer First Annual Seminar, University of Edinburgh, Edinburgh, Great Britain.

Visit to Edinburgh University, Edinburgh, U.K. (guest lecture).

Visit to Oxford University, Oxford, U.K. (guest lecture).

Visit to Manchester University, Manchester, U.K. (guest lecture).

Visit to University of Leeds, Leeds, U.K.

Visit to University of Glasgow, Glasgow, U.K

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

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

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

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

Visit to CWI, Amsterdam, Netherlands (guest lecture).

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

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

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

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

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

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

Patrik Eklund

1988 First Scandinavian Conference on Artificial Intelligence, Tromsö, Norway (paper).

Visiting researcher at University of Groningen, Department of Computing Science, Groningen, Netherlands.

CONPAR 88, Manchester, U.K. (paper).

International Conference on Computational Intelligence (CI88), Milan, Italy (paper).

Visit to Università degli studi di Trento, Facolta di Economia e Commercio, Istituto di Informatica, Trient, Italy (talk).

1989 Visit to Universität des Saarlandes, Institut für Informatik, Saarbrücken, W.-Germany (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).

Inger Eriksson

1988 Research semester at the University of Lund.

Method and Metaphor in Information Analysis, Qualitative Methods, Prof. Richard Boland, Paimio, Finland.

Lecture and work shop: In–formation of Information Systems, Boland and the Knowledge and Work Research Group, Prof. Richard Boland, University of Turku, Finland.

International Conference on Informatics "INFORMATICS '88", Havana, Cuba (paper).

Research visit, University of Lund.

NordDATA 88, Helsinki (paper).

POLAR-seminar, Denmark (paper).

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).

Jens Granlund

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

Summer School on User Interfaces, University of Tampere, Finland

Jukkapekka Hekanaho

1988 ICALP Summer School on Formal Methods in Programming, Orivesi, Finland.

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

Jan Komorowski

1988 Congress of the American Association for the Advancement of Science, Boston, U.S.A. (invited talk).

XIIth Symposium on Computer Applications in Medical Care, Washington, DC, U.S.A. (paper).

Visit to University of Massachusetts at Amherst, USA (guest lecture).

Visit to Northeastern University in Boston, USA (guest lecture).

Visit to University of Pittsburgh, Pittsburgh, USA (guest lecture).

Visit to Technical Research Centre, Tampere, Finland (guest lecture).

AAAI Spring Symposium Series, Stanford, CA, USA, March, 1988 (Panel Chairman).

French/Finnish Symposium on Biomedical Engineering, November 1988, Lyon, France (invited talk).

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).

Thomas Långbacka

1988 ICALP Summer School on Formal Methods in Computer Science, Orivesi.

1989 Summer School on Alternatives of Logic Programming, Kuopio

Tor-Erik Malén

1988 IEEE International Symposium on Circuits and Systems, Helsinki, Finland.

Transputer Course using TDS, arranged by INMOS Ltd., Bristol England.

1989 10th occam User Group Technical Meeting, Enschede, Netherlands.

Visit to Groningen University, Groningen, Netherlands.

Kaisa Sere

1988 15th ICALP, Tampere Finland.

ICALP Summer School on Formal Methods in Programming, Orivesi, Finland.

International Summer School on Constructive Methods in Computing Science, Marktoberdorf, West-Germany.

9th Occam User Group Meeting, Southampton, United Kingdom.

Symposium on Formal Techniques in Real Time and Fault Tolerant Systems, Warwick, United Kingdom.

Edinburgh Concurrent Supercomputer: first annual seminar, Edinburgh, United Kingdom.

HOL Users Meeting, Cambridge University, United Kingdom.

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, Netherlands (paper).

Hong Shen

1988 ICALP Summer School on Formal Methods in Programming, Orivesi, Finland.

IEEE International Symposium on Circuits and Systems, Helsinki, Finland.

Visit to Linköping University, Sweden.

1989 Visit to Linköping University, Sweden.

Dan-Johan Still

1988 Human Computer Interaction (HCI) and Expert Systems, Tampere, Finland

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

Summer School on User Interfaces, Tampere, Finland

Pål Sørgaard

1988 11th Information Systems Research Seminar in Scandinavia, Røros (paper).

ECOOP'88. European Conference on Object-Oriented Programming. Oslo (paper).

Organizational Competence in System Development, Tranum Klitgaard.

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

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

Talk at the University of Amsterdam, OOC programme.

Tietotekniikka 89, Jyväskylä (paper).

Aimo Törn

1988 Visit to the Lithuanian Academy of Sciences, Institute of Mathematics and Cybernetics, Vilnius, Lithuania (talk).

SIMS-88, Espoo, Finland (paper).

3rd Finnish-Polish Symposium on Methodology and Applications of Decision Suport Systems, Gdansk, Poland (paper).

The joint Finnish-Soviet Software Symposium, Helsinki, Finland (paper).

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

Patrick Waxlax

1988 ICALP Summer School on Formal Methods in Programming, Orivesi, Finland.

Course on Human-Computer Interaction and Expert Systems, Tampere, Finland.

1989 Summer School on User Interfaces, Tampere, Finland.

Chapter 12

Accepted Theses

The theses accepted at the department during the period January 1984 – May 1989 are listed below. The titles have been translated from Swedish to English.

12.1 Licentiate of Philosophy

1988 Eriksson, I., *The User's EDP-knowledge - A Necessity for Efficient Use of ISs*

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

12.2 Master of Science

1984 Ahlbäck, G., *Data security*

Jansson, P., *SOSED - a text editor for Univac-1100*

Ramberg, P., *Perspective plotting of functions with two variables*

Widjeskog, J., *The land tenancy system in the town of Karleby*

1985 Laaksonen, L., *TIMPS - an interactive time planning system*

Palenius, K., *Information systems:principles and tools*

Sjöblom, B., *A comparison of some methods for organizing tables*

Tammilehto, M., *Dialogue design*

Wåg, P., *A database for inventory and suppliers for the Town of Turku*

1986 Engman, R., Sölgén, J., *Systems for administrating loans*

Finne, J., *Early history of computing in Turku*

Haglund, P., *Important properties of code specifications*

Kujanen, R., *System development and the use of an application generator*

Söderback, T., *User system development*

1987 Aspnäs, M., *An implementation of joint actions in Modula-2*

Björkgren, C-J., *Expert systems for insurance contracts*

Raunio, A., *Modularization in action system*

Wikberg, S., *An expert system for pension calculations.*

1988 Aho, J., *Computer science education in Finland*

Boman, J., *Computer architectures for execution of production systems*

Eriksson, E., *Parallel implementation of the Rete algorithm*

Gammals, G., *A computer independent command interpreter*

Hassel, R., *Discrete object oriented simulation in Modula-2*

Hägg, T., *A graphical tool for design and execution of programs for discrete simulation*

Konttinen, K., *Use of information systems in working life*

Koski, P., *On measuring and quality in system design*

Kuusela, P., *Numerical fluid-flow calculations with the SIMPLER method on a multiprocessor system*

Malén, T-E., *Cosso - a configuration tool for a multiprocessor system*

Nygård, S., *A tool for constructing logical databases*

Pensar, U., *Uncertainty in MYCIN*

Sjöroos, K., *A study of efficiency of hashing with chaining*

Storsved, G., *Threats and security in computer systems and networks*

Sund, P-E., *Graphical user interfaces, man-machine communication*

Söderholm, E., *A Prolog based expert system for stock-keeping*

Walldén, M., *Full-text retrieval on multiprocessor systems*

Wennström, K., *PDPSIM - a PDP-11 simulator on DEC-10*

Åström, R., *Application generators: Help functions*

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*

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

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

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

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

Chapter 13

Publications 1984 - 1989

1984

Journals

[**Back and Mannila 84**] Back, R.J.R., Mannila, H., A Semantic Approach to Program Modularity. *Information and Control*, Vol 60, Nos. 1-3, January/February/March 1984. Academic Press, New York and London.

[**Eklund 84a**] Eklund, P., A Class of Completely Regular Spaces. *Internat. J. Math. & Math. Sci* 7 (1984) pp. 197-200.

[**Eklund 84b**] Eklund, P., Category Theoretic Properties of Fuzzy Topological Spaces. *Fuzzy Sets and Systems*, 13 (1984) pp. 303-310.

[**Törn 84**] Törn, A., Hashing with overflow indexing, *BIT* No. 24, 1984, pp. 317-332.

Proceedings

[**Back and Kurki-Suonio 84a**] Back, R.J.R., Kurki-Suonio, R., A case study in constructing distributed algorithms: distributed exchange sort. *Proc. Winter School on Theoretical Computer Science*, pp.1-3, Lammi, January 1984. Finnish Society of Information Processing Science.

[**Back and Rantanen 84**] Back, R.J.R., Rantanen, A., A case study in program modification: generalizing Lindstroms tree marking algorithm. *Proc. Winter School on Theoretical Computer Science*, pp.33-64, Lammi, January 1984. Finnish Society of Information Processing Science.

[**Back and Hietala 84**] Back, R.J.R., Hietala, P., A simple user interface for interactive program verification. In B. Schabel (ed.), *Human-Computer Interaction - INTERACT84: First IFIP Conference of Human-Computer Interaction*, pp. 391-395. North- Holland, Amsterdam, 1984.

Technical Reports

[**Back and Kurki-Suonio 84b**] Back, R.J.R., Kurki-Suonio, R., *Co-operation in distributed systems using symmetric multi-process handshaking.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 34, 1984.

[**Back et al 84**] Back, R.J.R., Eklund, P., Kurki-Suonio, R., *A fair and efficient implementation of CSP with output guards.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 38, 1984.

[**Eklund 84c**] Eklund, P., *Synchronizing Multiple Processes in Common Handshakes.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 39, 1984.

1985

Journals

[**Back and Mannila 85**] Back, R.J.R., Mannila, H., On the Suitability of Trace Semantics for Modular Proofs of Communicating Proceses. *Theoretical Computer Science* 39(1985), pp. 47- 68. Elsevier Science Publishers B.V. (North-Holland).

[**Törn 85**] Törn, A., Simulation nets, a simulation modelling and validation tool. *Simulation* 45:2, pp. 71-75.

Proceedings

[**Aspnäs et al 85**] Aspnäs, M., Eklund, P., Wärnström, A., Election in a Uni-directional Ring with a Variable Number of Processes. *Proc. 2nd Finnish Summer School on Theoretical Computer Science*, Pargas, August 1-16, 1985.

[**Salminen and Back 85**] Salminen, A., Back, R.J.R., A Relational Model for Documents. *Proc. International Symposium on New Directions on Computing*, pp. 50-59. IEEE Computer Society Press, 1985.

[**Solin et al 85**] Solin, U., Aspnäs, J., Levander, S., Raunio, A., A Distributed Election Algorithm Allowing Process Failures. *Proc. of the 2nd Finnish Summer School on Theoretical Computer Science*, Pargas, August 1-16, 1985, pp. 153-166. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 43, 1985.

Technical Reports

[**Back et al 85**] Back, R.J.R., Hartikainen, E., Kurki-Suonio, R., *Multi-process handshaking on broadcasting networks.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A. No 42, 1985.

[**Back and Kurki-Suonio 85**] Back, R.J.R., Kurki-Suonio, R., *Serializability in distributed systems.* Carnegie-Mellon University, Dept. of Computer Science. Research Reports CMU-CS-85-109, 1985.

[**Wikman 85**] Wikman, R., *Data Compression. Some Notes on the Effects of Combining Two Compression Methods.* Reports on Computer Science & mathematics, Åbo Akademi, Ser. A, No 47, 1984.

1986

Journals

[**Back86a**] Back, R.J.R., A computational interpretation of truth logic. *Synthese 66* (1986) pp. 15-34. D Reidel Publishing Company 1986. Finnish-Soviet Logic Seminar, Helsinki

1983. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 33, 1984.

[Eklund 86a] Eklund, P., A Comparison of Lattice-theoretic Approaches to Fuzzy Topology. *Fuzzy Sets and Systems* 19 (1986), pp. 81-87.

[Eklund 86b] Eklund, P., Categorial Fuzzy Topology. *Acta Acad. Aboensis*, Ser. B 46(1) (1986), pp. 1-13.

Collections

[Törn 86a] Törn, A., Simulation Nets - An application of Petri nets to system simulation. In: Ruschizka, M. (ed.), *Transactions on Scientific Computing -85*, Vol. 2, Computer Systems: Performance and Simulation (Elsevier), pp. 213-218. Also published in proceedings from 11th Imacs Congress on System Simulation and Scientific Computation, 5-9.8.1985, Oslo, pp. 187-190.

Proceedings

[Eklund and Gähler 86] Eklund, P., Gähler, W., Fuzzy Convergence. In: General Topology and its Relations to Modern Analysis and Algebra VI. *Proc. of the 6th Prague Topological Symposium 1986* (abstract), p. 662. Heidelmann Verlag, 1988.

[Eriksson and Kalmi 86] Eriksson, I., Kalmi, R., Interpretation of large ISs as a set of personal ISs and enrichening the users' knowledge with this view. In: Nissen, H-E. & Sandström, G. (eds), *Quality of Work versus Quality of Information Systems.* Report of the Ninth Scandinavian Research Seminar on Systemeering, Part 1, pp. 70-84. University of Lund, 1986.

[Törn 86b] Törn, A., Clustering methods in global optimization. *Preprints of the Second IFAC Symposium on Stochastic Control.* Vilnius, May 19-23, Part 2, pp. 138-143.

Technical Reports

[Back 86b] Back, R.J.R., *Formal methods in program construction.* Final report on a Research Project. Sponsored by Academy of Finland 1982-85. Åbo Akademi, Reports on Computer Science & Mathematics, No 48, 1986.

[Salminen and Back 86] Salminen, A., Back, R.J.R., *A Calculus for Document Handling.* University of Jyväskylä. Computer Science Reports, Working Paper WP-9. March 1986.

[Back and Hietala 86] Back, R.J.R., Hietala, P., *A Program Proof Management System.* Dept. of Mathematical Sciences. University of Tampere. Report No A136. December 1986.

[Solin 86] Solin, U., *Parallel Algorithm Animation.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 50, 1986.

1987

Journals

[Back and Kurki-Suonio 87] Back, R.J.R., Kurki-Suonio, R., A New Paradigm for the Design of Concurrent Systems, *Ada Letters VII*, No 6 (Fall, 1987), pp. 110-112.

Proceedings

[**Back 87a**] Back, R.J.R., A Calculus for Refinements for Program Derivations. *Proc. of the Workshop on Programming Logic*, Editors: Dybjer et al. Report 37, pp. 271-299. Chalmers University of Technology and University of Göteborg. Göteborg 1987.

[**Eriksson 87a**] Eriksson, I., A Program for Supporting End-Users' EDP-learning Composed by the Knowledge and Work Project. *First International Workshop "Activity, Work and Learning"*. Karis, June 1987. Nordic Society for Educational Research.

[**Eriksson 87b**] Eriksson, I., Learning Processes in the Context of Using ISs. In: Järvinen, P. (ed.), The report of the *10th IRIS (Information Systems Research Seminar In Scandinavia) Seminar*. Acta Universitatis Tamperensis, Ser. B. Vol. 27, Part 1, pp. 139-155, 1987. Also presented at the International Conference of Informatics "INFORMATICS 88", Havanna, Cuba, February 1988. To appear.

[**Sere 87b**] Sere K., Stepwise removal of virtual channels in distributed algorithms. *Proc. Distributed Algorithms, 2nd International workshop*, Amsterdam, Netherlands, July, 1987. Lecture Notes in Computer Science 312, pp. 408-428, (Revised version of [Sere87a]).

Technical Reports

[**Back 87b**] Back, R.J.R., *Procedural Abstraction in the Refinement Calculus.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 55, 1987. Submitted for publication.

[**Nurminen et al 87**] Nurminen, M., Eriksson, I., Finneman, A., Niemelä, J., Snellman, M., *Raportti työ- ja tietojärjestelmästä elintarviketehtaan lähettämössä.* University of Turku, Computer and Information Science , Report A 50, 1987. (In Finnish). Transl: Report of the work and information system of an inventory department in a firm belonging to foodstuffs industry.

[**Sere 87a**] Sere, K., *Stepwise removal of virtual channels in distributed algorithms: A case study.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 53, 1987.

[**Walldén 87**] Walldén, M., *A Case Study: Performance of a Distributed Algorithm.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. B, No. 5, 1987.

1988

Journals

[**Back 88a**] Back, R.J.R., A Calculus of Refinements for Program Derivations. *Acta Informatica*, Vol 25, No 6, 1988, pp. 593-624, (Revised version of [Back87a]).

[**Back and Kurki-Suonio 88a**] Back, R.J.R., Kurki-Suonio, R., Distributed Co-operation with Action Systems. *ACM Transactions on Programming Languages and Systems*, Vol 10, No 4, 1988, pp. 513-554.

[**Eklund 88a**] Eklund, P., Mapping Processes onto Processors. *Wissenschaftlische Zeitschrift*, TH, Leipzig, 12(1988), pp. 105-109. Also published in 4. Tagung, Modellierung, Optimierung und Steuerung von Systemen. Wissenschaftlische Berichte der Technischen Hochschule Leipzig. June 1987, pp. 36-37.

[**Eklund and Gähler 88a**] Eklund, P., Gähler, W., Basic Notions for Fuzzy Topology, I. *Fuzzy Sets and Systems*, 26 (1988), pp. 333-356.

[**Eklund and Gähler 88b**] Eklund, p., Gähler, W., Basic Notions for Fuzzy Topology, I. *Fuzzy Sets and Systems*, 27 (1988), pp. 171-195.

[**Komorowski 88**] Komorowski, J., A Declarative Logic Programming Environment. *The Journal of Systems and Software* 8, 77-89 (1988).

Collections

[**Eriksson 88c**] Eriksson, I., Simulation as a Learning Technique in Information Systems Development. To appear in *Organizational Competence in System Development.* Studentlitteratur, Lund, Sweden, 1989. Also published in Proceedings of the Nordic Seminar on "Organizational Competence in System Development". November 16-18, 1988. Tranum Klitgaard, Denmark, Institute of Electronic Systems. Aalborg University Centre, 1988, pp. 57-68.

[**Törn 88a**] Törn, A., Simulation modelling formalism: Extended Petri net graphs. In: Singh, M (ed.), *Encyclopedia of Systems and Control* (Pergamon Press, Oxford), pp. 4348-4350.

[**Törn 88b**] Törn, A., Systems modelling and analysis using simulation nets. In: Kulikowski, C.A. et al (eds), *Artificial Intelligence, Expert Systems and Languages in Modelling and Simulation* (Elsevier), pp. 283-288. Also published in preprints of International Symposium on AI, Expert Systems and Languages in Modelling and Simulation, Barcelona, June 2-4, 1987, pp. 31-36.

Proceedings

[**Appel et al 88**] Appel R., Komorowski, H.J., Barr, C.E., Greenes, R.A., Intelligent Focusing in Knowledge Indexing and Retieval – The Relatedness Tool. *Proc. XIIth Symposium on Computer Applications in Medical Care*, Washington, DC, November 1988. IEEE Computer Society.

[**Barr et al 88**] Barr, C.E., Komorowski, H.J., Pattison-Gordon, E., Greenes, R.A., Conceptual Modeling for the Unified Medical Language System. *Proc. XIIth Symposium on Computer Applications in Medical Care*, Washington, DC, November 1988, IEEE Computer Society.

[**Back and Kurki-Suonio 88b**] Back, R.J.R., Kurki-Suonio, R., Serializability in Distributed Systems with Handshaking. *15th International Colloquium on Automata, Languages and Programming*, Tampere, July 1988. Lecture Notes in Computer Science 317, pp. 52-66. Springer-Verlag. Heidelberg, 1988. (Revised version of [Back and Kurki-Suonio 85].)

[**Caianiello et al 88a**] Caianiello, E.R., Eklund, P., Squillante, M., Ventre, A.G.S., Formalism and Implementations of C-calculus. *Proc. of the International Conference on Computional Intelligence* (CI88). Milano, September 26-30, 1988. North-Holland. To appear.

[**Caianiello et al 88b**] Caianiello, E. R., Eklund, P. E., Squillante, M., Ventre, A. G. S., C-calculus: Sequential and parallel algorithms *SIAM meeting on Discrete Mathematics* (poster), San Francisco, June 13-16, 1988.

[Eklund and Malén 88] Eklund, P., Malén,T-E., Block Placement in Switching Networks. *Proc. CONPAR88*, Manchester, September 12-16, 1988, pp. 289-295. Cambridge University Press. 1988.

[Eklund 88b] Eklund, P., Modelling Configuration Expert Systems. In: *Proc. of the 1st Scandinavian Conference on Artificial Intelligence* (Danielsson, T. (ed.)). Tromsø, March 1988. International Organization Services. Amsterdam 1988.

[Eriksson 88a] Eriksson, I., Användarutbildning - erfarenheter och visioner. *Konferensdokumentation NordDATA 88*, part 2, pp. 188-192, Helsinki, 1988. Also published in DATA Nordisk DataNytt 5/1988, pp. 11-14. (In Swedish). Transl: End-user training - experiences and visions.

[Komorowski et al 88] Komorowski, H.J., Greenes, R.A., Barr, C.E., Pattison, E., Browsing and Authoring Tools for a Unified Medical Language System. *Proc. of RIAO'88, The Conference on User-Oriented Content-Based Text and Image Handling*, Cambridge, MA, March 1988.

[Solin 88] Solin, U., Animation Techniques for Parallel Algorithms. International *Proc. Conference on Parallel Processing and Applications*, September,1987, L'Aquila, Italy, pp. 437-445. Editors: E. Chiricorozzi, A. D'Amico. Elsevier Science Publishers B.V.(North-Holland), 1988.

[Sørgaard 88a] Sørgaard, P., A Framework for Computer Supported Cooperative Work. In: Kaasböll, J. (ed.), *Report of the 11th IRIS (Information System Research Seminar in Scandinavia)*, August 10-12, 1988, pp. 620-640. Research Report No. 116, Dept. of Informatics, University of Oslo, 1988.

[Sørgaard 88b] Sørgaard, P., Object Oriented Programming and Computerised Shared Material. Gjessing, S., Nygaard,m K. (eds), *ECOOP88 European Conference on Object Oriented Programming*, Oslo, August, 1988, pp. 319-334. Lecture Notes in Computer Science 322, Springer Verlag (Heidelberg) 1988.

[Törn88c] Törn, A.A., Parallel Monte Carlo with application to global optimization. In: K. Juslin and E. Silvennoinen (eds.), *Numerical Simulation of Processes, SIMS-88 Symposium*, The 30th Annual Meeting of the Scandinavian Simulation Society (SIMS), Espoo, Finland 21-22 April, 1988, VTT Symposium 84, (Technical Research Centre of Finland, Espoo) 156-165.

Technical Reports

[Back and Sere 88a] Back,R.J.R., Sere, K., *Stepwise Refinement of Parallel Algorithms.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 64, 1988. Submitted for publication.

[Back and Sere 88b] Back,R.J.R., Sere, K., *An Exercise in Deriving Parallel Algorithms: Gaussian Elimination.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 65, 1988.

[Back 88b] Back, R.J.R., *Data Refinement in the Refinement Calculus.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 68, 1988. Submitted for publication.

[**Öhman et al 88**] Öhman, G., Malén, T-E., Kuusela, P., *Numerical Fluid Flow and Heat Transfer Calculations on Multiprocessor Systems.* Report 88-3. Dept. of Chemical Engineering, Heat Engineering Laboratory, Åbo Akademi. 1988.

[**Sere 88a**] Sere, K., *Transforming Communication Topology in Distributed Algorithms.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 67, 1988. Submitted for publication.

[**Sere 88b**] Sere, K., *Deriving Action Systems for Processor Farms.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 69, 1988. Submitted for publication.

[**Törn 88d**] Törn, A., *Models of software accumulation.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 59, 1988. To appear in The Journal of Systems & Software.

[**Törn 88e**] Törn, A., *PICA - a graphical program development tool.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 60, 1988.

[**Walldén and Sere 88**] Walldén, M., Sere, K., *Design and Implementation of Full-Text Retrieval on Transputer Networks.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 63, 1988.

Ph.Lic.

[**Eriksson 88b**] Eriksson, I., *Användarens ADB-kunnande ett villkor för effektivt nyttjande av informationssystem.* (Users' EDP-knowledge - A Necessity for Efficient Use of ISs). Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1988.

[**Sere 88c**] Sere, K., *Stegvis utveckling av parallella algoritmer.* (Stepwise Refinement of Parallel Algorithms). Åbo Akademi, Dept. of Computer Science. Ph.Lic. thesis 1988.

1989

Books

[**Andersen et al 89**] Andersen, N., Kensing, F., Lassen, M., Lundin, J., Mathiassen, L., Munk-Madsen, A., Sørgaard, P., *Professional Systems Development.* Prentice Hall, Business Information Technology Series (U.K.) 1989. To appear.

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

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.

[**Caianiello et al 89a**] Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., Implementations of the C-calculus, *Connection Science*, 1 (1989), pp. 43-53.

[**Eriksson and Reijonen 89**] Eriksson, I., Reijonen,P., Training Computer-Supported Work by Simulation. To appear in *Education and Computing.* Also to appear in Proc. of the IFIP WG 3.4 Working Conference in Helsinki, July 31-August 4, 1989.

[**Eriksson et al 89**] Eriksson, I., Hellman, R., Nurminen, M., A Method for Supporting Users' Comprehensive Learning. To appear in *Education & Computing*. North-Holland. Also in: DeGross, J.I., Kriebel, C.H. (eds), Proceedings of the Eighth International Conference on Information Systems, ICIS, pp. 195-217, Pittsburgh, Pennsylvania, 1987.

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

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

[**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. (Revised version of [Walldén and Sere 88]).

Collections

[**Eriksson 89b**] 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.

Proceedings

[**Aspnäs and Back 89**] Aspnäs, M., Back, R.J.R., A Programming Environment for a Transputer-Based Multiprocessor System. To appear in *Proc. First Finnish-Hungarian Workshop on Programming Languages and Software Tools*, Szeged, Hungary, 7-11.8.1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 82, 1989.

[**Back 89a**] Back, R.J.R., Changing Data Representation in the Refinement Calculus. *Proceedings on Hawaii International Conference on System Sciences (HICSS-22)*, 3-6.1.1989, Kailua-Kona, Hawaii, (Revised version of [Back 88b]).

[**Back89b**] Back, R.J.R., Refining Atomicity in Parallel Algorithms. To appear in *Proc. Conference on Parallel Architechtures and Languages Europe*, Eindhoven, the Netherlands, June12-16, 1989. Lecture Notes in Computer Science. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 57.

[**Back and Sere 89a**] Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems. *Proc. of the Conference on Mathematics of Program Construction*, Groningen, the Netherlands, June 26-30,1989. Lecture Notes in Computer Science No 375, pp. 115-138. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 78.

[**Back and vWright 89a**] Back, R.J.R., v. Wright, J., A Lattice-theoretic basis for a specification language. *Proc. of the Conference on Mathematics of Program Construction*, Groningen, the Netherlands, June 26-30,1989. Lecture Notes in Computer Science No 375, pp. 139-156. (Short version of [Back and von Wright 89b].)

[**Caianiello et al 89b**] 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).

[Caianiello et al 89c] Caianiello, E. R., Eklund, P. E., Ventre, A. G. S., Error Propagation in C-calculus, *Proc. 2nd Italian Workshop on Parallel Architectures and Neural Nets*, Vietri sul Mare, Salerno, April 27-28, 1989.

[Caianiello et al 89d] 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 19-22, 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)*, Seattle, August 6-11, 1989, to appear.

[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 (1989), pp. 78-81.

[Eklund and Gähler 89a] Eklund, P., Gähler, W., General Structures and Fuzzy Filters, *Proc. 3rd Congress of the International Fuzzy Systems Association (IFSA)*, (invited talk), Seattle, August 6-11, 1989, to appear.

[Eklund and Gähler 89b] Eklund, P., Gähler, W., Set Functors and General Spaces, *Proc. 11th International Seminar on Fuzzy Set Theory, Applications of Category Theory to Fuzzy Subsets*, (invited talk), Linz, September 11-15, 1989, to appear.

[Eriksson 89a] Eriksson, I., Learning Process in the Context of Using ISs. IFIP WG 8.2. To appear in *Proc. Working Conference on Information Systems Development for Human Progress in Organizations* (abstract). Atlanta, Georgia, May 1987. North-Holland.

[Eriksson and Finnäs 89] Eriksson, I., Finnäs, A., Creating a Visual Simulation Model of an Inventory System. To appear in *Proc. of the 12th IRIS Conference*, Skagen, Denmark, August 13-16,1989.

[Komorowski 89] Komorowski, J., Towards Synthesis of Programs in the Partial Deduction Framework. *Proc. XIth International Joint Conference on Artificial Intelligence, Workshop on Automating Software Design*, Detroit, MI, USA, August 20-26, 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 81.

[Komorowski et al 89] Komorowski, J., Barr, C.E., and Greenes, R.A., Knowledge Modeling in the Unified Medical Language System. *Proc. Joint Scandinavian-Japanese Seminar on Information Modelling and Knowledge Bases*, Ellivuori, June 1989, Finland, University of Tampere.

[Rantala et al 89] Rantala, A., Raunio, A., Still, D-J., A Multiprocessor System for Fast Geometric Image Transformation. To appear in *Proc. 6th Scandinavian Conference on Image Analysis*, Oulu, June 19-22,1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 66, 1988.

[Shen 89a] Shen, H., A Fast Parallel Algorithm for Integer Sorting. To appear in *Proc. of Parallel Computing 89*, Leiden, The Netherlands, August 29 - September 1, 1989. North-Holland. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No 70, 1988.

[**Shen 89b**] Shen, H., Mapping Parallel Programs onto Transputer Networks. To appear in *Proc. of Australian Transputer and Occam User Group Conference*, Melbourne, Australia, July 5-6, 1989. Also published in Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 79, 1989.

[**Shen 89c**] Shen, H., Fast Path-disjoint Routing in Transputer Networks. To appear in *Proc. of First Finnish-Hungarian Worshop on Programming Languages and Software Tools*, Szeged, Hungary, August 7-11, 1989.

[**Shen 89c**] Shen, H., Self-adjusting mapping:a heuristic mapping algorithm for mapping parallel programs onto transputer networks. To appear in *Proc. 11th Occam User Group Technical Meeting*, Edinburgh, U.K., September 25-26,1989.

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

Technical Reports

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

[**Aspnäs et al 89b**] Aspnäs, M., Back, R.J.R., Malén, T-E., *The Hathi-2 Multiprocessor System.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 80.

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

[**Back and vWright 89b**] Back, R.J.R., v. Wright, J., *Duality in Specification Languages: A Latice Theoretic Approach.* Reports on Computer Science & Mathematics, Åbo Akademi, Ser. A, No. 77. Submitted for publication.

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

Working papers

[**Back 89c**] Back, R.J.R., Refinement of reactive action systems. To appear in *Proc. REX Workshop for Refinement of Distibuted Systems*, Nijmegen July 1989.

[**Back and Sere 89b**] Back, R.J.R., Sere, K., *Implementing action systems in Occam.* Manuscript 1989.

[**Back and vWright 89c**] Back, R.J.R., v. Wright, J., *On combining angels, demons and miracles in program specifications.* Manuscript 1989.

[**Back and vWright 89d**] Back, R.J.R., v. Wright, J., *Formalizing the refinement calculus in HOL.* Manuscript 1989.

[**Eriksson and Törn 89**] Eriksson, I., Törn. A., *Software quality concepts.* Dept. of Computer Science, Åbo Akademi, 1989.

[**Sere 89**] Sere, K., *Laws of Action Systems Programming.* Manuscript 1989.

[Sørgaard at al 89] Sørgaard, P., Nurminen, M., Forsman, U., *System Maintenance and Organizational Change.* Dept. of Computer Science, Åbo Akademi, 1989.

Törn 89b Törn, A., *Decision support by rapid simulation using simulation nets.* To be published in the journal Decision Support Systems.

[Törn and Žilinskas 89b] Törn, A., Žilinskas, A., *Global Optimization Algorithms in Optimal Design.* To be presented at the 14th IFIP Conference on System Modelling and Optimization, Leipzig, July 3-7, 1989.

Appendix A

Staff

Permanent positions and their deputies

Professor: Ralph-Johan Back
Deputy: Jan Komorowski

Professor: Vacant
Deputy: Aimo Törn

Associate professor (Administrative data-processing): Aimo Törn
Deputies: Pål Sørgaard (1.8.-31.12.1988), Ulla Solin (1.1.-31.7.1989)

Lecturer: Ragnar Wikman
Deputy: Paul Lindholm (1.1.-31.7.1989)

Lecturer (from 1.8.1989): Patrik Eklund

Assistant professor [1]: Vacant until 31.7.1989, from 1.8.1989 Pål Sørgaard
Deputies: Ulla Solin (1.8.-31.12.1988), Pål Sørgaard (1.1.-31.7.1989)

Assistant professor [1] (Computer science and mathematical statistics): Patrik Eklund, vacant from 1.8.1989
Deputies: Eva Söderholm (1.8.-31.12.1988), Jens Granlund (1.1.-31.7.1989)

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

Assistant: Ulla Solin
Deputies: Jukkapekka Hekanaho (1.8.-31.12.1988), Robin Rosenberg (1.1.-31.7.1989)

Assistant: Kaisa Sere, vacant from 1.8.1989
Deputies: Benita Heinonen (1.8.-31.12.1988), Thomas Långbacka (1.1.-31.7.1989)

Assistant (Administrative data-processing): Inger Eriksson
Deputies: Jens Granlund (1.8.-31.12.1988), Tony Riissanen (1.1.-31.7.1989)

Instructor: Paul Lindholm
Deputy: Sami Viitanen (1.1.-31.7.1989)

Instructor: Vacant
Deputy: Annamari Soini (1.8.1988-31.7.1989)

Laboratory Manager (from 1.1.1989): Mats Aspnäs
Deputy: John Aspnäs

Department secretary: Christel Engblom

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

Externally funded project staff

Mats Aspnäs (project researcher, leader of Millipede)

Ralph-Johan Back (project researcher, director of Millipede -, Centipede - and FINSOFT III Co-ordination projects)

Inger Eriksson (researcher, Ph.D. programme, SOLE)

Viking Högnäs (project researcher, Hathi, Centipede)

Gundel Höglund (project secretary, Millipede, Centipede, Co-ordination project)

Jan Komorowski (project researcher, Centipede)

Tor-Erik Malén (project researcher, Millipede)

Antti Raunio (project researcher, Hathi, leader of Centipede (1.4-31.12.1988)

Kaisa Sere (project researcher, leader of Centipede, Ph.D. programme)

Hong Shen (project researcher, Millipede, Centipede)

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

Undergraduates employed in projects

Peter Ahlskog (SOLE)

John Aspnäs (Hathi)

Peter Dahl (Centipede)

Jukkapekka Hekanaho (Hathi, Centipede)

Johan Lahtivuori (internal funding)

Stefan Levander (Hathi)

Thomas Långbacka (internal funding)

Mikael Norrbo (Millipede)

Yngve Nyman (Hathi, Millipede)

Mariella Ramstedt (internal funding)

Håkan Sarén (SOLE)

Dan-Johan Still (Hathi, Centipede)

Lena Ståhl (Hathi, Millipede)

Marina Walldén (Hathi)

Roger Wallin (SOLE)

Patrick Waxlax (Hathi, Centipede)

Appendix B

Course Descriptions

A.1 Languages

Language courses are given by the Language Service Centre at Åbo Akademi.

A.2 History of computer science, theory of science

Contents: The history of computer science. Theory of science with special emphasis on computer science.

A.3 Introduction to computer science

Contents: Computers, peripheral devices. Use of terminals, microcomputers, and standard software. First steps in programming.
Literature:
Goldschlager-Lister: Computer Science: A Modern Introduction. Prentice-Hall 1988.
Ford-Wiener: Modula-2 - A Software Development Approach. Wiley 1985.
Törn: Programmering - från problem till dokumentation. Studentlitteratur 1981.

A.4 Programming techniques I

Contents: Program development. Procedures. Modularisation. Testing, debugging. Programming in Modula-2. File handling.
Literature:
Ford-Wiener: Modula-2 - A Software Development Approach. Wiley 1985.
Törn: Programmering - från problem till dokumentation. Studentlitteratur 1981.

A.5 Introduction to systems development

Contents: The concept of information systems. System development from the user's perspective. First steps in project work. Description techniques.
Literature:
Flensburg: Systemutveckling med människan i centrum. Studentlitteratur 1987.
Andersen et al: Målinriktad projektstyrning. Studentlitteratur 1986.

B.1.1 Programming techniques II

Contents: Advanced program development. Modularisation. Introduction to parallel programming. Program transformations.
Literature:
Ford-Wiener: Modula-2 - A Software Development Approach, Part 2. Wiley 1985.
Wiener: Modula-2 Wizard. Wiley 1987.

B.1.2 Data structures

Contents: Data structures. Searching. Sorting. Analysis of algorithms
Literature:
Sincovec-Wiener: Data Structures Using Modula-2. Wiley 1986.

B.1.3 Programming methodology

Contents: Specifications. Developing, proving, and analysing programs. Semantics. Models and computations.
Literature:
Gries: The Science of Computer Programming. Springer 1981.
Backhouse: Program Construction and Verification. Prentice-Hall 1986.
Liskov-Guttag: Abstraction and Specification in Program Design. MIT Press 1986.
Berlioux-Bizard: Algorithms. Wiley 1986.
Wulf-Shaw-Hilfinger-Flon: Fundamental Structures of Computer Science. Addison-Wesley 1981.

B.1.4 Computer Architecture

Contents: Assembly language programming. I/O programming and interrupt handling. Assembly language programming on the Macintosh.
Literature:
Wakerly: Microcomputer Architecture and Programming. Wiley 1981.
Rosenzweig-Harrison: Programming the 680000: Macintosh Assembly Language. Hayden 1986.

B.1.5 Operating systems

Contents: The functionality of operating systems. Construction of operating systems.
Literature:
Peterson & Silberschatz: Operating System Concepts. Addison-Wesley 1985.
Deitel: An Introduction to Operating Systems. Addison-Wesley 1984.

B.1.6 Databases

Contents: Hierarchical databases. Relational databases. Network databases. Development and construction of databases.
Literature:
Korth-Silberschatz: Database System Concepts. Addison-Wesley 1986.
Howe: Data Analysis for Database Design. Arnold 1983.
Sundgren: Databaser och datamodeller. Studentlitteratur 1981.
Bubenko-Lindencrona: Konseptuell modellering - Informationsanalys. Studentlitteratur.

B.1.7 Systems Development

Contents: Jackson systems development. Organization theory. Prototyping.
Literature:
M.Jackson: System Development. Prentice-Hall 1983.
J.Galbraith: Designing Complex Organizations. Addison-Wesley 1973.

B.1.8 Seminar in computer science

Contents: To compose and present a seminar on a specific subject. To act as an opponent at another seminar.
Literature: Depending on the subject

B.2 Mathematics

Education in Mathematics is handled by the Department of Mathematics.

B.3 Statistics

Education in Statistics is handled by the Department of Statistics.

B.4 Business administration

Education in Business Administration is handled by the Faculty of Economics and Political Science.

Appendix C

Departmental Reports

The series Reports on Computer Science & Mathematics is, as the name indicates, published jointly with the Mathematics Department. Below only the reports in computer science are listed. Series A is for scientific reports, intended to be published also elsewhere, whereas Series B is for educational material and manuals.

Series A

1984

33. Back, R.J.R., A computational interpretation of truth logic.

34. Back, R.J.R., Kurki-Suonio, R., Co-operation in distributed systems using symmetric multi-process handshaking.

38. Back, R.J.R., Eklund, P., Kurki-Suonio, R., A fair and efficient implementation of CSP with output guards.

39. Eklund, P., Synchronizing multiple processes in common handshakes.

1985

42. Back, R.J.R., Hartikainen, E., Kurki-Suonio, R., Multi-process handshaking on broadcasting networks.

43. Solin, U., Aspnäs, J., Levander, S., Raunio, A., A Distributed Election Algorithm Allowing Process Failures.

44. Aspnäs, M., Eklund, P., Värnström, A., Elections in a uni-directional ring with a variable number of processes.

47. Wikman, R., Data compression: some notes on the effects of combining two compression methods.

1986

48. Back, R.J.R., Formal methods in program construction. Final report on a Research Project. Sponsored by Academy of Finland 1982-85.

50. Solin, U., Parallel Algorithm Animation.

1987

53. Sere, K., Stepwise removal of virtual channels in distributed algorithms: A case study.

54. Back, R.J.R., A Calculus for Refinements for Program Derivations.

55. Back, R.J.R., Procedural Abstraction in the Refinement Calculus. 1987.

56. Back, R.J.R., Kurki-Suonio, R., Distributed Co-operation with Action Systems.

1988

57. Back, R.J.R., Refining Atomicity in Parallel Algorithms.

58. Back, R.J.R., Kurki-Suonio, R., Decentralization of Process Nets with Centralized Control.

59. Törn, A., Models of software accumulation.

60. Törn, A., Parallel Monte Carlo with Application to Global Optimization.

61. Törn, A., PICA - a graphical program development tool.

63. Walldén, M., Sere, K., Design and Implementation of Full-Text Retrieval on Transputer Networks.

64. Back,R.J.R., Sere, K., Stepwise Refinement of Parallel Algorithms.

65. Back,R.J.R., Sere, K., An Exercise in Deriving Parallel Algorithms: Gaussian Elimination.

66. Rantala, A., Raunio, A., Still, D-J., A Multiprocessor System for Fast Geometric Image Transformation.

67. Sere, K., Transforming Communication Topology in Distributed Algoritms.

68. Back, R.J.R., Data Refinement in the Refinement Calculus.

69. Sere, K., Deriving Action Systems for Processor Farms.

70. Shen, H., A Fast Parallel Algorithm for Integer Sorting.

72. Sebastian, H-J., Brock, T., Bönewitz, S., Modelling and a First Realization of a Tool for Configuration Expert Systems.

1989

75. Aspnäs, M., Back, R.J.R., Kurki-Suonio, R., Efficient Implementation of Multi-process Handshaking on Broadcasting Networks.

76. Ståhl, L., Back, R.J.R., An Implementation of Multi-process Handshaking on Transputer networks.

77. Back, R.J.R., v. Wright, J., A Duality in specification languages: A Lattice-theoretic basis.

78. Back, R.J.R., Sere, K., Stepwise Refinement of Action Systems.

79. Shen, H., Mapping Parallel Programs onto Transputer Networks.

80. Aspnäs, M., Back, R.J.R., Malén, T-E., The Hathi-2 Multi-processor System.

81. Komorowski, J., Synthesis of Programs in the Framework of Partial Deduction.

82. Aspnäs, M., Back, R.J.R., A Programming Environment for a Transputer-Based Multi-processor System.

Series B

1987

5. Walldén, M., A Case Study: Performance of a Distributed Algorithm.

1989

6. Aspnäs, M., Malén, T-E., Hathi-2 Users Guide, version 1.0.

Series A

1988

67. *Kaisa Sere*, Transforming Communication Topology in Distributed Algorithms.
68. *R.J.R. Back*, Data Refinement in the Refinement Calculus.
69. *Kaisa Sere*, Deriving Action Systems for Processor Farms.
70. *Hong Shen*, A Fast Parallel Algorithm for Integer Sorting.
71. *Göran Högnäs*, A Note on the Semigroup of Analytic Mappings with a Common Fixed Point.
72. *Hans-Jürgen Sebastian, Tilo Brock, Stephan Bönewitz*, Modelling and a First Realization of a Tool for Configuration Expert Systems.

1989

73. *Aimo Törn*, An Efficient Procedure for Determining the Enabled Set.
74. *Göran Högnäs*, Nonlinear Autoregressive Processes.
75. *Mats Aspnäs, R.J.R. Back, Reino Kurki-Suonio*, Efficient Implementation of Multi-process Handshaking on Broadcasting Networks.
76. *Lena Ståhl, R.J.R. Back*, An Implementation of Multiprocess Handshaking on Transputer Networks.
77. *R.J.R. Back, J. von Wright*, Duality in Specification Languages: a Lattice-theoretical Approach.
78. *R.J.R. Back, Kaisa Sere*, Stepwise Refinement of Action Systems.
79. *Hong Shen*, Mapping Parallel Programs onto Transputer Networks.
80. *Mats Aspnäs, R.J.R. Back, Tor-Erik Malén*, The Hathi-2 Multiprocessor System.
81. *Jan Komorowski*, Synthesis of Programs in the Framework of Partial Deduction.
82. *Mats Aspnäs, R.J.R. Back*, A Programming Environment for a Transputer-Based Multiprocessor System.
83. *Hong Shen*, Fast Path-disjoint Routing in Transputer Networks.
84. *J. von Wright*, Stepwise Derivation of a Parallel Matrix Multiplication Algorithm.
85. *R.J.R. Back, J. von Wright*, Refinement Concepts Formalized in Higher Order Logic.
86. *R.J.R. Back, J. von Wright*, Combining Angels, Demons and Miracles in Program Specifications.
87. *Hong Shen*, Self-adjusting Mapping: A Heuristic Mapping Algorithm for Mapping Parallel Programs onto Transputer Networks.
88. *Dieter Ehrenberg, Patrik Eklund, Mario Fedrizzi, Aldo G.S. Ventre*, Consensus in Distributed Soft Environments.
89. *Inger Eriksson, Aimo Törn*, A Covering of IS Quality Concepts.

Series B

1989

6. *Mats Aspnäs, Tor-Erik Malén*, Hathi-2 Users Guide, version 1.0.
7. *R.J.R. Back, A. Törn*, Åbo Akademi, Department of Computer Science. 5-year Report 1984-1989.