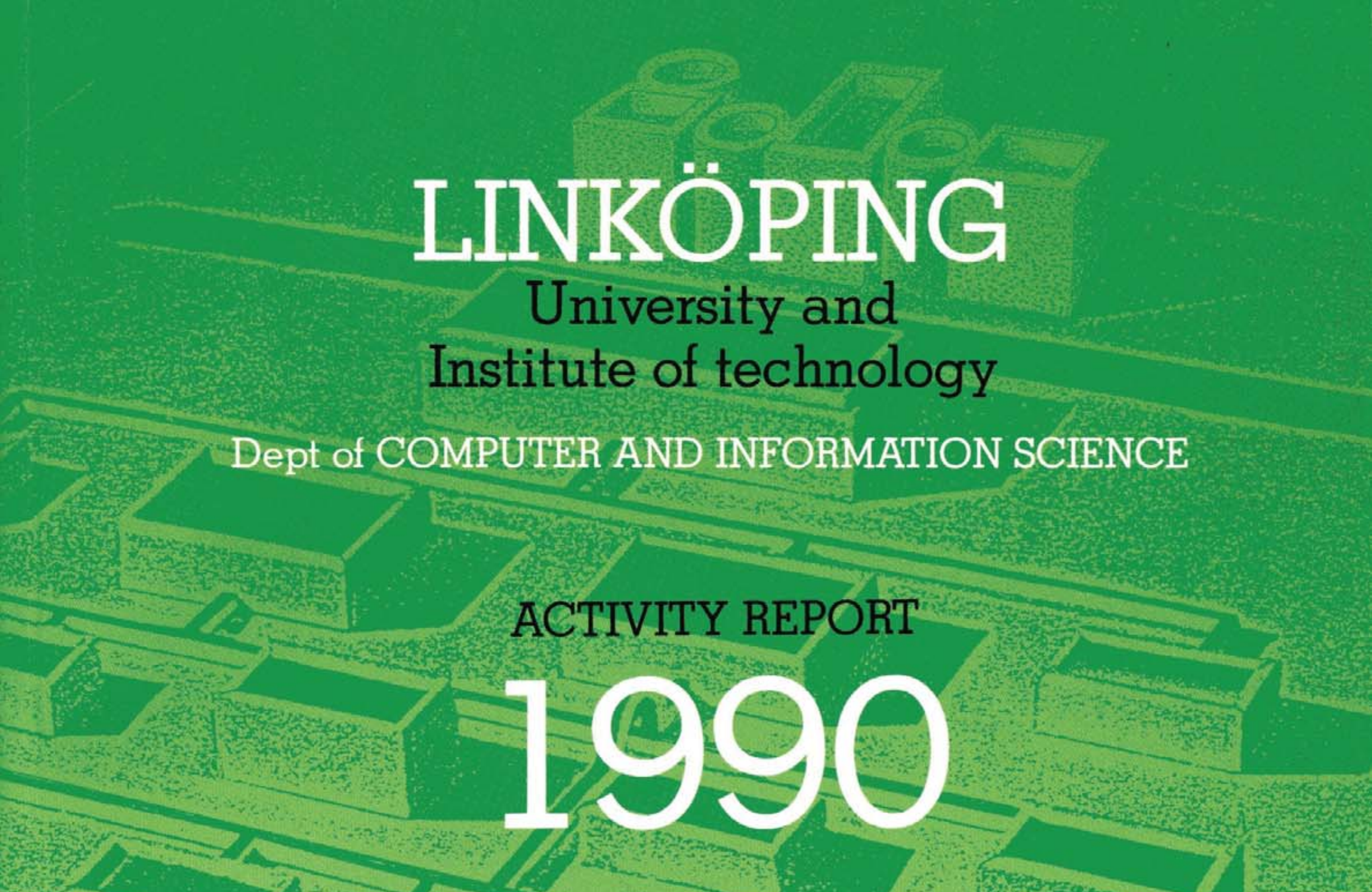
LINKÖPING

University and
Institute of technology

Dept of COMPUTER AND INFORMATION SCIENCE

ACTIVITY REPORT

1990



Department of Computer and Information Science

Activity Report 1990

LINKÖPING UNIVERSITY
INSTITUTE OF TECHNOLOGY

ISBN 91-7870-804-4
ISSN 0281-4250
Printed in Sweden by LJ Foto & Montage AB, Linköping/Norrköpings Tryckeri, Norrköping 1991

Acknowledgements

The editorial board of this report gratefully acknowledges the financial support provided by the following high-tech software companies engaged in our industrial cooperation programme:

ENATOR KUNSKAPSSYSTEM TELESOFT SoftLab

Back row: Douglas Busch, Lisbeth Linge, Birgitta Franzén
Front row: Arne Fäldt, Yngve Wassberg, Sture Hägglund, Lillemor Wallgren (chief editor), Tommy Olsson, Haraldur Bjarnasson.

The DTP job was done with FrameMaker on Sun.

Contents

Appendices

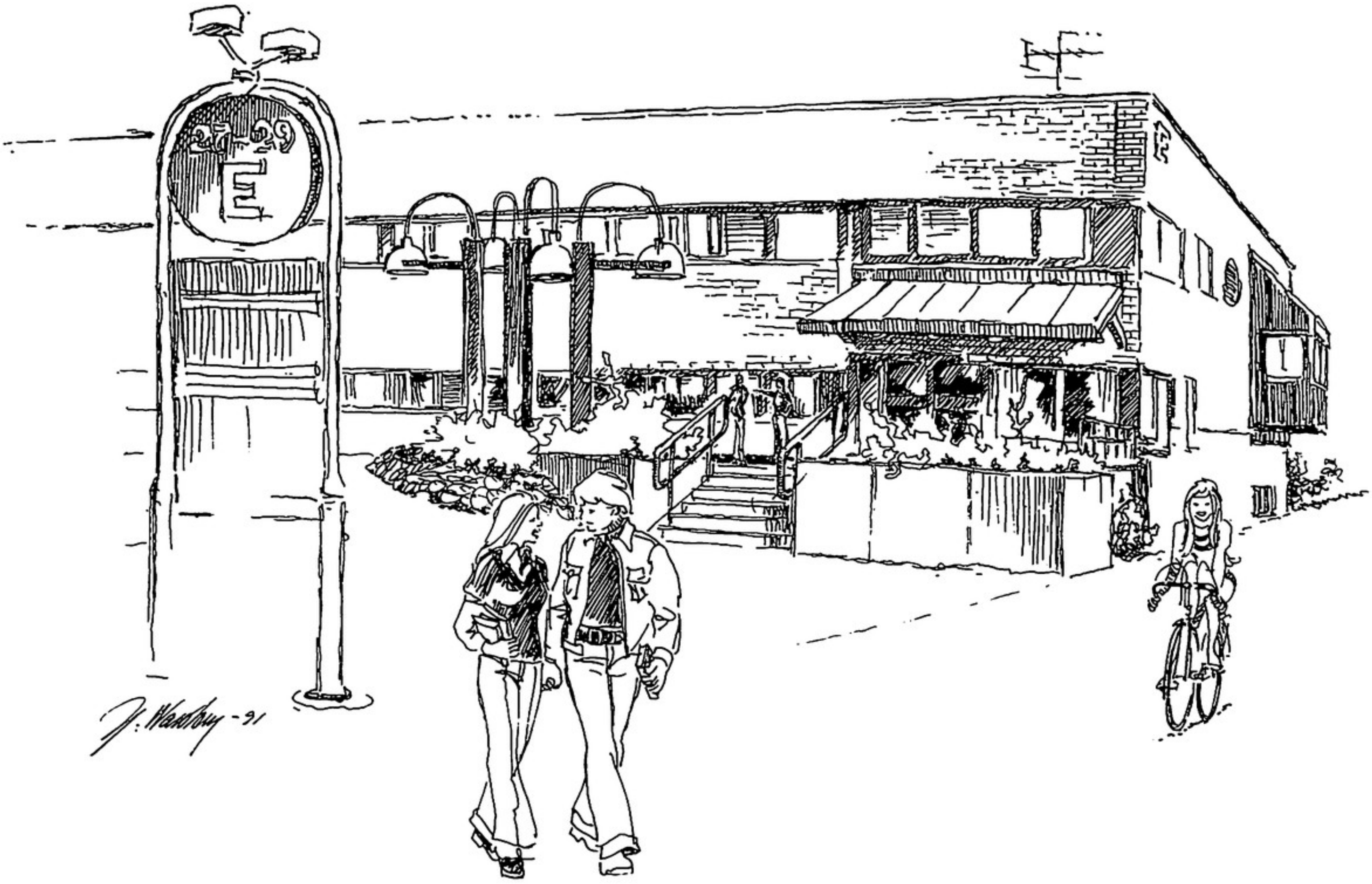


Figure 0-1. *Building E: Main entrance.*

1 Introduction and overview

The Department of Computer and Information Science (IDA) at Linköping University has now existed for seven years. During this period several educational programmes with a large number of courses have been developed for undergraduate education, new research laboratories have been established, and an effective infrastructure in terms of computer systems, system support staff, and organization in general has been created. Today IDA has about 150 employees (about 20 with a PhD), an annual budget of 52 MSEK ($9 million), about 100 undergraduate courses for 1200 students, and about 60 full-time graduate students. About 35% of the money and effort are for undergraduate education, 5% for continuing education and technology transfer activities, and 60% for research.

The research program has been designed to cover areas of strategic importance, both for our responsibilities in undergraduate education and for the needs in the society as envisaged by our funders. Currently there are twelve research laboratories. A lab is characterized by its long term commitment to development and maintenance of knowledge within a defined area, and by its long term responsibility for graduate students.

The current laboratories and groups are:

ACTLAB	(Fjällström)	for complexity of algorithms.
ASLAB	(Hägglund)	for application systems.
CADLAB	(Kuchcinski)	for computer-aided design of digital systems.
CAELAB	(Törne)	for computer assistance in engineering.
EIS	(Rapp)	for economic information systems.
IISLAB	(Padgham)	for intelligent information systems.
LIBLAB	(Hjerppe)	for library and information sciences.
LOGPRO	(Maluszynski)	for logic programming.
MDA	(Timpka)	for people, computers and work.
NLPLAB	(Ahrenberg)	for natural language processing.
PELAB	(Fritzson)	for programming environments.
RKLLAB	(Sandewall)	for representation of knowledge in logic.
IST Project	(Lennartsson)	for Industrial Software Technology.

The most significant recent change in the research organization is that from September 1990, the subject area Economic Information Systems with Professor Birger Rapp is included in our department. This group studies the use of information technology in organisations and material and production control with computer support.

The former LINCKS group has now established itself as a Laboratory for Intelligent Information Systems, IISLAB. Another group, MDA, for interdisciplinary studies of people, computers and work has also developed into a permanent research laboratory.

Two positions as full professor are currently open in the department, one in computer systems and one in engineering databases. Holders of these positions are expected to be installed during 1991.

Within the Center for Industrial Information Technology, CENIIT, our department has expanded its activities. In the laboratory for Computer Assistance in Engineering CENIIT projects are carried out regarding computer support for automation and engineering databases. A special project group for industrial software technology has also been created. Other projects concern computational geometry and natural language processing.

External cooperation has been intensified during recent years. IDA has taken a leading role in the PROART subproject for artificial intelligence within the large European Prometheus (future road traffic systems), where Volvo and Saab are the Swedish industrial participants. The Logic Programming Laboratory participate in an ESPRIT cooperation project, and several groups are engaged in COST activities. Within the national information technology programme (IT4), IDA has also been undertaking applied research in AI and human-computer interaction together with FOA (The Defense Research Institute). In order to create efficient forms for cooperation with industry and the public sector in applied projects, without undue interference with basic research goals, special R&D consortia have been created. For instance, LAIC in the area of artificial intelligence applications and FOKUS in the area of knowledge systems and human-computer interaction.

A special Knowledge Transfer Programme has been in operation for industrial liaison since 1984. Active cooperation exists with among others Pharmacia, Volvo, Saab-Scania and TeleSoft. Commissioned educational activities are also increasing, both directly for industrial companies and programmes delivered in cooperation with local universities and colleges. For instance, the 9 month knowledge engineering study programme was delivered in Västerås 1987–88 and in Gothenburg 1989–90.

The rest of this introductory chapter reviews the scope of our research programme and explains the organizational background for activities in the department. In later chapters the research labs and groups are presented, together with a short overview of our undergraduate programs. Appendices present more details on our administrative organization, educational programmes, and computer resources. A list of recent publications appears at the very end.

This annual research report is intended both for our colleagues internationally, and for Swedish readers in industry as well as other universities. When it comes to the sections about organization and about undergraduate teaching, different readers may have different frames of reference, and maybe also different interests. If a reader finds some parts of the following text redundant, then maybe that is a result of our attempts to cater to all readers at once.

1.1 IDA's organization in general

IDA's general organization is described in more detail in Appendix A. The department is led by a Department Board (institutionsstyrelse). From July 1, 1990, Anders Haraldsson succeded Bengt Lennartsson as Department Chairman ("prefekt"). Vice Chairman is Erik Sandewall.

The two main areas of activity are reflected in two subordinate committees, where the leadership also changed from the academic year 1990/91:

- the Undergraduate Teaching Committee, whose previous Chairman Anders Haraldsson has now been succeded by Bengt Lennartsson;
- the Research Committee, whose previous Chairman Erik Sandewall has been succeded by Sture Hägglund.

Research Committee members are appointed by the Department Board and the committee equals approximately the set of laboratory leaders. It is responsible for all aspects of the department's graduate education programs and research.

The organizational groups within the department are:

- the research laboratories, headed by the laboratory leaders;
- a number of undergraduate teaching groups, responsible for the teaching in both the School of Engineering ("tekniska högskolan"), and in the School of Arts and Sciences ("filosofiska fakulteten"). The teaching groups are each headed by a director of studies ("studierektor"). The teaching groups report to the Undergraduate Teaching Committee;
- a Technical Support and Service Group (TUS), which is headed jointly by Anders Aleryd and Mats S. Andersson and reports directly to the Department Board.

The department's resources are almost consistently measured in monetary units, kronor, and not as e.g. "positions" or "slots" for teachers. For example, the School of Engineering buys a number of courses from the department, for a price that is set in kronor. The "studierektor" uses the money partly for paying people in his own teaching group, and partly for subcontracting research labs to do some of the courses. The laboratory leaders see a number of distinct sources of income, such as sub-contracted courses, research grants, and industry cooperation, and must make ends meet.

Through this organization, we try to decentralize responsibilities within the department with a minimum of bureaucracy, and without sacrificing the advantages of joint strategical planning and continuous synergy effects between the different parts of the department. The organizational and economic structure defines a small set of "rules of the game", and the task of the laboratory leaders and laboratories is to maximize the lab's performance with respect to its research programme and graduate education responsibilities, within the constraints of the rules.

1.2 Research background

Our research is located in the University's School of Engineering (Tekniska högskolan i Linköping). Knowledge areas which are significant for Swedish industry at the end of this century, should have high priority for us.

The scope of IDA's interest is partly defined by the natural borderlines to other departments in the university, notably:

Electrical engineering
Mechanical engineering
Mathematics
Physics
Business administration
Communication studies
Technology and social change

(The last two of these are in the 'Themes' research organization which has an emphasis on social sciences).

It is typical for successful research groups in important areas that permanent university funding tends to lag behind. The situation for IDA illustrates this to a rather extreme degree. Only about 35% of the resources for research and graduate education comes from internal university funds.

Our main source of research grants is the Swedish Board of Technical Development (STU). Fortunately enough, STU has assumed a long-term responsibility for funding research in information technology, according to quality criteria that are commonly accepted in the international research community. In particular they promote the establishment and growth of "centers of excellence" in selected areas.

However, it is also important for our main sponsor that research results should be transferred to applications in industry, commercial users of computer systems, public administration, or in other areas of research outside computer science. These goals are sometimes competing, and possibly even contradictory. In IDA we have tried to balance our efforts so that both the basic research goal's and the applied goal's should be achieved reasonably well. We also recognize the importance of continuous interaction between basic and applied research in our field.

The research direction of our department is naturally determined by the roots and the traditions that it has emerged from. Our research profile has evolved from early Swedish efforts in the following areas:

artificial intelligence
programming environments
computer architectures
administrative data processing, data base and office systems

These areas still represent a large portion of the department's research, but they have also been complemented with research in areas such as:

mathematical logic for knowledge representation
logic programming

complexity theory
library science
information systems
human-computer interaction
industrial information technology
economic information systems

We do not wish to be a single-issue department, but at the same time we can not afford to spread out over all possible parts of computer science. The present research profile, as realized by the laboratories and groups summarized below and described in more detail in subsequent chapters, attempts to make a reasonable trade-off between concentration and breadth.

For a long time the Swedish Board for Technical Development, STU, has been a major sponsor for our research. STU has funded programmes for research and knowledge development in information processing since 1980. These programmes have been very important for strengthening computer science research in Swedish universities, both because the total amount of funding increased, and because it provided fairly stable funding during a lengthy period of time.

A special 3-year programme with joint activities between our university and the Defense Research Establishment was completed in 1990. IDA's contribution to this programme was mainly in the artificial intelligence (AI) area, with an emphasis on autonomous vehicles and mission planning. Another substantial effort in this area is the PROART project, which is the AI part of the Prometheus Eureka project (future road traffic systems). PROART is coordinated by Erik Sandewall and part of the research is done in our department.

1.3 Research organization

There are presently twelve research groups and laboratories in the department. Each laboratory consists of one or a few graduated (Ph.D.) researchers, five to ten (typically) graduate students, and some lab-specific support staff. From the department's point of view, the laboratories are the units which perform research projects, teach graduate courses, and are responsible for finding their own funding. From the graduate student's point of view, the laboratory is his or her organizational unit. The thesis project is done in one's own laboratory, but the graduate student must take courses across the range of all the laboratories.

The research programme and the graduate study programme are coordinated by the Research Committee. The committee supervises the performance of the laboratories and is responsible for appointing lab leaders, for admitting graduate students and for initiating appropriate changes in the lab structure when needed.

The laboratory system is an intermediate form between the flat university department and the formally structured one. In the flat department there is in principle no organization, just a number of professors each of whom is the advisor for a number of graduate students. The laboratory structure encourages, and makes visible those cases where several professors /researchers /advisors work jointly with their research and their students. In particular, faculty members who do a lot of work for undergraduate teaching find it convenient to be a member, but not a leader of a research laboratory. Also, a visiting scholar would be a member of an existing laboratory and would not form a new one.

The formally structured department is the one where the academic positions (several professorship levels, lecturer, etc.) define the hierarchical structure of the department. This has often been the traditional organization in Swedish universities. The laboratory structure at IDA is more uniform. It is also easier to change, since the department's decisions about changing laboratories (adding, deleting, splitting, or merging them) can be taken according to the needs of the research activities. The creation of a senior position does not automatically imply the creation of an organizational unit, nor is it automatic the other way around.

Within our department, we presently find it more important to strengthen the existing laboratories, before starting new ones. The ambition is to give special priority to areas of strategic importance where we believe that we can attain internationally competitive research of high quality, while observing the possibilities for coordination with other research in Sweden.

The recruiting situation is relatively good, both for faculty and for graduate students, and funding (in addition to the lack of office space) is therefore the primary constraint in most areas of our activities.

1.3.1 Research laboratories and groups

A short review of research areas covered in IDA is given below. A more detailed account is given in subsequent chapters in this report.

ACTLAB – Laboratory for Complexity of Algorithms

Per-Olof Fjällström

which is concerned with the design and analysis of efficient algorithms (sequential and parallel) and data structures for combinatorial and geometric problems arising in computer science and the study of the inherent complexity of these problems in simple models of computation. One application area, studied in the context of CENIIT, is efficient algorithms for three-dimensional geometrical problems and the planning of movements.

ASLAB – Application Systems Laboratory

Sture Hägglund

which is oriented towards the study of knowledge-based approaches to software development, including certain aspects of human-computer interaction. Major projects in the lab concern engineering environments for generic knowledge systems and expert critiquing approaches. Joint projects involve cooperation with industry in the knowledge transfer programme, and with several other research groups.

CADLAB – Laboratory for Computer-Aided Design of Digital Systems

Krzysztof Kuchcinski

which concentrates its research activities on computer-aided synthesis and verification of digital systems, especially those involving very large-scale integrated circuits (VLSI). The major concern is with the behavioural and structural aspects of digital systems specification, design, simulation, optimization, partitioning, synthesis and formal verification methods.

CAELAB – Laboratory for Computer Assistance in Engineering

Anders Törne

which is concerned with information technology to be used in industrial processes and products, in particular computer support for automation and engineering databases. This lab has its major funding from CENIIT (the Center for Industrial Information Technology) and STU. It is heavily engaged in cooperative efforts with for instance the mechanical engineering department. Close links to ASLAB are also maintained.

EIS – Economic Information Systems

Birger Rapp

which covers communication of information from people to/from systems or between people and the design of information systems supporting this communication. Research projects concern information resource management, business control and decision support systems. Research and graduate education is also coordinated with corresponding activities in the administrative data processing group.

IISLAB – Laboratory for Intelligent Information Systems

Lin Padgham

which studies theory and methods for distributed information management, including object-orientation and inheritance strategies. In a major implementation project, an object-oriented database, with a distributed architecture, support for history maintenance and parallel editing, has been developed.

LIBLAB – Laboratory for Library and Information Science

Roland Hjerppe

which studies methods for access to documents and the information contained in the documents, concentrating on catalogues and bibliographic representations, and on the human factors of library use. Recently new technologies, like hypermedia, and new methods of scientific communication and knowledge organization have become of primary interest to LIBLAB.

LOGPRO – Laboratory for Logic Programming

Jan Maluszynski

which has its research concentrated on foundations of logic programming, relations to other programming paradigms and methodology. The major current projects are concerned with “Logic programming and external procedures” and with “Systematic design of abstract machines through partial evaluation”.

MDA – People, Computers and Work

Toomas Timpka

conducts research into *information system development and use in working life* from the points of view of computer science, psychology, and social organization of work development. Within the MDA-group, activities at the Department of Computer and Information Science and the Medical Faculty have been coordinated to develop and evaluate experimental information systems.

NLPLAB – Natural Language Processing Laboratory

Lars Ahrenberg

which conducts research related to the development and use of natural language interfaces to computer software. The interests of the lab covers most aspects of natural language processing and computational linguistics, with theoretical research interests primarily in parsing and interpretation, in knowledge representation for NL understanding and in the characteristics of man-machine NL interaction.

PELAB – Programming Environments Laboratory

Peter Fritzson

which works with design of tools for software development, specific functions in such tools and theoretical aspects of programs under construction. Current projects are focused on two areas, namely on development of an architecture for a programming environment supporting design, implementation and maintenance of distributed systems, and on development and evaluation of tools for meta-level language support in a multi-level software architecture.

RKLLAB – Laboratory for Representation of Knowledge in Logic

Erik Sandewall

which covers issues and techniques such as non-monotonic logic, probabilistic logic, modal logic and truth maintenance algorithms and systems. Research in the lab combines fundamental theory development with practical work in applied projects. An important theme is the study of *plan-guided systems*, with applications for instance in mission planning for autonomous vehicles. External cooperation includes a significant role in the AI part of the European Prometheus (future road traffic) project.

IST Project – Industrial Software Technology Project

Bengt Lennartsson

which is a project within CENIIT, further described in chapter 14.3.

1.3.2 CENIIT – The Center for Industrial Information Technology

From July 1988 permanent research resources were allocated to Linköping University to create a center for research in industrial information technology. The purpose of the center's research is to improve the possibilities for utilizing advanced information technology in industrial processes and products. Researchers from different areas and departments will work together in projects within this center. Important areas are computer-aided engineering, computers in manufacturing, robotics, process control, human-computer systems and software engineering

The following areas are currently pursued as our department's contribution to CENIIT:

- *Geometrical algorithms,* (Per-Olof Fjällström)
- *Engineering databases,* (Area coordinator: Sture Hägglund)
- *Computer support for automation,* (Anders Törne)
- *Industrial software technology,* (Bengt Lennartsson)
- *Conceptual text representation for generation and translation,* (Lars Ahrenberg)

CENIIT projects in IDA are described in greater detail in chapter 14 of this report.

1.4 Educational programmes

Industry representatives often point out that teaching the next generation of engineers, "knowledge engineers" and systems analysts is the most important knowledge transfer activity for a university. For IDA, it accounts for roughly 35% of the total budget, whereas knowledge transfer directly to industry accounts for about 5%, and research accounts for the other 60%.

In addition to the review below, our undergraduate and graduate study programmes are described in chapter 15 of this report, with additional details given in Appendix B and Appendix C.

1.4.1 Undergraduate study programmes

Linköping is today the only university in Sweden which offers the three main 3–4.5 years undergraduate study programs in the area of computer science and systems analysis. This reflects Linköping's leading position when it comes to initiating and organizing new study programmes and its exceptionally good situation in terms of faculty. Other educational activities are also increasing, such as a continuing education programme in computer science for Swedish industry.

The Computer Science and Technology programme (Datateknik), which started 1975, is a 4.5 year programme with an annual admittance of 120 students leading to a Masters degree in Engineering (civilingenjörsexamen). This study programme has an equal emphasis on software and hardware related courses. During the last year several specializations are offered. One such specialization of special importance is in software engineering, where students do assignments in development and maintenance of large software systems in an industrial environment. Important issues in addition to software methodologies and programming support are project management, quality assurance, economic and legal aspects.

The computer science programme ("Datavetenskapliga linjen") started in 1982. It is a four-year programme focusing on software systems and theoretical computer science, leading to a Master of Science degree. The number of students accepted annually is 30. This programme differs from the curricula leading to engineering degrees, in that it contains more relevant humanities, such as psychology and linguistics, and less physics-related subjects. Discrete mathematics, logics and AI-related subjects, with courses in AI programming, knowledge representation, expert systems and in natural language processing, also play a significant role in this curriculum.

IDA also contributes to other study programmes with a computer science profile. Thus, for instance the mechanical engineering programme has been extended with a specialization that combines mechanical and computer engineering. We believe that especially research in artificial intelligence with application in robotics will be significant within this specialization.

The School of Humanities and Sciences has offered since 1977 a three-year programme in Systems Analysis (Computer Science and Business Administration). This programme aims at professional activities of design, evaluation and implementation of computer-based information systems. Consequently, systems analysis dominates the programme. Nevertheless great importance has been attached to other subjects in order to give the programme the necessary breadth and also to ensure that the students will become aware of the complexity of the community where computers can be used. IDA is responsible for the major part of this curriculum.

1.4.2 Graduate education

The department has given high priority to the task of organizing a comprehensive graduate study programme, with a coordinated set of courses offered to all graduate students. Support for thesis work is provided by the research groups and each student is assigned a supervisor committee consisting of one main and two assisting advisors.

It will naturally take a number of years before improvements in the educational activities show up in the form of increased production of PhDs. The current rate of degrees awarded is 3–4 PhDs and 8–9 Licentiate degrees per year. It is expected that, given the current size of the graduate student population, the number of PhDs awarded will increase significantly during the coming years, while the number of licentiates will presumably stabilize at the current level.

Information about recent graduate education activities is given in Appendix B.

1.4.3 Continuing education

For continuing education activities the most prominent part is suitably adapted academic courses given for people from industry. Recent continuing education activities include programmes in Process Programming and Operating Systems for Ellemtel and a Programming Methodology Block (including Data Structures, Computer Architectures, Process Programming and a project, etc.) for Ericsson as well as a number of Ada courses given for local companies in Linköping.

A special 28 week *knowledge engineering training programme*, which covers the theoretical basis in the area of AI/expert systems, has been offered since 1986. The programme is also offered outside Linköping, through the cooperation with local universities and colleges. It was delivered in Västerås during 1988 with participants mainly from ASEA Brown Boveri, and in Gothenburg 1989–90, with participants from Volvo and Chalmers. A next offering is planned for 1991 in the Stockholm/ Västerås area.

The department is also arranging shorter courses and seminars for the purpose of technology transfer. A series of software tutorials, SOFT, has been in operation for a number of years. The most recent tutorial, SOFT-6 was organized in the spring 1990 on the subject *Building Quality Software*.

1.5 Technology Transfer Activities

A research department produces and exchanges new knowledge. In order to flourish, it must itself produce new results, and also participate in the international “barter trade” for research results. The useful outcome of those activities, from the point of view of the taxpayers and the sponsoring agencies, is when the accumulated knowledge is transferred to practical use. We use the term “knowledge transfer activities” collectively for the various ways of transferring accumulated research knowledge.

An important task for a university department is to disseminate knowledge into the surrounding society, public sector, trade and industry. This means that the research organization should serve as a source of competence, bringing together and distributing not only its own results but also importing and collecting state-of-the-art information from the international research community.

The main channel for effectuating this task is obviously the knowledge transfer that results when people trained in undergraduate and graduate study programs enter working positions outside the university. Less efficient but equally important is the spreading of results through written reports and oral presentations by active researchers. A third way of achieving technology transfer is through cooperative work in joint projects.

In our department we have actively pursued these strategies, e.g. by issuing a special series of reports summarizing important results in central research areas specifically directed towards industry, by arranging and participating in tutorial conferences, (recent tutorials include *Artificial Intelligence, Software Development Environments, Prolog Programming Environments, AI and Expert Systems, Advanced Human-Computer Interaction, Software Engineering*), by developing continuing education programs for industry and by direct consultations and cooperation in applied projects.

However, we felt that in many cases these methods were too slow or too restricted to achieve an effective technology transfer in rapidly developing areas of strategic importance for industry. Thus we initiated some years ago a discussion with industry about this problem which led to the decision to start a special *knowledge transfer program*, KTP (KÖP), in 1984.

The goal of this program is to ’inject’ competence derived from research into the existing industrial organization. The method is that typically two persons, located on a middle level in the organization, come to our department for a period of one or a few years, in order to learn new technology, and return to their organization after that time. The participating company also pays a yearly contribution that helps pay for researchers (particularly guest lecturers) and equipment.

Since the start in 1984 a number of companies have participated in the knowledge transfer programme: SE-Banken, Ericsson, ASEA, Alfa-Laval, Philips and Sandvik Coromant, Volvo Cars, Volvo Data, Pharmacia LKB Biotechnology, Ericsson Telecom and TeleSoft (formerly TeleLogic Programsystem).

The programme has a budget of its own, but participating individuals are associated with one of the research laboratories in the department. Each participant is assigned a faculty supervisor and one graduate student, who is working closely together with the participant. Courses and other training activities are organized and coordinated at the department level. For companies where we already have established a foundation of common knowledge and understanding, other forms of project-based cooperation are also utilized.

IDA is strongly committed to the conviction that effective means of transferring research results to industry and knowledge of problems and needs in the other direction do not demand the creation of independent research institutes. Much is to be gained if basic research, educational activities, applied research and advanced development projects can be carried out within the university organization. In order to avoid problems of conflicting interests, we try to have a flexible organization which include the possibility of forming special R&D consortia, where research and development tasks respectively are clearly defined for the participants. Such consortia have for instance been formed in the area of applied AI, LAIC, and in human-computer interaction and knowledge-based systems, FOKUS.

IDA's policy is to accept industry contracts for knowledge transfer, i.e. for work where the customer wants (his employees) to acquire knowledge in some area, but to avoid direct consulting jobs or other projects where the customer wants specific software, hardware, or designs to be delivered. In many cases we can instead refer to existing spin-off companies, and we may also encourage IDA employees to form new spin-off companies in order to catch an opportunity.

The significance of university spin-off companies for industrial growth is well known. The rapidly growing Science Park contains a number of such companies, employing software specialists educated in IDA. The intensive communication with the many developing high-tech software companies around the university is a vitalizing force for the department. Examples of companies that were created as continuations of previous research are for instance, Epitec AB (now Enator Kunskapssystem AB), in the area of knowledge-based systems and Softlab AB, working in the areas of compiler design, certain aspects of database technology and advanced tools for software development.

1.6 International Cooperation

In computer science, like in most other disciplines, the most important international cooperation is informal. It takes place through personal contacts and visits, and at international conferences.

In addition, IDA or specific labs within IDA also participate in organized international projects, for instance:

- the PROMETHEUS project, which is part of the European EUREKA programme. Prometheus is concerned with future traffic and automobile systems and is a joint effort by the European car manufacturers. Erik Sandewall is the European coordinator for the AI subproject, PROART.
- ESPRIT projects, where for instance the Laboratory for Logic Programming participate in the basic research action 3020: INTEGRATION.
- COST projects. Current involvement in for instance in the CoTech project, studying computer-supported cooperative work. A previous effort was a European project on Computer Architectures for A.I., together with the Free University of Brussels (Belgium), the University of Rome ’La Sapienza’ (Italy), and Delphi S.A. in Pisa (Italy).

IDA also has extensive contacts with university and industrial research laboratories, primarily in the USA but also in Europe and to a certain degree also in the Far East. For instance, IDA regularly sends students to the Xerox Palo Alto Research Center in the USA for periods of 3–6 months.

1.7 Research Facilities

The main part of IDA is situated in the university’s E building. About one third of the department’s personnel was previously located to the adjacent B building, but moved in the autumn of 1990 to a new temporary building close to the E building. The premises are in general well planned and very nicely designed, but unfortunately at present heavily overcrowded.

With respect to equipment, IDA has recently gone through a transition phase where the DEC-system 2060s, which was used as the backbone of computing services for research and undergraduate education, now is replaced by about 200 networked SUN workstations. Further, there are about 24 Xerox AI workstations and also a large number of Macintoshes, PCs and some VaxStations in the department, as well as some special-purpose computers.

More details on equipment are given in Appendix D211.

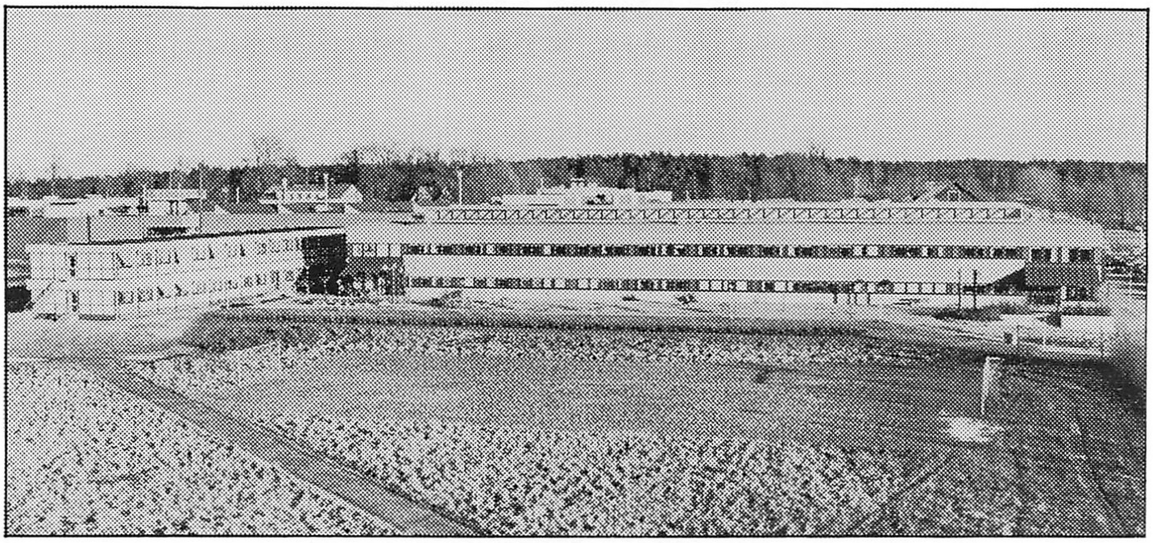
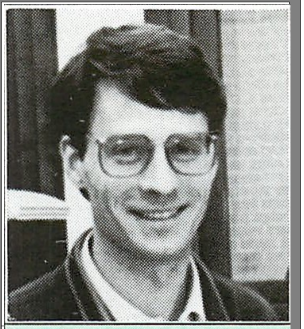


Figure 1-1. *The buildings of the Department of Computer and Information Science (E and E++, to the left).*

ACTLAB

2 The Laboratory for Complexity of Algorithms

Computational geometry
Analysis of algorithms
Data structures



Per-Olof Fjällström
director

Back row: Jan Petersson, Per-Olof Fjällström
Front row: David Andersson, Bodil Mattsson-Kihlström

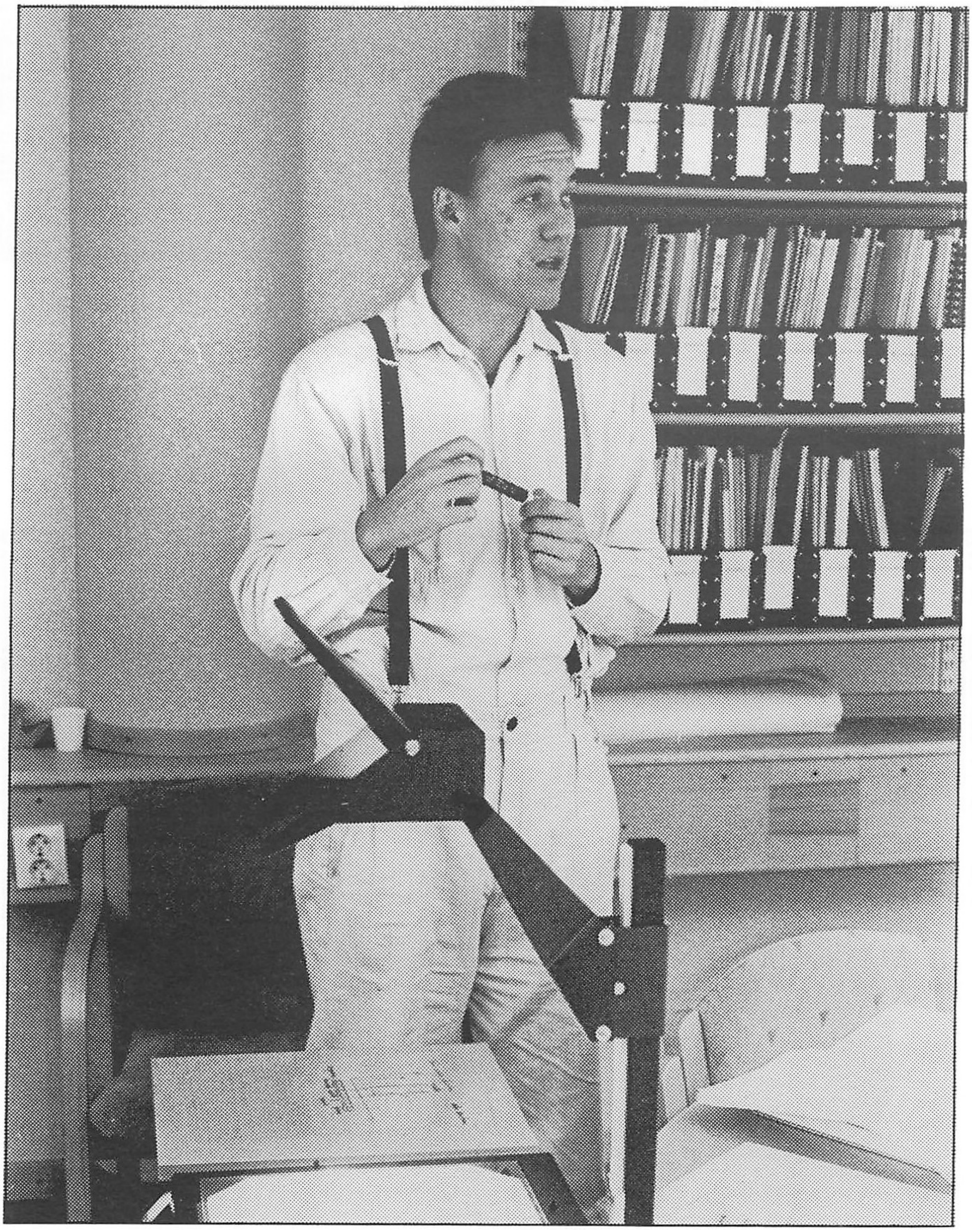


Figure 2-1. *Jyrki Katajainen, associated guest researcher to ACTLAB, giving graduate cource.*

2.1 Introduction

The Laboratory for Complexity of Algorithms is concerned with the design and analysis of efficient algorithms and data structures for combinatorial and geometric problems, and the study of the inherent complexity of these problems in simple models of computation. Members of the laboratory believe that work on efficiency of algorithms and data structures is no less important than the development of new programming methodologies, or new faster computers.

Although ACTLAB has already existed a number of years under the leadership of Andrzej Lingas, it must now be regarded as a new laboratory: all of the present members have joined the laboratory since 1989. As the current members share many of the interests and ambitions of the original members, it has, however, seemed natural to retain the name of the laboratory.

The research in the laboratory is presently limited to computational geometry. Computational geometry is a relatively new research area within computer science, and is concerned with the computational complexity of geometric problems within the framework of analysis of algorithms. A large number of applications areas, e.g. pattern recognition, computer graphics, computer-aided design, robotics, etc., have been the incubation bed for this new discipline. Although theoretical by nature, the results obtained by computational geometry now serve as an important base for further development within the above application areas.

The major funding 1990 is from CENIIT (the Center for Industrial Information Technology), a special programme for interdisciplinary research within Linköping University. Since July 1990, STUF has funded a project concerning parallel algorithms for certain geometric problems.

2.2 Laboratory Members

Laboratory leadership and administrative staff:

Per-Olof Fjällström, PhD, associate professor
Bodil Mattsson-Kihlström, secretary

Graduate students:

David Andersson (starting 1990)
Jan Petersson

The work in the Laboratory for Complexity of Algorithms is mainly supported by CENIIT (the Center for Industrial Information Technology), Linköping University.

2.3 CENIIT project: Geometrical Algorithms

The topic of this project is efficient algorithms for geometric problems, in particular *three-dimensional problems* and geometric problems occurring in *motion planning*. The project started in July 1988 under the leadership of Rolf Karlsson. Since 1989, Per-Olof Fjällström has been leading the project. Apart from investigating specific problems, an important goal of this project is to develop competence which can help to identify new research problems. The activities within the project, which are briefly described below, fall within three different research areas.

Motion Planning:

In collaboration with Erland Jungert, FOA, a series of seminars on motion planning was held during the autumn of 1989.

Parallel Algorithms for Geometric Problems:

In collaboration with previous members of the ACTLAB, a parallel algorithm for finding the convex hull of sorted points was developed, [FKLP90]. The work in this area also resulted in the STUF project, further described below.

We have also had some international contacts: Jyrki Katajainen (Turku University), Wojciech Rytter (Warsaw University), and Jörg-Rudiger Sack (Carleton University) have visited ACTLAB.

Polyhedral Approximation of Points in Space:

In many applications we are given a set *S* of points in the plane. Corresponding to *S* there is also a set of heights, *h(S)*, which contains the height at each point in *S*. We want to determine a continuous surface that approximates the given data. Usually this is achieved by triangulating *S*, and then, for each triangle, fit a triangular surface patch to the heights at the vertices. In order for this method to give good results, it is important for the triangulation to have certain properties. The so-called Delaunay triangulation is considered a good triangulation for this kind of application.

However, if there is a large number of points it can be quite expensive to store, and perform further calculations on this surface. To avoid this, some kind of data reduction may be necessary.

These observations lead us to formulate the following problem. Let *T(S)* be a Delaunay triangulation of *S*, and *F(T(S), h(S))* the set of planar, triangular faces which interpolate the given data. If we remove a subset *S'*, we obtain a modified surface, *F(T(S-S'), h(S-S'))*, which approximates the original data. Let *d(S')* be the maximum deviation between *F(T(S-S'), h(S-S'))* and *h(S') at S'*. We want to make *S'* as large as possible, without violating the constraint that *d(S')* must not be greater than a predefined value.

Both exact and approximative algorithms have been developed for this problem, [Fjä90]. In a preliminary study, [Fjä89], the corresponding two-dimensional problem was investigated.

2.4 STUF project: Parallel Algorithms for Geometric Problems

This project started July 1990. In the following we give a short introduction to the research area, and a description of the problems that we are specifically interested in.

During the last decade a large number of sequential algorithms have been developed to solve geometric problems. Since an increasing number of parallel computers have become commercially available, and many geometric problems are used for on-line applications where short response times are required, the design of parallel algorithms has lately received growing attention.

In contrast to sequential algorithms, there exists a variety of computational models which can be considered in designing parallel algorithms. In parallel computational geometry many of the published algorithms are designed for the *parallel random access machine* (PRAM), that is, a set of n processors which are connected to a global shared memory. The processors are allowed to simultaneously access any location of the memory in constant time. Some algorithms have also been developed for *network-connected processor arrays*, e.g. one- and two-dimensional mesh computers, and hypercubes. In, for example the two-dimensional mesh computer, n simple processing elements are arranged in a square lattice. In this model there is no shared memory, and unit-time communication can only take place between immediate vertical and horizontal neighbours.

All of the models referred to above are so-called *Single Instruction Multiple Data* (SIMD) machines. The characteristic of SIMD machines is that they consist of a set of identical synchronized processing elements capable of simultaneously performing the same operation on different data.

Most of the research on parallel algorithms for geometric problems has so far concentrated on two-dimensional problems. It is in general non-trivial to generalize a successful solution for a two-dimensional problem to cater for the corresponding higher dimensional problem. Unfortunately, many important problems in, for example, computer-aided design and robotics are of higher dimension. For this reason, we feel that it is important to investigate some three-dimensional problems. The following is a list of problems we plan to consider in this project.

1. Given two convex polyhedra P and Q, with L and M vertices respectively,
   - detect if any of their faces intersect,
   - if so, form the intersection of P and Q.

   Motivation: Several solid modelling systems use polyhedra to approximate primitive building blocks like spheres, cylinders, and etc. A careful approximation can lead to polyhedra with thousands of faces.

2. Given a simple polyhedron P with L vertices, partition this into convex polyhedra.

   Motivation: Partitionings are often used to decompose objects into simple objects, which are easier to handle in subsequent computations.

3. Given a set of points in three dimensions, find a triangulation of these points.

Motivation: This is also a kind of partitioning, with applications in numerical interpolation, the finite element method, etc.

2.5 External contacts

Per-Olof Fjällström: Participated in the *ACM Symposium on Computational Geometry,* held in Berkeley, USA, June 1990.

Jan Petersson: Participated in *Scandinavian Workshop on Algorithm Theory*, held in Bergen, Norway, July 1990.

Courses for Graduate Students

An important task of the group is to spread the knowledge of algorithm design and analysis among graduate students within the department. The following graduate courses has been offered for the academic year 1989-90:

Design and Analysis of Algorithms.
Data Structures and Graph Algorithms.
Computational Geometry.

2.6 References

[FKLP90] **P-O. Fjällström, J. Katajainen, C. Levcopoulos, O. Petersson:** A Sublogarithmic Convex Hull Algorithm. In *BIT, 30:3, pp 378-384, 1990.*

[Fjä89] **P-O. Fjällström:** Approximation of Polygonal Lines. *LiTH-IDA-R-89-30.*

[Fjä90] **P-O. Fjällström:** Polyhedral Approximation of Bivariate Functions. In preparation.

ASLAB

3 The Laboratory for Application Systems

Knowledge-based systems
Human-computer interaction
Database technology

Sture Hägglund
director

Back row: Henrik Eriksson, Kristian Sandahl, Thomas Padron-McCarthy, Eva Lundberg, Ivan Rankin, Torbjörn Näslund, Ting-Ting Zhang, Olof Johansson, Sture Hägglund.
Front row: Peter Eklund, Jonni Harrius, Jonas Löwgren, Birgitta Franzén.

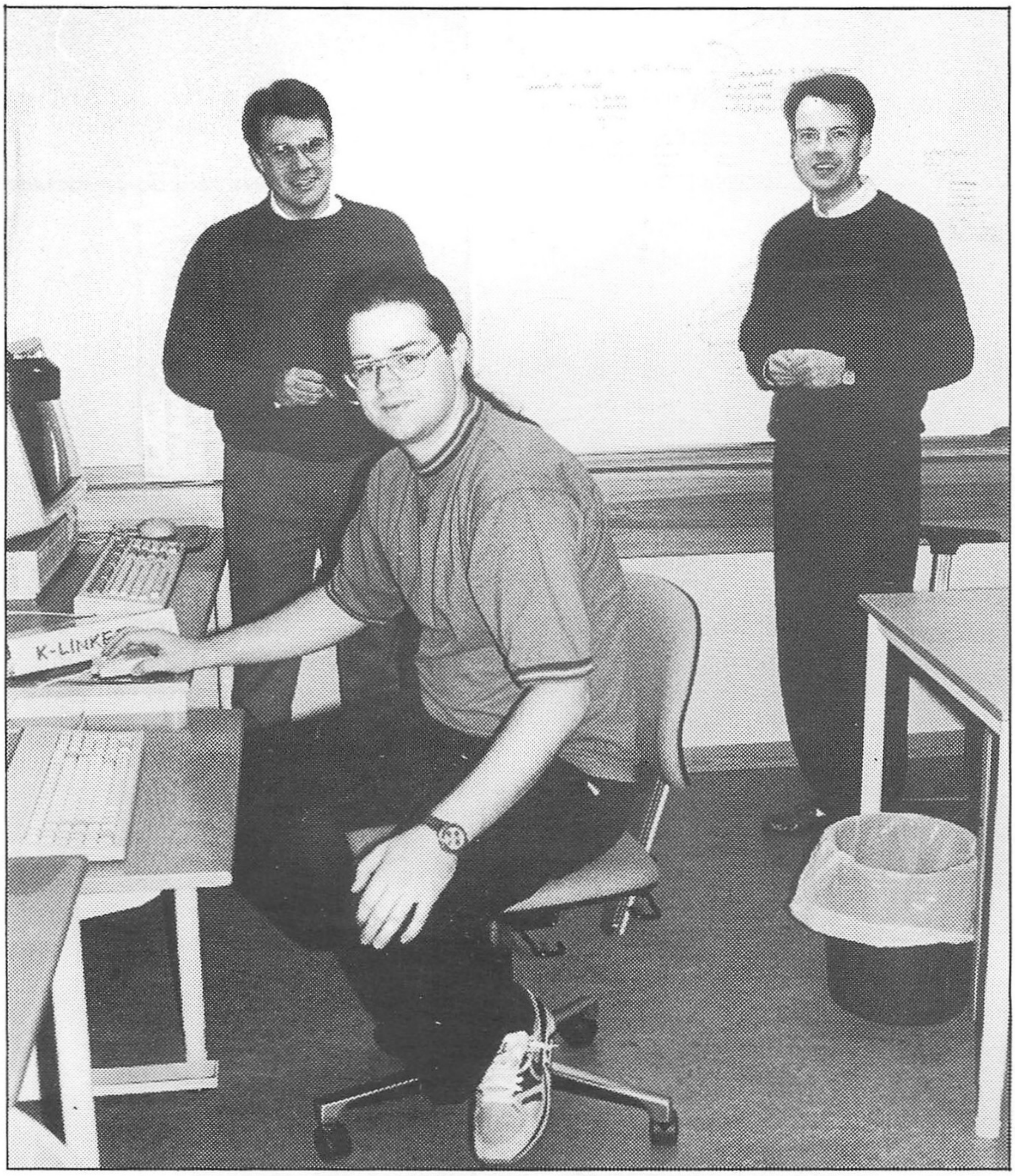


Figure 3-1. *Members of ASLAB in their research laboratory. Kristian Sandahl (left), Henrik Eriksson (right) and Thomas Padron-McCarthy (sitting).*

The research program in the Application Systems Laboratory (ASLAB) is oriented towards the study of theory, methods and tools, in particular knowledge-based approaches, for the development and maintenance of generic applications software aiming at a significant increase in productivity, maintainability, understandability and user control. A central theme for our research is the integration of applied AI techniques and expert systems methodology with more traditional information technology, in particular human-computer interaction and database technology. Projects usually take an experimental approach and emphasize participation in application-oriented projects with industry and the public sector.

3.1 Summary of current research

The current research in the laboratory is organized in two major activities dealing with the study of

- development methods and support environments for generic knowledge systems, in particular domain-oriented tools supporting knowledge acquisition, reusable knowledge representations for families of related applications, integrated support for tutoring and training, and migration to different delivery environments.
- certain aspects of the human-computer cooperation, especially consultation models for decision support, expert systems user interface management and text generation in expert critiquing systems.

Knowledge acquisition, i.e. the process of understanding, formulating and representing the relevant knowledge for solving problems in a particular area, is generally recognized as a key problem in knowledge system development. Our work focuses on methodological support for understanding the application domain, formulating a specification of the task to be solved, developing the appropriate knowledge representation and problem solving strategies, and finally finding an efficient implementation in the delivery environment. In the process of actually creating and maintaining a knowledge base, it is desirable that the domain expert should be as involved and in control as possible, the *active expert approach* (Sandahl). This calls for a meta-level tool support strategy (Eriksson), which allows the knowledge engineer to create a customized environment for a specific generic application domain (i.e. a conceptual framework for a ’family’ of knowledge systems) with minimal effort.

The major activity concerning development environments for generic knowledge systems, including intelligent tutoring support, during the last year has been based on a concentrated effort in an application domain defined together with Pharmacia Biosystems AB. Several prototype systems have been developed concerning planning of protein purification experiments. (Sandahl, Eriksson, Padron-McCarthy, Sokolnicki).

Another focus of research is on human-computer interaction, especially with respect to knowledge-based consultation systems. A generalized model for a UIMS (User Interface Management System) architecture which supports dialogue interfaces to expert systems has been developed (Löwgren). Our approach identifies two different

The work in ASLAB is mainly supported by STU, The Swedish Board for Technical Development. Additional funding comes from CENIIT, HSFR, FOA and others.

structures in an expert system user interface: the *surface dialogue* and the *session discourse*. The management of the session discourse provides a possibility to make the dialogue structure independent of the underlying reasoning process of the system, by using a planning formalism based on operators defined by speech acts.

A special interest in our group is devoted to expert critiquing systems, i.e. consultative systems that review and comment upon proposed decisions and plans for action, indicating oversights, risks, alternatives and other relevant observations. In particular we study generalized methods for text generation in expert critiquing systems (Rankin) and related knowledge representation problems (Harrius). The focus is on deep generation of text, which involve establishing the goals the text is to achieve, determining a level of detail of information to include and organizing the various parts of the text to form a cohesive unit.

In a related work a critiquing/negotiating approach is applied to the problem of designing conceptual structures in the form of semantic networks for a particular domain. An acquisition interface system is built in order to support this approach (Eklund).

The bordering area between knowledge systems, database technology and computerized decision support plays an increasingly important role. We cooperate with other groups in the department on these subjects, for instance IISLAB/LINCKS (Padgham) concerning diagnosis reasoning by encoding classification knowledge in type hierarchies with inheritance (Zhang). This work is carried out in the context of our participation in research on engineering databases (Johansson). Another area of study is information management in the context of crisis management and emergency decision making, where we cooperate with the Defense Research Establishment (FOA) on knowledge-based decision support in command and control systems (Näslund).

3.2 ASLAB personnel

The following list presents persons contributing to project activities in ASLAB during 1989–1990.

Laboratory leadership, administrative and technical staff:

Sture Hägglund, PhD, professor.

Birgitta Franzén, secretary (from spring 1990).

Britt-Marie Ahlenbäck, secretary (until autumn 1989).
Mats Andersson, senior research engineer.

Graduate students:

Henrik Eriksson, MSc, Tekn. Lic.
Peter Eklund, B.Math, M.Phil.
Jonni Harrius, MSc.
Jonas Löwgren, MSc, Tekn. Lic.
Henrik Nordin, MSc, Tekn. Lic. (until autumn 1989).
Torbjörn Näslund, BSc.
Thomas Padron-McCarthy, MSc.

Ivan Rankin, BA, Fil. Lic.
Kristian Sandahl, MSc, Tekn. Lic.
Tomas Sokolnicki, MSc.
Tingting Zhang, MSc.

Associated persons:

This list includes persons who have actively contributed to ASLAB projects during the last year, either as cooperating researchers in other departments or as project participants not permanently employed in the lab. (Cooperation with researchers in other labs in our department is extensive, but not included in this list for the sake of brevity.)

John Brewer, Pharmacia Biosystems AB.
Östen Dahl, Dept. of Linguistics, Stockholm University.
Christina Hellman, Dept. of Linguistics, Stockholm University.
Eva Lundberg, IDA and FOA 5.
Jalal Maleki, IDA.
Tommy Nordqvist, FOA 5.
Greger Sahlberg, Pharmacia Biosystems AB.
Anne Steinemann, Stanford University.
Yvonne Waern, Dept. of Psychology, Stockholm University.
Bengt Österlund, Pharmacia Biosystems AB.

3.3 Engineering Environments for Generic Knowledge Systems

In the K-Linker project we are studying issues related to knowledge acquisition, experiment planning and integrated approaches to the development of real-world knowledge-based systems. The project emphasizes practical investigations with iterated implementations, where architectural issues are tested and experiences generalized into revised designs.

The current reference domain is advisory systems for the planning of biotechnical experiments, with practical applications developed in cooperation with Pharmacia Biosystems AB. Experiments involve the use of complicated laboratory equipment and domain experts are regularly consulted both for initial advice and if experiments fail.

The K-Linker project strives for a solution which shows how effective domain-dependent development support can be readily created, once a generic application domain has been identified. This includes customized support for knowledge acquisition and maintenance, transformation from the development into the delivery environment and user interface support including tutorial use of the knowledge. The feasibility of the approach will be demonstrated in the biotechnical experiment planning domain, but the claim is that results will be applicable to many other application areas as well.

3.3.1 Knowledge system design with active expert participation

Kristian Sandahl

This work focuses on the methodological problem of transferring knowledge from domain experts to non-expert end users. Experiences from a number of previous expert system projects, where Sandahl has been the project leader or else participated, form the basis for the current design of the K-Linker support environment as well as for a forthcoming PhD thesis.

The active expert approach is essentially a two-phase methodology, where the conceptual and problem-solving framework for a generic application domain is created during the first phase under the active supervision and support of a knowledge engineer. During the second phase, the domain expert is in control and uses a customized knowledge acquisition environment for development and maintenance purposes. The tools needed to create such environments will be discussed in more detail in the next section.

Some specific problems related to the active expert methodology are the following:

- Problems of detecting, preventing and resolving inconsistences between concepts or specific parameters introduced by the expert. Such problems are common in practice, due to inherent ambiguity in every-day concepts, inexperience among domain experts concerning complete formalization of terminology and in particular problems arising when multiple experts are involved.
- Knowledge base transformations in connection with migration to the delivery environment. This is motivated both by the desire to postpone decisions regarding implementation details of individual systems as long as possible, and to allow adaption to local needs for a delivered system.
- Problems related to the design of appropriate abstraction mechanisms for domain experts in a given domain and for specific tasks. An example of this problem was encountered in connection with the design of a two-level planner for biotechnology experimentation.
- Problems in connection with iteration over the two development phases, where revisions in the domain model framework lead to demands for modifications in the customized knowledge acquisition environment. The degree to which such changes should be allowed and supported is a crucial issue also from a methodological point of view.

The active expert approach to knowledge engineering and an analysis of experiences from a series of applied projects is the topic of Sandahl's forthcoming PhD dissertation.

3.3.2 Meta-level tools for domain-oriented knowledge acquisition

Henrik Eriksson

Meta-level environments, i.e. tools which create other knowledge acquisition (KA) tools could be a way of reducing the cost of domain-dependency. Our hypothesis is that it is more effective (by time and quality measures) to use a meta-tool to create a tailored KA tool which in turn is used to implement the target system, than to have a general KA tool which covers a wider range of applications to build the final system.

The *domain-oriented* approach assumes that (a) experts can utilize a KA tool better if it is specialized to his/her area and (b) distortion and loss of knowledge in the KA process can be reduced if the tool design is influenced by the application domain. Furthermore, the KA tool should be based on the expert's conceptual model or a conceptual model shared by a group of experts, i.e. the tool should be *model-based*.

The KA tool P10, customized to the domain of protein purification planning in particular membrane proteins, was implemented as a step towards gaining an understanding of the requirements of such environments. P10 supports direct entry of knowledge by domain experts through a set of graphs and menus. Entered knowledge can later be transformed into a target knowledge base for delivery. In P10, a set of transformation rules is used for that purpose. In a comparison between tool-generated and hand-crafted knowledge bases, P10 was used by an expert to create a new knowledge base for another application. The amount of terminal time spent by the expert was about six hours; the original system (P8) was developed by conventional means in six man-months.

In our approach to meta-support, the meta-tool should not resemble a KA tool, but rather a supportive environment (or tool shell) for the knowledge engineer. We suggest that the meta-environment should be used in the process of eliciting a conceptual model of the domain; the expert should be able to run and comment upon several prototype versions of the KA tool, i.e. the conceptual model and the KA tool should be developed incrementally and in parallel.

A meta-tool (DOTS) which can be used by knowledge engineers to develop and maintain realistic KA tools is currently being implemented. DOTS allows entry and maintenance of an explicit representation of the adopted conceptual model of the domain. From this domain model and additional specifications (such as interaction models, and transformational rules) the program can produce a domain-oriented KA tool.

DOTS is task and domain-independent in the sense that it does not assume any particular domain/task orientation for KA tools it generates. Thus it will be subject to evaluation by applying it to several different domains and generic tasks. Its design and preliminary results of evaluation is reported in Eriksson's forthcoming PhD dissertation.

3.3.3 Justification in knowledge systems

Thomas Padron-McCarthy

In our experience, the knowledge communication aspect is of major importance in many expert systems applications. This means for instance that the possibility to motivate and justify results and recommendations from a system is a crucial issue. Within the K-Linker project, Thomas Padron-McCarthy is studying problems in connection with justifying reasoning steps in a knowledge system, in particular the role of "negative" knowledge in this context.

This area is of course closely related to the problem of realizing intelligent tutoring services, as described in the next section. It is also an essential issue for expert critiquing approaches, which is another major theme of study in our group. The work of Padron-McCarthy on the use, representation and application of negative knowledge in knowledge-based systems is still in a preliminary stage, but we believe the ability to build and maintain justification structures to be an important problem for the continued project.

3.3.4 Reusing knowledge bases for tutoring

Tomas Sokolnicki

An important aspect of the K-Linker approach is the emphasis on generalized support for knowledge management, including also the communication of knowledge to the end user. In particular, Sokolnicki studies the potential for realizing intelligent tutoring services based on knowledge primarily elicited for problem-solving purposes.

This work concentrates on strategies and practice for coaching partial plans in the domain of application. As a concrete case in the K-Linker context, planning of protein purification experiments are studied. The objective is to:

- supply a training environment as an extension to a problem-solving and decision-supporting system.
- develop and evaluate a modular architecture for intelligent tutoring systems based on a domain-oriented, yet flexible, approach.
- prepare and adjust the knowledge acquisition methodology so that tutoring knowledge can be included with a minimum of extra effort.
- evaluate coaching as an instructional paradigm: its feasibility, integrability aspects and user acceptance.
- provide a basis for supporting guidance and explanations not only for pure training, but also for user assistance in general.

A first design based on initial experiments was presented in a licentiate thesis, 1990. This work also includes consideration of “weak” coaching, understood as computer-based training based on case management simulations. Experience from the use of interactive video and from training programs with manual supervision in the application domain is also considered.

3.4 Knowledge-based human-computer dialogue models

Managing the user interface presents a special challenge in the context of knowledge systems, where the system as well as the user is actively generating and pursuing plans which initiate communicative actions. One main theme of our research is to extend ideas from standard UIMS (user interface management systems) into this area by introducing a session discourse manager, which mediates between the reasoning and the dialogue processes.

Another major theme in our research is the interest in the expert critiquing approach, which assumes that an advisory system should evaluate and comment upon suggestions from the user rather than dictate the solution model to be applied to a specific problem. Critiquing is an interesting paradigm, which provides a valuable complement to traditional transaction-oriented dialogue systems and also to the first generation expert systems, where the dialogue is generated by the inference engine as a side effect of the problem solving. Critiquing is particularly interesting with respect to its potential for supporting the user's competence development, where we take a special interest in allowing explicit tutoring and training to be included in the user support facilities.

In order to ensure that our work is based on a sound theoretical foundation regarding cognitive aspects of human-computer interfaces, we have established separately funded cooperation with the Psychology Department, Stockholm University (Yvonne Waern). Our joint project studies computer-mediated knowledge communication as a phenomenon (the CAFKA project; *Communication, Application and Forwarding of Knowledge in Artificial Systems*). For problems in text generation, cooperation is established with NLPLAB and with the Linguistics Department, Stockholm University (Östen Dahl).

3.4.1 User interface management

Jonas Löwgren

User Interface Management Systems (UIMSs) are today widely recognized as bringing several advantages to the process of software development. By separating the user interface from the functionality of the application being developed, maintainability, reusability and prototyping is greatly enhanced. Unfortunately, UIMSs of today are not directly applicable to expert systems development. The reason for this is that in expert systems, the application functionality has traditionally been regarded as the determining factor for the structure of the user interface. We are developing a user interface management architecture for expert systems, analogous to a UIMS, which gives the expert system developer the possibility to design the user interface of the system as a separate entity without affecting the problem-solving capabilities of the expert system.

The expert system user interface management architecture comprises two modules: a surface dialogue manager and a session discourse manager. The surface dialogue manager is essentially a conventional, object-oriented UIMS which is used for designing the lexical and syntactic properties of the user interface. In a model-world interface to an object-oriented knowledge base, the surface dialogue manager defines the views or appearances of the domain objects along with their lexical/syntactic behaviour. For the conversational interface case, the objects of the surface dialogue give surface form to the discourse acts received from the session discourse manager (see below).

The session discourse manager is what enables the system to carry out a dialogue which is possibly distinct from the line of reasoning in the problem domain. It is a plan-based tool where the operators of the planning system are similar to speech acts. Plan structures are defined in terms of their applicability conditions, decompositions into subplans, and effects. There are two kinds of plans: domain plans and discourse plans. The discourse plans are domain-independent and embody the kind of conversational structures that can arise in a dialogue irrespective of the topic discussed. Examples of discourse plans include to ask for clarification, to change the topic and so on. The domain plans are specific for each application and are used at runtime to create an overall hierarchical plan, describing the discourse structure for the whole session. A plan recognition module ensures that the system is responsive to user deviations from this overall structure.

The separation of domain reasoning from discourse management makes it possible for the expert system developer to keep the knowledge base clean, which is likely to greatly enhance portability and maintainability. We also expect that this separation will yield the other advantages which, in the software engineering community, are put forward as effects of employing UIMS technology. Among these are capabilities for prototyping and making available different user interfaces for different user categories.

Another special interest within the area of knowledge-based systems and human-computer interaction is the possibility to develop knowledge-based support for dialogue design, providing critique or constructive advice. The knowledge base might contain general standards as well as local design rules, published guidelines for dialogue design, more formal style guides, rules for consistency control, etc. An explicit representation of the user interface is also needed. Some notion of task and user modelling is essential for a deeper analysis and an effective evaluation of a dialogue design.

In a joint project with FOA and Enator/Epitec these and other related issues are studied (Löwgren et al., 1989). The KRI user interface critic contains a rule base with evaluation knowledge, a database with interface design guidelines and a user interface evaluation taxonomy. The demonstration prototype developed so far allows the user to customize an evaluation plan for a given dialogue design and then have a critique generated. Comments may concern menu layouts, dialogue sequencing, consistency of design decisions, etc. Continued studies emphasize improved evaluation methodologies, integration of KBS and UIMS technologies, user interface representation issues, and implications for the proper codification of dialogue design guidelines to be stored in a knowledge base. The KRI project and these continued studies are reported in a forthcoming PhD dissertation by Löwgren together with an implementation of the aforementioned approach to expert system user interface management.

3.4.2 Expert critiquing and text generation

Ivan Rankin, Jonni Harrius

In an *expert critiquing* system the user is assumed to take the initiative and suggest a decision or course of action. The system then reviews this suggestion relative to known circumstances and tries to evaluate the decision, to comment upon its necessary prerequisites, its risks, costs, reasonable alternatives and their merits, etc.

A high-performance critiquing system must be able to solve decision-making problems independently of the user in order to provide a basis for a profound criticism of the user's decision proposal. Thus it needs the power of a problem-solving expert system and at the same time the reasoning ability and a knowledge base broad enough to ensure an adequate analysis of various decisions suggested by the user. This indicates that building a system for expert critiquing is an order of magnitude harder than creating an ordinary expert system.

> Patients being treated for cystitis should be considered individually.
>
> The urine culture should be carried out, but it is normal practice to run urography, cystoscopy and routine blood tests as well.
>
> A return visit in 5 weeks conforms to standard practice.
>
> Trimetroprim 330 mg 1*1 is the most appropriate medicine in this case, though it should be taken for 7 days, not 3.

Example 3-1. A critique generated by the CRIME system in response to a suggested treatment plan for a diagnosed urinary tract infection.

However, there is also another side of the coin. We can expect an expert critiquing system to be useful also with partial proficiency in an area, while a notorious problem with traditional expert systems is that typically a very high degree of competence must be documented before the system is of any practical value. Sometimes this level is never achieved and at least it is often a very demanding task to acquire all the knowledge needed for peak performance.

Supporting an expert critiquing mode is important not only as a means to improve user acceptance, but also in situations where we can not guarantee a complete and proficient knowledge base. It is also interesting to note that critiquing may constitute a transition mode of use during an interactive process of build-up and validation of an expert system. More precisely, the usage may be restricted to critiquing services during early development stages with a still imperfect knowledge base. When confidence in the system is established, users can be allowed to rely more on the system's advice.

The work by Rankin demonstrates how text generation based on general-purpose linguistic principles can be achieved in the context of a critiquing system. This project has investigated the utility of more straight forward approaches in a medical application (the CRIME system). In the generalized approach, a system (AREST, [Harrius 89]) was implemented based on Mann's *rhetorical structure theory.* Rankin emphasizes the problem of deep generation, rather than concern with the surface text realization. Understanding how to establish the goal the text is to achieve, determining the appropriate level of detail, plan for the presentation and finally to organize a cohesive message presents a stimulating challenge for research. Continued work in cooperation with Stockholm University (Dahl, Hellman) will further develop a theory for rhetorical relations, suitable for generation of critiquing and explanatory texts.

Expert critiquing is obviously closely related to the problem of providing explanations and justifications in expert systems. It has been argued that an effective explanation should be realized as a conversational process, where the required understanding is established as a result of a dialogue. Such an approach connects our work on critiquing with the previously discussed model for user interface management, where a session discourse manager supervises the dialogue guided by discourse plans. Thus we plan to extend the use of rhetorical structures to dialogue planning in addition to generating monolithic texts.

The work of Harrius, who joined the project last year, is oriented towards the requirements on the underlying knowledge representation for critiquing. One possibility, studied by Harrius, is to combine RST with a knowledge representation scheme based on conceptual graphs (according to Sowa's definition) in order to support text generation.

3.4.3 Negotiating Network Hierarchies

Peter Eklund

To make proper use of conceptual structures, network representations need to be free of ambiguous or misleading interpretations. For this reason this work produces critiques of networks in terms of their logical consequences.

Critiquing or negotiating systems build on traditional knowledge-based technologies to emphasize the support of client decisions suggesting recommendations. In the case where a client proposes a conceptual structure, two potential solutions are proposed for the basis of a critique of a network hierarchy;

1. the domain is genuinely characterized by an incomplete or ambiguous model,
2. the client has incorrectly characterized domain knowledge.

If the domain is incorrectly characterized, the strategy in semantic network hierarchies is to isolate nodes which yield contradictory conclusions. The client then addresses a node's classification or, alternatively, reaffirms commitment to the domain model. If the later applies, the soundness of the topology is explored. This is done by isolating network paths yielding mutually exclusive states. By ascribing a "domain" level priority to these competing paths, undesirable alternatives are eliminated from the model. If competing extensions are preferred by the client, the network topology is reconfigured to reflect this preference.

If the domain is genuinely incomplete or ambiguous, and the user model agrees that the client's domain model is correctly described, then the strategy is to apply a more general theory of default reasoning. An interesting aspect of this work is the selection of an appropriate theory of default reasoning for inheritance hierarchies.

A formal cognitive basis for user misconception and system misinterpretation in expert critiquing gives a clear mechanism for recognizing the appropriate discourse situation when ambiguous hierarchical topologies are present. On the basis of this recognition, the system can then act to produce a domain model critique or reason with ambiguous or misleading knowledge. The construction of an acquisition interface for editing semantic networks is one way of verifying the effectiveness of the expert critiquing paradigm.

This work will be reported in the forthcoming PhD dissertation by Eklund.

3.4.4 Support systems for decision making

Torbjörn Näslund

Iterative development and prototyping is important for decision support systems, due to the difficulties to specify in advance what constitutes good support for a particular decision and decision-maker. Such a development activity should be a carefully planned investigation in order to gain empirical knowledge for reducing uncertainties which complicate the design process. The role of evaluation is important in this development paradigm. It is the evaluation activity that helps us reduce uncertainties of the effectiveness of the system, and provides information for design decisions and for a continuous adaptation of the system to its intended use.

Evaluations are performed by comparing some aspects of a system (or a proposed system) to a norm. The norm should ideally be deduced from the set of objectives for the system. The relevant aspects should in some way be measured on the proposed system. These measurements can be collected by use analysis, inquiries, studies, and experimentation.

Evaluations of decision support systems can be performed at different levels, depending on where the uncertainties are. We believe that one common deficiency among support systems is that the systems per se are correct, but that there are disturbances in the interaction between the system and its user. For a support system, it is vital that the user is actually supported. When the user is not able to utilize the intended idea underlying the system, we call this a shadowed idea. One use of evaluation is to assess this shadow, in order to be able to reduce it.

Our research involves taking a rather active role in the evaluation of real support systems under development. This work is primarily oriented towards command and control systems and carried out in cooperation with the Defense Research Establishment, involving also issues of the human-computer interface (Lundberg). Here we will be able to collect successive evaluation results, and to show how the development process can benefit from these evaluation results. Thus we intend both to develop a normative framework of how to perform evaluations and describe cases where this framework have been used, and hopefully have improved the development of the systems.

3.4.5 Classification models for knowledge-based diagnosis

Tingting Zhang

The aim of this project is to further develop a model of typing and default reasoning to include classification and diagnosis in the context of object-oriented databases. Our model for typing was originally developed as a knowledge representation theory (Padgham, IISLAB) with several advantages over other existing theories with the same purpose. Our goal is now to apply this model to the task of making diagnosis classification in domains where causes of relevant disorders can be organized into an inheritance lattice. Applications can be found in technical trouble shooting, engineering design and medicine, for instance. A characteristic feature of our approach is that it can deal with exceptional and incompletely described items in a natural way, while at the same time standard sources of classification schemata can be used as the main input in building the knowledge base.

It is our belief that in order to make progress in the area of routinely building large knowledge-based systems it is important to identify general purpose representations of domain knowledge and to combine this with reasoning algorithms which are also relatively general-purpose and tractable. It is also clear that a knowledge-based system needs to incorporate default reasoning about objects (such as is typical of the knowledge representation approach to inheritance schemas).

This work is carried out in cooperation with IISLAB. In a first investigation, we have built a medical diagnosis system for urinary tract infections. Our system, LIC (A Lattice-based Inheritance Classifier), uses the lattice based inheritance model developed by Lin Padgham. The purpose of the implementation is to evaluate the use of this representation, both from the point of view of the adequacy of the resulting system for diagnosis, and also from the point of view of organization of the knowledge base, ease of obtaining and maintaining the knowledge and use of the knowledge base for tasks other than diagnosis.

3.5 Joint projects and external cooperation

ASLAB projects emphasize joint efforts with other groups and industry. The following is a list of current involvements where we are actively cooperating with companies subscribing to the knowledge transfer programme provided by our department:

- Pharmacia, Uppsala. cooperation on knowledge-based systems in support of experiment planning in bio-technology started during the spring, 1988.
- Volvo PV, Gothenburg. Joint activities on expert systems started in December 1987 with a special emphasis on systems for fault diagnosis and technical maintenance.
- TeleSOFT, (formerly Programsystem AB) Linköping. cooperation with a special emphasis on human-computer interaction in general and UIMS in particular. This company is developing the TeleUSE UIMS, which is one of a few candidates listed by OSF as the basis for a potential UIMS standard.
- Volvo Data, Gothenburg. This cooperation is presently in a planning phase. ASLAB's involvement is primarily in areas of AI planning and methodologies for knowledge acquisition.

We also cooperate with the National Defense Research Establishment (FOA) and Epitec/Enator in a R&D consortium in the area of knowledge-based systems and user interfaces, with a special emphasis on evaluation methodology. In joint projects with Psychology and Linguistics at Stockholm University (Yvonne Waern and Östen Dahl), we study the cognitive and linguistic aspects of the communication that goes on in the interaction with knowledge-based systems. Within the department, cooperation is also extensive, e.g. with CAELAB, IISLAB, NLPLAB and LIBLAB where joint projects are in progress.

Another major channel for industry contacts is *Sveriges Mekanförbund*, where we carry out commissioned study projects in cooperation with groups of companies. Recent studies include *Knowledge Systems for Product Development: A Feasibility Study in Ten Companies and Dialogue Ergonomics – Effective Human-Computer Interaction*. These projects also contribute to the funding of our group, in addition to providing effective means of communication with personnel from industry. A similar study is made in the area of knowledge-based systems for process control.

Numerous seminars for industry have been given and we have also participated in the development of continuing education programmes in *knowledge engineering*. Thus we are responsible for the expert systems courses in the 14-month half-time programme offered by our department. (The programme was delivered in Västerås for ABB in 1988–89 and in Gothenburg for Chalmers and Volvo 1989–90.)

International contacts are mainly oriented towards the US. We regularly send students to Xerox PARC as summer interns. Other important contacts are found in for instance Information Science Institute, USC and at Stanford. In Europe we maintain contacts with e g Xerox EuroPARC, Univ. of Limerick, Ireland (Kevin Ryan) and Siemens in Munich (Michael Reinfrank). We have also joined a COST project on Computer Supported cooperative Work.

3.6 Publications

For a complete listing of published papers, see Appendix E. Below a list of selected recent publications by lab members is given for an easy reference.

Forthcoming PhD dissertations:

Jonas Löwgren: Knowledge-Based Design Support and Discourse Management in User Interface Management Systems. March 1991.
Henrik Eriksson: Meta-Tool Support for Knowledge Acquisition. April, 1991.

Licentiate theses 1989/90:

Henrik Eriksson: A Study in Domain-oriented Tool Support for Knowledge Acquisition, 1989.
Jonas Löwgren: Supporting Design and Management of Expert System User Interfaces, 1989.
Ivan Rankin: The Deep Generation of Text in Expert Critiquing Systems, 1989.
Tomas Sokolnicki: *Coaching Partial Plans: An Approach to Knowledge-Based Tutoring*, 1990.

International publications 1989/90:

P. Eklund: Using Path Algebras to Encode Directed Graph Knowledge-bases. *4th Annual Workshop on Conceptual Graphs*, Detroit 1989.

P. Eklund: Negotiating Conceptual Structures. *5th Annual Workshop on Conceptual Graphs*, Stockholm 1990.

P. Eklund, J. Kellett: Prospects for Conceptual Graphs in Acquisition Interfaces. *3rd European Workshop on Knowledge Acquisition*, Paris, 1989.

P. Eklund, D. Driankov, A Study of Induction Learning and Statistical Methods for Attribute Prediction, to appear in *Proc. of the 3rd International Workshop on AI and Statistics*, Fort Lauderdale, FL, 1991.

H. Eriksson: Meta-Tool Support for Customized Domain-Oriented Knowledge Acquisition. *Fifth Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Nov. 1990.

H. Eriksson: Exploring the Meta-Level – A Supportive Environment for Implementation of Domain-Oriented Knowledge Acquisition Tools. In: *Proceedings of AAAI-90 Workshop on Knowledge Acquisition: Practical Tools and Techniques*, Boston, MA, 1990.

H. Eriksson: Specialized Knowledge Acquisition Tool Support Compared with Manual Development – a Case Study. To appear in *Proc. of the 7th IEEE Conf. on AI Applications,* Miami Beach, FL, 1991.

J. Harrius: Text Generation in Expert Critiquing Systems Using Rhetorical Structure Theory and Conceptual Graps. *5th Annual Workshop on Conceptual Graphs*, ECAI 90, Stockholm 1990.

S. Hägglund: Emerging Systems for Computer-Based Knowledge Processing in Office Work. In *Office: Technology and People,* 4, pp 135–145, 1989.

S. Hägglund: Iterative Development and Adaptive Maintenance of Knowledge-Based Systems, in *Office Information Systems: the Design Process*, B. Pernici and A.A. Verrijn-Stuart (Eds.), North-Holland, 1989.

S. Hägglund: The impact of intelligent systems on office procedures and knowledge management. Invited paper, *Proc of the Int. Conf. on Opportunities and Risks of Artificial Intelligence Systems*, Hamburg, 1989.

P. Kuczora, P. Eklund: A Conceptual Graph Implementation for Knowledge Engineering. In: *"Human and Organisational Issues in Expert Systems*", A. Hart and D. Berry (Eds), Chapman and Hall, 1990.

J. Löwgren: An Architecture for Expert System User Interface Design and Management. In: *Proc. SIGGRAPH Symposium on User Interface Software and Technology (UIST'89),* Williamsburg, VA, 13–15 Nov. 1989. New York: ACM Press.

J. Löwgren, T. Nordqvist: A Knowledge-Based Tool for User Interface Evaluation and its Integration in a UIMS. In: *Proceedings of Human-Computer Interaction – Interact'90,* Cambridge, UK, August 27–31 1990. Amsterdam: Elsevier.

I. Rankin: Rhetorical Structure Theory in Text Generation. In: *Proceedings of the Second Nordic Conference on Text Comprehension in Man and Machine*, Täby, Sweden, 1990.

K. Sandahl, A. Brogren, L. Reshagen: Transferring Knowledge from an Active Expert to an End-User Environment. *Fifth Banff Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff, Nov. 1990.

K. Sandahl, H. Eriksson, T. Padron-McCarthy, T. Sokolnicki, B. Österlund: Meeting the Requirements of Knowledge Management. Accepted for publication in *Expert Systems with Applications*.

T. Sokolnicki: Towards Knowledge-Based Tutors: A Survey and Appraisal of Intelligent Tutoring Systems, accepted for publication in *the Knowledge Engineering Review,* 1990.

Y. Waern, S. Hägglund, J. Löwgren, I. Rankin, T. Sokolnicki, A. Steinemann: Communication Knowledge for Knowledge Communication. Accepted for publication in *Int. Journal of Man-Machine Studies*.

T. Zhang, L. Padgham: A Diagnosis System Using Classification. *5th International Symposium on Methodologies for Intelligent Systems*, Knoxville, Tennessee, 1990.

A selection of additional reports 1989/90:

P. Eklund: Job Shop Scheduling and AI Planning. ASLAB Memo 90-02, 1990.

H. Eriksson: A Domain-Oriented Tool for Knowledge Acquisition and a Comparison with Manual Development. ASLAB Memo 90–04, 1990.

S. Hägglund, P. Larsson, Kunskapssystem vid produktframtagning, Sv. Mekanförbund, 1989.

S. Hägglund, m fl, Dialogergonomi – Effektiv interaktion människa-dator, Sv. Mekanförbund, 1989.

S. Hägglund, A Note on the Communication Paradigm of Knowledge Systems. ASLAB Memo 89–06, 1989.

S. Hägglund: Kunskapssystem och intelligenta handböcker. – Lärande och kompetensutveckling i datorstött arbete. In: *Proceedings NordDATA 90,* 1990.

J. Harrius: The Constructive Use of Rhetorical Structure Theory in Text Generation, Master's thesis, Linköping University, 1989.

J. Löwgren, T. Nordqvist, S. Löf: Knowledge-Based Support for User Interface Evaluation in User Interface Management Systems, Report LiTH-IDA-R-89-32, 1989.

J. Löwgren: Phenomenology Philosophy and its Implications for AI. ASLAB Memo 89-01, 1989.**T. Näslund:** Utvärdering av beslutsstödsystem. In: *Proceedings NordDATA 90, 1*990.

T. Näslund: Evaluation as a means for improving development of decision support systems. Presented at the doctoral consortium, *11th Int. Conf. on Information Systems*, Copenhagen 1990.

T. Padron-McCarthy, IXEO – A System for Symbolic Representation and Interpretation of Ion Exchange Chromatograms. ASLAB Memo 89-02, 1989.

T. Padron-McCarthy: Negative Knowledge – On the Use, Representation and Application of Negative Knowledge in Knowledge-Based Systems (Extended Abstract). In: *Proceedings of SAIS-90, Ninth Annual Workshop and Meeting of the Swedish AI Society,* Stockholm, 1990.

T. Padron-McCarthy: Metoder för korrekturläsning. ASLAB Memo 89-10, 1989.

K. Sandahl, H. Eriksson, T. Padron-McCarthy, T. Sokolnicki: A Strategic Framework for Knowledge Management. ASLAB Memo 90-01, 1990.

A. Steinemann, Acquiring and Using Knowledge from Multiple Experts; Potential of Critiquing Systems. ASLAB Memo 89-07, 1989.

W. Swartout, H. Nordin, C. Paris, S. Smoliar, Toward a Rapid Prototyping Environment for Expert Systems, invited paper German Workshop on AI, Geseke, 1989. (Work by Nordin at ISI/USC.)

CADLAB

Krzysztof Kuchcinski
director

4
The Laboratory for Computer-Aided Design of Digital Systems

Digital systems synthesis and analysis
Computer-aided design
Computer architecture
VLSI

Back row: Zebo Peng, Bodil Mattsson-Kihlström, Erik Stoy, Tony Larsson.
Front row: Mikael Patel, Björn Fjellborg, Mats Larsson, Krzysztof Kuchcinski.

4.1 Introduction

The laboratory for Computer Aided Design of Digital Systems, CADLAB, is concerned with the computer aided synthesis and verification of digital systems, especially those involving very large scale integrated circuits (VLSI). The major effort of our research work concentrates on the behavioral and structural aspects of digital system specification, design, simulation, optimization, partitioning, synthesis and formal verification methods.

In the reported time CADLAB was broadly concerned with several aspects of the automatic hardware design problems. The main area of interest concentrated on the *high-level synthesis* problem which is defined as the process of translating a behavioral description of a digital system into a register-transfer level VLSI implementation. Such a transformation should be carried out so as to preserve the semantics of the algorithm and at the same time to meet certain cost/performance constraints. Therefore it is a quite complex problem and in practice exhaustive searching for optimal implementation is impossible. To address the complexity problem, several methods are proposed. First, the design space can be cut if we assume a given class of system architectures. Second, an intermediate representation of the design can be introduced to form a base for different optimization strategies. Finally, some stepwise refinement method can be utilized.

Our research activities concentrate mainly on different methodologies and computer-aided design tools which constitute a complete hardware design environment. The main field of interest include synthesis and verification problems. The synthesis is defined by us as a high-level synthesis and includes the general synthesis methods which can be applied to high-level behavioral descriptions as well as special design styles, like synthesis of pipeline structures and design for testability. The verification part is intended to create methods for proving the correctness of manually or automatically synthesized hardware against its specification.

4.2 Laboratory Members and Guests

The following persons were involved in the CADLAB research project during the reported time:

Laboratory leadership and administrative staff:

Krzysztof Kuchcinski, Ph.D., acting professor
Bodil Mattsson-Kihlström, laboratory secretary

Laboratory members having a Ph.D. degree:

Zebo Peng, Ph.D., assistant professor

Graduate students:

Björn Fjellborg, Tech. Lic.
Mats Larsson, MSE

The work in CADLAB is mainly supported by STU, The Swedish Board for Technical Development

Mikael Patel, Tech. Lic.
Göran Rydqvist, MSE
Erik Stoy, master thesis student

Associated persons:

Tony Larsson, Ph.D.

In addition the following guests from Department of Computer Engineering, Tallinn Technical University, participated in some research at CADLAB:

Raimund Ubar, Ph.D., *guest researcher* (January 15 – February 15, 1990)
Peeter Ellervee, *guest graduate student* (March 15 – June 15, 1990)

The following degrees were awarded to CADLAB members during 1989/90:

Tony Larsson completed his Ph.D. dissertation entitled *A Formal Description and Verification Method*.
Björn Fjellborg completed his licentiate degree. The title of the thesis was *An Approach to Extraction of Pipeline Structures for VLSI High-Level Synthesis*.

18 technical papers were accepted for publication or presented on local and international conferences by CADLAB members during 1989/90.

4.3 The Objectives of the Project

Digital systems are becoming more and more complex due to advances in VLSI technology. The complexity of today's microprocessors reaches 1 million transistors and it is expected that their complexity will reach the level of 50 million transistors in the year 2000 with a projected performance of 2000 MIPS. As the complexity of such systems grows, it can be expected that existing design methods will not be able to cope with complex VLSI designs. New methods must be developed for digital systems design which will support higher levels of digital systems design than logic level or even register-transfer level.

The problems pointed out above can be solved by raising the synthesis level for hardware systems. Traditional synthesis systems are usually able to perform synthesis on a register-transfer or gate level and generate an optimized design together with masks (or equivalent specification) for production. The input specification and the synthesis method for such systems are tightly dependent on the technology used. To avoid technology dependent specifications and synthesis methods at an early design stage we investigate a synthesis level called *behavioral* or *high-level synthesis*. The synthesis system on this level will accept a behavioral specification in a form of an algorithm and generate a register-transfer implementation. This implementation can be further synthesized using traditional synthesis methods.

The high-level synthesis issue is the main part of CADLAB's research project *Automatic Synthesis and Verification of VLSI Digital Systems.* The lower synthesis levels, like logic and layout synthesis, are not considered in the framework of the project. We assume the availability of back end synthesis tools which can perform the synthesis task on this level. In this framework our synthesis tools form a front end to a complete synthesis system.

The inputs to a high-level synthesis system usually consist of three components:

- a high-level behavioral specification of the VLSI system to be designed;
- a set of design constraints concerning, for example, cost, performance, testability, etc., and a cost function which specifies what the criterion for optimization is;
- a module library which captures the available hardware modules to be used and their cost/performance attributes.

The output of the high-level synthesis process usually consists of a register-transfer level data-path structure and a description of the control function. It may also include some implementation attributes which can be used by the lower-level design tools.

To achieve the high-level synthesis goal, i.e., to generate a register-transfer implementation structure, a set of basic issues must be addressed by a system. The first issue is *operation scheduling*. It deals with the assignment of each of the operations in the behavioral specification to a slot on the time axis corresponding to a clock phase or time interval. The aim of event scheduling is to minimize the number of control steps or amount of time needed for completion of the specified algorithm.

The second issue to be addressed is *resource allocation*, which selects hardware structures to implement the given functions. For example, an ALU can be allocated to perform an addition operation. The allocation can be carried out by two allocators, the data-path and the control allocators. The former is responsible for the selection of registers, memories, operators, and their connection to form a data-path which is able to perform the specified functions. The control allocator, on the other hand, analyses the description of the hardware data path and description of the desired behavior to produce a description of a control mechanism which evokes the data path in an order consistent with the behavioral description.

The third issue is *design transformation*, i.e., to change a design to achieve a goal or meet a constraint. Design transformations can be done at various stages of the synthesis process as well as in different domains of the design representations. The basic objective of design transformations is to improve the design. In the extreme case, the design is transformed to (globally) optimal solution, which is another issue of the high-level synthesis. The issue of optimization is often though distributed among other synthesis activities and spans the whole synthesis spectrum.

The other issues of the high-level synthesis are *composition* and *decomposition*. The former deals with the situation when there is no direct mapping between functions and structural components. The latter is used to divide a design into a set of modules so that each module can be implemented independently.

Different ways to address these basic issues have resulted in the development of different high-level synthesis systems. A large body of knowledge and techniques has also been developed to solve some of the problems. For example, the CMU-DA project at Carnegie-Mellon University has reported a set of working algorithms addressing either an individual issue or several issues together. However, problems remain. One of the most difficult problems is that the different issues of high-level synthesis are closely interrelated and depend on each other. If they are solved separately as most of the existing algorithms do, the global optimization cannot be achieved.

The other part of CADLAB's research project deals with hardware verification issues. The verification, as defined in the framework of our project, can be seen as complementary to the previously described synthesis. The automatic synthesis will, by definition, produce correct hardware as long as all synthesis methods and algorithms can be proved to be correctness preserving. On the other hand if we will perform some synthesis steps manually or we would like to prove the correctness of automatic synthesis system we need formal methods to support verification of hardware.

There exist many different verification methods. They can be used in many different ways to prove special properties of the designed hardware or to show that an implementation follows the specification. The work done in this project can be used for both purposes but is mostly used to prove equivalence between two descriptions of the same hardware namely its specification and implementation description. The approach forms a basis for mechanized verification of hardware systems. The verification system has been built and successfully used for hardware examples described on different abstraction levels.

4.4 Current Research Projects

Our research activities at CADLAB in the synthesis subject have so far been concentrating on how to improve the chance of getting a globally optimal solution during a synthesis process. Two projects, the CAMAD system and the TAO approach addressed this particular problem. Another work on pipeline extraction from ordinary algorithm descriptions forms a preprocessor for the CAMAD system. It resulted in a prototype system, PiX (Pipeline Extractor). The results produced by the system are very encouraging and form a basis for further research. Some work has been done also on the issue of design for testability.

Another part of our activities include the work on formal methods which can be used for hardware verification. The prototype verification system, which uses an extension of predicate logic with a temporal operator, has been implemented and successfully used to verify a number of different hardware examples.

4.4.1 Design Representation and Synthesis

The different tasks which must be performed during a high-level synthesis process, such as design partitioning, operation decomposition, resource allocation, scheduling, and resource binding, are usually performed in a sequence. This, however, does not guarantee good results since the tasks are interrelated. Our approach, in some sense, generalizes previous approaches. We make a clear distinction between transformations of the design and the optimization algorithm used. The transformations can be applied to the design representation in any order but they are chosen by the optimization algorithm. The transformation can capture either one aspect of the synthesis, for example, register allocation, or they can deal with several problems at the same time, such as allocation, scheduling and module binding.

To make this strategy possible we have developed a design representation which can be used throughout the whole design process. Actually we have two different design representations which are under validation in our research environment. The first one, Extended Timed Petri Nets (ETPN) has been developed by Zebo Peng and is used in the CAMAD design system (see for example [2], [3]). The second one, the TAO graph design representation (Task, Algorithm, Operation graphs) is under development by Mikael Patel [4]. It will make a basis for another design system. Both representations are in some sense complementary. The TAO graph, for example, may be transformed into an ETPN and thus use CAMAD as a back end.

The CAMAD Synthesis System

The CAMAD system is based on a unified design representation, the Extended Timed Petri Net (ETPN) notation, which consists of separate but related models of control and data parts. Petri nets are used here to represent the control flow, which allows explicit expression of concurrency and parallelism. The data part of the design representation, on the other hand, is modelled as a directed graph. Nodes of the graph are used to represent operations on different levels of granularity. The arcs are used to model the data flow between the operations.

The synthesis process of CAMAD is formulated as a sequence of simple transformations each of which preserves the semantics of the designed system. The problem of selecting a particular transformation to be used in each synthesis step can then be formulated as an optimization problem. To solve this optimization problem, a design space exploration strategy and some heuristic algorithms have been developed. The basic idea is to use the critical path for performance to guide the selection of transformations, which is supplemented by other heuristics concerning, for example, locality of computation, communication frequency, resource bound, etc.

The formulation of the unified design representation model also provides a framework for incorporating a set of different design methodologies and tools in a coherent way. In our approach, different design tools all interact with the ETPN design representation which functions as a common design data base. Therefore they can freely communicate with each other and the effect of one design algorithm can immediately be visible to the others.

The formulation of the ETPN design representation and the basic framework of CAMAD were described in Zebo Peng’s Ph.D. dissertation. The current work on CAMAD is intended to integrate more design decisions into the optimization process. Examples of such design decisions deal with synchronicity/asynchronicity trade-offs, module-library component selection, design verification, and testability issues. Some results of the current work have been published in several papers by Zebo Peng and Krzysztof Kuchcinski (see for example [2], [3], [5]).

The TAO Design Representation

The TAO (Task, Algorithm, and Operation graphs) notation is a design representation for high level synthesis, which captures the different types of parallelism on separate levels so that the synthesis process is simplified. TAO is targeted to handle algorithm specification written in languages that allow declaration of concurrent tasks, such as

VHDL, Ada, and others. By separating the types of parallelism and using a design representation which is hierarchical, it reduces the time complexity of many of the graph algorithms used throughout the synthesis process.

The design representation TAO is a hierarchical structure of graphs (see for example [4]). The highest level, *the task level*, expresses a set of communicating concurrent processes. The behavior of each task is described by the next level, the *algorithm level*. This level expresses the overall control structure of the algorithm. Again the behavior of each node in the algorithm graph is described by a lower level. The next, and last level, is *the operation graph*. This graph expresses the partial ordering of primitive operations.

The TAO design representation introduces a new level of abstraction, namely the task level. The motivation for this level is to achieve higher levels of parallelism where independent, asynchronously communicating tasks can be run in parallel. This kind of parallelism, which can be specified directly by a designer (by for example VHDL processes), will improve the whole system performance much more than low level parallelisation of data independent operations.

Each of the three levels in the TAO representation allows different sets of transformations during the synthesis process. The lowest level, operation graphs, allows functional parallelism to be exploited. Several operations may be scheduled within the same clock cycle if they do not violate design constraints. By transforming the algorithmic level the size of the operation graphs may be increased and thus the level of available parallelism. Also classical block level compiler optimization techniques are applicable. The highest level, the task level, allows partitioning into physical devices and links by evaluating feasible clusters of the tasks to form physical units.

4.4.2 Pipeline Extraction

One of the concerns in high-level synthesis is how to efficiently exploit the potential concurrency in a design. The approach to synthesis by applying a set of transformation represents a general method for performing architectural trade-offs. It does not include, however, very specialized design styles, such as pipelines which can achieve a high degree of concurrency and a certain structural regularity through exploitation of locality in communication. This fundamental technique for high speed computer system design is, however, sensitive to the system environment and prone to design errors. The method for automatic extraction and synthesis of parts feasible for pipelining from high level behavioral design descriptions can improve a designed system's performance and quality. This subproject is a step in this direction.

Pipeline extraction localizes parts of the design that can benefit from pipelining. Such extraction is a first step in pipeline synthesis. While current pipeline synthesis systems are restricted to exploitation of loops, this project addresses the problem of extracting pipeline structures from arbitrary designs without apparent pipelining properties. Therefore, an approach that is based on pipelining of individual computations is explored. Still, loops constitute an important special case, and can be encompassed within the approach in an efficient way. Due to the combinatorial explosion of the design space, an iterative search strategy is proposed to perform the pipeline extraction task.

The implementation of the above ideas resulted in the PiX (Pipeline Extractor) system [6]. A specific polynomial-time algorithm based on an iterative search strategy, using several additional heuristics to reduce complexity, has been implemented in the system. It operates as a preprocessor to the CAMAD design system. The input to PiX is an algorithmic description in a Pascal-like language, which is translated into the Extended Timed Petri Net (ETPN) design representation. The extraction is achieved by analysing and transforming ETPN descriptions [7]. Preliminary results from PiX show that the approach is feasible and useful for realistic designs. The designs which have been processed by PiX include a FIR filter, a square root algorithm and the Data Encryption Standard algorithm.

4.4.3 Design for Testability

An important aspect of the design process is design for testability. It means a special design style allowing easier testing of the final circuits. There exist well known techniques for improving the testability of designs but they usually apply only to lower levels of the design representation. There are no results which can be directly used by high-level synthesis systems. In the framework of our project we are proposing to include design for testability into our high-level synthesis system. The first step in this direction is the definition of testability factors and an algorithm which computes them. Having these factors computed we can improve the system testability by performing specialized transformations in parts of the design with the highest testability factors (most difficult to test parts).

The testability analysis is achieved by defining a quantitative measurement of testability for the ETPN design representation [5]. The measurement gives both the controllability and observability of each connecting point in the data path. Both the controllability and observability are in turn defined by a combinational factor and a sequential factor. The combinational factor measures the complexity of the hardware circuits involved to control a specified data (for controllability) or to observe a specified output value (for observability) at some given point. The sequential factor, on the other hand, measures the corresponding time steps (e.g., number of clock cycles) needed to control or observe the given point. In the ETPN notation, the combinational factor is mainly determined by the structure of the data path. The sequential factor, on the other hand, is based on the properties of the Petri net. We have developed a set of heuristics for computing testability factors.

The proposed method for computing testability factors is intended to be used in the early stage of the design process to make proper decisions regarding testability. These factors can be used together with other design quality measures, such as cost and performance, for making proper optimization decisions. From the high-level synthesis' point of view, the testability factors for the ETPN design representation create the basis for improving observability and controllability of the design by the use of some existing testability improvement methods, such as scan techniques.

4.4.4 Hardware Description and Verification

Design of correctly working hardware systems involves the description of functional, structural and temporal aspects at different levels of abstraction and the verification of the requested equivalence between descriptions. This process is usually very time consuming and its simplification is a desirable aim.

To provide a good verification environment it is important that the description language and the verification method can be used at as many abstraction levels as possible and that previous results can be reused. Further in order to support formal reasoning about hardware circuits and their correctness, it is preferable if the description method is based on a well-founded formalism.

The work in this subproject have used an extension of predicate logic with a temporal operator which make possible to specify both functional, temporal and structural properties of hardware circuits. The temporal reference operator makes it possible to describe and reason about relationships between the streams of values observable at the ports of a hardware circuits. The intended meaning of this temporal operator has been defined by a set of axioms and by giving it an interpretation vis-a-vis a temporal model. Based on these axioms it is possible to formally reason about temporal relationships.

This led to the major goal, i.e. to provide support for a further mechanized verification of hardware based on transformation of boolean, arithmetic, relational and temporal constructs expressed in the description language (see for example [1]).

Important contributions of the work are methods for multi-level hardware description and methods for mechanized verification including functional, structural, and temporal aspects that can be used as a complement to existing theorem proving systems. A prototype implementation of the description language based on the generalized (untyped) predicate logic has been done together with an implementation of a verification system as a part of this research. A detailed description of this work can be found in Tony Larsson’s Ph.D. dissertation.

We have also studied the integration of symbolic simulation into the formal verification system. By this we will unify static and dynamic reasoning in a coherent environment. Part of this work has been defined as Mats Larsson's Licentiate Thesis project.

4.5 Related Activities and External Cooperation

CADLAB is involved in the graduate courses program of IDA. During the reported time our laboratory contributed to the program by giving three courses. Tony Larsson gave a course “Formal Hardware Description and Verification”. During the spring 1990 the course “Introduction to Petri Nets” was given by Zebo Peng and Krzysztof Kuchcinski. Finally, Prof. Raimund Ubar from Tallinn Technical University gave a course “Test Generation Methods for Digital Systems” during his visit in our laboratory. Additionally, a series of research seminars on digital systems design automation were given every week by members of CADLAB.

In terms of international cooperation we have established contacts with Tallinn Technical University. Professor Raimund Ubar visited CADLAB for one month at the beginning of 1990. Peeter Ellervee, a graduate student, from the same department has been visiting us during the spring for a period of three months. Dr. Wolfgang Schade from Karl-Weierstrass-Institut für Mathematik der Akademie der Wissenschaften der DDR visited CADLAB for several weeks in May, 1990.

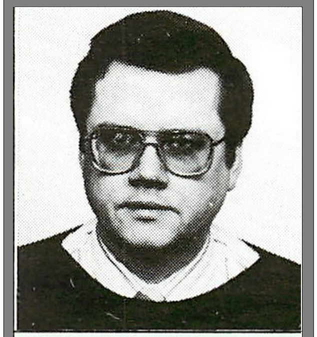
4.6 References

The following are selected CADLAB publications that are referenced in the text. For the full list of publications, please refer to Appendix E.

1. **T. Larsson:** Hardware Verification based on Algebraic Manipulation and Partial Evaluation, *Proc. of IFIP WG 10.2 International Working Conference on "The Fusion of Hardware Design and Verification", Glasgow, Scotland, July 3–6, 1988.*
2. **Z. Peng, K. Kuchcinski, B. Lyles:** CAMAD: A Unified Data Path/Control Synthesis Environment, *Proceedings of the IFIP TC10 Working Conference on Design Methodologies for VLSI and Computer Architecture, Pisa, Italy, September 19–21, 1988, Ed. D.A. Edwards, North-Holland, 1989.*
3. **Z. Peng, K. Kuchcinski:** A Multi-Level Optimization Strategy for Behavior-Entry Synthesis, *Proc. of the Fourth International Workshop on High-Level Synthesis, Kennebunkport, Maine, USA, October 15–18, 1989.*
4. **M. Patel:** A Design Representation for High Level Synthesis, *Proc. European Design Automation Conference (EDAC'90), Glasgow, Scotland, 12–15 March 1990.*
5. **K. Kuchcinski, Z. Peng:** Testability Analysis in a VLSI High-Level Synthesis System, *Microprocessing and Microprogramming, The Euromicro Journal, vol. 28, no. 1-5, March 1990, pp. 295–300.*
6. **B. Fjellborg:** A High-Level Synthesis Tool for Exploiting Pipelines in Special-Purpose Systems, *IFIP Working Conference on Logic and Architecture Synthesis*, Paris, 30 May – 1 June 1990.
7. **B. Fjellborg:** A Petri Net Model for Pipeline Scheduling, *Sixteenth Symposium on Microprocessing and Microprogramming,* EUROMICRO'90, Short Notes Session, Amsterdam, August 27-30, 1990.

CAELAB

5
The Laboratory for Computer Assistance in Engineering



Anders Törne
director

Design
Manufacturing
Industrial Robots
Engineering Databases

Sture Hägglund, Anders Törne, Olof Johansson, Peter Loborg, Jerker Wilander, Martin Sköld.
Missing: Gunilla Lingenhult.

Figure 5-1. *Exterior view of the E++ building.*

5.1 Summary

The laboratory for Computer Assistance in Engineering is a new research group at IDA and was formally started in July 1989. The research of the laboratory is interdisciplinary between Computer Science and AI in particular and Engineering, especially mechanical engineering. In this context, the engineering process is the representation, generation, communication and storage of the necessary information to control, supervise and support the manufacturing process.

In the laboratory the work consists of applying results from Computer Science research to assist this process and from the experience initiate and inspire research in Computer Science. The laboratory has, initially, two directions of research:

- Engineering databases
- Computer Support for Automation.

Within the laboratory there are two projects supported by CENIIT, a special program for interdisciplinary research within Linköping University and one project, in cooperation with the Department of Physics and Measurement Technology on robotics and sensory fusion/integration. The latter project is supported by STU, the Swedish Board for Technical Development. Furthermore there is the ARAMIS-project which functions as a research platform for some of the other projects in CAELAB.

5.2 The laboratory research

5.2.1 General

An industrial production process can be divided into two main activities:

- The physical activities of production machines and processes (production or manufacturing)
- The generation of the needed information to control the physical activities (engineering)

Engineering, in the above sense, can furthermore be divided into design and production engineering activities and may be further subdivided. Most of the traditional engineering sciences contribute to the understanding of how to generate the needed information in the engineering process.

The research in our laboratory is focused on the architectures and representations needed in systems for computer support in the engineering process or more specifically, principles for software construction in this area of computer applications. This does not include numerical algorithms. In practice the work is done by case studies, the subsequent generic analysis of the useability of the architectures and representations and the development of special architectures and representations for this application domain.

The work in CAELAB is mainly supported by CENIIT (The Center for Industrial Information Technology), Linköping University and by STU (The Swedish Board for Technical Development).

Our general view of the engineering process is one of a coupled network of subprocesses, each one generating a refined description of the information needed for the control of the physical activity. In one end (the top) of this network the preliminary design is made from market needs of customer orders, and in the other (the bottom) the final control information, like operations lists and NC-programs, are generated. In this network there is also a flow of information in the opposite direction, from the physical process. This is information about performance, changes in equipment etc., which may initialize regeneration or correction of the information flowing in the other direction.

The management of these information flows is studied in the CENIIT-project “Engineering Databases”. This includes both database architectures and tools for the presentation of the information in the database. One important aspect of the presentation which is studied is the support for browsing and search in the database. Another is the use of the database as a part of the control system for the physical process or, as we prefer to call it, a real-time database.

The focus of the CENIIT-project “Computer Support in Automation” is on the control of the information flows described above. One of the great problems in the manufacturing industry is to maintain the consistency of the information at all levels in the network. When a change is made in a design, the control information for the physical process must be updated in those parts which depend on the changes made in the design. On the other hand, if the physical environment changes it might be necessary to change the control information for optimal performance. This project studies architectures for support systems to simplify the regeneration of information when these inconsistencies appear. An example of this would be a configuration expert system, a generating tool for robot programs or general mechanisms in a database where the information resides.

A part of the latter project above has received STU-support, within the program for “Adaptive Manufacturing Equipments”, to realize a task programmed industrial robot with sensory fusion. The project is called “Instruction and Control of Adaptive Manufacturing Equipments – Task and Execution Adaptivity”. The work is done in cooperation with Prof. Alexander Lauber at the department of Physics and Measurement Technology at Linköping University.

5.2.2 Method

The research consists of case studies, the generalization of the experiences from the case studies into models for the architecture and representation used in support tools and further development of the models used.

The case studies may be made in cooperation with other departments at the university, other institutes and industry. A lot of the research going on in the field consists of taking methods from computer science and applying them to known problem in the field of manufacturing engineering. These methods are originally developed for other applications and it is not directly inferred that they can be used in their original form. An example is the use of databases to store technical information, which poses different questions than the traditional use of databases, e.g. long transaction times and structured dataobjects. The straightforward application research is of course legitimate from the view of the application field. The research of CAELAB, however, stresses the feedback of experience from applications into computer science research. Presently the focus is on AI and object-oriented methods.

The work in the laboratory demands interest for and understanding of a broad variety of traditional subjects – from control theory and logic to sensor physics and production technology. Although broad knowledge is not necessary for a newcomer to the laboratory, it will be naturally acquired in the research projects and the post-graduate education.

5.3 Laboratory members

Laboratory leadership and administrative staff:

Anders Törne, PhD, associate professor
Gunilla Lingenhult, secretary

Graduate students:

Olof Johansson, MSc.
Peter Loborg, MSc.
Martin Sköld, MSc.

Associated persons:

Sture Hägglund, PhD, professor
Jerker Wilander, associated researcher (20%).

5.4 Projects

5.4.1 ARAMIS – taskoriented programming of manufacturing environments

Peter Loborg, Martin Sköld, Anders Törne

Although much effort has been spent on the problem of how to program an industrial robot in a flexible manufacturing cell, no suggestion so far seems to give essential benefits in operator useability and transparency compared with traditional methods. In our opinion this is due to the fact that the capabilities of the intended user have been ignored or at least not prioritized.

It is not a trivial problem to instruct an industrial robot in an environment consisting of several machines like transports, processing machines and other manipulators. The machines cooperate and synchronize in a complicated way which makes the problem similar to the problem of process programming in a multi-process computer system. In fact, instructing the robot movements might be a lesser part of the total operator task of coordinating events and actions in the machine environment. This is true in particular for flexible manufacturing cells, where changes in the machine configuration and the task itself makes the programming task exceedingly complex for an operator. It is also not surprising that languages and tools, developed for programmers and system analysts, have not had a large impact in industrial applications of industrial robots, as the users in this case are not too familiar with modern programming tools.

On the other hand such a language must have the expressiveness needed to specify and to give an overview of the behaviour of a flexible manufacturing cell. Since this involves real-time processing primitives like those in normal process programming languages, it is non-trivial how these two demands, expressiveness and transparency, can be satisfied simultaneously.

The broad similarity to computer process programming does not imply that some or the other computer process programming language is preferable for describing the control of the manufacturing process. There are essential differences in what aspects of the programming/instruction task should be supported or not.

We have in the ARAMIS-project decided to investigate how these differences influence the language and tool design, rather than to focus on the application itself. Since user transparency is essential we have chosen to take an AI-based approach, where the system is viewed as a knowledge base and the operator task is to combine the available knowledge of how to process material by the machines in a manufacturing environment. Graphical presentation is considered to be essential for the transparency and the modelling of the ‘program’ should not differ too much from models that are familiar to the operator/production engineer.

This project is used as a platform for other research activities within the laboratory. The STU-project below may be seen as an integration of the ARAMIS-system with a sensor controlled industrial robot at the department of Physics and Measurement Technology. In the CENIIT-project “Computer Support in Automation” it may be seen as the low end programming system for a manufacturing equipment group. This corresponds to a subprocess in the engineering process which takes operation lists for the manufacturing task and generates a control program for the equipment. A change in the operation list or in the equipment imply changes to the program – the aim is to minimize the effort needed to introduce these changes into the program.

5.4.2 CENIIT project: Computer Support in Automation

Anders Törne

The focus of the project is on the control of information flow in the manufacturing industry and specially for the “flexible” manufacturing area. The control aspect includes:

- How to separate the environment independent information from the dependent and what restrictions must be put on the language for describing the information at different stages?
- How, when and why information is not correct, and how corrections are to be made when the executing environment (the manufacturing environment) changes.

Particular interesting areas are computer support for product configuration, process and operations planning, factory scheduling, and control of the physical process at the cell level.

The information flow in the engineering process is modelled as a set of cooperating autonomous agents. Each agent performs a task without direct interaction or intervention from the other agents. The needed interaction is achieved by negotiation between agents about tasks. A task is defined by the presence of one or several information blocks and the non-presence of a corresponding “output” block for a specific type of agent. If several agents can generate the missing block they must negotiate about who is to perform the task. Each one of the activities above is modelled by such an agent.

To establish the research, three issues have been extracted:

- Tools for autonomous control of multimachine environments
- Representation and compilation for control in manufacturing
- Support systems for product configuration

The first issue is related to process control, but for a batch process. This research will involve man-machine interfaces, non-programmer transparency and real time compatibility, focusing on sequential control and exception handling. This issue is currently in part included in the STU-supported project below.

The second issue has to do with formal methods and models for automation. The central question is: How to express manufacturing information in such a way that when the environment changes, the necessary changes to process plans and programs can be directly found and inferred? The issue is related, at least conceptually, to the field of adaptive control for “continuous” processes. Methods exists in computer science for what is called truth maintenance which would correspond to what is needed but on a smaller scale.

The third issue has to do with the merging of functional requirements with production constraints that is done when an order is processed and the product design is more of a framework for configurating the delivery than a conventional design. This involves in particular the application of expertsystem technology and heuristic optimization.

Research in these issues is a first step to achieve understanding for how the goals of the project can be achieved.

5.4.3 CENIIT project: Engineering Databases

Sture Hägglund, Olof Johansson, Jerker Wilander

This area covers those aspects of database research, which are relevant for databases used in support of design, development and maintenance of (large) technical systems. Design support applies in this context to mechanical engineering, as well as to software engineering, electronics design and knowledge engineering in general. Crucial issues for research are effective methods for management of complex structures in a database, extensions to relational databases, object-oriented approaches, conceptual modelling and knowledge engineering for design objects and related processes, and the relationship to administrative databases and information retrieval systems.

Current research focuses on object-oriented techniques with an emphasis on distributed solutions, on intelligent support for engineering design and on technical information systems, including knowledge-based approaches (“intelligent handbooks”). This research is done in cooperation with ASLAB and IISLAB.

During 1990, a new position of professor of engineering databases was created. It is expected that this position will be filled by the autumn semester 1991. The proposed research is indicated under the heading *mediating database access* below.

Object-orientation in database design

In the database area, relational databases have become established as the dominant technology in later years. It has been shown that high efficiency for many classical administrative applications can be combined with an approach, which is based on a general and theoretically well-founded database concept. However, normalized relational databases do not satisfy all demands in typical engineering applications, with a need to support design activities and objects with a complex component structure. Increased interest has thus been devoted to a new generation object-oriented approaches, which also relate to models for knowledge representation used in AI. This includes also support for type hierarchies and mechanisms for inheritance of implicitly defined object properties.

Many theoretically hard problems still remain to be solved in this area, e.g. regarding optimization of data storage and retrieval, definition of a transaction protocol which preserves consistency of concurrent updates, algorithms for distributed databases, strategies for inheritance in type hierarchies, integration of different user/application models, etc.

Most practical CAE systems of today are based on customized database management or simple file management systems. This results in problems with integration and communication of common data. In the long run it is desirable to be able to use generalized database management to a considerably higher degree, allowing re-use of data and well-defined communication of information between different applications.

Our current work in the area involves both fundamental studies of issues in engineering databases and theoretical work on mechanisms for inheritance in object hierarchies (Padgham). Engineering databases are characterized by a large number of object categories related in product and component families, which makes efficient management of such hierarchies important. For practical purposes, however, strict inheritance is not realistic since there are always exceptions which violate the general rules. Padgham’s approach recognize this need by using both necessary and typical properties to characterize objects and relating them in a lattice structure. This work is further described in chapter 7 on IISLAB activities.

Intelligent CAE frontends and user interface issues

Database management in an engineering environment refers to the process of organizing the information which is needed for various tasks and making it available in an effective fashion for different programs and users. A generalized approach includes also the management of different services available in the CAE environment, such as for instance library software and application programs. This represents an interesting challenge in terms of the need for adaptation and integration of techniques from traditional database systems and knowledge-based systems respectively.

An example of this situation has been studied in cooperation with the department for mechanical engineering. The domain is crack growth analysis for calculation of the length of the life of parts used in aircrafts. The task for the knowledge-based frontend is to assist the user in entering information about the crack geometry, the analysis model and a few other things. The frontend checks for completeness and consistency and generates input data and command sequences to existing calculation programs. Such a system has been implemented, in the form of a prototype using KEE on an Apollo workstation. The prototype contains more than 300 frames and 300 rules, which express domain knowledge and other facts.

Partly based on experiences from this project, Johansson's forthcoming licentiate thesis discusses architectural issues in engineering environments for knowledge-based design systems. His work is based on a model-based approach to design support and includes also issues of user interface generation for large object-oriented databases.

Mediating database access

Future computing environments will have large number of work stations connected with communication networks. Workstations will have their own powerful computation capacities which store, maintain, and do inferences over local data- and knowledge-bases, or *information bases*. Each information base is maintained locally by some human operator and is autonomous from other information bases. Data resources, servers, and applications are heterogeneous. The environment needs to support frequent changes and additions to these data and computation resources. Information and control often has to be exchanged between different information bases. We will have a large distributed *information system*, through which it is possible to access data stored in a variety of local forms, integrate the data, and obtain information that is now very difficult to obtain.

In such a dynamic environment it will be inflexible to develop static programs that directly access explicitly named database tables. We will need *mediators* between data sources and the applications that locate information, transform information, combine information, observes changes in information, etc. Our current plans are to investigate methods to model, search, combine, update, and monitor information through mediators in such a dynamic information system.

5.4.4 Project: Instruction and Control of Adaptive Manufacturing Equipments –Task and Execution Adaptivity

Peter Loborg, Martin Sköld, Anders Törne

Adaptivity in manufacturing equipments does not only mean that the equipment may execute operations/tasks in an adaptive manner, i.e. to consider and react on changes or differences in the environment while executing the operation. Another essential adaptive property is the task adaptivity, i.e. fast reconfiguration for different tasks, known or unknown from the beginning. The known tasks can be programmed before-hand and the setup, loading etc. for a new task is taken care of by the runtime system. The new (unknown before) tasks must be added by programming the new task.

The project connects these two notions of adaptivity and is a cooperation project between the research group of Alexander Lauber at the dept. of Physics and Measurement Technology at Linköping University.

The long term goal of the project is to find:

- algorithms and software architectures for sensor behaviour and integration
- language and tools for generation of such software
- structuring of the knowledge about sensor behaviour
- methods for describing and integrating direct sensor/actuator dependence
- languages and tools for task instruction
- principles for transformations from taskprogram to sensor/actuator dependence

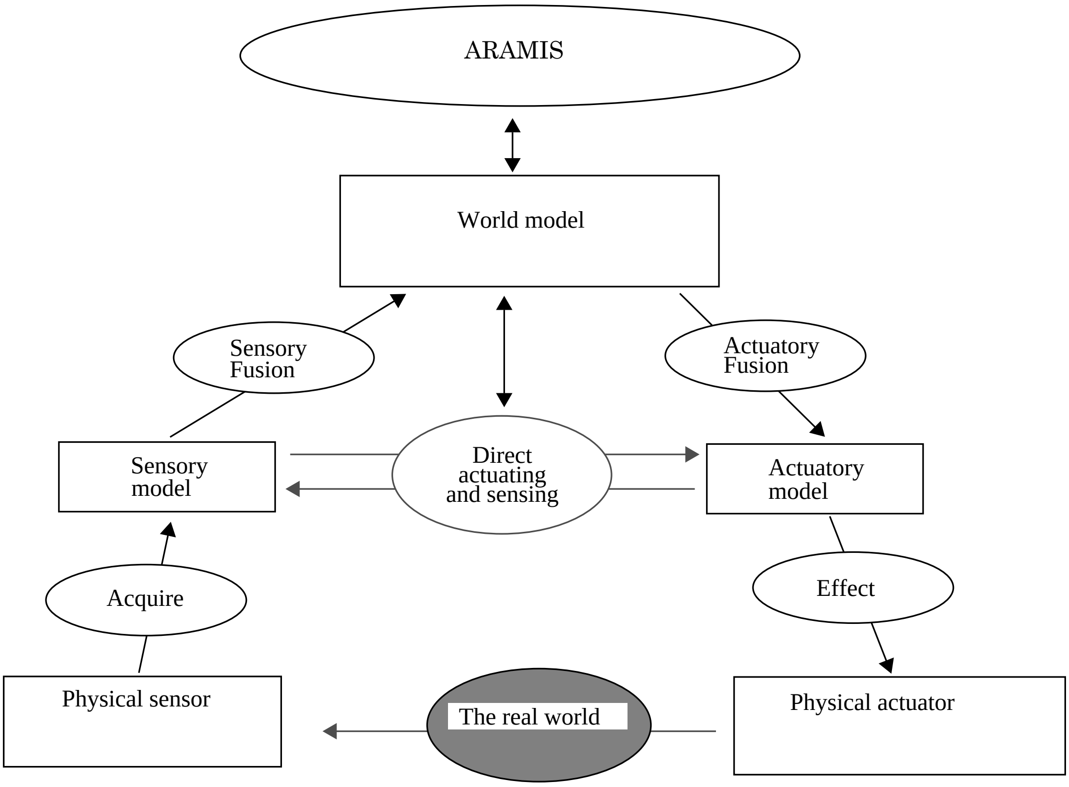


Figure 5-2. *A model of the dataflow and the different subprocesses when connecting ARAMIS to an executing environment.*

The project started 1990 and the initial phase will last until 1991. The goal is initially to connect the ARAMIS-system to an executing environment supplied by the Physics group. This environment consists of a PUMA-robot, a Motorola based control system, a vision system, a tactile and a sonar sensor, and a gripper. The ARAMIS-system is currently being implemented on Sun Sparc-stations and the environment will be increased with a stand-alone Sun-station, connected to the university network.

The project has received support from STU.

5.5 Other activities

A commissioned study project on "Knowledge systems for product development" has been carried out in cooperation with IVF, Epitec AB, Infologics AB and ten manufacturing companies (Volvo, Ericsson, Electrolux, Atlas Copco, etc.). The study was administered by Mekanförbundet and financed by IT4. In the final report 10 case studied in participating companies were reported and discussed. (Sture Hägglund)

A post-graduate course on "CIM - Data and process models" was carried out during spring 1990. The course surveyed different approaches to CIM, i.e. models of corporate information flow, CAD-tools, automatic process planning, factory scheduling and industrial robot programming techniques.

Current activities involve a seminar series on the problems and possibilities for intelligent control of autonomous systems (autumn 1990). This is done in line with the regular laboratory meetings.

Topics of some seminars:

- The Prometheus project – Erik Sandewall, IDA
- Real time databases – Jerker Wilander, IDA/Softlab AB
- Modifying an ABB robot control system – Klas Nilsson, Dept. of Control, Lund University

A joint study group on CIM (Computers in Manufacturing) has been organized by Anders Törne. Other participants come from the dept. of Production Technology at the university of Linköping and the Swedish Institute for Production Engineering Research.

A workshop on "Realtime systems – models, architectures and methods" was organized in Sept. 1990, with about 50 participants from different areas and from several universities in Sweden. About 10 were representatives for swedish industry. The workshop is well in line with software architecture questions in robotics and with the cooperation project with Alexander Lauber. The workshop was supported by STU.

5.6 Relevant publications since 1989

Peter Eklund: Prospects for Conceptual Graphs in Engineering Interfaces, in *Proc. of the 2nd Int. Conf. on Industrial and Engineering Applications of AI and Expert Systems*, Tullahoma, 1989.

Sture Hägglund, Peter Larsson: Kunskapssystem vid produktframtagning: Möjligheter studerade i tio företag. Svenska Mekanförbund, 1989.

Olof Johansson: A Perspective on Engineering Database Research, Report IDA/CENIIT, LiTH-IDA-R-89-14, Linköping University, 1989.

Olof Johansson: An Experiment with a Neural Network for Handwritten Character Recognition, Report IDA/CENIIT, LiTH-IDA-R-89-44.

Peter Loborg, Anders Törne: A hybrid language for the control of multimachine environments, accepted to EIA/AIE-91, Hawaii, June 1991.

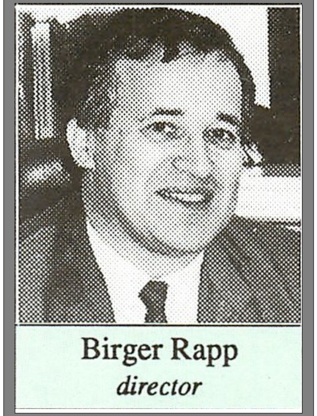
Anders Törne: Computer Assistance in Automation, Report IDA/CENIIT, LiTH-IDA-R-89-51, Linköping University, 1989.

Anders Törne: The Instruction and Control of Multi-Machine Environments, LiTH-IDA-90-07, (also in *Applications of Artificial Intelligence in Engineering V, vol. 2, proc. of the 5th Int. Conf.* in Boston July 90, Springer-Verlag and presented at the AAAI 90 workshop on intelligent diagnostic and control systems for manufacturing, Boston, July 90.

Tingting Zhang, Lin Padgham: A Diagnosis System Using Classification, Report IDA/CENIIT, LiTH-IDA-R-90-13, Linköping University, 1990. Also in *Proc. of the 5th Int. Symposium. on Methodologies for Intelligent Systems*, Knoxville, Tennessee, 1990.

EIS

6
Economic Information Systems



Birger Rapp
director

Management Information Systems
Decision Support Systems
Accounting Information Systems
Manufacturing, Planning and Control Systems
Implementation and Evaluation, IT-strategy
Activity Based Costing
Telework, Organisation Communication
Audit

Back row: Carl-Gunnar Hansson, Lars Mattson, Torbjörn Näslund, Eva Elfinger, Birger Rapp, Maria Olsson, Camilla Sjöström, Merhan Noghabai, Anette Stolt.
Front row: Peter Carlsson, Jonas Lind, Björn Helander, Jörgen Andersson, Anna Moberg, Jaime Villegas.
Missing: Göran Goldkuhl, Lennart Ohlsén.

Figure 6-1. *Interior view of the E++ building.*

The subject area Economic Information Systems has become the focus of increasing interest throughout the last decade, reflecting the growing awareness of the value of information and the importance of adequately developed Information Systems. An expression of this new awareness has been the foundation of a professorship in Economic Information Systems (Linkoping,1987) and two adjunct professorships, one in Linköping (1985) and one in Gothenburg (Economics of Information Technology,1990). The evolution of society towards a "knowledge society" is also creating new conditions of employment and forms of organization, and allocating new roles to the individual. Information Manager and Division Controller are two examples of new kinds of employment area which are relevant to Economic Information Systems. Research is going on, both internationally and in Sweden, in order to reach a better understanding of how information technology can be adapted to the conditions of people and organizations. The role of information technology and its consequences have been given a new emphasis in connection with the strategic questions facing companies. New attention is being paid to the direction of companies and the control of the flow of goods and services, with the help of the new information technology.

Economic Information Systems has connections to both technological and economic subjects. There is also a natural strong common area of interest with the field Computer and Information Science. Furthermore, understanding of information technology is important. Because of the subject's orientation towards applications; areas such as Production Economics, Accounting, Marketing and Transportation Economics will spontaneously have connections to Economic Information Systems. Immediate connections to the subject areas Organization and Psychology exist, because of the questions that arise concerning the way in which people and organizations use different forms of information transfer.

The subject area Economic Information Systems involves, among other things, communication and transfer of information from between and to people, as well as the development of suitable information systems for this purpose. The subject also deals with the use of modern information technology and the development of structures within organizations, together with the effect of information technology on people and organizations. This involves both questions concerning economic direction and control, and the capacity of people to take in and use information.

Some examples of questions which lie within the subject are: what information should a decision maker have access to, and how and in what form should that person acquire the information. Another question is how an information system (a system for economic control, for example) should be developed to fit in with a company's overall approach to business.

At present EIS has at its disposal an IBM 9370 computer and a network of PC's using OS/2. An environment where various large economic systems, office systems, total systems and executive systems can be studied in operation is currently being built up. Big investments in hardware, but even more in software, will be needed to maintain research and undergraduate teaching at a high international level.

At present the subject Economic Information Systems is undergoing a period of intensive expansion. Many doctoral students are applying, and there is a big interest externally (from industrial companies and organizations). There is an obvious need to recruit new staff, and to increase the number of doctoral positions.

6.1 Research Projects in Economic Information Systems

Research within EIS is being conducted under the following main headings: information support, economic control and accounting systems, material and production control, information technology and organizational solutions. Some of the research projects are being carried out in conjunction with Swedish and international researchers and companies.

6.1.1 Information Support

Society and working life are going to change in the coming years. One reason for this is breakthrough of information technology. The “information society” is developing. Consequently, it is important to understand how available information and technology can be used to support decision-making and to know how to evaluate the consequences. Within this project lie questions concerning the introduction, use and evaluation of information systems.

A sub-project is devoted to the introduction of information systems, in particular accounting and office information systems.

Another sub-project is devoted to questions of evaluation. Applications are expected above all in the area of FMS-investments (Flexible manufacturing systems) and office information systems.

A third sub-project is primarily devoted to DTP (Desktop Publishing). Suitable parameters are being sought, which will make it possible to describe changes within DTP and to evaluate the use of DTP within companies.

6.1.2 Economic Control and Accounting Systems

Good economic control is required to be able to run a company or an organization effectively in accordance with stated goals. This requires, among other things, such activities as planning and following up activities. In principle economic control should be able to give ongoing information support about particular matters such as organizational units, products and projects. This information should be available to different interested parties with differing measuring scales, such as returned interest on investment, costs, volume and quality requirements.

An essential support for economic control is accounting, with following up and reporting. Nowadays this is often computerized. Many companies have systems dating back to the middle or early seventies. These accounting systems are naturally primarily organised to suit the situation that prevailed at that time. This entails that they are primarily organised for financial accounting (the external accounting that includes among other things, balance and profit/loss sheets). The laws and practices in force at the time also affected the design of these systems.

Other important supports for economic control are budgeting and management accounting. This includes both product costing and investment calculation. Throughout the eighties companies have changed their organization. There has been a pervasive trend towards subdivision and decentralisation. Decentralisation has created new users of economic systems with responsibility for results, who require different kinds of reports than the traditional reports. Some studies indicate that it is precisely decentralisation (or plans for decentralisation) which is the main force behind the demand for new systems.

The spread of new Administrative Data Processing technology has lead to increasing demands on and criticism of traditional accounting systems and the types of models underlying the application of such systems. The complaints about such systems are that they are inflexible, hard to grasp as a whole, hard to operate, and expensive to use. Many people feel that we are heading for new approaches to economic control, with the help of non-traditional approaches and accounting systems.

Changes in economic control are being studied from this perspective.

A subproject is studying the consequences for accounting systems in various countries of developments in the European Community in 92/93.

Another sub-project is investigating the possibilities of applying relational data-bases in accounting systems.

The possibilities for using Activity Based Costing are being investigated in a number of companies.

6.1.3 Material and Production Control

Information technology has developed quickly in the last years of the eighties, with many changes as a consequence. A number of technological breakthroughs have become practically useable. A number of examples have been reported which show the successful employment of information technology within companies. It is only natural to expect that this development will also affect the development within the area Material and Production Control.

In one project there is a discussion relative to a system of reference (Olhager, Rapp 1985) of trends within modern material and production control. Building from observations of changes within the production function in companies and new technology for MPC systems (Material and Production Control), the project discusses the properties of future MPC systems and the expected content of different modules.

Another project is devoted to joint economic and production control. At present the project is being conducted so that a number of detailed parts within economic, material and production control are being studied.

6.1.4 Information Technology and Organizational Solutions

Our society is undergoing constant change. There is an ongoing concentration of places of work to increasingly bigger urban areas. This leads to a deterioration in the social quality of the environment and diminished productive time for people, who have longer journeys to work and longer waiting times. However there are examples today showing how this trend can be broken, through the use of modern information technology. There has been a growing interest in finding solutions for people which give an acceptable social environment. Besides that, society has an interest in being able to use the working individual effectively throughout the whole working day.

One project is devoted to the organizational effects of the new information technology. Its first phase focuses on remote working. Initially this involves a survey of current research and a study of activities which can be classified as remote working. Remote working is nothing new, in itself. It already existed before the rise of industrialisation. Then people were forced to move to bigger work-places, because of the capital-intensive investments made there. In the project, the concept “remote work” will be given a restricted sense: work that is carried out at a place which is geographically remote from the location of the person(s) responsible for the undertaking. This implies also that new information technology can be used. Remote teaching by satellite or telephone are examples. Other examples are remote programming and factory units which are spread out to several locations and co-ordinated by a central control unit which uses information technology.

In the next phase, the project will investigate the presuppositions and conditions of information technology with respect to the mobility of the individual and the geographical independence of the organization. The purpose is to delimit peoples’ possibilities for using information technology in different working situations, and to study the possibilities and limiting factors affecting the geographical and organizational spreading out of organizations and companies. This can also lead to new organizational solutions.

6.1.5 Development of information systems and supported activities

Göran Goldkuhl

The systems development process does not only involve development of the system itself but also a development of the surrounding/supported activities. It is of great importance that the development process is governed by explicitly stated activity goals (business objectives). There is a need for theories and methods capable of dealing with organizational and information systems issues in an integrated manner.

A research group dealing with this issues is going to be established under leadership of docent Göran Goldkuhl.

One research interest is in the area of change analysis. Change analysis means investigation of problems and goals and decision on change actions, where information systems development might be one such action (“feasibility study”). The research is directed towards studying a specific method (SIM). Empirical studies of methods use and further methods development are performed.

Another research interest is in information requirements analysis. We are working on a method for integrating development of information systems and human activity structure (business functions).

A third research topic is design rationale in systems development. This means how design options can be rationally justified through explicit criteria. Research questions are: How people think in design and how they can be supported by appropriate design concepts and notation.

Other research projects are being planned in the following areas:

- computer-support in systems development (CASE-tools, and especially CASE-shells)
- evaluation of computer-supported activities
- principles and strategies for information systems architecture (how to divide and relate different information systems to each other)

6.2 EIS personnel

Group leadership and administrative staff:

Birger Rapp, professor
Göran Goldkuhl, associate professor
John Andersson, adjunct professor
Eva Elfinger, secretary

Employed graduate students:

Jörgen Andersson, B.Ec.
Peter Carlsson, B.Ec.
Carl-Gunnar Hansson, MSc
Björn Helander, B.Ec.
Jonas Lind, MSc
Lars Mattsson, MSc
Anna Moberg, B.Ec.
Mehran Noghabai, MSc
Lennart Ohlsén, MSc
Maria Olsson, B.Ec.
Camilla Sjöström, B.Ec.
Annette Stolt, B.Ec.
Torbjörn Näslund. B.Sc.

Graduate Students:

Jaime Eduardo Villegas, PhD.
Mikael Isaksson, B.Ec.
Rolf Larsson, B.Ec.
Arne Kjellman, Systems Analyst

6.3 Selected publications 1988-

Mattson L., Olhager J., Ovrin P., Rapp B.: Computerization of Manufacturing Planning, *Engineering Costs and Production Economics*, 17, 1989, 71-78, Elsevier Science Publishers B V, Amsterdam.

Moghabai M.: A Study on the Manufacturing Planning and Control in a Beef Cutting Plant of a Swedish Company, *Engineering Costs and Production Economics*, 17 (1989) 43-53, Elsevier Science Publishers B.V., Amsterdam.

Olhager J., Rapp B.: Setup Reduction in Inventory Turnover Rate, *Int. J. Prod. Res.*, Vol. 29, No. 1, 1991, pp 95–105.

Olhager J., Rapp B.: Modern Concepts for Computer-Based Manufacturing Planning and Control System, *Operational Research ’90*, Ed. H.E. Bradley, Elsevier Science Publishers B V, Amsterdam, 1990.

Olhager J., Rapp B.: Setup Reduction in Production Systems: Review and Extensions, *Operational Research ’87*, Ed. G.K. Rand, Elsevier Science Publishers B V, Amsterdam, 1988.

Ovrin P., Olhager J., Rapp B.: Computer-Based Manufacturing Planning and Control Systems in Small Companies, *European Journal of Operational Research 45*, 1990, pp 106–112.

Rapp B.: Forecasting and Stock Control, *Computer Aided Production Management*, Ed A. Rolstadsås, Springer-Verlag, 1988, Ch. 19, 289–302.

Rapp B.: Operations Research in Sweden, *OMEGA*, *Int. J. of Mgmt. Sci.*, Vol. 16, No 3, pp 189–195, 1988.

IISLAB

Lin Padgham
director

7
The Laboratory for Intelligent Information Systems

Default reasoning
Inheritance
Reasoning with time
Object-oriented systems

Back row: Ralph Rönnquist (Vanitha Padgham), Sven Moen, Patrick Olnäs, Michael Jansson, Christian Ökvist, Bodil Mattsson-Kihlström, Mikael Ohlsson, Ulrik Eriksson, Patrick Lambrix.
Front row: Lin Padgham.

7.1 Overview

The Intelligent Information Systems Laboratory began as a sub-group of Erik Sandewall's Laboratory for Representation of Knowledge in Logic in 1986, where it was known as the LINCKS group. The group became independent in June 1989 and changed name to the Laboratory for Intelligent Information Systems in autumn 1990. Intelligent Information Systems is an area which arises largely in the intersection of the established areas of Databases and Artificial Intelligence. During recent years there has been an increasing interest in this particular interface. Database researchers, users and developers are becoming aware of such things as the need for increased representational flexibility if databases are to be used for applications outside the traditional area of business applications. At the same time many A.I. researchers are finding that complex A.I. problems require access to large amounts of information which must be stored and maintained and in many cases shared between several users and applications. The traditional knowledge bases of e.g. expert systems are not suitable for such things as multiple users and larger amounts of information.

There are many generic problems regarding such things as how to represent and use non-homogeneous and often multiple purpose information in systems which include reasoning components. Such problems and issues, as well as the design of systems which maintain and use shared information for reasoning software, are our focus of interest.

A major part of the work in the group is centered around the LINCKS project. The aim in this project is to gradually build up a sophisticated and complex information system, exploring a number of issues such as appropriate representations for knowledge and information, distributed information, integration of expert system services using a shared database of knowledge and information, user interfaces to complex systems, planning and reasoning about actions, default reasoning and intelligent support for a variety of user tasks. It is also expected that the system built up will provide a testbed system for exploration of future research issues.

There are several strands of theoretical work in the group – particularly default reasoning and reasoning about change. These both receive stimulus from and provide stimulus to the more system oriented project. The theoretical work and the LINCKS project/system are explored in more detail in later sections.

The work in the Laboratory for Intelligent Information Systems is mainly supported by STU, The Swedish Board for Technical Development, with some additional funding also from CENIIT, The Center for Industrial Information Technology, and from STUF, the basic research organ of STU. The group has also received money from FRN (national research advice committee) for equipment under 1990–1992

7.2 Members

The following people have been members of the group during all or part of 1989/90:

Laboratory leadership and administrative staff:

Lin Padgham, PhD, assistant professor.
Bodil Mattsson-Kihlström, secretary.

Graduate students:

Peter Åberg, licentiate June 1989, working with Dr. Bob Neches at ISI since October 1989, planning to return summer 1991.
Michael Jansson, graduate student.
Ralph Rönnquist, licentiate degree 1987.
Patrick Lambrix, graduate student.

Visitors during extended periods:

Nilgün Mat, guest researcher August 1989 – October 1990.
Timothy Teitenberg, guest researcher August 1989 – November 1990.
David Partain, visiting student, summer 1990, returning August 1991.

Employees:

Ulric Eriksson, programming assistant (half-time).
Sven Moen, research engineer, short-term.
Patrick Olnäs, research engineer.
Michael Ohlsson, programming assistant (half-time).
Christian Ökvist, programming assistant.
Ewa Rauch, engineer (until April 1990).

7.3 LINCKS System

A bottom layer for the LINCKS system consisting of an object-oriented database with some basic support for history maintenance and parallel editing was essentially completed in late 1987.

During 1989–1990 there has been another intensive implementation effort to realise some of the group’s ideas regarding user editing in a broad purpose shared database. We expect this work to result in a relatively stable next layer of the LINCKS system during late 1990 or early 1991. This layer provides facilities for extracting and editing conceptual objects from the database of interconnected information chunks. Each information chunk may participate in several conceptual objects, used by different users. The information extracted from, and the appearance of an information chunk in the database varies depending on the conceptual object in which the information is used. Information regarding the appearance and structure of objects as well as the location of information parts in the database is itself stored in declarative form in the database. This layer is intended to give a flexible tool for experimenting with issues regarding multiple views and uses of information. It also provides a powerful window oriented user-interface to the database, making it possible to start experimenting with the notion of information which is both relatively structured for interpretation by computer software (A.I. programs) and relatively free (e.g. documents) for use by human users. The following subsections describe very briefly some of the areas of research interest which are addressed by the current LINCKS system.

7.3.1 History

History information – that is information regarding the development of objects over time – is information which is very often needed if one is to do even quite simple reasoning. However almost all database systems contain information regarding only a single point in time, namely the present.

The LINCKS system maintains information regarding the historical development of each individual information chunk. From this it is then possible to build the development history of conceptual objects which are made up of such information chunks. The history structure is ordered according to logical time rather than strict real time, and is therefore a partial ordering rather than a strictly sequential ordering. Cases where an object is changed on two different workstations without any possibility of one person seeing the other's change (i.e. it is not yet placed in the database) are regarded as cases of parallel change – i.e. the history structure shows no ordering between the two changes, even though it may be possible to determine a real time ordering based on time stamps. We regard this concept of logical time as important for reasoning processes, rather than time-stamp time. The difference becomes most important when one has disconnected workstations which operate on portions of the database during a period of time (perhaps quite long) when they are disconnected.

7.3.2 User Interface Management Systems

User Interface Management Systems focus on declarative specification of a user interface and software for interpreting that specification and integrating the user interface with the application functionality. Such systems usually provide a mechanism for specifying various kinds of interface objects (e.g. menus, text windows, buttons, etc.) and their behaviour. The manipulation of the contents of these interface objects typically resides however within the application.

Our system takes this model one step further and provides the mechanisms for declarative specification of the contents of user interface objects (as well as their type). The system then allows for dynamic manipulation of these contents and consequent updating of the underlying database plus propagation to other screen occurrences of that information. This provides a significantly higher level of flexibility than traditional UIMS's and allows for building of more complex interactive systems.

The ability to declaratively specify and manage behaviour attached to display objects is not yet implemented within our system, but is clearly a part of our model and will hopefully be implemented soon.

7.3.3 Hypertext

Hypertext, or more generally hypermedia is an area of growing research interest. The basic structure of the LINCKS database, combined with the display layer, provides a natural and flexible system for building hypermedia applications. One prototype application (a hypertext medical book) has been developed successfully within LINCKS. The advantages of LINCKS compared to other hypertext systems (e.g. Hypercard, HAM, etc.) are that there is an underlying multi-user database and that the LINCKS system allows greater flexibility in terms of number and kind of link types and attributes. The number of links, link types and attributes is unlimited within the LINCKS system.

7.3.4 Structured Editing

Structured editing is editing which supports and is in some way based on the notion of the internal structure of the edited objects (documents). It is not in itself new and there exist a number of commercial and research systems which support a view of documents as structures of smaller pieces such as sections, paragraphs, titles, etc. However such systems typically require the user to select and deal with one particular ‘chunk’ of the document at a time. Thus the integrated or holistic view of the document as it will eventually appear is lost.

In LINCKS we have a text editor which allows viewing and editing over the entire document (in the style of emacs, wordstar, etc.) at the same time as the notion of underlying structure is maintained and ‘understood’ by the system. Thus we combine the advantages of both a structured and an integrated view of objects. The underlying structure of different object types is stored declaratively within the database, and is interpreted at runtime. This provides a dynamic and flexible system where the structure of a particular object type can be developed interactively.

7.3.5 Object Oriented Databases

We have built a database where it is possible to store small information chunks of different kinds (they can contain text, graphics, animation sequences, sound, etc.) which can then be connected together into larger conceptual objects. For instance a document may be built up of a number of text pieces and graphics pieces with a structure which includes such things as author, title and subtitles, etc. Each information chunk can participate in any number of objects allowing for a high degree of information sharing between objects. This means that when an information chunk is updated in a particular context, it is also updated in other contexts in which it is used.

The database is multi-user and makes use of a distributed architecture of fileserver(s) and workstations. As mentioned previously the database maintains historical information regarding all of its objects.

One of the interesting aspects of the LINCKS database model is that we do not have a hard coupling between actions (or methods or messages) and object types. Rather we see types, actions and contexts as three orthogonal parameters, all of which can be represented as objects and together determine behaviour and presentation of information (or data) objects. This allows for explicit representation of actions as objects together with information regarding those actions. It also helps to avoid anomalies in the type hierarchy resulting from the need to inherit methods [PAD87].

The report ‘LINCKS – An Imperative Object Oriented System’ [PAD87] gives further details regarding differences between our view of object orientation and views that are more commonly propagated based on the message passing paradigm. We argue that we fulfill the essential requirements of object-orientation and that decoupling of types from actions is an important step in obtaining a clean model which is easily manageable for end-users.

7.4 Inheritance, Default Reasoning

The LINCKS group have developed a model for representation of types based on the notion of a type core which contains all necessary characteristics of the type, and a type default which describes the typical member of the type. This model was used as the basis for a theory of default reasoning developed in Lin Padgham’s PhD thesis in June 1989. The theory developed is able to obtain intuitively desirable solutions in a simple manner for many of the classical problem examples from the literature of default reasoning with inheritance. It allows for distinctions between necessarily inherited characteristics and characteristics inherited by default. Such a distinction is critical if one wants a mapping to e.g. default logic. The developed theory also allows for default reasoning about contra-indications, and types probably excluded. The information that a particular subclass is typical for its superclass can also be represented, which is unusual for inheritance network representations. This information is expected to be important in such things as learning systems.

An important aspect of the theory for default reasoning with inheritance is that it has been shown (in Padgham’s thesis) to be implementable using an essentially linear algorithm. During the past year a prototype has been implemented according to the algorithms described in the thesis and it performs as expected. We hope to be able to integrate this prototype into the ongoing LINCKS system within the coming year.

During the past year we have extended the work on default reasoning with inheritance hierarchies to explore classification reasoning using the same representation of type cores and defaults. The classification work has then been extended further and used as the basis for diagnosis in a medical domain. Results appear promising but are preliminary as yet. A prototype classifier and diagnosis system is in the final stages of implementation and we expect to test and report on the system during the coming year. The work on diagnosis has been done in co-operation with ASLAB and the system has been implemented by Tingting Zhang.

7.5 Information System Dynamics

Taking the LINCKS system as starting point, Ralph Rönnquist has been studying formal methods for characterising the dynamics of information systems. The working hypothesis is that information systems differ significantly from physical systems, and that therefore it is not obvious that methods developed for physical systems apply for information systems. Some of the more distinct differences are:

- that an information system is primarily digital, i.e. discrete both with respect to time and to system states. The changes that occur in an information system stem from computations, which means that only some small portion changes at a time and that propagation of changes plays an important role in the dynamics;
- that information entities are less distinct than entities in physical systems, in the sense that the same component may be shared among views and thus appear in many different contexts. The actual appearance of an information component is dependent on the context in which it appears.
- that an information entity may exist in parallel versions, e.g. in different languages or as different alternatives.

This work is aimed towards characterisations of dynamics in information systems, including interface layer changes such as window updates, application layer changes such as maintenance of data dependencies, as well as changes that occur at database level to ensure modelling invariances. One result so far is a first-order temporal logic (LITE) which directly correlates to the structure implemented in the NODE database, and at the same time provides a formal framework for declarative descriptions of processes. The LITE logic is based on the conception that objects occur as time structures of instances. In other words, a functor or object referent in a sentence refers to a set of formal individuals that constitute the various appearances over time of an object. Interpretation is then done relative to a temporal context through which the object referent is resolved into one of the object instances.

By using the LITE logic, we are able to formalise the notion of historical object references (i.e. referring to objects as they were at some time before) as well as providing strict formal semantics for transition graph (or rule based system) execution. The latter may be applied e.g. for characterising communication protocols in such a way that a model structure is a communication session.

7.6 Reports and Papers

The following theses and papers have been written or published during ’89/90. For a fuller list of reports from the group please see appendix.

7.6.1 Theses

Lin Padgham, Non-Monotonic Inheritance for an Object-Oriented Database, 1989, ISBN 91-7870-485-5 (PhD thesis).

Peter Åberg, Design of a Multiple View Presentation and Interaction Manager, LiU-Tek-Lic 1989:17. (licentiate thesis).

7.7 Internationally Published or Accepted Reports

Padgham L., Negative Reasoning Using Inheritance, *Proceedings of IJCAI '89*, Aug. 20–25, 1989, Detroit, Michigan, USA. vol.2 p. 1086.

Padgham L., A Lattice Based Model for Inheritance Reasoning, *Proceedings of the Workshop on Inheritance in Programming Languages and Knowledge Representation*, Viareggio, Italy, February 6–8, 1989.

Rönnquist R., Reasoning about Control of Dynamic Systems, *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, Charlotte, N.C., USA October 12–14, 1989.

Timpka T., Introducing hypertext in primary health care: A study on the feasibility of decision support for practitioners. *Comp Prog Biomed* 1989;29:1–13.

Timpka T, Padgham L, Hedblom P, Wallin S, Tibblin G., A hypertext knowledge base for primary care: LIMEDS in LINCKS. *Proc. of SIGIR* '89, Boston, ACM Publ., 1989.

Padgham L., A Lattice Based Model for Inheritance Reasoning, in *Inheritance Hierarchies in Knowledge Representation and Programming Languages*, John Wiley and Sons, England (to appear).

Padgham L., Defeasible Inheritance: A Lattice Based Approach, C*omputers and Mathematics with Applications*, special issue on Semantic Nets (to appear).

Mat N., Padgham L., Åberg P., Rönnquist R., Jansson M., Teitenberg T., User Interface and Information Management for an Object-Oriented Database, T*echno-Data '90*, Berlin 1990.

Zhang T., Padgham L., A Diagnosis System Using Inheritance in an Inheritance Net, P*roceedings of the International Symposium on Methodologies for Intelligent Systems, Knoxville, USA, Oct. 24–27 1990.*

Rönnquist R., A Logic for Propagation Based Characterisation of Process Behaviour, *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, Knoxville, USA, Oct. 24–27 1990.

LIBLAB

8 The Laboratory for Library and Information Science

Roland Hjerppe
director

User interfaces
Document description
Knowledge organization
Hypermedia
Information management

Barbara Ekman, Arja Vainio-Larsson, Roland Hjerppe, Lisbeth Björklund.
Missing: Andreas Björklind, Birgitta Olander.

Figure 8-1. *Exterior view of the building E.*

8.1 Introduction

Libraries have for a very long time been the main social mechanism for providing access to publications – the experiences and knowledge documented in text and made available to the public.

Information technology, its possibilities for and its effects on library and information work are the broad areas of interest for LIBLAB. One of the objectives is to be able to actively influence and anticipate applications of new technology to the field of information supply and management, including libraries. New media, new forms of publications, and new methods of scientific communication and knowledge organization, and their interaction with library functions are hence of primary interest to LIBLAB.

8.1.1 LIBLAB's Research program

There are two main themes in our research program:

- users and information systems, especially bibliographic systems, and
- document description and representation.

The first theme is oriented toward the use and users of interactive information systems. The user's central role as producer and consumer of information services is emphasized. Within this theme we are focusing on two research areas:

a) user participation and user behaviour, and

b) orientation in databases: maps and other tools.

The second theme is concerned with descriptions and representations of documents and collections and their relations at different levels. The background is computerized catalogues as tools for access to documents. Within this theme we have identified three research areas:

c) the convergence of hypertext and multimedia, Hypermedia,

d) HyperCatalogs, and

e) formalisms for document description within documents and catalogues

8.1.2 Project HYPERCATalog

Project HYPERCATalog is a long-term project where we try to synthesize all the various research efforts of LIBLAB. The HYPERCATalog is a vision, or a metaphor for a complex yet flexible way of giving users access to the information and the structures that are more or less implicit in document collections. Our vision is an information system that is an integrated part of a familiar working environment, a tool for information management rather than merely information retrieval.

For the period 1983–88 LIBLAB was funded by the "Delegationen för vetenskaplig och teknisk informationsförsörjning (DFI)". From July 1, 1988, LIBLAB has been funded by "Forskningsrådsnämnden (FRN)".

8.2 Laboratory Members

Laboratory leadership and administrative staff:

Roland Hjerppe, MSc, researcher
Barbara Ekman, secretary

Graduate students:

Arja Vainio-Larsson, MA., doctoral student since 1984.
Andreas Björklind (starting 1991)
Lisbeth Björklund, BSc, doctoral student since 1985.

Associated member:

Birgitta Olander, BA, MLS., research assistant. Doctoral student at the Faculty of Library and Information Science, University of Toronto, Canada, since 1984, on leave during 1989/90.

8.3 Current Research

The research activities during 1989/90 can be grouped into three broad areas, namely Document architecture, Knowledge organization, and User interaction.

8.3.1 Documents and Their Architecture

Current research in this area, which is part of the programme on document description and representation, is oriented towards two issues: the generalization of the concepts document, text and reading, and the implications of that for the interrelationships between formalisms for mark-up of written documents and formalisms for description in the context of collections and catalogues.

8.3.1.1 Generalizing documents – texts – reading

Information is today conceived of as residing mainly in texts and numbers and the word document evokes associations of writings, letters on paper.

Texts, written forms of language, have since the invention of writing systems, and especially after the revolutions caused by the printing press, become the major form for transmission of information across the boundaries of space and time. The carrier of a text is called a document although document in general denotes “something written, inscribed, etc.,which furnishes evidence or information upon any subject, as a manuscript, titledeed, coin etc.” (The Shorter Oxford English Dictionary on Historical principles 3rd. Ed.)

In the rest of this section on documents and their architecture the words document, text and read will, however, be used in a generalized sense:
document = something which furnishes evidence or information, the carrier of a text
text = something which can be read
to read = to make out the/a meaning of

Documents in this more general sense can be subdivided into cultural documents – the result of cultural processes, and natural documents – the result of natural processes. An example of a natural document would be the stratigraphical features of an escarpment, readable for a geologist. (A collection of natural documents, e.g. mineralogical specimens, can as a collection be regarded as a cultural document.) All the documents are, with different proportions, carriers of a mixture of explicitly accessible, readable, information, experiences (pleasure, entertainment etc.), and cultural heritage.

Libraries and archives are the social institutions that have evolved to cope with a special kind of cultural documents – the multitude of written documents generated through the ages. The contents of libraries and archives are an important part of the cultural heritage but societies can and do sustain without them and the cultural heritage includes much more than that which can be found in written texts – documents in the narrow sense.

Artifacts, small and large, as well as processes, e.g customs and procedures, and structures, e.g. of organization, are important cultural documents carrying texts which, however, are not as easily read as ordinary – written – text. The information that can be elicited from them by a trained reader is usually transcribed into "ordinary" documents, viz. the notes, photographs, sketches etc. collected by an ethnographer in a field study.

Museums are the social institutions that for (small) artifacts (and objets trouvées) and natural documents perform the same functions as libraries and archives. For large artifacts, which by their nature cannot be collected and organized in a manner similar to that of museums, there are usually also social institutions, e.g. a Department of Antiquities.

8.3.1.2 Document collections and catalogues

The fundamental task of libraries, archives and museums is to acquire artifacts that function as cultural documents, describe and organize them for retrieval, store and provide them upon request. For every collection there is a development policy (implicit or explicit) that guides the decisions on which items to include in the collection.

Within these institutions there have over time evolved different approaches to describing the items and organizing the collections. These differences are of course in part due to the different nature of the items but also in part due to a different view of the institutional goals and their operationalizations. All of them do, however, provide two very different tools for finding documents, namely the shelving/display organization, with the users walking around in the document collection, and the catalogue, in which the users' fingers do the walking, in the document description collection

Libraries have by and large been quick to embrace new technology and the catalogues of all major and many minor libraries are now computer based. Archives and museums have been slower in their adoption but are now also moving towards a similar state. In this movement it has become apparent that much of the work done in libraries on standardization (which is a necessity for cooperation involving computerization) as well as the experiences gained by them in this process can be utilized by archives and museums.

Catalogues contain representations of items. The representations have the functions of identifying the items in the collection, describing specific aspects of the item, and of being a token or substitute.

A catalogue is hence a listing of items (or classes of items) in a collection.

The explicit relations between documents and collections of libraries, archives and museums have until now been weak and few. The implicit relations are, however, strong and many. The conversion of catalogue records to electronic form, coupled with the increasing hospitality of electronic documents towards different forms of description, representation and presentation means that it is now possible to start considering multimedia catalogues that can integrate and relate hitherto separated and unrelated items of information from the domains of libraries, archives and museums.

Electronic documents (documents is here used in the general sense) and hence databases of electronic documents can and should therefore be designed to be an integrated collection of related items.

8.3.1.3 Electronic documents

In the world of electronic documents the documents and their descriptions have functionalities that are very different from those of print on paper and all tools can (at least in principle) be used for collections of both types. Documents and their variable (in size and type) descriptions can, furthermore, be seen to comprise a continuum from an one-line reference to a complete document. One of the objectives in designing the architectures of electronic documents must thus be to enable the algorithmic derivation and generation of a variety of document descriptions from the documents themselves.

Electronic documents are also accommodating in an increasingly integrated fashion most forms of transcription and presentation of information. An electronic document can besides text contain e.g. sound and animated images in colour. The hypermedia concept, of open ended non-sequential, reader determined use of integrated multimedia document databases also has the consequence that our generalization of document can be given a wider interpretation than the one discussed so far. A specific path through a large hypermedia database is also a document.

Large shares, in terms of frequency and volume, of the generation and handling of traditional, and electronic, documents are, nevertheless, taking place in other contexts than libraries, archives and museums, viz. in offices and administration. Most institutions and organizations are, or have components and processes which are, clustered around the generation and handling of written documents. A very large part of these documents are now being created using computer based tools, but the print-on-paper written document is still the end product (although the database inquiry and/or update dominates in many transaction oriented systems, e.g. banks, with a print-on-paper document as a side product.)

8.3.1.4 Three Issues

Information is managed at different levels of access: personal, group(s), institution(s), societ(y/ies). The volume of information managed grows by orders of magnitude with each step away from the personal level. Each level has also until now been separated from the next by increasing physical distances. Those physical distances vanish for the reader in front of a screen if his computer is tied into the computer communication networks.The volume of information then accessible at the personal level is (in principle) enormous but the barrier of physical distance has instead been replaced by more abstract barriers of intangibility and need for guidance through the electronic mazes.

Finding or being provided with pertinent information in the right form at the right time has always been a problem and will remain so. The desideratum “the right form” is today furthermore taking on a number of added possibilities. It can be “a compatible mix of media” and each “with a functional as well as pleasing appearance” and “at an appropriate level of detail” and “with expedient interface” etc.

8.3.1.4.1 Document description and representation

Electronic documents can and should contain descriptions of themselves and they should be designed in a manner that enables an algorithmic generation of catalogue records with varying degrees of detail.

The work done on abstract editors and in generating standards such as SGML and ODA, X.400, EDIFACT and associated standards on the one hand and the work done in libraries, archives and museums on cataloguing and document description on the other are two different approaches to the same issues. An attempt is being made at LIBLAB to reconcile the creation oriented approaches of the standards with the collection and organization approach of libraries, archives and museums.

So far the efforts in the standardization work have, however, been concentrated mainly on the generation and distribution processes and on the documents as instances of document types. The longer perspective, of documents as items in large collections, seems to be lacking.

The same kind and amount of care and consideration that today is spent on language and layout in creating a document should also be given to facilitating its retrieval in the future. Retrieval of a document from a collection and of specific parts from a comprehensive document are different sides of the same issue.

Uniform and adequate methods for description of collections are lacking in both. An approach is needed that can be used both for individual documents and collections of documents. In general a collection is not a set of isolated items. Various types of relations exist between the items in the collection. The structures of collections as compared to the structures of the subjects and literatures needs special analysis. The role of knowledge organizing schemes such as classifications and thesauri as structuring mechanisms is hence also being studied in the light of hypermedia databases, see the next section on knowledge organization.

Using a number of existing documents that have been marked up using e.g. the AAP application of SGML and/or a similar application of ODA as well as using the TEI guidelines a study is in process of in what respects modifications will be necessary for present cataloguing rules (AACR2) as well as the applications in order to reach a mutual agreement between the two approaches to document description.

8.3.1.4.2 Users and uses of documents

Users – tools/system(s) – documents form a triad of interaction in document generation, retrieval and use.

The interest in and work on standards such as those mentioned above is based on a recognition of the need for an approach to the problems of formal communication (through document generation, transmission, reading and handling) that recognizes that users in addition to reading documents also want i.a. to be able to reuse them or parts of them. Reuse implies tools for manipulating the documents and thence access to explicit information about the document's content, appearance, structure etc.

Tools that have functionalities based on the characteristics of documents that embody the formalisms mentioned are lacking and those formalisms have likewise been designed more with consideration given to accommodating existing types of written documents than to the users and uses of documents.

8.3.1.4.3 Documents as databases

Each electronic document is potentially a hypermedia database. The focus in the standardization work so far has been mainly on documents as carriers of written text. Graphics and other entities are treated as undifferentiated objects that are adjuncts to the writing. The technical problems of connecting the various components of multimedia systems are still in the centre of attention.

The problems of integrating diverse media from a structural and interaction point of view are only slowly beginning to receive attention. One of the major problems is that very little has been accomplished in terms of retrieval of non-verbal entities. Research is needed both on the description in verbal terms of non-verbal entities and on methods for formulating non-verbal queries without resorting to written descriptions and methods for matching such queries with corresponding objects.

8.3.2 Knowledge Organization in Hypertext/Hypermedia Environments

A major problem with most hypertext and hypermedia systems is that the user very easily gets lost. Following links and trails makes her loose her orientation in the network. Systems like these must offer ways to find out the user's current position and show possible ways to go from there. The possibility to visualize the network becomes more important in systems based on the “go” approach than in systems based on the “show” approach. In the first category of systems, where the context is actually changed when a link is followed, the tendency to get lost is much higher than in the last category, where something is added to the former context. Many systems have a mix of “go” and “show” approaches.

An interesting research area is to study the possibilities of using traditional indexing and classification techniques as knowledge organizing structures and tools for navigation in hypermedia systems. These techniques have been used for a long time in order to organize documents and to help people find what they are looking in bibliographical environments. Bringing together the best parts of current IR-techniques with hypertext/hypermedia techniques brings forth both effectiveness and flexibility.

There are several areas that need to be studied:

- The structure(s) of classification and indexing systems. The basic structures of this kind of knowledge organizing structures must be considered. Are there any common structures at all, and if so, how can we make use of them in order to facilitate the use of hypermedia information systems?
- Representation and presentation. The problem of using indexing and classification techniques as navigation aids in hypermedia environments, can be seen as both a representation and a presentation problem.If we choose a solution where a traditional classification scheme serves as one navigation tool in a hypermedia environment, what is the best way to represent it, and how should it be presented to the user? Are there benefits of hypertext systems versus other kinds of systems for representation and presentation of this kind of structure?
- Evaluation of hypertext/hypermedia systems. Traditional measures as precision and recall, which are used when evaluating IR-systems, cannot be used when evaluating systems that combine IR and hypermedia techniques. Methods for comparing traditional IR systems with combined IR/hypermedia systems, and for comparing combined IR/hypermedia systems with pure hypermedia systems need to be developed. For the future, there is also a need to study what features in currently available systems are the most interesting ones in this context and what is missing.

8.3.2.1 Application

Previous work at LIBLAB has been focused on the library context. As a natural expansion, we are now also studying problems of knowledge organization in an archival environment.

8.3.2.1.1 KAM

The project Knowledge-based System in an Archival Environment (KAM) was initiated as a part of a larger project called Archival Information 2000 (AI 2000). AI 2000 is a joint project between the Swedish National Archive, the Central Board of National Antiquities, Linköping University, and The Provincial Government of Västernorrland. The KAM project is now a sub-project of the National Archival Database (NAD) project and a cooperation between the Swedish National Archive and the Dept. of Computer and Information Science at Linköping University.

The goal for the KAM project is to design, implement and evaluate a system that will help the user in accessing information about:

- the creator of an archive.
- the contents of the archive, and the extent of it.
- how the information in the archive can be used.

A search strategy for this kind of retrieval has been formulated, and different knowledge bases needed for this purpose has been identified. At least three more or less separate parts of a knowledge base are needed for retrieval. These different parts support:

the decision on Authority type, based on Function, Object, Time and knowledge about rules and regulations for the public administration.

a correct geographical statement, based on time and knowledge about changes in the administrative division of the country.

the decision on document type, based on decision 1 and 2 together with information from the current Archive Guide.

Several parallel activities have been initiated such as:

- experimental prototyping to study the interaction with and use of such a system.
- studies on how to represent geographical time-dependent information.
- studies on content description of the material.

A first prototype has been built by Robert Jakobsson. This prototype is to be evaluated, in order to decide which parts to develop first, and how deep we should penetrate the different areas.

A special group is working on a knowledge base for Authority type, discussing what functions and objects are relevant to the Swedish material, and how much of the knowledge about the administration is needed and can be implemented.

Another group is working on the geographical base. Their main objective is to investigate how much information there is already available in machine-readable form, that we could use in this system. Another major problem is how to represent the changes over time, concerning the administrative division, and connect that with the maps.

The main project group will discuss further plans for new prototypes after the evaluation of the first one. This evaluation will be performed at different archives in Sweden. The first user group to test it will be archivists. Depending on the result of this, there may also be a broader evaluation, including other users.

Discussions will also include what type of technique should be used for forthcoming prototypes. A combination of hypermedia and expert systems techniques, making use of experiences from the HYPERCATalog project and other activities at LIBLAB has been proposed.

8.3.3 A Framework for Analyzing and Designing User Interaction

Effective human-computer interaction presupposes that there is a natural connection and a consistency between the objects and operations that a system supplies and the objects and operations that its users demand. There has been a striving through the 1980s to bring the context of use into design and to utilize methods and techniques such as direct manipulation, object orientation, prototyping, user interface management, etc. in order to support the design of truly usable computer systems. However, a common, explicit theory of HCI that could effectively support the design process remains to be formulated.

In the ambition to find the semantics of HCI we attempt at LIBLAB to represent such a theory so that it fosters our understanding of its pragmatics, i.e. our understanding of the artifacts themselves and the context in which they are used.

8.3.3.1 Formal Descriptions, Metaphors and Conceptual Models

In an earlier project a top down approach for the hierarchical decomposition and description of complex systems has been described. In this model higher level descriptions act as abstract specifications of lower level ones and not only as the usual where lower level descriptions are more detailed instances of higher level descriptions.

This top down approach, where the system is viewed from its metaphors, also work as a user oriented description in terms of qualitative aspects as compared to more traditional descriptions (e.g. requirement specifications). These are to a large extent quantitative measures and thus easier to formalize than their more over-riding conceptual and metaphorical ones. The system designer has to realize and actualize this implicit correspondence between quantitative and qualitative descriptions. Since every level results in design alternatives to be chosen on succeeding levels this is not only a simple matter of choice, but also a question of seeing these options in terms of higher level requirements.

8.3.3.2 Hypermedia Applications

One of the research themes of general interest within the LIBLAB group is hypertext/hypermedia applications and has been so since 1985. A related project is hence to study the interactivity and the usability of this kind of integrated multifunctional tools. Hypertext/hypermedia is, in this context, seen as a generic form of human- computer interaction that above all stress the usability of computer systems.

The (re)presentation of complex information in a manner familiar to users becomes even more intricate in hypermedia environments. Users often find it cognitively too difficult to correctly interpret the hidden properties contained within a chosen representation. The task seems either to be too abstract or too concrete for users to manage properly without much difficulty.

This growing complexity of computer applications increases the need for alternative user interface metaphors that support not only the process of reading and writing *per se* but higher levels of cognitive processes as the dynamic generation, collection. organization and communication of ideas.

8.3.3.3 Evaluation, Specification and Design

This is another sub-theme closely linked to the main research theme of defining a framework for analysing and designing user interaction. It concerns the investigation and testing of methods for evaluating usability in order to establish effective procedures for usability testing of hypermedia environments. The objective is also to incorporate usability considerations with functionality considerations for hypermedia environments.

8.3.3.4 Applications

8.3.3.4.1 Usability Evaluation in Practice

This project was initiated in 1988 as a joint undertaking between Swedish Telecom and the Department of Computer and Information Science to study methods for the evaluation of human-computer interfaces. The purposes were:

- to investigate and test methods for evaluating usability,
- to evaluate a prototype for network operation and control, and
- to establish a procedure for usability testing.

The project was divided into two parts:

1. An extensive literature study with the aim of investigation reported uses of HCI evaluation methods.
2. An actual evaluation of a prototype for a system to support the operation and control of telecommunication networks.

In part two some of the methods reported in part one were tested in order to answer questions regarding:

- the benefits and shortcomings of the methods used,
- the benefits and shortcomings of combining various evaluation methods,
- the need to develop automated tools to assist in the effective collection, compilation and analysis of evaluation data, and
- the feedback of knowledge gained from the actual use of a prototype system into the system development process.

8.3.3.4.2 Training Within Multimedia Environments

During 1989 and 1990 a joint project between Telepedagogik, British Telecom and the Department of Computer and Information Science was carried out with the aim of evaluating the introduction, organization and planning of education utilizing new technology in order to obtain a specification of the different needs for users and enterprises in interactive multimedia environments.

In a case study in '89/'90 two different interactive video training packages were tested. The evaluation methods comprised a combination of seminars, questionnaires, video recordings and interviews. In total 125 users participated in the study.

8.3.3.4.3 Open Learning Systems

Yet another project carried out in 1989/90 was to assist in the specification of the user environment for hypermedia based learning systems as part of the European DELTA program (Developing European Learning through Technological Advance), the EPOS project (the European PTT Open Learning Systems).

8.4 Publications

External refereed publications:

Björklund, L.: HYPERCATalog – the Researcher's Tool for Information Management. In: *Information Technology and the Research Process Proc. of a Conf. held at Cranfield Inst. of Technology, UK, July 18–21, 1989. M. Feeney and K. Merry, Eds. Bowker-Saur, 1990. pp.95–103.*

Björklund, L., Olander, B.: The Mystery Cat – the Modern Magician. In: *Information and Innovation.: Proc. of the 7th. Nordic conference for Information and Documentation, Århus, Danmark, August 28–30, 1989. H. Clausen,Ed. Aarhus Univ, 1989. pp. 55–62.*

Björklund, L., Olander, B., Smith, L.: The Personal HYPERCATalog. In: *Proc. of the 52nd Annual Meeting of the American Society for Information Science, Washington DC, October 29 – November 2, 1989. pp.115–120.*

Björklund, L.: HYPERCLASS. Four hypertext applications of a classification scheme. In: *Tools for Knowledge Organization and the Human Interface: Proceedings of the First International ISKO-Conference, Darmstadt, 14–17 Aug, 1990. R. Fugman, Ed. INDEKS Verlag. Frankfurt/Main FRG. 1990 pp.190–198.*

Björklund, L, Kristiansson, G.: Problems of Knowledge Organization in an Archival Environment – KAM. In: *Tools for Knowledge Organization and the Human Interface: Proceedings of the First International ISKO-Conference, Darmstadt, 14–17 Aug, 1990. R. Fugman, Ed. INDEKS Verlag. Frankfurt/Main FRG. 1990 pp.199–205.*

Hjerppe, R.: HYPERCAT at LIBLAB in Sweden: a progress report. In: *The Online Catalogue: Developments and Directions*, Charles R. Hildreth, Ed. The Library Association, 1989, pp. 177-209.

Hjerppe, R.: A Framework for Characterizing systems for Knowledge Organization: a first basis for comparisons and evaluation. In: *Tools for Knowledge Organization and the Human Interface: Proceedings of the First International ISKO-Conference, Darmstadt, 14–17 Aug. 1990. R. Fugman, Ed. INDEKS Verlag. Frankfurt/Main FRG. 1990 pp.21–34.*

Hjerppe, R.: The Role of Classification and Indexes in Hypertext. Issues in implementing Roget's Thesaurus as a Hypertext. In: *Tools for Knowledge Organization and the Human Interface: Proceedings of the First International ISKO-Conference, Darmstadt, 14–17 Aug, 1990. R. Fugman, Ed. INDEKS Verlag. Frankfurt/Main FRG. 1990 pp.206–215*

Smith, L.: 'Wholly New Forms of Encyclopedias': Electronic Knowledge in the Form of Hypertext. In: *Information * Knowledge * Evolution: Proceedings of the forty-fourth FID Congress held in Helsinki, Finland, 28 August – 1 September 1988, Sinikka Koskiala and Ritva Launo, Eds. North Holland, 1989, pp. 245–250.*

Vainio-Larsson, A, Orring, R.: Evaluating the Usability of User Interfaces: Research in Practice. Paper presented at *INTERACT 90, The 3rd IFIP Conference on Human-Computer Interaction, Cambridge, England, August 1990*

Other external publications:

Björklund, L., Kristiansson, G.: Projektet Kunskapssystem i Arkivmiljö (KAM). *Delrapport 1. Riksarkivet, Januari 1990, 5p.*

Vainio-Larsson, A.: Hypertext/hypermedia från ide till praktisk tillämpning .In: *Dialogergonomi: Effektiv interaktion människa-dator, Mekanresultat 89005, October 1989 (Also available as LiTH-IDA-R-89-1)*

Vainio-Larsson, A.: Människa-dator interaktionsforskning i USA: en Teldok studieresa april '89. *STIMMET, no 2, '89.*

Vainio-Larsson, A.: Rapport från ännu ett hypertext/hypermedia seminarium. *STIMMET, no 2, '89.*

Vainio-Larsson, A.: Human-Computer Research in Europe and USA '88 / '89. *EACE Newsletter, '89.*

Internal publications:

LiTH-IDA-R-89-31 **Vainio-Larsson, A., Åberg, P.:** Utvärdering av människa-maskin gränssnitt (Delrapport II). *September 1989, 31p.*

LiU-LIBLAB-R-90-20 **Vainio-Larsson, A.:** Training Within a Multimedia Environment: Education Based on Interactive Video. *September 1990, 18p.*

LiU-LIBLAB-WP:43 **Björklund, L, Hjerppe, R, Vainio-Larsson, A.:** LIBLAB Planer för verksamheten under bå 1990/91. *March 1990, 5p.*

LiU-LIBLAB-WP:44 **Hjerppe, R, Höglund, L, Persson, O.:** Informationsanvändning – behov och teknologi i samspel. *March 1990, 5p.*

LiU-LIBLAB-WP:45 **Hjerppe, R, Jungert, E.:** Virtuell och faktisk. Blandade verkligheter som interaktionsmodell för datorstödd samverkan mellan flera personer. *April 1990, 7p.*

LiU-LIBLAB-WP:46 **Hjerppe, R.:** The HYPERCATalog: The Role of Document Architecture and User Interaction for Finding Information in Hypermedia Databases *May 1990*.

LiU-LIBLAB-WP:47 **Björklund, L., Hjerppe, R.:** Från ärendehantering till arkivåtkomst: Dokumentarkitektur och kunskapssystem för beslutsstöd *May 1990, 10p. + 20p.*

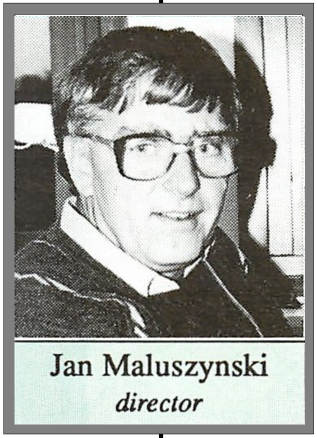
LiU-LIBLAB-WP:48 **Hjerppe, R.:** Roget's International Thesaurus. Experiments in Hypertext Using Hypercard. (En Hypercard stack) *May 1990.*

LiU-LIBLAB-WP:49 **Hjerppe, R.:** Implementing Roget’s International Thesaurus as a Hypertext. Design and Other Issues *June 1990.*

LiU-LIBLAB-WP:50 **Hjerppe, R.:** A Note on Going and Showing *June 1990.*

LOGPRO

9 The Laboratory for Logic Programming


Jan Maluszynski
director

Foundations of logic programming
Integration of programming paradigms
Program transformation

Back row: Andreas Kågedal, Ulf Nilsson, Anders Haraldsson.
Front row: Xinli Gu, Staffan Bonnier, Jan Maluszynski, Jonas Wallgren, Feliks Kluzniak.
Missing: Johan Boye, Wlodzimiertz Drabent, Gunilla Lingenhult.

Figure 9-1. *Environment view in the campus.*

9.1 Introduction

The Laboratory for Logic Programming was formally created in spring 1985 though research activities in logic programming at the department started much earlier. The research concentrates on the foundations of logic programming systems and on the relation of logic programming to other computational paradigms.

An important objective of the laboratory is also to contribute to the research activities of the other laboratories by offering courses and seminars on logic programming, theory of programming and formal language theory.

The following persons were involved in the activities of the group:

Laboratory leadership and administrative staff:

Jan Maluszynski, Ph.D., professor
Gunilla Lingenhult, secretary

Associated member:

Anders Haraldsson, Ph. D., associate professor

Guest researcher:

Wlodzimierz Drabent, Ph.D.

Graduate students:

Staffan Bonnier
Xinli Gu
Simin Nadjm-Tehrani
Ulf Nilsson
Torbjörn Näslund
Jonas Wallgren

Some of the research has been carried out in external cooperation with researchers at INRIA, France, University of Utah, USA and Institute of Computer Science of the Polish Academy of Sciences.

The main research activity has been concentrated around the projects *“Logic Programming with External Procedures”* funded by the National Swedish Board for Technical Development (grant STU 87-2926) for a three year period commencing from July 87 and *“Systematic Design of Abstract Machines through Partial Evaluation”* (STUF 88-376) starting from July 88 ranging over a two year period.

The work in the Logic Programming Group is mainly supported by STU, The Swedish Board for Technical Development and by NFR, the Swedish Natural Science Research Council.

9.1.1 The Objectives of the Present Research

Logic Programming with External Procedures

It is often claimed that logic programming has great potential for reducing the cost of software development. One of the reasons supporting this claim is the declarative nature of logic programming. Since the control information need not be specified, the size of the code is often dramatically reduced in comparison with the size of algorithmic programs. However, the cost of software development also depends heavily on the possibility of re-using of existing modules, and on the methodology of programming. Thus, the problem how to combine logic programs with existing software is of great practical importance, but unfortunately its ‘‘ad hoc’’ solution may destroy the declarative reading of programs. This in turn may create serious problems in reasoning about such programs and decrease their reliability.

This project has two primary objectives:

- to develop a theoretical basis for a solution of the problem of re-usability of existing software in logic programming preserving the declarative nature of logic programs;
- to contribute to the methodology of development of correct logic programs with external procedures.

Constructive Negation

The negation-as-failure rule [Cla78] and SLDNF-resolution allow a limited form of negation in definite programs. Unfortunately negation as failure can be used only as a test. It is not possible (and in some sense not even allowed) to invoke a negated goal unless the goal is completely instantiated. This makes negation as failure a rather weak form of negation and, more seriously, since few existing Prolog systems check that negative goals are fully instantiated it is not unusual that incorrect results are produced! The general idea of *constructive negation* [BMPT87, BMPT89, Cha88] is to overcome some of the most serious deficiencies of the negation-as-failure rule and SLDNF-resolution. The most notable difference as compared to the standard approach is to allow (a limited form of) negated goals which contain variables, and to compute bindings for those variables.

The objectives of the project are twofold:

- to develop an extended negation-as-failure rule for definite programs which allows to compute solutions to non-ground negative goals.
- to extend the approach to normal programs, i.e. logic programs which allow negative literals in the bodies of clauses, and to amalgamated programs.

Systematic Design of Abstract Machines through Partial Evaluation

In recent years Prolog implementations have undergone substantial improvements both in terms of time- and space-efficiency. This is due mainly to a new generation of abstract execution models presented in the literature. Most notably the Warren Abstract Machine (WAM). Unfortunately the reader of such specifications is given very little intuition as to why an abstract machine has a particular appearance. Instead (s)he is often faced with a set of registers, data structures and machine instructions. This kind of ad hoc presentation of abstract machines not only obscures the reader's understanding of the machine but also fails to reflect the connection between the source language (Prolog) and the target language (the produced machine instructions), something which obstructs the construction of simple and correct compilers.

On the other hand a large number of ‘‘Prolog’’-like languages have been presented lately. For each such language a new abstract machine must be constructed.

The objectives of this project are the following:

- to study and propose a systematic way of designing abstract machines for ‘‘Prolog’’-like languages by means of partial evaluation;
- To apply the proposed methodology to the logic programming language with external procedures.

9.2 The Research Topics

9.2.1 Amalgamation of Logic Programs with Functional Procedures

The objective is to develop a systematic approach to writing logic programs which use external procedures. The motivation for this is twofold:

- to allow for re-using of existing (possibly imperative) software while still preserving the declarative nature of the top-level logic programs;
- though the Horn clause logic provides a universal computational paradigm it seems often quite unnatural to express functions in the relational formalism.

In recent years there have been a number of suggestions concerning combination of functional and logic programming in a single framework (see e.g. [GL86] or [BL86] for a survey).

The approaches can be classified into two categories:

- Integration of existing programming languages and logic programs (well-known examples of this type are LOGLISP, QLOG, POPLOG, APPLOG and Quintus Prolog's foreign functor interface);
- Construction of new languages which allow one to define functions and relations and to combine functional and relational definitions. (Well-known examples include EQLOG, LEAF, SLOG and FUNLOG.)

The main objective within the first approach is often to give access from logic programs to specific features of the underlying programming language, or programming environment. This aspect is usually more important than concern about the declarative semantics of the amalgamation. It may be rather difficult to give such a semantics if low-level features of the underlying system are accessible in the resulting language. On the other hand, many of the languages defined within the second approach have both declarative and operational semantics and some completeness results are also presented.

Our general perspective is different: we assume that we have given two arbitrary languages – one (not necessarily functional!) with functional procedures and the other one being a logic programming language. It is our belief that a good combination of these two should result in a language which is as ‘‘conservative’’ as possible. Old programs (both their evaluation and meanings) should not change. It will allow us not only to think in a well established manner about things which have not changed but also to save a lot of work because we do not have to rewrite old programs in a new language. On the other hand we want to be able to build new logic programs employing calls to functional procedures and integrate them with the old ones. A basis for construction of an interface between the two different systems is the assumption that terms are their common data structures. A call of a functional procedure is itself a term. Since its execution is assumed to return a term we can view the underlying programming system as a term rewriting system. This permits the use of the theory of logic programming with equality (see e.g. [JLM84]) to give a clean declarative semantics of the amalgamated language, and for application of E-unification in its interpreter. However, since we are not specific about the language of the functional procedures we have no access to the rewrite rules used by the system and we cannot use them for construction of E-unifiers. In [LM86] we suggest a way of overcoming this difficulty. It is a new unification algorithm, called S-unification, which is a special incomplete case of E-unification. Let us compare our suggestion with the two kinds of approaches mentioned above.

The distinction between our work and the first approach is that the underlying programming system is considered to be a black box: low-level features can be used in the underlying programs but not on the logical level. This makes it possible to give a relatively simple declarative semantics of the amalgamation. The difference between our suggestion and the second approach is that we are primarily interested in re-using functional procedures written in other languages, regardless of the type of the language (be it a pure functional language, or an algorithmic language which admits functional procedures, like Fortran, Pascal or Ada) in a logic programming environment. In contrast to the systems of this category we assume existence of the term machine and we are able to use it in the top-level computational mechanism without being specific about its construction.

9.2.2 Methodology of Amalgamated Programming

We are searching for concepts and methods that facilitate development of correct logic programs including calls to external procedures. This work is based on existing ideas concerning logic programming.

Correctness of programs is usually defined with respect to some initial formal specification. Often it is suggested to develop correct programs from this specification by some formal transformations. In case of logic programs this approach has been studied in many papers (see for example [HT82], [Cla81], [Hog81]). A (pure) logic program is a logic formula and due to its declarative reading may be sometimes considered a specification of the problem. Therefore it is sometimes suggested to organize development of logic programs as construction of a derivation in the first order logic, where the initial formula is a complete specification of the problem and the final one is a logically equivalent logic program. However, a general difficulty with complete specifications is that they often do not properly reflect user's intuition and must be subject to changes. Therefore, testing of the program is necessary not to discover bugs, since the ideal development technique would guarantee correctness of the program with respect to the specification, but rather to confront user's informal understanding of the problem with the formal specification. The aspect of programmer's intuition is reflected in Shapiro's work on algorithmic debugging [Sha83]. The debugging process relies on an ‘‘oracle'' which answers the questions generated by the debugging system. These answers are statements concerning properties of the intended model of the program. They should be satisfied by the program under development.

Usually it is the programmer that plays the role of the oracle. The questions she has to answer concern the satisfiability and validity of given atomic formulae in the intended model. The answers are ‘‘yes'' or ‘‘no''; in some cases she is required to provide those instances of a given formula that are true in the model. So the language of the dialogue is rather restricted; the programmer communicates her knowledge about the intended meaning of the program by means of examples. It is interesting to investigate possible improvements to this communication. We study a possibility of supplying partial specifications of the intended model. Such a specification formalizes a property of the intended model and is equivalent to some (usually infinitely many) oracle answers. This generalizes both complete specifications and the usual oracle answers. If a partial specification is executable, it may be used to automatically answer part of the questions posed by the debugging algorithm, thus decreasing the number of questions to be answered by the programmer. In our approach, partial specifications are logic programs and are called assertions. (They are different from assertions used in our earlier work [DM88] for describing run-time properties of programs). We intend to explore the practicality of the approach for debugging non-trivial Prolog programs. This requires extending the framework to non-pure logic programs and building a (prototype) implementation.

9.2.3 Constructive Negation

The negation-as-failure rule and SLDNF-resolution are well-known techniques for dealing with negative information in logic programs. The former is a meta-inference rule allowing to prove the negation of a ground atom A when the goal $\longleftarrow A$ finitely fails. Several results concerning soundness and completeness of this rule have been established (e.g. [Cla78, JLL83, She88]).

The major drawback of logic programming systems employing the negation-as-failure rule is the non-symmetric treatment of positive and negative information – the rule can be used solely to check universally quantified negative literals, not to compute solutions to existentially quantified negative ones. The aim of our research is to extend the negation-as-failure rule in such a way that it is possible to find solutions also for this case. The problem can be formulated as follows: For a given negative goal $\longleftarrow\neg A$ find a (complete) set of (maximally general) substitutions Θ such that each goal $\longleftarrow A\Theta$ finitely fails. The substitutions are called *fail* substitutions. Logical correctness of the answers obtained in that way follows from soundness of the negation-as-failure rule.

9.2.4 Systematic Design of Abstract Machines through Partial Evaluation

Existing proposals for abstract machines for the Prolog language simply consist of a collection of machine registers, areas of memory, data structures and machine instructions (see e.g. [War84]). It is not so easy to see the connection between source and object language.

However, in [Kur87] it was shown that some of the machine instructions in the WAM can be inferred in a systematic manner by means of partial evaluation. The main objective of this research is to refine and extend the technique outlined in [Kur87]. Roughly speaking, a methodology for design of abstract Prolog machines would look as follows:

Given an operational description of the source language (i.e. an interpreter) and a collection $P_1, \ldots, P_n$ of source programs, partially evaluate the interpreter with respect to the programs. The result is a collection $O_1, \ldots, O_n$ of programs in the same language as the interpreter. Next identify pieces of code in $O_1, \ldots, O_n$ which occur frequently (or which are instances of code which occur frequently). Abstract the code into a machine instruction and make the corresponding abstraction in the interpreter. Then repeat the process until acceptable machine code is produced for the sample programs. Then the collection of all abstractions constitutes the set of machine instructions and their definitions.

9.3 The Results

The following main results were obtained from July 1989 to June 1990:

9.3.1 Amalgamating Logic Programs with External Procedures

In order to allow the use of external functional procedures in logic programming an incomplete E-unification algorithm, called S-unification is developed. The algorithm is based on the ideas presented in [LM86] and [BM88]. It has been proved that S-unification always terminates with one of two possible results for given arguments: (1) A complete set of E-unifiers (which due to our restrictions either is a singleton or is empty), (2) A don’t know message, indicating that S-unification is unable to solve the E-unification problem. Moreover, S-unification has been shown to be *weakly complete* [B89]. This means that the proposed algorithm is as complete as any unification algorithm which could be used as a substitute for S-unification. More precisely, if such a unification algorithm finds a complete set of E-unifiers, then a complete set is also found by S-unification. A completeness check based on notions of abstract substitutions is also developed. The check provides a sufficient condition for the interpreter to be complete for a given program and a class of goals.

9.3.2 Constructive Negation

Constructive negation aims at extending the negation-as-failure rule, so that solutions to existentially quantified negative literals can be computed. The suggested approach is restricted to the case of definite programs and is presented in [MN89]. The solution is based on systematic traversal of the search space (i.e. the SLD-tree) of the goal $\leftarrow A$ and "pruning" of its selected branches. Pruning consists of selecting a pruning point of the branch and finding a solution β of the term inequality associated with the point. If there is no solution, another pruning point should be chosen. The search space of $\leftarrow A\beta$ is isomorphic to a subset of the search space of $\leftarrow A$ with the selected branch pruned in the selected point. The answer is obtained by composition of pruning substitutions of all non-fail branches of the original SLD-tree. Alternative answers are constructed by alternative selection of pruning points.

9.3.3 Algorithmic Debugging with Assertions

Algorithmic debugging is based on communication between the debugging system and an oracle (usually the user). The oracle delivers information about the intended model of the program. In [DNM89] we suggest a generalization of the language used to communicate with the debugger. In addition to the usual ‘‘yes’’ and ‘‘no’’ answers, formal specifications of some properties of the intended model are allowed. The specifications are logic programs and are called assertions. Assertions employ library procedures and are developed interactively in the debugging process. The debugging system described in [DNM89] employs algorithms which, in contrast to most other systems, do not expect the oracle to provide true instances of a goal. This is achieved by generalization of the oracle in the incorrectness algorithm, and by adopting a new insufficiency algorithm. A formal proof of correctness and completeness of the new insufficiency algorithm is presented.

A prototype debugging system incorporating these ideas has been implemented. [N90] reports on this implementation and the preliminary results obtained while debugging a sample of non-trivial Prolog programs. Additionally, [N90] discusses extensions of the method for programs with cut, negation, and some Prolog built-in predicates.

9.3.4 A Scheme for Abstract Interpretation of Logic Programs

Static analysis of logic programs is an important tool for inferring run-time properties of programs. However, since most interesting properties are undecidable one cannot hope to find both sound and complete tools for inferring properties. Abstract interpretation provides a solid mathematical framework for program analysis. In [Nil89] a scheme based on this framework is presented. The paper describes a ‘‘base’’ semantics for logic programs based on its (very simple) operational semantics. Because of its simplicity the ‘‘base’’ semantics facilitates development of nonstandard (abstract) interpretations of programs. Depending on what abstract interpretation is used, different properties may be inferred in finite time. Sufficient conditions are formulated to guarantee that properties inferred are sound.

9.3.5 Systematic Design of Abstract Machines through Partial Evaluation

Recently logic programming has been enriched by a large number of logic programming languages. Since these languages often have a similar basis but come with different flavours the question arises: Is it possible to single out some common methods for the design of specialized abstract machines for this family of languages? In [Nil90] we have shown a simple set of transformations to derive machine instructions for the control flow of logic programming languages à la Prolog (i.e. depth first search with backtracking). The major transformation technique used is that of partial deduction but also other (non-standard) program transformations are exploited.

9.3.6 Miscellaneous

- During the autumn of 1989 three licentiate thesis [N89, Nil89, B89] were defended by LOGPRO students.
- During the spring of 1990 LOGPRO co-organized the *Second International Workshop on Programming Language Implementation and Logic Programming* held in Linköping, Sweden, August 1990.
- In june 1990 LOGPRO was formally accepted as a member in the ESPRIT Basic Research Action 3020: INTEGRATION.
- As a side-effect of previous research and teaching, a monograph on logic programming was developed [NM90]. The objective of the book is to provide a coherent and intuitive introduction to both foundations of logic programming and programming in logic.

9.4 References

The following are papers quoted above, a complete list of LOGPRO publications can be found in the list of IDA publications in Appendix E.

LOGPRO papers:

[BM88] **Bonnier S. and Maluszynski J.**, Towards a Clean Amalgamation of Logic Programs with External Procedures. In *Proc. of Fifth International Conf/Symposium on Logic Programming*, Seattle (1988). Also in *Proc. of PLILP'88*, LNCS 348, Springer-Verlag.

[DM88] **Drabent W. and Maluszynski J.**, Inductive Assertion Method for Logic Programs. *Theoretical Computer Science* 59 (1988).

[DNM89] **Drabent W., Nadjm-Tehrani S. and Maluszynski, J.**, Algorithmic Debugging with Assertions. In Meta-Programming in Logic Programming, *MIT Press*, 1989.

[LM86] **Leszczylowski J. and Maluszynski J.**, Logic Programming with External Procedures: Introducing S-Unification. *Rep. TR-86-21*, Dept. of Computer Science, Iowa State University, Ames (1986).

[MN89] **Maluszynski J., and Näslund T.**, Fail Substitutions for Negation as Failure, in: *Proceedings of the North American Conference on Logic Programming*, MIT Press 1989.

[N90] **Nadjm-Tehrani S.**, Debugging Prolog Programs Declaratively. In *Proceedings of the Second Workshop on Meta-Programming in Logic Programming*, 1990.

[Nil90] **Nilsson U.**, Towards a Methodology for the Design of Abstract Machines of Logic Programming Languages. *Research Report LiTH-IDA-R-90-12*, Linköping University (1990). Submitted for publication.

[NM90] **Nilsson U., and Maluszynski J.**, Logic, Programming and Prolog, John Wiley & Sons, 1990.

Licentiate theses 1989/90:

[B89] **Bonnier S.**, Horn Clause Logic with External Procedures: Towards a Theoretical Framework, *Licentiate Thesis No. 197*, Linköping 1989.

[N89] **Nadjm-Tehrani S.**, Contributions to the Declarative Approach to Debugging Prolog Programs, *Licentiate Thesis No. 187*, Linköping 1989.

[Nil89] **Nilsson U.**, A Systematic Approach to Abstract Interpretation of Logic Programs, *Licentiate Thesis No. 196*, Linköping 1989.

Other References:

[BL86] **Bellia M., Levi, G.**, The Relation between Logic and Functional Languages: A Survey. *J. Logic Programming*, 3 (1986).

[BMPT87] **Barbuti R., Mancarella P., Pedreschi, D., Turini, F.**, Intensional Negation of Logic Programs: Examples and Implementation Techniques. In *Proc. of TAPSOFT '87*, LNCS 250, Springer-Verlag, 1987.

[BMPT89] **Barbuti R., Mancarella P., Pedreschi D., Turini F.**, A Transformational Approach to Negation in Logic Programming. *Journal of Logic Programming*, 8(3), 1989.

[Cha88] **Chan D.**, Constructive Negation Based on the Completed Database. In *Proc. of the Fifth International Conf/Symposium on Logic Programming*, Seattle (1988).

[Cla78] **Clark K.**, Negation as Failure. In Gallaire, H. and Minker, J. editors, *Logic and Databases*, Plenum Press, New York, 1978.

[Cla81] **Clark K.**, The Synthesis and Verification of Logic Programs. Research Report 81-36, Dept. of Computing, Imperial College, London (1981).

[GL86] **DeGroot D., Lindstrom G.**, (editors) *Logic Programming, Functions, Relations and Equations*. Prentice-Hall, (1986).

[HT82] **Hansson Å., Tärnlund S-Å.**, Program Transformation by Data Structure Mapping. in Clark, K. and Tärnlund, S-Å (editors). *Logic Programming*, Academic Press (1982).

[Hog81] **Hogger C.J.**, Derivation of Logic Programs. *J. ACM* 28, 2 (1981).

[JLL83] **Jaffar J., Lassez J-L., Lloyd J.**, Completeness of the Negation as Failure Rule. In *Proc. of IJCAI-83*, Karlsruhe, 1983.

[JLM84] **Jaffar J., Lassez J-L., Maher M.**, A Theory of Complete Logic Programs with Equality. *J. Logic Programming* 1(3) (1884).

[Kur87] **Kursawe P.**, How to Invent a Prolog Machine. *New Generation Computing*, 5 (1987).

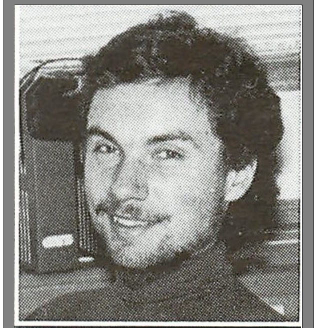
[Sha83] **Shapiro, E.**, Algorithmic Program Debugging. *MIT Press* (1983).

[She88] **Shepherdson J.**, Negation in Logic Programming. In Minker, J. editor, *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann, Los Altos, 1988.

[War84] **Warren D.H.D.**, An Abstract Prolog Instruction Set. *Technical Note 309*, SRI International (1984).

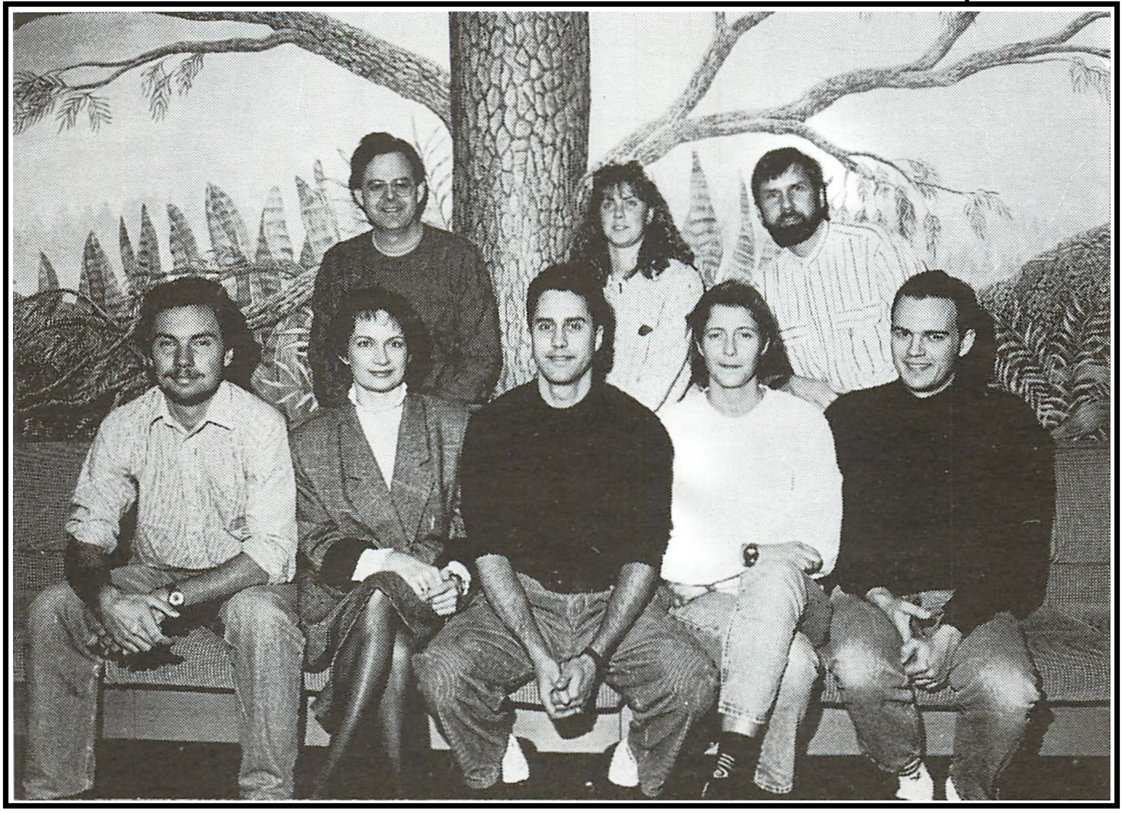
MDA

10 People, Computers and Work



Toomas Timpka
director

Computerization in work life
Hypermedia
Action research
Video studies
Qualitative methods



Back row: Arne Fäldt, Lena Wigh, Hans Holmgren.
Front row: Toomas Timpka, Ewa Rauch, Mikael Peolsson, Cecilia Sjöberg, Per Hedblom.
Missing: Tom Buur, Lars Reshagen, Ying Shu, Tom af Klercker.

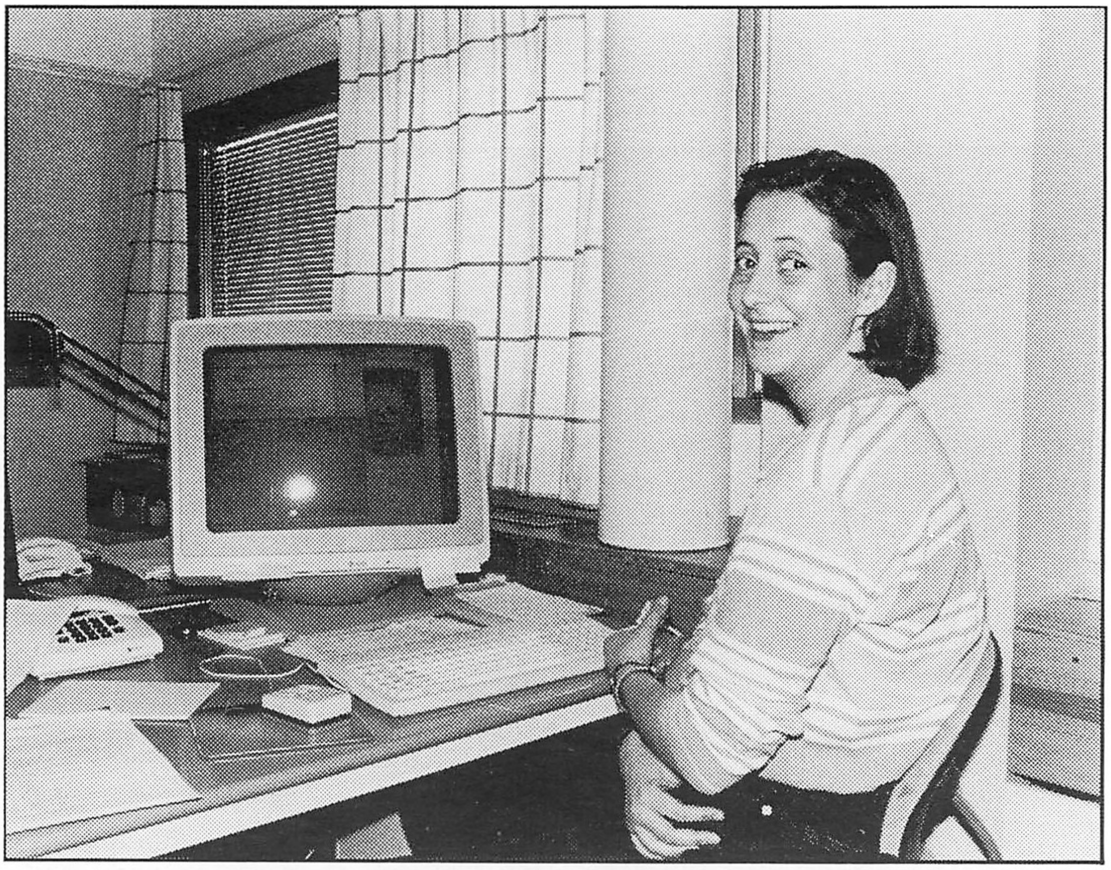


Figure 10-1. *Systems development within the MDA-group is based not only on computer science, but also on knowledge from psychology, graphical design, and directly from professionals practicing in the application field. Cecilia Sjöberg has this area as topic for her thesis project.*

10.1 Introduction

The MDA-group conducts research into *information system development and use in working-life* from the points of view of computer science, psychology, and the social organization of work development. Within the MDA-group, activities at the Department of Computer and Information Science and the Medical Faculty have been coordinated to develop and evaluate experimental information systems. The actual research since the formation of the group in 1988 has been directed towards four interdisciplinary areas: action research, hypermedia, studies of work practices, and the role of video in computer and information science.

10.1.1 Action research

The notion of *action research* is the basis for the activities of the MDA-group. In action research the aim of the research is to establish *forms and processes of development* that people can use to change their own lives and working conditions. Hence, the researcher involves the organization under study in acting upon subjective and objective descriptions of their work. Action research can be described by five characteristics:

- collaboration between researchers and the members of the target organization in terms of aims and performance of the research;
- approval of the parallel goals of practical problem solving and knowledge building;
- continuous feedback to participants of research findings, as well as results obtained by outside researchers in order to stimulate reflection;
- recognition of *group dynamics* in the research process. Attention is paid to values, power structure, opinion leaders, accepted experts, etc.; and
- cyclical research process. Description, action, and evaluation phases succeed each other in an uninterrupted cycle.

10.1.2 Hypermedia

The principle behind hypermedia is to use computers for easily making and following links between stored information, weather this information consist of text, images, graphics or sound. With hypermedia technology, the user is in total control of the interaction with the computer. Consequently this technology, more than most, can be made to fit into habitual practices. Hypermedia is thus interesting for working-life applications since this technology, with proper assessment and evaluation, has the potential to better support work routines of professionals than other computer techniques, such as expert systems.

A special issue is the role visual material has or should have in information systems, especially in hypermedia systems. Reasoning in many areas in work life, for instance in health care and engineering, is based on spatial abstraction of structures, and for this reason issues of visualization need to be brought into the design of hypermedia

MDA (Människor, Datateknik och Arbetsliv) is the swedish for People, Computers and Work.

systems to be used for decision support. Integrated images and real-time video at least gives the *potential* to design computer systems which address issues more conventional media can not. Hence, development of such systems should be based not only on computer and information science, but also on knowledge from psychology, graphical design, and directly from professionals practising in the application field. We believe reflection through action research is essential to find the *forms* for such an inter-disciplinary development process.

10.1.3 Studies of work practices

Qualitative research methodologies are important for empirical studies performed in the working-life setting since these methods can be used to pick up useful data other methodologies can not. This is because traditional methodologies for development of computer systems tend to work from fixed, predetermined models (input, processing, and output of data). Even when pre-tested, their categories tend to be fixed. On the other hand, qualitative methods attempt to elicit what is perceived significant and important to the organizations and individuals we study. Consequently, there is less chance of missing important issues. In short, the qualitative method allows us to pick up and understand what for our subjects is taken for granted and routine practice. The importance this has for studies of working-life and fitting technology into habitual practice and the practices of working life is considerable.

10.1.4 Video in Computer Science research

Another of our interests is the use of video as a tool in computer and information science research. Video can, firstly, be used for documentation in empirical studies to understand human-computer and human interaction. It constitutes a direct and contextually rich means for capturing observable behaviour. However, one major problem lies in analysis of the recorded material. Use of qualitative methods is an interesting alternative from our perspective. Secondly, video can be used directly in the system design process for visualization of design alternatives. Design can be seen as a social activity where video in many situations is the most efficient medium for documentation and communication of ideas.

10.2 The group

The members of the MDA-group at the Department of Computer Science are (October 1990).

Group leadership and administrative staff:

Toomas Timpka, MD, assistant professor
Lena Wigh, secretary

Graduate researchers:

Tom Buur
Shamsul Chowdhury

Graduate students:

Per Hedblom
Hans Holmgren
Mikael Peolsson
Ying Shu

Technical staff:

Arne Fäldt
Leili Lind
Ewa Rauch
Lars Reshagen
Cecilia Sjöberg

Göran Goldkuhl from the ADP-group has been associated with the group since its formation in 1988. At the Department of Community Medicine have Per Bjurulf, Elisabeth Arborelius, Tom af Klercker, and Ann-Charlotte Nilsson been involved in the MDA-group. Jim Nyce, PhD., Brown University, USA, has visited the group twice and also hosted visits from group members during 1989-90.

10.3 Funding

The MDA-group is mainly supported by a grant financed on an equal basis by the Swedish Work Environment Fund and the Swedish National Board for Technical Development. An equipment grant comprising 8 workstations with colour displays was received from the Swedish Council for research in the Natural Sciences. Additional funding has been received from the Swedish Telecom and Östergötland County Council.

10.4 Projects

The MDA-group has been involved in three projects investigating, respectively: computer use during consultations in the service sector, the impact of physiological modelling on decision-making in hemodialysis, and computer support for statistical analyses.

10.4.1 ‘‘Datorstö(r)d’’ – computer use during consultations

Between 1988 and 1992, the project “Datorstö(r)d vid konsultation?” is the main focus for the MDA-group. “Datorstö(r)d” is an inter-disciplinary research project aimed at investigating computer support for pairs of users, one of whom is giving professional advice to the other. The particular working environment which is studied is consultation situations in primary health care. Computer systems, mainly using hypermedia techniques, are being developed specifically aimed at supporting the cooperation between health professional and patient. The category “health professional” is to be understood in a wide sense. Physicians and nurses are essential here, but paramedicals and medical secretaries are examples of other groups whose professional activities are of direct interest for the project. However, conclusions and results are expected to be directly relevant for many other types of consultation situations within the today rapidly growing service sector.

Project work is carried out in cooperation with the staff of the primary health care center in Mjölby, situated 40 km west of Linköping. The project is also discussed with representatives from the County Council of Östergötland, who are responsible for the provision of health care in Mjölby. Close contacts are maintained with the groups of Professors Werner Schneider and Gösta Tibblin, both at Uppsala University. Project supervisor is Professor Per Bjurulf from the Department of Community Medicine.

10.4.1.1 Results

A hypermedia system for primary health care is under development. Both software design and the introduction of the system at the health care center are framed within action research. The system development is based on video studies of the actual work practices.

10.4.1.1.1 Video studies

No scientific methods have been available for study of the difficulties, dilemmas, experienced by professionals during day-to-day consultations. Such methods have relevance for designers of computer systems, educators, and administrators. Hence, based on a theoretical background in ‘‘continental’’ phenomenology, the practical details have been worked out of a method where a video-recording is reviewed for stimulated recall of perceived difficulties (Timpka and Arborelius 1989, Arborelius and Timpka 1990). The stated dilemmas are categorized in two ways: by an inductive approach, and by a schema derived ‘From Jurgen Habermas’ epistemological theory. The method has been employed in studies of both General Practitioners (GPs) (Timpka and Arborelius 1990, Arborelius and Timpka 1990, Arborelius, Bremberg and Timpka 1990) and Primary Care Nurses (PCNs) (Timpka and Arborelius 1990). The studies of the GPs showed that medical information dilemmas were encountered during three out of four consultations. Dilemmas in the communication with the patient occurred during two consultations out of three, while dilemmas in the organizational environment and dilemmas challenging the GP’s personal competence occurred during one consultation out of three, respectively. A phenomenological analysis of the comments showed that few dilemmas were described as problems during ‘‘hypothetico-deductive’’ reasoning. In many cases the GP found it difficult to *understand* the situation as a whole. Analysis of the dilemmas of the PCNs showed a similar picture. Medical information dilemmas were the most frequent, occurring in two consultations out of three. Dilemmas in the area of inter-personal communication frequently concerned distrust, either in what the patient presented, or in what the patient said to have understood.

10.4.1.1.2 Action research

Little has previously been published on methods for development of information systems as action research, and even less has been concerned with evaluation of this approach in practice. The MDA-group has, firstly, evaluated a graphical method for encouraging professionals to reflect over their work routines, and, secondly, produced a method for software development taking its point of departure in workplace routines.

SIM (Goldkuhl and Röstlinger 1988) is a method, or “methodological toolbox”, for system development, whose originators have been inspired by the action research paradigm. SIM includes detailed recommendations on three levels: methodological attitude, analysis tools (forms of documentation) and project management, all of which should support an action research style of information systems development. One of SIM’s methodological formalisms is called “activity graphs”, a flow-oriented graphical notation to be used to document work activities. In an evaluation, these graphs were used cooperatively by health care professionals and researchers to describe aspects of communication and information flow in daily work in order to *promote reflection* among the professionals, and to *create understanding* on the part of researchers (Holmgren and Timpka 1990). The activity graphs were readily accepted by the staff and was found to have descriptive strength. Passive competence (capacity to interpret a finished graph) was quickly acquired by everyone involved. However, staff members exposed to the description technique acquired very little active competence in using it themselves. Moreover, there was little evidence that descriptive activities had stimulated any deeper reflection on the part of staff members.

One major problem with close-to-work methods for software design, and with user participation in general, has been the *making use* in the actual software development process of the vast amount of empirical data and experiences collected in the first phases of design. Our solution to this problem is to extend the methods to *action design* where an interdisciplinary group functions as a reference source for exploration and experimentation at the workplace, and where object-oriented programming environments (OOEs) are used as framework for the software development (Timpka, Hedblom, and Holmgren 1990). However, the main point of action design is still that *real* users should reflect and act upon early versions of software while doing *real* work. By using an OOE where functionality and user interface are very tightly coupled it is possible to run applications almost from the start of software development. It is also possible to make rapid changes, even going deep down in the structure of the application.

10.4.1.1.3 Hypermedia in primary health care

The studies of the work practices of GPs and PHCNs showed, hence, that their methods of information processing had clear similarities with those employed by scholars in the humanities, e.g., by historians. Thus, since hypermedia has showed to be useful in the humanities, it is to be relevant for diagnosis support in primary health care as well. ‘‘Gösta’s book’’ is a collection of hypermedia documents for this purpose, developed in co-operation with the Department of Family Medicine at Uppsala University (Timpka, Hedblom, and Tibblin 1990). The authoring of this hypermedia collection is a continuous process, where three steps are iterated: content authoring, structure editing, and systems implementation. The structure editing includes the authoring of *overview nodes* and *guided tours* through the collection. Overview nodes define a document in the collection by both describing a problem area and linking to all basic text nodes related to the area. Authoring of these navigation aids is integrated with structure editing and has to be performed by the content and structure authors together. The resulting collection is divided into 22 hypermedia documents, consisting of 500 basic text objects and connected by 3000 links. *Gösta's book* has this far been implemented in two versions: firstly, as an hypertext interface to an object-oriented database, NODE (Timpka et al. 1989); and secondly, as a stack in Hypercard (TM). A third implementation is under way using an object-oriented programming environment. The implementations are all focused on the user interface and have been developed in close co-operation with health care professionals.

10.4.2 Impact of physiological modelling on clinical practice

Modelling and simulations of physiological systems in the human body provide health care professionals with new information, previously not available for clinical decision-making. The aim of a collaboration led by Tom Buur between the Department of Nephrology and the MDA group is to investigate the impact on clinical routines at a dialysis unit, at both individual and organizational level, by making available computer-based modelling of the generation and removal of a marker blood molecule (urea).

10.4.2.1 Results

A study of the decision-making at a hemodialysis unit was performed and a program, DiaKin, for physiological modelling of urea kinetics was developed. Thereafter, an evaluation was made to find the forms for the introduction of physiological models in routine decision-making.

10.4.2.1.1 Hemodialysis decision-making in practice

To examine hemodialysis decision-making in the practice setting, four physicians independently assessed 62 patients on maintenance hemodialysis (Timpka and Buur 1990). Deviation from normal was determined for adequacy of dialysis, protein intake, and metabolic state. The kappa-index, by which chance agreement is adjusted for, was used to analyze inter-observer variation for each monitoring diagnosis. Low agreement was found on decisions concerning adequacy of dialysis, while agreement was higher about protein intake, and metabolic state. Routines for review of recent medical history also differed significantly between the physicians. The conclusion is that measures are needed to increase the reliability of decisions regarding the monitoring of chronic hemodialysis. Medical audit as a part of the clinical routines, and use of additional sources of information, such as physiological modelling, are possible steps to take.

10.4.2.1.2 DiaKin

A program for the IBM PC performing physiological modelling of relevance for the prescription of hemodialysis treatment was developed (Buur 1990). The program has a spreadsheet-like interface, and results can be presented graphically. The present implementation (October 1990) covers 89 algorithms/equations, all based on the assumption of single-pool kinetics. Some of these are detailed for the first time, including considerations on their implementation using a generalizing, structured approach.

10.4.2.1.3 Evaluation in the practice setting

Four physicians evaluated dialysis adequacy, daily protein intake, and metabolic stability in 62 patients (Buur and Timpka 90). Physiological modelling (urea kinetics) was done in parallel, but the results were not revealed. Clinicians' decisions were then compared with the model's measures of effective dialysis and protein catabolic rate. Detection of inadequately treated patients was poor (28%). Combining protein catabolic rate and consensus decisions, 59 patients could be assigned a ‘‘probable’’ daily protein intake. Using this as a *gold standard*, the average clinician detected 10 of 13 patients with low daily protein intake. Simulated decision making suggested that combining the model with clinical evaluation in a logical way would lead to detection of most patients with low daily protein intake. The conclusion is that physiological modelling should be used routinely.

10.4.3 Support in statistical analysis

With increasing organizational decentralization and ‘‘flattening’’ of hierachies in work life, statistical analysis of data is becoming a routine for many more professionals than the traditional managers and clerks. A statistically less experienced professional may need to be supported in any one of the three phases of data analysis and interpretation, namely:

- to provide a way to choose the appropriate statistical technique/s after examining the data;
- to attain a more fruitful data analysis by providing guidance in the analysis processes; and
- to improve the understanding of the results through interpretation.

In a collaboration led by Shamsul Chowdhury between the Department of Medical Informatics and the MDA group, the use of Artificial Intelligence (AI) techniques are investigated which provide interfaces to support professionals in the process of selecting the statistical method and corresponding software, in preparing parameters and data, and in interpreting the results of the computation. The emphasis is on the statistical aspects of a computation rather than the computational, thereby providing more close-to-practise statistical software.

10.4.3.1 Results

The developed software, MAXITAB, is termed a statistical consultation system since users can interact with it for guidance when analyzing data, for help in choosing and correctly applying the chosen techniques as well as for help in interpreting and presenting the results. MAXITAB has been evaluated in two settings in work life, both connected to development of decision support systems: in the validation of database maintenance measures, and in an empirical study.

10.4.3.1.1 The MAXITAB system

Computer support for data analysis and interpretation could be developed using one of the following methods.

1. write/develop the entire analysis system using an AI approach;
2. write/develop the support module in an AI environment and interface with existing statistical packages; or
3. build the support module within the environment provided by the statistical package.

The MAXITAB system was developed based on the second approach (Chowdhury, Wigertz, and Sundgren 1989). The front and back end interface program MAXITAB provides support in data analysis and interpretation. MAXITAB has been programmed as an interface to the MINITAB statistical package. It does not automatically interpret the results, but provides support to a domain expert to conclude whether or not the analysis shows a statistical or causal dependency between the variables.

10.4.3.1.2 Application in working-life

The facts in a system for decision support are usually acquired from experts and the literature. But the facts can also be extracted from a database of observations and from interpretation of those observations. The resulting system would be more accurate in the latter case, especially if it is intended to operate in decision support in the same setting as the data were collected. MAXITAB was utilized in two studies conducted to investigate how retrospectively collected data could be used in this process. In the first study a database of liver disease patients was used. Necessary manipulations such as the detection of outliers, the estimation of missing values were studied to improve the usability and reliability of the database. The data material used in the second study was collected from Kronan Health Care Center in suburban Stockholm. The problem here was to empirically test the hypothesis of a possible causation between hypertension and diabetes. Statistical expertise is usually required to structure this kind of analysis to follow an appropriate path.

In the first study the ‘‘K nearest neighbour’’ technique was found to be an easy and efficient method for estimation of missing values for cases with only a few missing values in the database. Inductive learning technique for prediction of missing values could perform well in the prediction of missing values, where the prediction intervals are known with certainty from the domain. The results of the second study support the assumption that there is a relationship between diabetes and hypertension but the question of the direction of this relationship remained unsolved, as did the question of direct causality.

10.4.4 Courses and supervision

The MDA-group provides post-graduate courses and supervision of Master’s students in our fields of interest. The intention is to bring to the students an inter-disciplinary approach to research, and to expose them to problems faced by practising professionals outside the University. Courses given are, firstly, *Hypermedia: History, concepts, and applications* where the aim is to give a basic knowledge of hypermedia systems and their use in working-life and education. This course was given for the first time 1989/90. Secondly, a course called *Empirical research methods* is given in 1990/91 which provides a basic knowledge of quantitative and qualitative methods for empirical research.

10.5 Publications

Dissertations:

Tom Buur: Computer methods and kinetic methods in hemodialysis. *Linköping Medical Dissertations nr. 318*, Linköping University, 1990.

Shamsul I. Chowdhury: Computer-based support for knowledge extraction from clinical databases. *Linköping Studies in Science and Technology. Dissertations nr. 240*, 1990.

Papers in international scientific journals:

Timpka T, Ekström M, Bjurulf P: Information needs and information seeking behaviour in Primary Health Care. *Scand J Prim Health Care 1989*;7:105–9.

Timpka T: Introducing hypertext in primary health care: – A study on the feasibility of decision support for practitioners. *Comp Prog Biomed 1989*;29:1–13.

Timpka T, Bjurulf P: The semantics of diagnosis and management of genitourinary infections: A cross-specialty study. *Family Practice 1989*;6:279–85.

Chowdhury SI, Wigertz O, Sundgren B: A knowledge-based system for data analysis and interpretation. *Methods of Information in Medicine 1989*;28:6–13.

Timpka T, Bjurulf P, Buur T: Audit of decision-making regarding female genitourinary infections in outpatient practice. *Scand J Inf Dis 1990*;22:49–57.

Arborelius E, Timpka T: General Practitioners' comment on video recorded consultations as an aid to understanding the doctor-patient relationship. *Family Practice 1990*;7:84–90.

Timpka T, Arborelius E: The GP's Dilemmas: A Study of Knowledge Need and Use During Health Care Consultations. *Methods of Information in Medicine 1990*;29:23–29.

Buur T, Timpka T, Lundberg M: Urea kinetics and the clinical evaluation of the hemodialysis patient. *Nephrology, Dialysis and Transplantation 1990*;5:347–51.

Timpka T, Buur T: Inter-observer variation in decision-making regarding patients on chronic hemodialysis. *Methods of Information in Medicine 1990*;29: 153–7.

Buur T, Timpka T: Improvement of decisions regarding hemodialysis patients by urea kinetic modeling. *Medical Decision Making 1990*;10:274–82.

Timpka T, Hedblom P, Tibblin G: A hypermedia document collection for primary health care: Why, What, and How. *Artificial Intelligence in Medicine 1990 2*;179–92.

Timpka T, Arborelius E: The primary health care nurse's dilemmas: a study of knowledge use and need during telephone consultations. *Journal of Advanced Nursing 1990*;15:1457–65.

Arborelius E, Bremberg S, Timpka T: What is going on when the general practitioner doesn't grasp the consultation? *Family Practice 1991*, in press.

Arborelius E, Timpka T: In what way may videotapes be used to get significant information about the patient-physician relationship. *Medical Teacher 1990*;12:194–208.

Arborelius E, Timpka T: A comparison of patients' and physicians' comments on video-recorded consultations. *Scand J Primary Health Care 1991*, in press.

Timpka T, Arborelius E: A method for study of dilemmas during health care consultations. *Medical Informatics 1991*, in press.

Timpka T, Buur T: Medical reasoning and patient requests in decision-making for female genitourinary infections. *Methods of Information in Medicine 1991*, in press.

Books:

Timpka T: Design of decision support for General Practitioners. *Health Services Studies*, WHO Linköping Collaboration Centre, 1990.

Papers in conference proceedings and other scientific periodicals:

Timpka T, Arborelius E: Study of the Practitioner's Knowledge Need and Use During Health Care Consultations. Part 1: Theory and method. *Proc. of MedInfo' 89*, North-Holland, Amsterdam, 1989.

Timpka T, Padgham L, Hedblom P, Wallin S, Tibblin G: A hypertext knowledge base for primary health care. LIMEDS in LINCKS. *Proc. of SIGIR '89*, ACM Press, New York, 1989.

Arborelius E, Timpka T: Study of the Practitioner's Knowledge Need and Use during Health Care Consultations. Part 2: The dilemma spectrum of the GP. In *Proc. of MedInfo '89*, North-Holland, Amsterdam, 1989.

Waern Y, Bannon L, Timpka T, Schneider W: User participation in HCI research – Effects on processes and results. *Proc. of Interact '90*, Elsevier, Amsterdam, 1990.

Holmgren H, Timpka T: System development in health care as action research. *Proc. of Medical Informatics in Europe '90*, Springer Verlag, Berlin, 1990.

Timpka T, Hedblom P, Holmgren H: Action design: Using an object-oriented environment for group process development of medical software. *Software Engineering in Medical Informatics*, Elsevier, Amsterdam, 1991.

Timpka T: Information system development as action research: Action design. *Proc. of the Seminar on Medical Database Systems – Computer Aided Management and Economics in Health Care*. Warsaw: International Center of Biocybernetics, 1990.

Chowdhury SI: Need and relevance of AI methods in statistical analysis and interpretation of data. In P*roc. of the 9th International Workshop on Expert Systems and their Applications*, Avignon, France, 1989, 627–643.

Chowdhury SI, Wigertz O, Sundgren B: Artificial Intelligence Methods in Data Analysis and Interpretation. In *Knowledge, Data and Computer-Assisted Decisions* (eds. M. Schader and W. Gaul), Springer-Verlag Berlin Heidelberg 1990, 199–208.

Arborelius E, Bremberg S, Timpka T: Videoinspelade läkarbesök – vad gör läkaren när han blir ställd? En studie av de svårigheter distriktsläkare upplever i kommunikationen med patienten. *Läkartidningen 1991*, in press.

Arborelius E, Timpka T: En jämförelse av patienters och läkares kommentarer av videoinspelade läkarbesök. *Allmänmedicin 1990;11*:163–6.

Peterson H, Almquist G, Hesselius I, Schneider W, Timpka T: Aspekter på säkerhet och sekretess vid datoranvändning inom hälso- och sjukvården. (Panel). Svenska Läkarsällskapets Riksstämma. *Hygiea 1989;98*. (Abstract).

Timpka T, Buur T: Beslutsvariation vid hemodialys. Svenska Läkarsällskapets Riksstämma. *Hygiea 1989;98*. (Abstract).

Holmgren H, Timpka T: Från systemutveckling till aktionsforskning: ett nytt synsätt på användarmedverkan. Svenska Läkarsällskapets Rikstämma. *Hygia 1990* (Abstract).

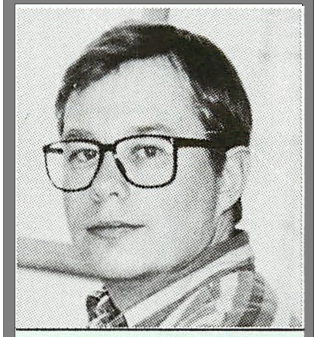
Buur T, Denneberg T, Lundberg M, Timpka T: How well is the elderly haemodialysis patients doing? *Nordiske nyredage 1991*, in press (abstract).

Research reports:

Goldkuhl G: Förändringsanalys på en vårdcentral: Erfarenheter från ett organisationsutvecklingsprojekt. Institutionen för Datavetenskap, Universitetet i Linköping. *Rapport LiTH-IDA-R-90-08,* 1990.

NLPLAB

11 The Laboratory for Natural Language Processing



Lars Ahrenberg
director

Natural language processing
Natural language interfaces
Discourse representation
Unification-based grammar
Parsing
Semantic interpretation
Generation

Back row: Lena Andersson, Nils Dahlbäck, Arne Jönsson, Mats Wirén.
Front row: Stefan Svenberg, Richard Hirsch, Lisbeth Linge, Lars Ahrenberg.
Missing: Åke Thurée.

Figure 11-1. *Lisbeth Linge, secretary of NLPLAB and technical advisor for the administrative staff.*

11.1 Introduction

The Natural Language Processing Laboratory was formed in 1986 with five members. Since then the group has grown slowly but steadily and research interests have been extended in several directions; from grammar formalisms to general knowledge representation languages, from parsing to generation and general inference techniques and from dialogue to text understanding. The group receives external support from the Swedish National Board of Technical Development (STU) and from the Board of Research in the Humanistic and Social Sciences (HSFR). There is also a project within the local interdisciplinary research program concerned with industrial information technology, CENIIT.

Laboratory leadership, administrative and technical staff:

Lars Ahrenberg, PhD, associate professor
Lisbeth Linge, secretary
Bernt Nilsson, research engineer

Employees:

Richard Hirsch, PhD, lecturer (as from November 1990)
Arne Jönsson, MSc, BA, Tekn.lic, lecturer
Åke Thurée, MSc, research assistant

Graduate students:

Lena Andersson, MSc
Nils Dahlbäck, BA
Magnus Merkel, BA, Fil.lic (until June 1990)
Stefan Svenberg, MSc
Mats Wirén, MSc, BA, Tekn.lic.

11.2 Overview of current research

11.2.1 Natural Language Dialogue Systems

The design of natural language dialogue systems for use as interfaces to information systems has been a long term research area for us. We have investigated this area from several perspectives. On the one hand we have been interested in the architecture of a “shell” for a NL dialogue system that could be adapted to background systems setting different requirements on the interface as regards domain knowledge and communicative behaviour. We have designed such a system, called the Linköping Natural Language Interface, or LINLIN, for short. We have also been interested in issues of knowledge representation for the system, concerning the representation of static knowledge, both linguistic and world knowledge, as well as the dynamic knowledge-bases needed to support a coherent dialogue. A third important goal has been the characterization of the sublanguage of man-machine communication in natural language, on the assumption that this language differs from the language used in dialogues between humans in many respects.

11.2.1.1 The LINLIN architecture

To be of general use as an interface system a dialog system must meet a number of requirements. Only some of these are actually connected to the system’s ability to understand and produce natural language, but even if we restrict ourselves to such problems, it is unlikely that general-purpose systems can be developed. This is so because the language requirements are different in different applications. Thus we favour a sublanguage approach to the design of a NLI. For instance, it is an advantage if meanings of a word that do not occur in the specific knowledge domain of the application are not listed as alternatives in the dictionary used. But the specific linguistic requirements are not limited to vocabulary for the expression of domain concepts, but also concerns syntactic constructions, the speech acts likely to occur in interactions with the system and the ways context is exploited.

The declarative knowledge-bases of the system which can be changed to suit the needs of a given application thus comprises not only the dictionary and the domain concepts, but the grammar and the dialogue objects, i.e. the possible moves (speech acts) and exchanges as well. An overview of the system is presented in Figure 11-2.

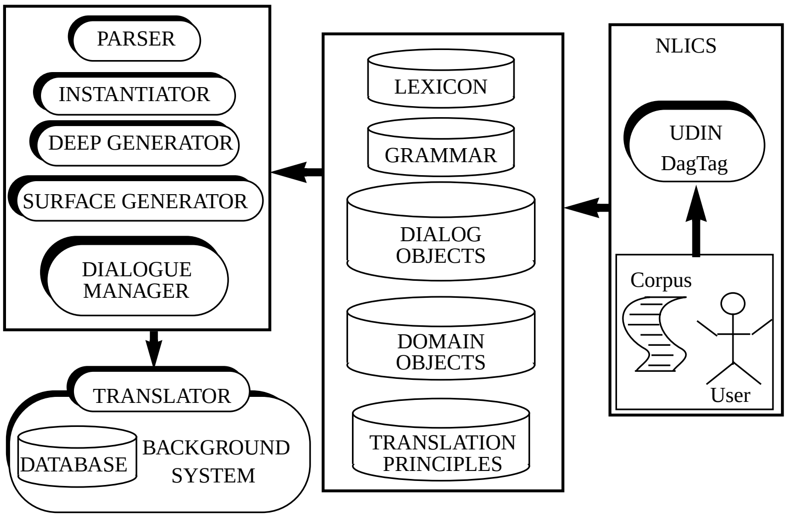


Figure 11-2. *The LINLIN architecture*

All knowledge of the system is represented in the same structure, and in the same representation language. This makes it possible to develop the linguistic knowledge and the domain knowledge simultaneously in the same environment. It also makes it possible in principle to integrate syntactic and semantic processing, although this has not been done in the system so far.

While the system is thus representationally integrated to a large extent, the representation language is a non-principled mixture of two traditions, the restricted unification-based formalisms developed in computational linguistics with the unrestricted object-oriented representational languages developed in AI. Interestingly enough, we can see a development on the linguistic side towards extending the formalisms in various ways (cf. below) while at same time there are directions on the AI-side to restricting the representation languages to make the subsumption relation tractable. A goal for our current research is thus the development of a representation language which is rich enough for the expression of a variety of linguistic and general constraints, but still computationally tractable in the average case.

The central processing module of the system is the dialogue manager, DM, which receives user inputs and controls all processing of the system. (Figure 11-3). The parser consists of a chart-parser (Wirén, 1988a) and a unification package (Andersson, 1989) that handles a variant of the PATR-II formalism (Shieber et al., 1983) with several extensions. The instantiator is a module that interprets the output of the parser as an instance of a speech act in the manner of Ahrenberg (1989a). The DM receives such instances from the instantiator and decides how to proceed on the basis of their properties and the current state of the dialogue as represented by the system's dynamic knowledge structures. For the construction of output the DM calls the deep generator. The deep generator basically merges information in stored templates for speech act descriptions with information received from the DM and the background system. Output from the deep generator is given to the surface generator which outputs text on the basis of grammatically complete speech act descriptions. The surface generator uses the same grammar and the same chart techniques as the parser on the model of Shieber (1988).

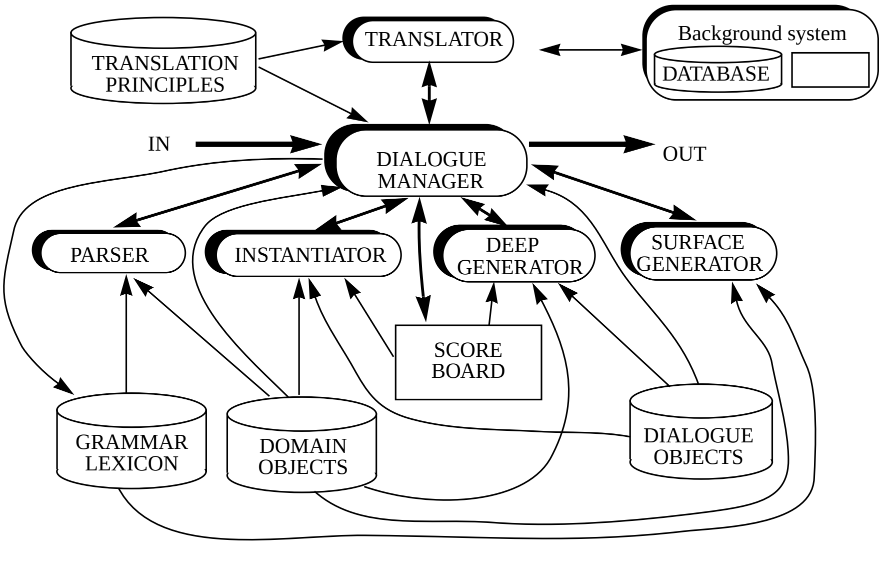


Figure 11-3. *Control and information flow in LINLIN.*

Internally the dialogue manager maintains three dynamic structures for monitoring the dialogue. First there is a scoreboard where results of interactions are kept. The score board is a kind of interface to the other modules and contains information on speaker, hearer, salient objects etc., which is used by the instantiator and deep generator. There is a dialogue tree which represents the dialogue as it proceeds in the interaction. The nodes of the dialogue tree are instances of dialogue objects. Dialogue object descriptions have two parts, one part concerned with attributes such as speaker, hearer, type, topic, context and different types of salient objects, and the second part concerned with the internal actions associated with the object: an action plan. These are combined in the third dynamic structure, the action plan stack where the actions to be performed by the DM are stored.

Three different categories of dialogue objects are currently used to hierarchically structure the communication. The smallest unit is the move. Normally two moves constitute an exchange of information, an Initiative-Response-unit (IR-unit), where the first introduces a goal, which is satisfied, or in some other way eliminated, through the response. Finally, there is a unit corresponding to the dialogue as a whole.

An important feature of the DM is its distributed control. The actions of the action plan are distributed on the nodes of the dialogue tree that are still open. This means that if, say, a user move does not succeed in parsing the input, the user move node creates an instance of an IR-unit for clarification request, which will control the dialogue during the clarification. This IR-unit consists of two parts, one part for prompting the user with a clarification request and another to interpret the user input. Finally the user response is integrated into the dialogue tree.

The distributed design has the advantage that we can use quite simple, local plans, which are combined by means of the dialogue tree. A more detailed description of the dialogue manager can be found in Ahrenberg, Jönsson and Dahlbäck, (1990).

There is as yet no implementation of the system as a whole, but most of the modules exist as independent systems. We are currently working on an implementation using a relational data base as background system.

Thesis work

In his thesis work Arne Jönsson develops a design for a dialogue manager as part of a natural language interface shell to be used by a language engineer when creating an NLI for a specific application. The approach, as summarized above, is based on the idea of sublanguages, and the results from the Wizard of Oz simulations that he and Nils Dahlbäck have been conducting. The work is also concerned with the problems of customizing a natural language interface to an application.

11.2.1.2 Wizard-of-Oz studies

In the project “Models for Human-Computer Interaction in Natural Language” we aim at finding characteristics of the sublanguage of man-machine communication in NL, which would be useful for the design of NL-interfaces. Empirical studies of this kind of dialogues have been undertaken for some time now in our group using so-called Wizard of Oz experiments, i.e. letting users communicate with a background system through an interface which they have been told is a natural-language interface, but which in reality is a person simulating such a device (Dahlbäck & Jönsson, 1989; Jönsson & Dahlbäck, 1988). The work is still in progress, both as regards data collection and analysis. The most recent published analysis is based on 21 subjects using five background systems of different complexity. There are 1047 utterances in this corpus. Among the results obtained this far the following can be mentioned: Only 48% of the utterances can be interpreted in isolation, ellipsis (64%) and definite descriptions (29%) being the most common anaphoric devices, whereas the use of pronouns is limited (16%); the task structure and the number of possible user actions seem to influence the structure of the dialogues more than the type of computer background system (real or simulated) used; indirect speech acts are few and seem to performed using only a small set of standardized expressions.

Current work in this project is focused on two areas. First, we are developing methods of analysis for discourse structure and referential expressions. Second, we will extend the experiments to make it possible to study the differences between dialogues with people and with computers.

For the first task our aim is to make it possible to analyze dialogues based on the principles of the LINLIN architecture. We will therefore use the same IR-categories that are used in that project. The advantage of using such a conceptual scheme is not only that it makes it possible to let the formal-computational and the empirical projects benefit from each other. Another advantage is that it makes it possible to compare the dialogue structure of different types of dialogues, by not being tied to a specific content domain.

Our own work, as well as similar studies by others, indicates that dialogues with computers in written natural language differ from dialogues between people. It is still, however, an open question to what extent these differences are due to assumed and real differences between people and computers as dialogue partners, or due to the qualities of the communication channel. To make it possible to answer this question, we will run an experiment where half of the subjects will be run in a Wizard of Oz situation, and the other half will be told that they are communicating with a person. We will use two or three different types of background systems to avoid the risk of obtaining results with limited generalizability.

Thesis work

The thesis work of Nils Dahlbäck is concerned with empirical studies of connected discourse. The first part starts out from Johnson-Laird's theory of mental models, and evaluates it as a theory of discourse representation both theoretically and empirically. The main result is that the term 'mental model' is used in two different senses; as an image-like representation used for processing text with a strong visual content, and which seems to be under voluntary conscious control, and as a term for a discourse representation structure richer than a mental representation of the text's surface structure, but with less constraints on its format than in the former case. This result is also related to the general discussion within AI and philosophy of mind on the so-called representational theory of mind. In the second part the characteristics of human-computer dialogues in natural language is studied using the Wizard of Oz technique as described above.

11.2.1.3 Analysis of the corpus

For the analysis of a corpus of dialogues an interactive tagging system called DagTag has been designed (Ahrenberg & Jönsson, 1988, Jönsson & Ahrenberg, 1989). DagTag uses attribute-value matrices as tags and is more general than most existing tagging systems in its ability to deal with constituents above word and sentence level. The relation between units on different levels is one of constituency, so that, say, a dialogue can be analyzed into a sequence of segments, which in turn can be analyzed into a sequence of utterances, and so on. Thus, a tree structure is imposed on the dialogue during the analysis.

The system can be used for finding out quite subtle information from the analyzed dialogue, e.g. correlations between grammatical and functional properties. A set of tagged dialogues can be searched by means of *a model dag*, an attribute-value matrix containing arbitrary information. The search will return all units whose tags are subsumed by the model dag.

11.2.2 Unification-Based Grammar Formalisms

Most NLP-applications require that a fairly large subset of the morphemes and constructions of a natural language can be recognized and interpreted. For some applications, such as interactive systems, it is also important that interpretation is reasonably fast. For this to be possible we require a grammatical formalism which is both powerful enough to express the complexity of natural language constructions, yet sufficiently restricted so as to allow recognition and parsing by fast algorithms. We have chosen to use declarative, unification-based grammar formalisms for the purpose. Our work has been concerned both with developing adequate linguistic descriptions, primarily for Swedish, and investigating the logical and computational properties of these formalisms.

11.2.2.1 The need for extensions

Unification-based grammar formalisms in their purest form use term or graph unification as the sole information-combining operation e.g. PATR-II. However, on descriptive grounds several extensions have been proposed, for example disjunction, negation, implication, set values, infinite disjunctions expressed by regular paths (functional uncertainty), generalization, and subsumption. These extensions are intended to preserve the declarativity of the formalism and the monotonicity of unification. Other proposed extensions, such as multiple inheritance for the purpose of avoiding redundancy in large lexical and grammatical definitions, introduce non-monotonicity into the formalism. Extensions have also been proposed for the expression of constituent structure, such as the rules of partial ordering (LP-rules) used in GPSG and HPSG to express word order generalizations.

In our own work on grammar development we have explored several of the proposed extensions and added some of our own. We report on two topics of current interest: the proper treatment of disjunctions and negations in feature structures and the representation and manipulation of field structure in the Scandinavian grammatical tradition in feature structures for the expression and control of word order constraints in configurational languages.

11.2.2.2 Disjunction and negation in feature structures.

Lena Andersson explores various possible and proposed extensions to unification-based formalisms, also including their applications to the Swedish language. The goals of the project are to survey a various possible and proposed extensions, to provide a formal description and characterization of them, and to develop a prototype implementation of a system for manipulating feature structures. So far the work has mostly concerned disjunction and negation. In this work various known unification algorithms for disjunction have been implemented and integrated with negation. This work also includes the development of a new algorithm for disjunctive unification with better computational performance.

Thesis work

The work on extensions to unification-based formalisms is the subject for Lena Andersson's thesis. The aim of the thesis is to develop a formalism that is as powerful as a unification-based grammar with many of the extensions proposed in current literature. The formalism should preserve the nice properties of unification, such as declarativness. Part of the work is concerned with finding a subset of extensions that is as small as possible, but in which other proposed extensions can be expressed. For this set a formal description and characterization will be developed. The work also includes implementation of a system that could be used as a tool by other systems. This means that the question of complexity of the formalism is also an important part of the work.

11.2.2.3 Field structure

In several works the Danish linguist Paul Diderichsen described the structure of clauses and other major categories in the Scandinavian languages with reference to a *schema*, i.e. a finite structure of sequentially and hierarchically ordered elements called *fields* or *positions*. What is interesting about Diderichsen's schemas is, first of all, that they say something true of every subtype of the category to which they apply. Thus, the clause schema is true of all clauses, whether they are declarative or interrogative, passive or active, finite or non-finite. Second, positions differ in the number and category of fillers that they allow and different subtypes of a category may be distinguished on the basis of the topological constraints that they satisfy, i.e. what categories they accept or require in a certain position. Third, there is a correlation between positions and grammatical function, so that a noun phrase can be interpreted as a subject or object on the basis of the position it stands in.

We have used Diderichsen's ideas in a formalism, termed Field-and-Category Grammar, which can be viewed as a variant of LFG (Kaplan and Bresnan, 1983; Ahrenberg, 1990a, 1990b). Apart from the syntactic notions recognized in the phrase-structure part of an LFG, such as category, dominance and linear precedence, it also employs the notion of a field and two important relations between fields and categories: (i) an associative relation between a major category and the schema that is responsible for the precedence relations of its daughters, and (ii) an occurrence relation between fields and the categories it contains. The formalism also makes use of category definitions rather than rewrite rules, so that the categories are organized in a network in which functional as well as topological constraints that are common to a class of categories can be inherited from a common, abstract supercategory.

Current work includes the specification of a Swedish Field-and-Category Grammar (Ahrenberg, 1989b) and the development of a chart-parser that exploits topological constraints in predictions. We are also looking at how fields and schemas can be integrated in a framework such as HPSG (Pollard and Sag, 1987).

11.2.3 Dynamic Language Understanding

Natural language understanding can be thought of as a dynamic process in which an agent changes his information-state on the basis of interpretive action. Our work so far on understanding in the context of a dialogue has captured some aspects of the dynamics of interpretation by maintaining a dynamic discourse representation as described above, but in a deliberately simplified way to meet requirements on modularity and customization. Moreover, interpretive action has been sequenced in a syntactic/semantic part and a pragmatic part with very little communication between the two processes, i.e. the system's knowledge of the discourse cannot be used for guiding syntactic/semantic processing, and, conversely, the syntactic/semantic processing of the input has no effect on the discourse representation until it is complete. A higher degree of interaction between these two aspects of interpretation is clearly desirable. In our continued work we want to find more principled solutions to the interpretation problem. This work is undertaken in the project Dynamic Natural Language Understanding, financed jointly by STU and HSFR.

In this work we will continue to merge processing and representation techniques from computational linguistics and AI, though with the aim of a principled comparison and evaluation of these techniques. Some initial results have been obtained and are elaborated below.

11.2.3.1 Incremental chart parsing

Incremental parsing and interpretation may be desirable features for several reasons. First, they constitute a vital property of many types of real-time and interactive natural-language applications, for example, language-sensitive text editing and speech-to-speech translation of telephone dialogues. Systems of this kind require a capability of analysing text on-line, as it is received, and preferably also to cope with changes of previous input without reverting to exhaustive recomputation. Furthermore, it is sometimes useful to be able to give linguistic status to sentence fragments, for example in multi-media systems, which mix, say, deictic and natural-language input, and in systems that attempt to diagnose ill-formed input. Incremental parsing is also needed in order to realize anticipation feedback loops in incremental text generation.

Secondly, a number of psychological inquiries provide support for the view that umans understand sentences incrementally. Models of incremental parsing and interpretation may therefore be used in order to simulate the internal, mental processes underlying this behaviour (cognitive modelling) and to *mimic* this behaviour (in artificial intelligence).

Mats Wirén studies the problem of incremental parsing and interpretation. In Wirén (1988b, 1988c, 1989), he has developed an algorithm for incremental parsing based on chart parsing which does not obey the usual restriction to piecemeal, ‘‘cumulative’’ analysis but which handles *arbitrary* changes of the input. He is currently working on improving and generalizing this algorithm, in particular to enable incremental analysis of running discourse rather than just single sentences. This extension is in part being based on an encoding of Kamp's discourse-representation theory in the manner of Johnson and Klein (1986).

Furthermore, this parser has been combined with an interactive (on-line) parsing system. A previous prototype implementation, called LIPS, is currently being partly reprogrammed and improved as part of a Master's thesis by Ulf Dahlén. The system, which can be thought of as a rudimentary language-sensitive text editor, allows the user to freely display-edit a text. The changes are analysed incrementally and on-line by the parser which, at each moment, tries to keep an accurate analysis of what the user has typed in so far. For demonstration and debugging purposes, the system may be told to display the charts incrementally, drawing and erasing individual edges in tandem with the parsing process.

Other ongoing activities of this work are to extend it to incremental interpretation and to investigate the relation between incremental parsing/interpretation and reason maintenance. A preliminary account of the latter is developed in Wirén (1990).

Thesis work

The subject of Mats Wirén's Ph.D. thesis is incremental parsing and interpretation. At the heart of the work is an algorithm for incremental parsing which is grounded in a chart-parsing framework, and which retains the flexibility in terms of control and grammatical formalism that this framework provides. The framework is then extended to incremental interpretation, drawing on work by Haddock (1987), Johnson and Klein (1986), and Pereira and Pollack (1988). An additional aspect of the work is to try to develop the relationship between incremental parsing --interpretation and reason maintenance.

11.2.4 Conceptual Text Representation for Generation and Translation

In the interlingua approach to machine translation the source language text is analyzed into an interlingua structure, i.e. a language-independent representation of the content of the text. This representation is then used as the basis for generation of the target language text. The interlingua approach has been criticized because of its lack of attention to grammatical structure, in particular structural correspondences between the language pair(s) of concern, but in recent years it has become quite popular again, then often by the name of knowledge-based machine translation (KBMT) as AI knowledge-representation techniques have increasingly come to be used (Carbonell and Tomita, 1987).

In the project Conceptual Text Representation for Automatic Generation and Translation, funded by CENIIT, we are investigating a knowledge-based approach to text generation and translation. The project works with texts from Volvo service manuals and Swedish and English as target languages. The primary purpose of the project is to develop a representation language for the descriptive texts of such manuals and test its usefulness by generating actual Swedish and English text from sample representations.

A demand on the representation language is that it should be rich enough to represent the components that the texts describe and their properties and relations. Thus, a user of the system could point to an object, displayed as a node of a network or by an image, and get answers to questions such as "What is this?", "What does it do?", “What parts does it have?”. Another demand is that it should comprise the necessary features for the expression of linguistic knowledge.

The text to be generated is specified as a sequence of smaller text objects. Every text object is specified for textual structure, propositional content, topic and (illocutionary) function. Grammatical and lexical information is added at the sentence and clause levels by unifying the semantic/pragmatic information contained in the description with the grammar(s) of the target language(s). The format of the grammatical information is in extended PATR-II, similar to that used in other projects.

Internal references:

The following are articles and reports written by members of NLPLAB that are referenced in the text above. For a full set of recent NLPLAB publications, please see Appendix E.

Ahrenberg, Lars (1989a). A Constraint-Based Model for Natural-Language Understanding and a Pilot Implementation. Research Report LiTH-IDA-R-89-22.

Ahrenberg Lars (1989b). A Formal Field Grammar. Research Report LiTH-IDA-R-89-46.

Ahrenberg, Lars (1990a). Topological Fields and Word Order Constraints. *Edinburgh Working Papers in Cognitive Science, Vol. 6: Parametric Variation in Germanic and Romance, Proceedings from a DYANA Workshop,* September 1989, pp. 1–19.

Ahrenberg, Lars (1990b). A Grammar Combining Field Structure and Phrase Structure. *Proceedings of the 13th International Conference on Computational Linguistics,* Helsinki, August 20–25, 1990, Vol. 2, pp. 1–6.

Ahrenberg, Lars and Arne Jönsson (1988). An Interactive System for Tagging Dialogues. *Literary and Linguistic Computing,* Vol. 3, No. 2, pp. 66–70.

Ahrenberg, Lars, Arne Jönsson and Nils Dahlbäck (1990). Discourse Representation and Discourse Management for a Natural Language Dialogue System. To appear in *Proceedings of the Second Nordic Conference on Text Comprehension in Man and Machine,* Stockholm, April, 1990.

Andersson, Lena (1989). A Study in Linguistically Motivated Extensions to Unification-Based Grammar Formalisms. Master’s thesis LiTH-IDA-Ex-8918.

Dahlbäck, Nils and Arne Jönsson (1989). Empirical Studies of Discourse Representations for Natural Language Interfaces. *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics,* Manchester, April 10–12, pp. 291–298.

Jönsson, Arne and Nils Dahlbäck (1988). Talking to a Computer is not like Talking to Your Best Friend. In *Proceedings of the SCAI,* Tromsö, Norway, 1988, pp. 53–68.

Jönsson, Arne and Lars Ahrenberg (1990). Extensions of a Descriptor-based Tagging System into a Tool for the Generation of Unification-Based Grammars. Paper presented at *the 16th Int. ALLC Conference and the 9th ICCH Conference* Canada, June 6–10, 1989 and to appear in *Readings in Humanities Computing,* Vol. 1.

Wirén, Mats (1988a). A Control-Strategy-Independent Implementation of PATR. *Proceedings of the 1st Scandinavian Conference on Artificial Intelligence*, Tromsö, Norway, March 9–11, 1988, pp. 161–172.

Wirén, Mats (1988b). An Incremental Chart Parser for PATR. Research report LiTH-IDA-R-88-11, Department of Computer and Information Science, Linköping University, Linköping, Sweden.

Wirén, Mats (1988c). *On Control Strategies and Incrementality in Unification-Based Chart Parsing.* Linköping Studies in Science and Technology, thesis no. 140, Department of Computer and Information Science, Linköping University, Linköping, Sweden. 1988.

Wirén, Mats (1989). Interactive Incremental Chart Parsing. *Proceedings of the 4th Conference of the European Chapter of the Association for Computational Linguistics,* Manchester, April 10–12, 1989, pp. 241–248.

Wirén, Mats (1990). Incremental Parsing and Reason Maintenance. *Proceedings of the 13th International Conference on Computational Linguistics,* Helsinki, August 20–25, 1990, Vol. 3, pp. 287–292.

External references:

Carbonell, Jaime G. and Masaru Tomita (1987). Knowledge-based machine-translation, the CMU approach. In S. Nirenburg (Ed.) *Machine Translation*, Cambridge University Press, pp. 68-89.

Haddock, Nicholas J. (1987). Incremental Interpretation and Combinatory Categorial Grammar. *Proc. Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy: 661–663.

Johnson, Mark and Ewan Klein (1986). Discourse, Anaphora and Parsing. *Proc. 11th International Conference on Computational Linguistics*, Bonn, Federal Republic of Germany: 669–675.

Kaplan, Ronald and Joan Bresnan (1982). Lexical-Functional Grammar: A Formal System for Grammatical Representation. In J. Bresnan (ed.) *The Mental Representation of Grammatical Relations.* MIT Press, Cambridge, Mass.

Pollack, Martha E. and Fernando C. N. Pereira (1988). An Integrated Framework for Semantic and Pragmatic Interpretation. *Proc. 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York: 75–86.

Pollard, Carl and Ivan Sag (1987). Information-based Syntax and Semantics. *CSLI Lecture Notes No. 13*, Chicago University Press.

Shieber, Stuart M., Hans Uszkoreit, Fernando C. N. Pereira, Jane J. Robinson and Mabry Tyson (1983). The formalism and implementation of PATR-II. In B. Grosz M. Stickel (eds.) *Research on Interactive Acquisition and Use of Knowledge* SRI Final Report 1894, SRI International, Menlo Park, California.

Shieber, Stuart M. (1988). A uniform architecture for parsing and generation. *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest, Hungary, pp. 614–619.

PELAB

Peter Fritzson
director

12 The Laboratory for Programming Environments

Programming environments
Scientific computing
Debugging tools
Incremental compilation technology
Compiler generation
Compilers for parallel hardware
Distributed systems

Back row: Mariam Kamkar, Johan Fagerström, Peter Fritzson.
Middle row: Mikael Pettersson, Gunilla Lingenhult, Lars Viklund.
Front row: Rober Bilos.
Missing: Johan Herber, Johan Ringström, Nahid Shahmehri, Lars Strömberg.

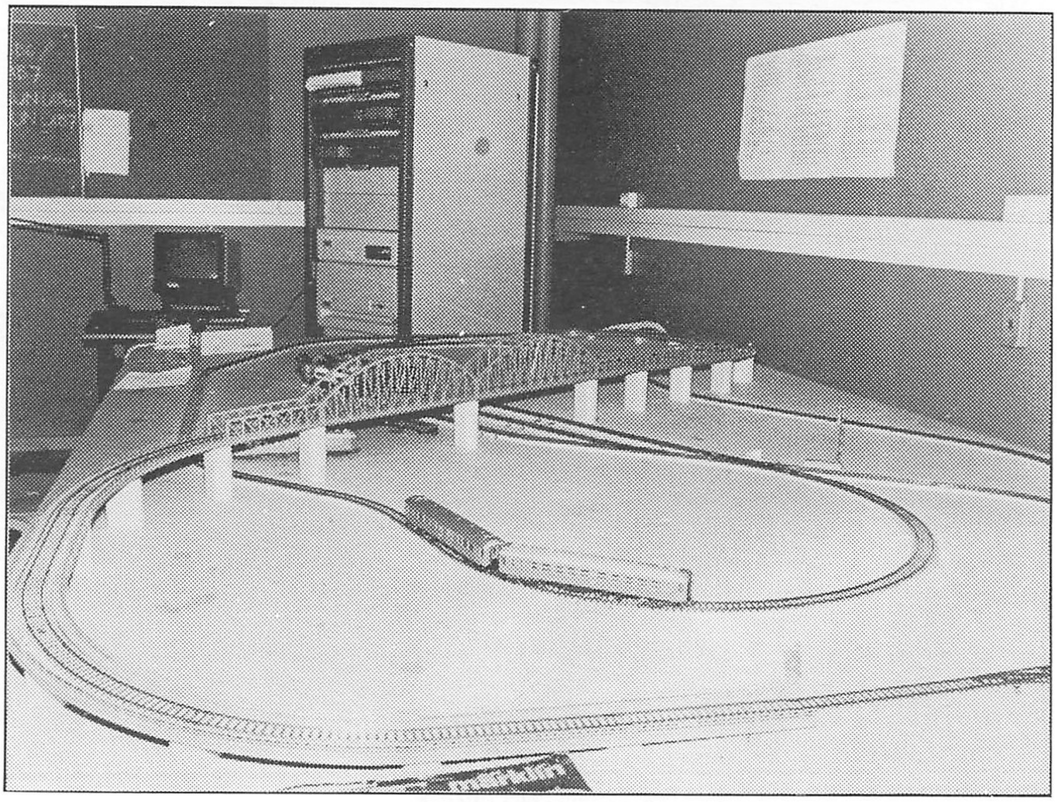


Figure 12-1. *Real-time computing demo lab.*

12.1 Introduction

PELAB, the Programming Environment Laboratory of the Department of Computer and Information Science at Linköping University, is concerned with research in the area of tools and methods for the development and maintenance of computer programs. Some examples are: debuggers, incremental programming environments and compilers, compiler generators, tools for debugging and maintenance of distributed and real-time systems, program transformation systems, etc.

Our view on programming environment research is rather pragmatic. We are primarily interested in developing and investigating new methods and tools that have potential practical applications, e.g. in support systems for software specialists. Developing such tools is very important, since most of the rising cost of computer systems is devoted to development, debugging and maintenance of software.

During the report period, our main research efforts and results have been in the area of debugging support tools. Results have primarily been achieved in the following three areas:

- Semi-automatic debugging support through generalized algorithmic debugging
- Support for debugging distributed and parallel systems
- Practical compiler generation from denotational semantics specifications

In addition, two subprojects have been initiated in new areas:

- Parallel programming and compilation support for parallel hardware
- Programming environment support for scientific programming in equations

12.2 Laboratory Members and Guests

The members of PELAB share their time between undergraduate education and research. The research part is 10 to 80 per cent of full time, varying from person to person, and also from one year to another.

Laboratory leadership and administrative staff:

Peter Fritzson, PhD., associate professor.
Gunilla Lingenhult, secretary.

Laboratory members having a Ph.D. degree:

Johan Fagerström, PhD., assistant professor.

Employed graduate students:

Rober Bilos, Lic.
Mariam Kamkar, Lic.
Mikael Pettersson, MSc.
Johan Ringström, MSE.

The work in PELAB is supported by STU, The Swedish Board for Technical Development

Nahid Shahmehri, Lic.
Lars Strömberg, Lic.
Lars Viklund, MSE (employed 1991).
Johan Herber, MSE (employed 1991).

Guest Researchers:

Petr Kroha, PhD., Technical University, Prague, Feb.–June 1989.
Michael Albinus, PhD., Academy of Sciences, Berlin, GDR, Aug. 1990.

Associated persons:

Zeno Dumbrava, master student in PELAB.
Azadeh Ghaemi, TeleSoft AB.
Yngve Larsson, Softlab AB.
Sven Moen, previous PELAB member, grad. student at Brown Univ., (until 90).
Tommy Olsson, lecturer IDA.
Kjell Post, graduate student at University of California, Santa Cruz.
Öystein Santi, Norsk Hydro.
Dick Schefström, TeleSoft AB.
Dan Strömberg, Swedish Defence Research Institute (FOA).

12.3 Degrees and papers

The following degrees were awarded to PELAB members during 1989/90:

Yngve Larsson completed his licentiate degree. The title of the thesis is: *Dynamic Configuration in a Distributed Environment*.

Lars Strömberg completed his licentiate degree. The title of the thesis is: *Postmortem Debugging of Distributed Systems*.

Seven technical papers were accepted for publication or presented at international conferences by PELAB members during 1989/90. In addition to these, eight technical reports were written.

12.4 Current Research Projects

Our long range goal is to enhance and simplify the programming process, both during the *development* and *maintenance* phases.

Previous PELAB projects have improved the development process, by providing an efficient incremental environment – the DICE system, [Fritzson-83]; and by providing an environment that support development of well-structured distributed applications – the PEPSy project, [Fagerström-88].

Another approach is to raise the language level, i.e. to provide more powerful programming constructs. This is most natural in the context of specialized application areas. Three of our projects use this approach. The goal of one project is to compile language specifications written in DML, Denotational Meta Language, into an efficient language processors that generates quadruples. A first version of this compiler generator is nearing completion during the autumn of 1990. A second project aims at raising the level of parallel programming. The goal is to provide a machine independent way of parallel programming that can be efficiently compiled onto different parallel hardware. A third project aims at constructing a high-level programming environment for scientific programming, which provides support for high-level programming in equations, instead of Fortran.

The debugging aspects of program maintenance were the second focus of the DICE and PEPSy projects. This work is extended in our project on debugging tools for distributed and real-time system. However, the ultimate goal for debugging tools is to be able to both localize and correct bugs. A step in this direction is the algorithmic debugging technique, where the debugging system itself localizes the bug. The project on algorithmic debugging generalizes and improves this technique in several respects.

12.4.1 Generalized Algorithmic Debugging and Program Slicing

PhD project for Nahid Shahmehri and Mariam Kamkar. Supervisor: Peter Fritzson.

Debugging accounts for a large fraction of the total programming expense, and it is not surprising that attempts have been made to automate this task. Several artificial intelligence based tools [Seviora-87] use various ad-hoc pattern matching techniques in order to find and correct bugs. Templates or chunks of code are matched against the program to be debugged. These techniques have severe limitations, both in the small size of the programs handled, the number of templates that need to be supplied, and imprecise program semantics.

However, the *Algorithmic Debugging Technique*, introduced by Shapiro 1983, has a precise semantics and does not need any templates. Instead, the programmer supplies a partial specification of the program during the bug localization process, by answering questions. However, so far this technique has been limited to programs without side-effects, and has only been applied to Prolog programs.

The goal of this project is to generalize and improve algorithmic debugging in several respects.

Algorithmic Debugging for Imperative Languages

So far in this project we have generalized the algorithmic debugging method to programs which may contain side-effects and which can be written in imperative languages, e.g. Pascal. Our method combines program transformations with results from data flow analysis to achieve this goal. Programs which contain side-effects are transformed or mapped to programs without side-effects. These transformations are guided by data flow analysis results. The conventional algorithmic debugging technique is used on the transformed or mapped program, but the debugging process is presented to the user in terms of the original program. Thus, our version of algorithmic debugging consists of a program transformation phase followed by a conventional algorithmic debugging phase.

During the actual bug localization phase of algorithmic debugging, the system traverses the execution tree of all activation records. Thus, a trace has to have been produced during the execution of the call which produced an incorrect result, i.e. a bug symptom. The system asks an oracle (usually the programmer) if certain calls are correct. In Shapiro’s original system the user could only answer yes or no, but the system remembers all answers. An improved version, [Drabent-88], also allows the user to answer with assertions about procedure input/output semantics. This reduces the number of questions asked by the system, and thus makes the algorithmic debugger more practical to use.

A prototype Pascal algorithmic debugger including transformations has been implemented within the DICE system. This prototype will be extended to support interactive addition of assertions to the algorithmic debuggers oracle database in order to limit the number of user interactions required. These assertions can be efficiently compiled using the DICE incremental compiler.

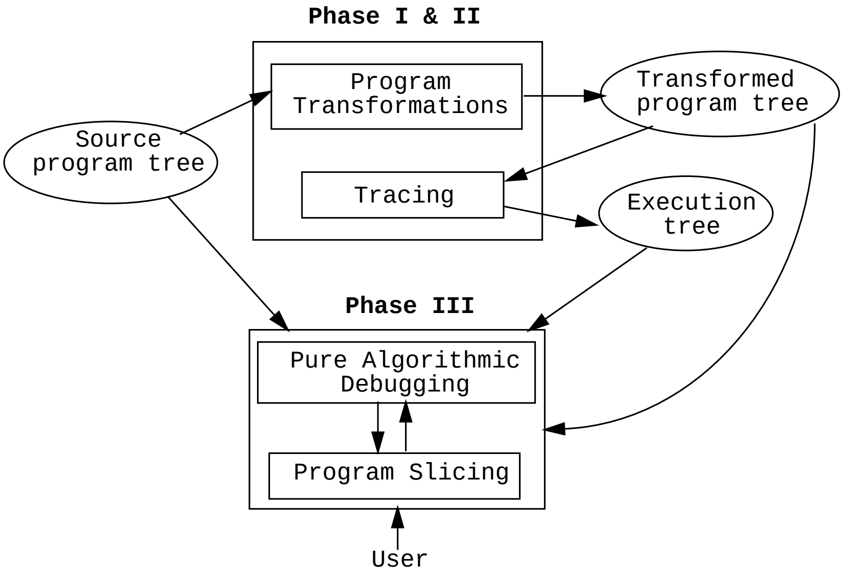


Figure 12-2. *The functional structure of the algorithmic debugging system. Arrows denote information transfer.*

Focusing bug localization through program slicing

In [Kamkar,Shahmehri,Fritzson-90], we show how program slicing, a data flow analysis technique, can be used to focus the search process during bug localization with algorithmic debugging. The slicing is applied at each interaction where the user or the algorithmic debugger can point out an erroneous data value. It eliminates the parts of the execution tree which are independent of the erroneous value, and thus irrelevant for the search.

Possible future work

This work seems very promising, and we would like to extend it further in several directions:

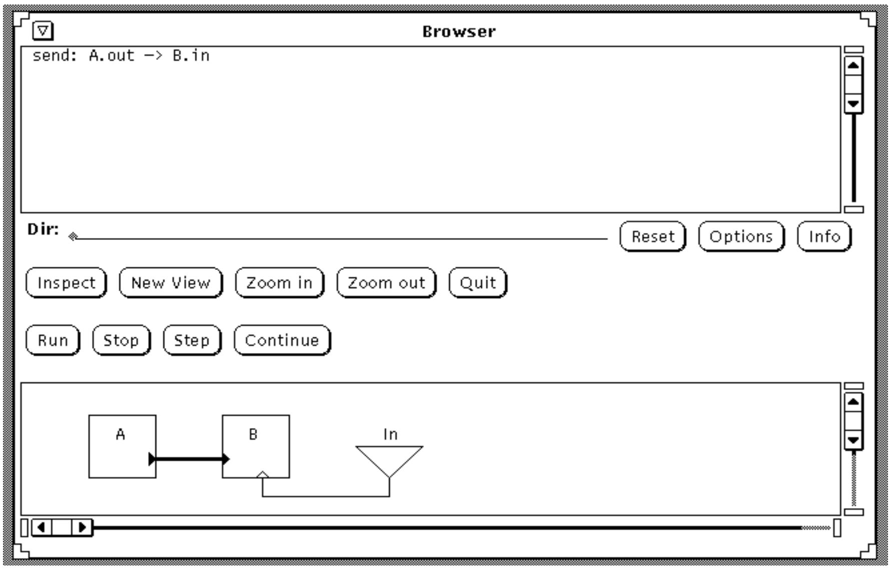
- Temporarily allow *don't know* answers during the bug localization process. This would allow more flexible debugging, but complicates the bug localization logic.
- Extend Algorithmic Debugging to Object-Oriented Languages.

- Investigate how the Algorithmic Debugging technique can be generalized to handle debugging multi-process parallel systems. Initially, we will use a remote procedure call model of inter-process communication, which is close to our current model of algorithmic debugging.
- Program transformations in order to handle Algorithmic Debugging in the presence of pointer related side-effects, which we currently cannot handle.

12.4.2 Debugging Tools for Distributed and Real-Time Systems

Lic project for Lars Strömberg. Supervisors: Johan Fagerström, Peter Fritzson.

The current activity within this project is described in the licentiate thesis by Lars Strömberg The topic is the design and implementation of a postmortem debugger for parallel programs executing on a system of loosely coupled processors. The debugging system consists of a trace collecting runtime component linked into the debugged program, and two window based tools, a *browser* and an *inspector*, for inspecting the traces. The main advantages of this work over some existing work in the area are: overview of the interprocess communication structure, a minimal amount of irrelevant information presented in the inspection tools, and possibilities of working at different levels of detail. The debugger also supports animation of traced events. The primary types of program errors that can be found using the debugger are errors in the communication between processes. The system is expected to be used together with a traditional sequential debugger for debugging the internal algorithms of the processes.

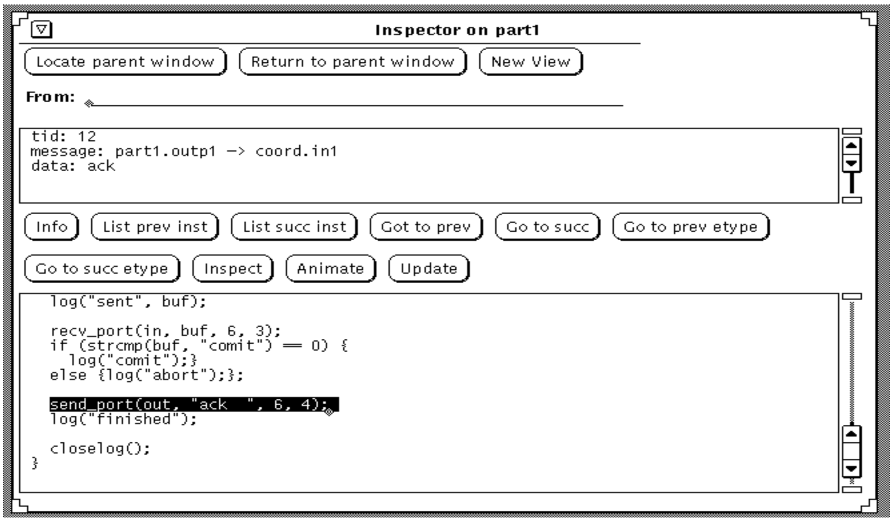


```
send: A.out -> B.in
```

Figure 12-3. *The browser during an animation.*

The debugger works by logging process related events onto trace files. This can be done with little extra overhead. There is one trace file for each process or module. The trace file can be inspected using two tools, called *Browser* and *Inspector.* The browser tool has a graphical picture of the internals of one module (submodules, ports, and connections between ports). It is possible to zoom into a submodule, replacing the current picture with a picture representing the submodule. There is also a corresponding function for zooming out. The graphical tool is intended to give an overview of the system. Since the configuration can change dynamically, the graphical view shown reflects the system at one specific moment in the trace file of a module. At start-up, the browser shows the configuration at the end of the execution.

Inspecting the code of a module is done in the inspector. It is started by selecting a module in the browser giving the command “Inspect”. The inspector runs in a window separate from the browser. The inspector permits the user to look closely into the history of a single module. The source code of the module is shown in the inspector, and the programmer can see all events that have taken place at a certain point in the code. The programmer can continue forward or backward from an event to the next (or previous). This makes it possible to follow the execution in order to find the cause of a failure. We have also included an animation facility which makes it easier to examine a sequence of events, especially if the sequence is repetitive. Animation can also be used as a fast means for positioning the tool in the execution.



```
tid: 12
message: part1.outp1 -> coord.in1
data: ack
```

```
  log("sent", buf);

  recv_port(in, buf, 6, 3);
  if (strcmp(buf, "comit") == 0) {
    log("comit");}
  else {log("abort");};

  send_port(out, "ack  ", 6, 4);
  log("finished");

  closelog();
}
```

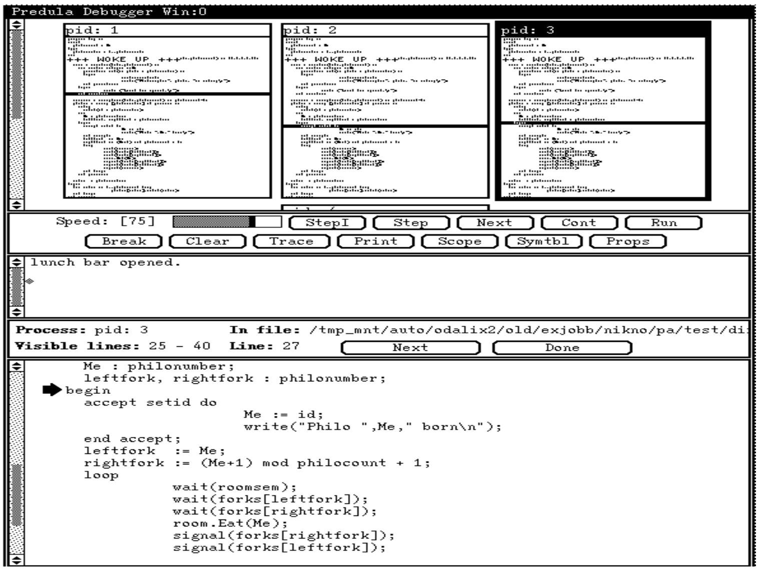
Figure 12-4. *Inspecting a participating process.*

An interesting future project would be to integrate the algorithmic debugging technique with this kind of distributed debugging tool.

12.4.3 The PREDULA Parallel Programming and Debugging Environment

During the past 1.5 years, a multi-paradigm parallel programming environment and language called PREDULA, has been developed within PELAB.

This work started as the master thesis project of Johan Ringström. The primary motivation is to make parallel programming primitives from different paradigms available within a single environment for teaching purposes. For example, primitives such as semaphores, monitors, channels, remote procedure calls and rendezvous are all available within the system.



```
lunch bar opened.
```

```
Process: pid: 3          In file: /tmp_mnt/auto/odalix2/old/exjobb/nikno/pa/test/di
Visible lines: 25 - 40   Line: 27
```

```
     Me : philonumber;
     leftfork, rightfork : philonumber;
 ➡ begin
     accept setid do
                         Me := id;
                         write("Philo ",Me," born\n");
     end accept;
     leftfork  := Me;
     rightfork := (Me+1) mod philocount + 1;
     loop
               wait(roomsem);
               wait(forks[leftfork]);
               wait(forks[rightfork]);
               room.Eat(Me);
               signal(forks[rightfork]);
               signal(forks[leftfork]);
```

Figure 12-5. *The PREDULA debugger in action*

The PREDULA language is compiled into an intermediate language called ANTS, which is then interpreted. A debugger is under development as a master thesis project. The system was used successfully by students in the process programming course during the autumn of 1989 and 1990.

12.4.4 Generation of Practical Compilers from Denotational Semantics

Lic project for Mikael Pettersson. Supervisor: Peter Fritzson.

Generating compilers from formal specifications of programming languages has for a long time been a research goal in the compiler-writing community. Several efforts to generate compilers from denotational semantics specifications (mostly by theoreticians) has resulted in compilers and code that run very slowly (usually a factor of 1000 slower than for commercial compilers), and do not interface to product-quality compiler generation tools such as YACC, or to commercial optimizing code generators.

Recent work by Peter Lee [Lee-89] has demonstrated that it is indeed possible to generate realistic compilers from denotational semantics definitions. However, his system, called MESS, is monolithic and does not interface well with standard parser generators and code generators. Also the code generator in his system need to be handwritten in Prolog.

In comparison with [Lee-89], we are going one step further. We are building the first realistic compiler generator that interfaces well with standard tools, and also automatically generates an intermediate code generator down to the quadruple level.

Current results

An initial small prototype was implemented during the autumn of 1989, which is reported in [Pettersson-89] and [Pettersson-90]. It was able to generate an interpreter and a compiler that produces quadruple code, for a language which is a tiny subset of C. The quality of code produced by the generated compiler is close to that of commercial compilers. In this experiment C++ was used as a meta-language, which provides type checking and efficient compilation. However, this approach has several limitations, as C++ does not support local procedures and closures, and does not have tail-recursion removal. Also language specification was a bit clumsy, with an overhead of C++ declarations.

The next step was the design and implementation of a full-scale compiler generator without the above mentioned limitations. Here we provide DML (we call it Denotational Meta Language – mostly a subset of ML), as a specification language for denotational semantics. This choice of syntax is similar to Peter Lee's approach, and avoids the cryptic syntax of Oxford school denotational semantics. However, our DML also has the enhancement of inline concrete syntax with pattern matching operators, which simplifies access to structures and gives shorter and more readable specifications.

The implementation of this tool has been done as follows: First a type checker for DML and a DML to Scheme translator are implemented in Scheme, since Scheme can handle local procedures and closures. Scheme is then compiled to C, using Bartlett's Scheme to C translator. To enhance run-time efficiency we have implemented tail-recursion removal for C on the MC68020 and Sparc processors. The final result is a DML to C translator, which can be compiled into C. Finally, DML will be bootstrapped in itself, by rewriting the semantics for DML in DML.

The first version of this tool, called SOCCER – Semantic Oriented Compiler Compiler, is being completed during the autumn of 1990. During the spring 1991 it will be bootstrapped in itself, and further developed and tested. Finally, the practicality of this compiler generation tool will be tested by specifying the semantics of a full-scale language, e.g. full C.

There seems to be substantial general interest in this kind of compiler generation tool. For example, when Peter Fritzson recently visited MIT, giving a talk on this project, Richard Stallman expressed great interest in the future availability of this tool.

12.4.5 Very High Level Parallel Programming and Compilation to Parallel Hardware

Lic project for Johan Ringström. Supervisors: Johan Fagerström, Peter Fritzson.

Programming of parallel computers is very complicated today. The programmer has to deal both with complexities caused by the parallel hardware in addition to normal implementation details. Also, most of the commonly used programming languages such as C or Pascal require implementation details that constrain compilers too much when generating efficient code for parallel hardware. A better way would be to use VHLL languages (Very High Level Languages), e.g. Refine, SETL or Paralation Lisp, where it is possible to let the compiler automatically supply low level implementation details during the translation process. The class of VHLL languages we are interested in can also be denoted *collection-oriented languages*, since much of their power comes from their ability to conveniently specify operations on collections of data objects rather than single objects.

This new project aims at developing a transformation-based programming environment for programming parallel hardware. The programmer specifies his algorithm in a very high level language which is subsequently translated into an efficient parallel implementation.

This is a rather ambitious long-range goal, and should be approached in steps of manageable difficulty. Initially we have chosen to focus on the *Paralation model* for parallel programming, described in Gary Sabot's PhD thesis. This model is more general than programming with arrays or sets, and is closely related to programming with relations. Another attractive feature is the possibility of achieving efficient compilation to parallel hardware by describing the communication structure through the Paralation *shape* facility.

In order to compare the Paralation model with another collection-oriented parallel programming model we implemented a simple application both in Paralation Lisp and on the Connection Machine at KTH. The resulting program was significantly shorter and more readable in Paralation Lisp, which strengthened our decision to use this model.

There are currently two important research issues in this project: The first is how the Paralation model can be integrated in other languages than Lisp. For example: functional languages such as Lazy ML, which have great inherent potential for parallelism. The ability of lazy languages to handle infinite data structures also offers some hope of generalizing the Paralation model to remove the fixed length limitation of Paralations.

The second research topic we will pursue is to make it possible to actually use the shape facility to guide a compiler for parallel hardware, and to implement such a prototype compiler. The *shape* concept was described by Guy Sabot, but has so far not been used in any compiler according to its inventor.

12.4.6 High-level Programming Environment Support for Equational Modeling in Scientific Computing

Lic project for Lars Viklund, Johan Herber. Supervisor: Peter Fritzson. In cooperation with Dag Fritzson, SKF Engineering Research Centre, Holland.

The goal of this project is the development of a high-level programming environment for scientific programming that support programming in equations instead of low-level procedural programming. The high-level equational representation also gives better chances to utilize the inherent parallelism of a problem to generating efficient code for parallel hardware. As an initial example application domain in scientific programming, we have chosen mechanical analysis in machine elements. We would also like to mention that this work has some connections with the work on qualitative reasoning on physical systems done by RKLLAB, and object oriented modeling of control systems [Andersson-90]

Application domain: modeling in mechanical analysis

At the current state of the art, there are two main classes of activities within research projects or advanced development projects in mechanical analysis:

- Theory development
- Numerical computation

During theory development, a mathematical model is developed that describes the relevant properties of the investigated machine element. This is normally done manually, using only pen and paper.

In order to perform numerical computations, the model must be translated to some programming language, or to input specifications for some existing numerical modeling program, such as for example a finite element analysis program. Normally, existing high level tools can at best be used for limited parts of the total computational problem, because it usually is too complex.

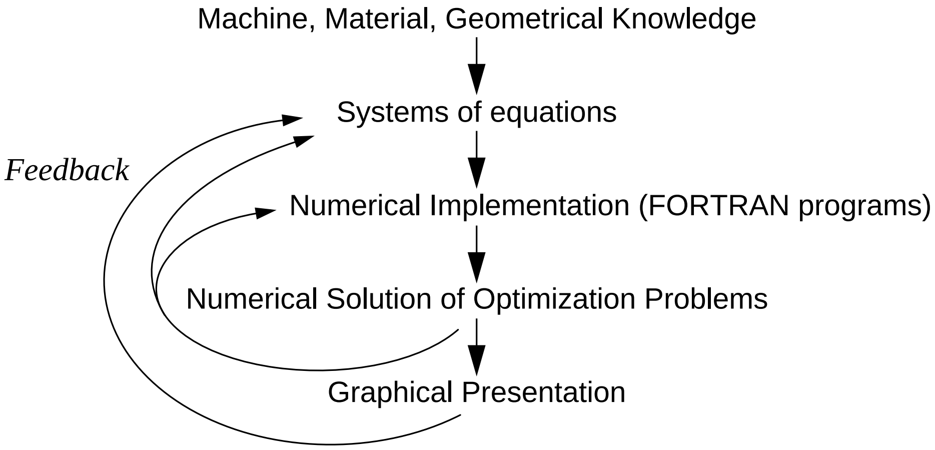

Machine, Material, Geometrical Knowledge
Systems of equations
Feedback
Numerical Implementation (FORTRAN programs)
Numerical Solution of Optimization Problems
Graphical Presentation

Figure 12-6. *The iterative process of modeling in traditional mechanical analysis. Often 50% of the time is spent on FORTRAN programming.*

The ideal programming environment: programming in high-level equations

The ideal high level programming environment would automatically transform systems of equations into efficient symbolic and numerical programs. It would select optimization routines with good convergence properties for the given problem. The environment would also aid in formulating equations given geometrical constraints and transforming equations between different coordinate systems. However, fully automatic versions of some of these capabilities will be hard to achieve. It is more realistic to assume that the user will work in dialogue with the interactive system, and that the user can supply valuable hints and information that will guide the system to choose the right algorithms and transformations. Some desired capabilities are listed below:

- Support in expressing systems of equations, e.g. handling geometrical constraints and coordinate transformations
- Integration of object-oriented techniques in equational modeling

- Algebraic transformations of equations
- Compilation/transformation of model equations to efficient numerical programs
- Transformation of equations for computation on parallel hardware
- Convergence and selection of optimization methods
- Evaluation of Numerical Experiments
- Graphical presentation

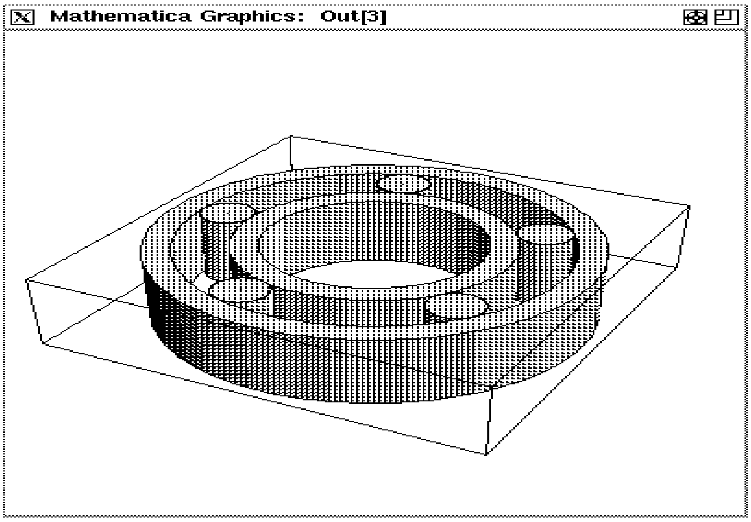


Figure 12-7. *A three-dimensional view of the roller bearing, automatically generated from the model example equations.*

Current status

As a test case we have chosen a bearing example model, containing about 200 equations. A first version of an object-oriented environment for scientific programming is currently prototyped in Mathematica, and applied to this example.

12.5 List of publications

The following are publications by PELAB members during the 18 month period 1989-90. For the full list of publications, please refer to Appendix E.

Licentiate Theses

[Larsson-89] **Yngve Larsson:** Dynamic Configuration in a Distributed Environment. 1989
[Strömberg-89] **Lars Strömberg**: Postmortem Debugging of Distributed Systems. 1990

External publications

[Fritzson-89] **P. Fritzson:** Incremental Symbol Processing. *Lecture Notes in Computer Science, LNCS-371, Springer-Verlag,* 1989.

[Fritzson,et.al-90] **P. Fritzson, T. Gyimothy, M. Kamkar, N. Shahmehri: Generalized Algorithmic Debugging and Testing**. In *Proc. of SIGPLAN'91 Conference on Programming Language Design and Implementation,* Toronto, June 24–28, 1991. Also as report LiTH-IDA-R-90-42, Dec. 1990.

[Moen-90] **S. Moen:** Drawing Dynamic Trees. *IEEE Software, vol. 7,* no 4, July 1990.

[Kamkar,Shahmehri,Fritzson-90] **M. Kamkar, N. Shahmehri, P. Fritzson:** Bug Localization by Algorithmic Debugging and Program Slicing. In *Proc. of the Int. Conf. on Programming Language Implementation and Logic Programming*, Aug 1990, *LNCS 456*, Springer-Verlag, 1990. Also presented at the *Fifth Nordic Workshop on Programming Environments*, Trondheim, June 1990.

[Kroha,Fritzson-90] **P. Kroha, P. Fritzson:** A Compiler with Scheduling for a Specialized Synchronous Multiprocessor System. In *Proc of the 3rd International Workshop on Compiler Compilers*, Schwerin, GDR, October 22–24 1990. Proc in the *LNCS* series by Springer-Verlag.

[Pettersson-90] **M. Pettersson**: Generating Efficient Code from Continuation Semantics. In *Proc of CC'90 – the 3rd International Workshop on Compiler Compilers*, Schwerin, GDR, October 22–26 1990. Proc in the *LNCS series by Springer-Verlag*. Also presented at *The fifth Nordic Workshop on Programming Environments*, Trondheim, June 1990.

[Shahmehri, Fritzson-90] **N. Shahmehri, P. Fritzson:** Algorithmic Debugging for Imperative Languages with Side-effects.In *Proc of CC'90 – The 3rd International Workshop on Compiler Compilers*, Schwerin, Germany, October 22–24, 1990. The proceedings are published by Springer Verlag in the *LNCS* series.

[Shahmehri,Kamkar,Fritzson-90] **N. Shahmehri, M. Kamkar, P. Fritzson:** Semi-Automatic Bug Localization in Software Maintenance, In *Proc of the IEEE Conf. on Software Maintenance,* San Diego, Nov. 1990.

Research reports, etc.

[Bilos-89] **R. Bilos:** A survey of Visibility Control Models. *LiTH-IDA-R-89-18*, 1989.

[Fritzson, Kroha-89] **P. Fritzson, P. Kroha:** An Object-Oriented Database Approach to the Symbol Processing in an Incremental Compiler, *LiTH-IDA-R-89-27.*

[Fritzson, Fritzson-91a] **P. Fritzson, D. Fritzson:** The Need for High-Level Programming Support in Scientific Computing – Applied to Mechanical Analysis. Forthcoming report, spring 1991.

[Joas-89] **K. Joas:** Interprocess Communication Primitives for a Distributed Debugger under Unix. *LiTH-IDA-R-89-48.*

[Kroha-89] **P. Kroha:** An Extension of the Single Instruction Machine Idea, *LiTH-IDA-R-89-25.*

[Pettersson-89] **M. Petterson:** Generation of Interpreters from Denotational Specifications using C++ as a Meta-language. *LiTH-R-89-52*

[Pettersson-90] **M. Petterson:** DML – A Meta-Language for Compiler Generation from Denotational Specifications, *LiTH-R-90-43, Dec. 1990*

[Ringström-90] **J. Ringström:** PREDULA –A Multi-Paradigm Parallel Programming Environment, *LiTH-IDA-R-90-41*, Dec. 1990. Presented at *The Fifth Nordic Workshop on Programming Environments*, Trondheim, June 1990.

[Santi-89] **Ö. Santi:** Retargeting of an Incremental Code Generator to MC68020, *LiTH-IDA-R-89-13*, 1989.

Other references

[Andersson-90] **M. Andersson:** Omola – An Object-Oriented Language for Model Representation. *Lic Thesis,* Dept. of Automatic Control, Lund Institute of Technology, 1990.

[Drabent-88] **Drabent et. al.:** Algorithmic Debugging with Assertions. *LiTH-IDA-R-88-04,* 1988.

[Fagerström-88] **J. Fagerström:** A Paradigm and System for Design and Test of Distributed Systems, *Ph.D. Thesis, Dept. of Computer and Information Science*, Linköping University, 1988.

Fritzson-83] **P. Fritzson:** Symbolic Debugging through Incremental Compilation in an Integrated Environment. *The Journal of Systems and Software* 3, 285–294 (1983)

[Lee-89] **P. Lee:** Realistic Compiler Generation, *MIT Press*, Cambridge, Massachusetts USA, 1989.

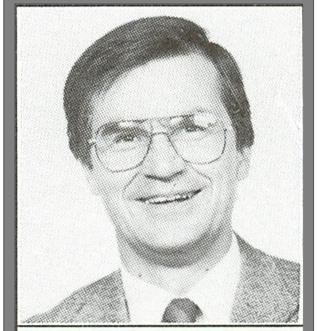
[Sabot-89] **G. Sabot:** The Paralation Model. MIT Press.

[Seviora-87] **R. Seviora:** Knowledge-Based Program Debugging Systems, *IEEE Software*, May 1987.

[Shapiro-83] **E.Y. Shapiro:** Algorithmic Debugging. *MIT Press*, 1983.

RKLLAB

13 The Laboratory for Representation of Knowledge in Logic



Erik Sandewall
director

Artificial intelligence
Real-time systems
Non-monotonic logic
Planning
Autonomous agents
Temporal reasoning
Qualitative physics

Back row: Magnus Morin, Lennart Staflin, Leif Finmo, Simin Nadjm-Tehrani, Per Österling, Christer Bäckström, Hua Shu.
Middle row: Douglas Busch, Jacek Malec, Erik Sandewall, Ingrid Sandblom, Lars Degerstedt.
Front row: Keith Downing, Reza Hamidi-Farahani, Ulf Söderman, Tommy Persson, Dimiter Driankov.
Missing: Patrick Doherty.

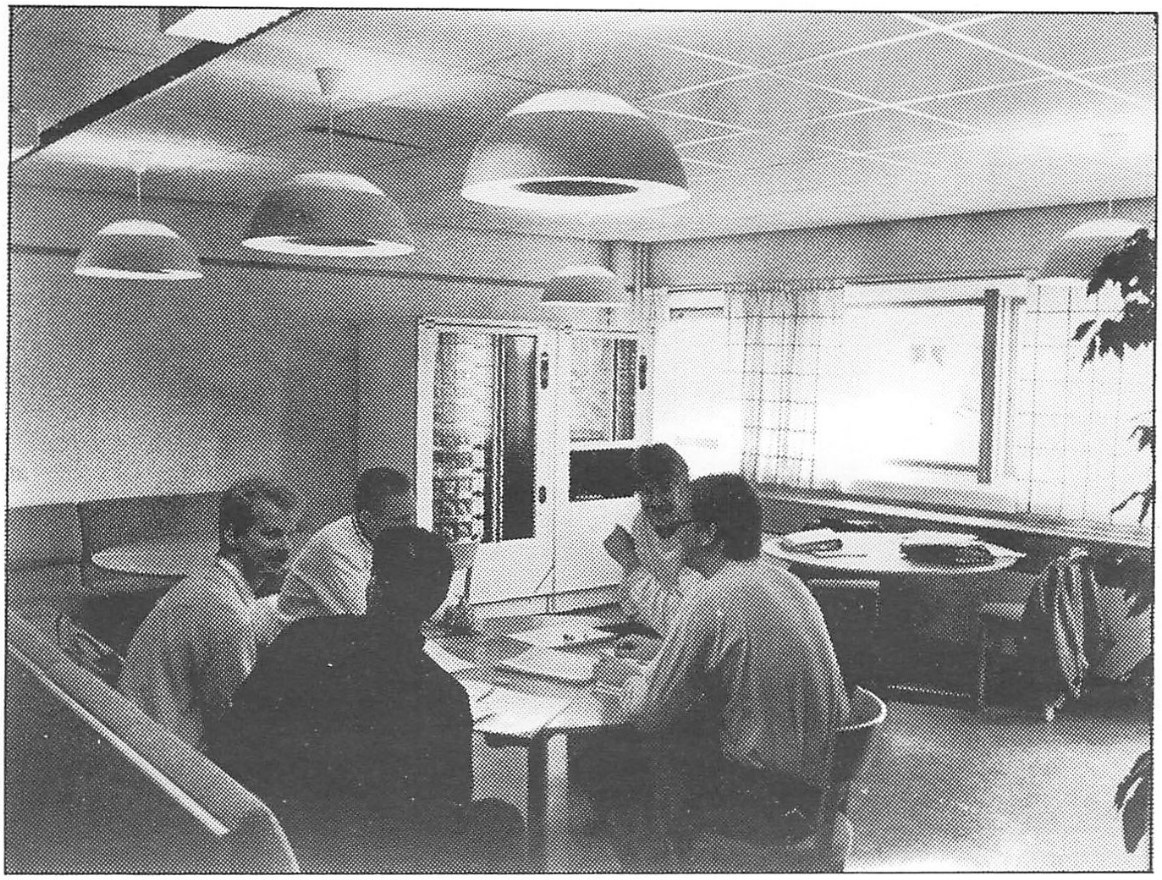


Figure 13-1. *Student coffee area.*

The area of interest for RKLLAB is theoretical aspects of knowledge based systems. The activity of "knowledge engineering", or the design of expert systems and other knowledge based systems, is often a rather ad hoc activity. Logic (and discrete mathematics) with suitable extensions, may be applied to strengthening the theoretical basis for knowledge engineering. It is the objective of RKLLAB to contribute in this respect.

13.1 Researchers and Projects

13.1.1 Activities

The activities of RKLLAB during the academic year 1989–1990 have been in the following, overlapping and interacting areas:

Non-standard logics and their implementations, in particular:

1. non-monotonic logic and reason maintenance
2. temporal logic, and associated algorithms for temporal reasoning and planning
3. logic of uncertainty

Plan-guided systems, i.e. autonomous vehicles and autonomous agents, including the following research topics:

1. software architectures for autonomous agents
2. description of physical change, in particular by combining logic and differential calculus
3. modelling of the behavior of vehicles in road traffic

Work on plan-guided systems has been done in the framework of several applications areas, in particular the pan-European Prometheus project where most of the (West) European automobile manufacturers cooperate in the development of information technology for future cars. Also some of the work has been done in cooperation with the FOA research institute in Linköping. The application projects have been funded by the IT4 research program and were organized through the Linköping A.I. Center (LAIC).

13.2 Laboratory members

The following researchers have participated in RKLLAB's activities during the academic year 1989–1990:

Laboratory leadership, administrative and technical staff:

Erik Sandewall, PhD, professor
Lillemor Wallgren, secretary (until autumn 1991)
Ingrid Sandblom, secretary (from autumn 1991)
Leif Finmo, research engineer

Laboratory members having or completing a Ph.D. degree during the period:

Douglas Busch, PhD, associate professor
Dimiter Driankov, PhD, assistant professor
Jacek Malec, PhD (joined RKLLAB in May 1990)

Visitors during extended periods:

Petr , PhD, Czechoslovak Academy of Sciences, Prague
Witold , PhD, Polish Academy of Sciences, Warsaw
Michael Reinfrank, PhD, Siemens AG, Munich

Laboratory members having, or completing a licentiate degree during the period:

Christer Bäckström
Patrick Doherty
Christer Hansson
Simin Nadjm-Tehrani
Hua Shu

Other graduate students:

Reza Farahani
Magnus Morin
Tommy Persson
Lennart Staflin
Ulf Söderman
Per Österling

The following degrees were awarded to RKLLAB members during 1989–1990:

Michael Reinfrank completed his Ph.D. The title of the thesis was: “Fundamentals and Logical Foundations of Truth Maintenance”.

Patrick Doherty completed his licentiate degree. The title of the thesis was: “A Three-Valued Approach to Non-Monotonic Reasoning”. He now continues the same work towards a PhD, which is expected to be completed during 1991.

Christer Hansson completed his licentiate degree. The title of the thesis was: “A Prototype System for Logical Reasoning about Time and Action”. Christer is now working with Asea Brown Boveri in Västerås.

Also, two recent RKLLAB members received degrees from other groups based on their previous work there:

Simin Nadjm-Tehrani completed her licentiate degree in Computer Science based on her research contributions in the Logic Programming Laboratory (see chapter 9 in this book). The title of the thesis was “Contributions to the Declarative Approach to the Debugging of Prolog Programs”.

Hua Shu completed her licentiate degree in Mechanical Engineering. The title of the thesis was “A Prototype Expert System for Pressure Vessel Design”. The concluding part of the work was done while she was employed in our group, and she now continues towards a Ph.D. in computer science.

A total of 15 papers by RKLLAB members were published, or accepted for publication, in international journals or conferences during the same period, as listed in of this volume.

13.3 The practical and the theoretical visions

In order to make progress you must define a point far ahead of you as the target that you are working towards. In RKLLAB we have chosen two such "visions" for the research, one theoretical and one applied.

The applied "vision" is the eventual development of *plan-guided systems*, i.e. a system such as an unmanned vehicle, or an "intelligent" automatic manufacturing cell, which is able to accept an assignment or request, make a plan for how to carry out the assignment, execute the plan, recognize problems which may impede the plan execution, revise the plan if necessary, and report success or failure.

It is clear that A.I. and knowledge engineering techniques are only one part of what is needed for designing plan guided systems. Automatic control, sensor technology including sensor data fusion, and often computer vision are also needed. Our point is however that the successful design of plan-guided systems requires a tight connection of results from these various fields. It is not sufficient to let the specialists in the various fields build each their part of the total system. In particular the perspectives of control theory/automatic control and of knowledge engineering must be combined in order to build intelligent plan-guided systems.

The theoretical "vision" is to eventually combine temporal logic and temporal reasoning, planning (in the A.I. sense of the word), qualitative reasoning, and the logic of uncertainty into a unified theory. In the contemporary A.I. literature these are seen as distinct sub-fields. We foresee that in the future they will have a common theoretical base, consisting of a *theory of change* that includes both "physical change" (change which occurs spontaneously and continuously) and change that is brought about by the actions of agents, operating in the world at hand. Qualitative reasoning hopefully will be based on such a theory of change, or conversely: one of the requirements on such a theory of change is that it must be sufficient as a basis for what we today call qualitative reasoning about physical systems. Similarly, knowledge-based planning should be explainable as a logical operation (abduction, presumably) that is well defined in the theory of change.

The first steps towards a theory of change are taken model-theoretically, by defining the logical language and its formal semantics. However from a computer science standpoint, the project has not been completed until we also have developed efficient algorithms and other computational methods for the various kinds of analyses which are needed in such a logic: temporal prediction and postdiction, planning and plan revision, plan recognition, identification of mechanisms and faults in them (i.e. diagnosis), etc.

It is also a reasonable guess, based on current research in this area, that non-monotonic reasoning and the logic of uncertainty should be used in the theory of change.

The theoretical "vision" of a comprehensive logical theory of change, combines naturally with the applied "vision" of plan-guided vehicles and other plan-guided systems.

We hope and believe that theoretical progress in this area will be useful in the work towards the applied goal. We feel likewise that experimental work on relatively simple plan-guided systems can provide useful insights and guidance also for the emerging theory, and can help formulating what are the most important questions that the theory must answer.

For these reasons we let the research in RKLLAB (including our participation in LAIC) be guided by the two visions jointly. This is not to say that all efforts *only* have that goal in mind: the research on logic of uncertainty, for example, has many other potential applications besides for use in the eventual logic of change and in the plan-guided systems. The same holds for non-monotonic logic and for reason maintenance systems. The ideas of plan guided systems and of a theory of change do not serve as a straight-jacket, but rather as inspiration, challenge, and a frame of reference for the lab.

In the following sections we summarize work on several research topics which have been motivated by these two visions or long range goals. Because of the format of the present volume, the exposition is by necessity brief. Please refer to the publications listed in Appendix E for more details.

13.4 Non-monotonic logic and reason maintenance

Erik Sandewall, Douglas Busch, Patrick Doherty, Petr , Witold , Michael Reinfrank.

Several approaches to non-monotonic logic have been proposed and studied in A.I., but the method of circumscription is now the dominant one. It has definite advantages over competing methods, but does have the disadvantage of being quite complicated for practical use.

We study a new approach to non-monotonic logic called NML3 which is characterized by the use of an explicit default operator. In our notation, the rule that if *a* is known and *b* is not contradicted, then *c* may be concluded as default, would be written as the wff

$$La \wedge Mb \Rightarrow Dc$$

to be read as “if *a* is known and *b* is possible, then default-*c* follows”.

The semantics of such formulas is defined using partial or three-valued interpretations, where proposition symbols (etc.) are assigned either of the truth-values “true”, “false”, or “unknown”. A *model frame* is defined as a pair $\langle \Delta, u \rangle$, where u is a set of partial interpretations, and $u \in \Delta$ and is information-minimal in Δ. Informally, the u component of the model frame is a partial description of the current world, and the other members of Δ are preferred, alternate descriptions of the world which one may switch to as new information is learned.

The semantics of wff:s also containing the operators *L*, *M* and *D* as in the example above, is now defined in terms of such model frames. Doherty and have done this work and shown that the resulting logic is a cumulative logic in the sense of Kraus, Lehmann and Magidor, and in particular that it has the properties of Reflexivity, Left Logical Equivalence, Right Weakening, Cut, and Cautious Monotonicity.

Other results regarding NML3 concern the development of a decision procedure based on analytic tableaux, as well as its use for the purpose of inheritance reasoning.

In the area of *reason maintenance,* Michael Reinfrank defended his Ph.D. thesis in January, 1990. Please refer to last year’s progress report for a summary of the thesis topic, or to the references in Appendix E for the complete material. Dr. Reinfrank is now working with Siemens AG in Munich. He continues his participation in RKL-LAB activities by means of recurrent visits.

13.5 Temporal logic, reasoning about actions, and planning

Erik Sandewall, Christer Bäckström, Christer Hansson

During the period reported here, we have pursued two approaches to reasoning about actions.

The **action structures** approach was initiated by Erik Sandewall and Ralph Rönnquist (now in the IIS laboratory) during earlier years, and has been further pursued by Christer Bäckström. The key idea is to characterize each type of actions by a number of partial states, i.e. mappings from atomic formulas to either of the truth- values “true”, “false”, or “unknown”. In the simplest case, each action type is characterized by three such states, namely the preconditions (must apply at the time where the action starts), the postconditions (must apply at the time when the action ends), and the prevail conditions (which must apply throughout the duration of the action).

During the last year, Bäckström has cooperated with Inger Klein in the Division of Automatic Control in the study of special cases of action-structure planning, where additional constraints are imposed on the structure of the actions, in such a way that the planning problem becomes computationally tractable. They defined a planning algorithm where, under the specified constraints, the worst time complexity is $O(\left|M\right|^3)$ where M is the set of properties by which the planning world is characterized.

The **explicit temporal logic** approach has been developed by Erik Sandewall and Christer Hansson. This approach is similar to the one used by Shoham, but with another preference relation on models than his. In our approach we can represent processes over time containing actions which occur sequentially or concurrently, and which have direct or indirect effects on properties of the world. Temporal prediction, temporal postdiction, explanation and planning can be obtained as special cases of non-monotonic deduction.

A computational method for the ETL approach has also been developed. The basic idea is to represent the known atomic facts as a partial interpretation, where time is partially ordered, and where each property may have the value true, false, or unknown during each time interval. A time interval is defined as a pair of time-points which are known to occur in temporal succession. The algorithm operates on a set of such partial interpretations, and its key idea is to gradually strengthen the information content in the partial interpretations until they satisfy all the axioms and are minimal with respect to the model preference orderings.

Christer Hansson has improved the method, implemented the algorithm, and analyzed its complexity properties. The work was defended as a licentiate thesis in February, 1990.

13.6 Logic of uncertainty

Dimiter Driankov

The following approach to the logic of uncertainty is pursued, for the simple case of propositional formulas. A *set-up* is defined as a mapping which to each atomic formula assigns a belief-disbelief pair. Each of the two components of such a pair is taken from a finite set of belief degrees (certain, extremely likely, good chance, etc.). An *epistemic state* is defined as a set of set-ups for a given set of proposition symbols.

Informally one may think of an epistemic state as analogous to a formula in disjunctive normal form, and a set-up is then one of the conjunctions from which the dnf is formed. The conversion from a conjunction of atomic formulas, to a corresponding set-up is obtained by mapping each atomic formula to the appropriate belief-disbelief pair, depending on whether the formula itself, its negation, or neither is present in the conjunction. The generalization is obtained by going from those three "truth-values" to the full range of belief-disbelief pairs.

In the present logic the epistemic states are however not a kind of formulas, but are used as models for formulas. A number of interesting issues arise. In particular, evaluation of formulas is not compositional: the value of a composite expression is not a function of the values of its components, but must be determined in a more complex fashion.

Dimiter Driankov has developed this logic in a number of publications (see Appendix E) and in this Ph.D. thesis.

The use of partial interpretations (set-ups) and sets of partial interpretations (epistemic states) recurs in several other of our research topics, including the work on non-monotonic logic and in temporal reasoning described above.

13.7 Reasoning about physical change

Erik Sandewall, Tommy Persson, Lennart Staflin, Ulf Söderman

A complete logical theory of change must deal with both "physical" (spontaneous, continuous) change, and change brought about by the actions of agents. The modelling of physical change is studied in qualitative reasoning, but logicist methods are not currently being used there.

We have taken some first steps towards a logical theory of physical change, by showing how differential calculus can be embedded in a temporal logic. The resulting hybrid system can characterize piecewise continuous physical systems, where the discontinuities are induced by features of the physical environment: particles bouncing (discontinuous velocity), particles going over an edge and starting to fall (discontinuous acceleration), etc. Differential calculus (D.C.) is used as usual for characterizing the parameters in their continuous regions. Logic is used for two purposes: for specifying the regions where each D.C. description is valid, and for specifying the behavior of the parameters around their discontinuities.

A second paper extends the approach to also include actions (performed by an agent) into the logical system.

Our first axiomatizations of application scenarios had to contain some axioms which were not intuitively required, but which were needed for technical reasons in order to constrain the set of models sufficiently. More recent work by Tommy Persson and Lennart Staflin has shown how the approach can be modified in order to obtain more natural axiomatizations of the applications.

One important part of our activity on "reasoning about physical change" is also a discussion seminar together with some members of the Division of Automatic Control in our university, including professor Lennart Ljung.

13.8 Modelling traffic behavior

Erik Sandewall, Jacek Malec, Simin Nadjm-Tehrani, Hua Shu, Per Österling.

The major application area in our research is within the Prometheus project, a pre-competitive research cooperation project by the West European automobile manufacturers. The application goal there is really driver support, rather than autonomous vehicles in the strict sense, but the technical problems are very similar at least from our point of view.

One of the important issues for Prometheus is the modelling of what happens in traffic: how our own car (the car in which the driver support system is mounted) behaves, and how neighboring cars can be understood to behave. The model of traffic behavior is needed as a basis for plan recognition, identification of dangerous situations, planning (planning the route, or planning how to handle a danger), etcetera.

Several approaches to traffic behavior modelling are being tested. They include logic-based approaches along the lines described above, a state-space based approach, and an approach which views each car control/driver support system basically as a reactive agent.

13.9 Software architecture for autonomous agents

Erik Sandewall, Magnus Morin, Reza Farahani.

In designing the complete autonomous vehicle, or other plan-guided real-world systems, the choice of an adequate software architecture is a crucial issue. It is very important from a conventional software engineering perspective; it is also very important if we hope to combine contributions from A.I., automatic control, sensor technology and others into a coherent system.

Although we feel that the logic-based methods described above could be very useful for the design of a plan-guided system, the software work does not *only* have the goal to implement the given theoretical methods. The emphasis is instead on identifying a software architecture which will satisfy the needs of all the participating technologies, including the logic-based A.I. contributions.

The group has developed a multi-layer software architecture as an answer to these questions. The lowest layer is essentially an “engine” which updates a state vector at fixed-time intervals, using transition rules that can be redefined dynamically. Sensor data are inserted into the vector; actuators take their data from the state vector. Besides the process layer “engine”, there are various tools which make it possible for higher, logic-oriented layers of the architecture to influence the operation, and redefine the structure of the lowest layer.

The architecture is being implemented in a target system based on a Motorola 68000 processor, using the PSOS$^+$ operating system and the C programming language for its lowest layers. A Sun workstation, connected to the target system by a special protocol, is used as the development vehicle. The first version of the lowest two layers were completed in mid-1989.

13.10 International activities

RKLLAB’s activities have a very international flavour. More than half of the participants in this year’s activity are foreign nationals, guest researchers included. Several members from abroad joined RKLLAB during the year, both as permanent members and as guest researchers for extended periods, as reported above.

Also RKLLAB participates in the Eureka project Prometheus. In particular, Erik Sandewall is European co-coordinator for ProArt, the A.I. branch of Prometheus.

13.11 Publications

RKLLAB members were authors of the following external publications (papers published in books, journals, or international conference proceedings) during 1989–90, using the numbers in Appendix E.

14
CENIIT – The Center for Industrial Information Technology

Since1988, our University's School of Engineering has funded a special research programme on *Industrial information technology*. This choice was based on the observation that information technology is both the underlying technology for the information industry (computers, software, telecommunications, electronic components), and also is one very important enabling technology for other industries. The term 'industrial information technology' that was adopted by the School of Engineering, refers to that second aspect of the use of information technology.

The actual work in the new *Center for Industrial Information Technology, CENIIT*, started in mid-1988. Government funding is currently in the order of 7.5 MSEK per year. The departments for computer science, electrical engineering, physics and mechanical engineering are presently the main participators, but the programme may eventually involve most of the departments of our school.

CENIIT research is currently concentrated in the following areas:

- *Autonomous systems*, defined as technical systems which exhibit independent, adaptive, plan-guided and learning behaviour in different environments. This research area has industrial relevance for instance in automated manufacturing processes and for autonomous vehicles.
- *The usage of information technology for product development and production planning*, for instance in connection with computer-aided design and process planning.

In addition, activities in the following research areas are planned:

- Process and product control for environment and energy optimization.
- Industrial documentation technology, including also issues of hypermedia technology, 'intelligent handbooks', natural language processing, machine translation etc.

The inter-disciplinary research programme in CENIIT is coordinated and planned by a steering committee, with representatives also from industry. During 1988–89 a primary area for new research efforts. The current committee consists of:

Erik Sandewall, Department of computer and information science (chair)
Tore Gullstrand, Saab-Scania AB
Lennart Ljung, Department of electrical engineering
Benny Odenteg, BT Systems AB

The work in CENIIT projects is supported by the School of Engineering at Linköping University.

Hasse Odenö, Department of mechanical engineering
Lars G. Rosengren, Volvo AB
Christer Svensson, Department of Physics

The following projects (most of which are further described elsewhere in this report) are currently undertaken as our department's contribution to CENIIT:

14.1 Computer support for automation

This project started during the spring of 1989 when Dr. Anders Törne was recruited from ABB Corporate Research in Västerås. The project now forms the core of the CAELAB (*Laboratory for Computer Assistance in Engineering*). The research is oriented towards the study of architectures and representations needed in systems for computer support in the engineering process. In particular, the principles for software engineering in this application domain and methods for management of the related information flows. The research takes an interdisciplinary approach and emphasizes cooperation with the mechanical engineering department.

A more detailed description of the research area and project goals is given in chapter 5 of this report.

Area leader: *Anders Törne*

14.2 Engineering databases

This area is concerned with aspects of database research, which are relevant for database support in design, development and maintenance of (large) technical systems. In particular database support for mechanical engineering, knowledge engineering and software engineering. Central research issues concern effective methods for management of complex structures in the database, including object-oriented approaches, conceptual modelling of database contents, and integration with special types of representations, such as graphics, text and knowledge bases. The relationships to database technology for administrative applications and information retrieval systems are also of interest.

Initially research has been focused on object-oriented approaches with a special concern for distributed solutions. Connections to expert systems and knowledge engineering is also strong. Within our department close connections to ASLAB and IISLAB are established. Current activities and results are described in greater detail in chapter 5 of this report.

Area coordinator: *Sture Hägglund*

14.3 Industrial Software Technology

The objective in the project Industrial Software Technology (Industriell programvaruteknik, IST) is to study the programming-in-the-many and the programming-in-the-large aspects of software development and maintenance. Since the large scale effects can't easily be studied in an academic environment, the approach is to start with observations and case studies in the software industry. The focus is on the development of embedded systems.

This project has a basic funding from CENIIT, with the funding period starting in July 1989. The activity is conducted by Bengt Lennartsson, and one graduate student, Ulf Cederling, has a thesis project in industrial software technology. Ulf Cederling's research is also supported by the University of Växjö.

From an earlier study, funded by The Trade Association of the Swedish Mechanical and Electrical Engineering Industry, 'Mekanförbundet', we have got an overview of activities in Swedish software industry. Case studies in this area are unusual in Sweden.

Ulf Cederling's thesis project has the following components:

1. Study and describe the software development process sufficiently sharply to understand the functional requirements on a software development environment supporting software quality and reuse.
2. Identify the information needed for software reuse.
3. Study and analyse the influence from different development methods on software reuse and on software quality in general.
4. Develop a conceptual frame-work for the development of reusable software.
5. Develop methods for the identification of information needed for the development and maintenance of software.

The initial study of the software development process in some selected organizations: Bofors Electronics AB, Saab Scania AB, SATech Electronics AB, Telub AB, and Kockums Automation AB, is almost completed.

Bofors Electronics AB has recently been awarded a contract on a new national combat control system, StriC. The development of StriC, where reuse of existing software is a key issue, is planned to be the major information source for Ulf Cederling's thesis work.

In May 1990 a two day seminar, SOFT-6, given by Robert Glass, on 'Building Quality Software' was organized. There were 45 attendants from Swedish software industry.

Project leader: *Bengt Lennartsson*

14.4 Geometrical algorithms

The purpose of this activity is to develop competence and perform basic research in the area of *efficient algorithms for three-dimensional geometric problems and motion planning*. Sequential as well as parallel algorithms are considered.

A more detailed description of this work is given in chapter 2 of this report.

Area leader: *Per-Olof Fjällström*

14.5 Conceptual representations for automatic text generation and translation

The purpose of this project is (i) to develop a representation of the content and structure of text which is language-independent and possible to integrate with the logical representation of the objects and relations in the domain from which the topics of the text are taken, and (ii) to develop parsers, generators and translation systems that employ such representations.

While the long-term theoretical goal of this kind of research is to develop representational schemas that could be applied to arbitrary text, it is not possible to achieve this within the present state-of-the-art. The project works with restricted forms of texts taken from manuals in Swedish and English. In-depth studies of restricted text types contribute to the general goal while also having potential applications in improved techniques for the production of documentation of various kinds.

At present we are working on an algorithm for multi-lingual generation (realization) of text from specifications of their content and function. These specifications have the form of directed graphs and the generation is performed through guided unification of language-specific grammatical and lexical information. Crucial in this process are rules that relate conceptual structures of various illocutionary and propositional types to types of linguistic constructions. One goal of the work is to develop rules which are as general and powerful as possible. Finally, the text is derived from the resulting language-specific information structures.

Further work includes the implementation of an analysis algorithm with the conceptual representations as target structures for use in a translation system and for high-level proof-reading of a text of the given type. That is, apart from the spelling-checking that is found with ordinary word processors, the system would check grammar, terminology and logical content of the text.

Area leader: *Lars Ahrenberg*

15
Educational Programmes

Undergraduate education in Sweden is organized in study programmes. Each is made up of a set of courses from different departments and subject areas. Figure 15-1 below shows the levels of degrees in the Swedish university system.

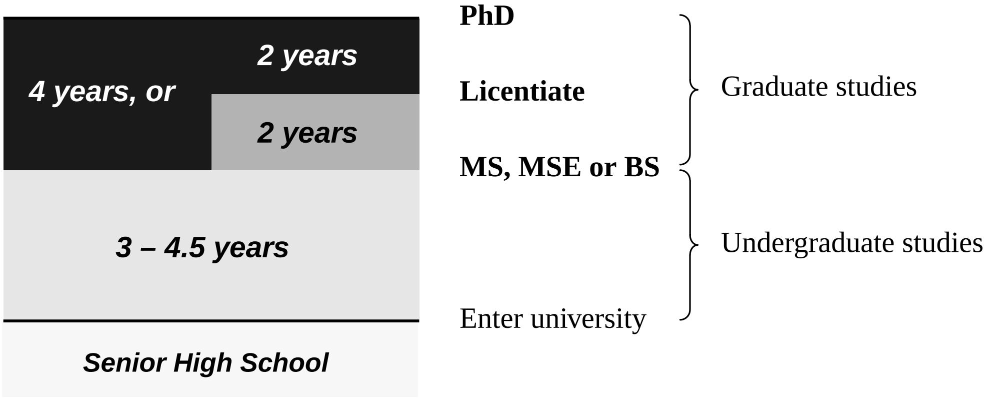


Figure 15-1. *Levels of degrees in Sweden.*

Each undergraduate student is assigned to a *study programme:* e.g. mechanical engineering, computer engineering, computer science, international economy, or system analysis, taking from three years to four and a half. All the students in such a study line take the same courses (with minor exceptions) during the first two years, and have a free(-er) choice from the third year onwards.The role of the department in these study programmes is to sell and deliver courses to the study programme committees. The courses given by our department during 1989/90 for the undergraduate education, up to and including the master’s degree level, are listed in Appendix C.

The graduate study programme provides the studies *from* the level of master of engineering, *to* the licentiate and/or PhD degrees. The graduate students are accepted and cared for by the department. A summary of recent courses and seminars in the graduate study programme is given in Appendix B, together with a presentation of faculty engaged in research and graduate education.

15.1 Graduate education

Graduate studies in the department of Computer and Information Science are organized as a program consisting of courses and project participation. The course program is organized at the department level and consists of *regular courses*, each of which is given approximately every second or third year (if possible), and *occasional courses* which depend on the profile and interests of current faculty and visiting scientists. Thesis projects are always done within or in association with the laboratories or research groups.

Courses and seminars are normally given in English (unless all participants are fluent in Swedish). Licentiate and PhD theses are (with few exceptions) written and defended in English. Thus English is the working language of the graduate study programme.

The programme leads to one of the following degrees:

Licentiate of technology or philosophy. The requirements include 50 points (one point equivalent to one week full time studies) for completed courses and 30 points thesis work. For a licentiate of technology, a master of science/engineering ('civilingenjör', 4.5 years of study) is normally assumed as a prerequisite.

Doctor of technology or philosophy. The requirements are 80 points for courses and 80 points for thesis work. Most of the PhD students take the licentiate degree as an integral part of their doctoral studies.

Although formally not part of the graduate study programme, a *Master of Science* degree is also offered within the department. For this degree, 40 course points are required, where a selection of courses from the undergraduate study programme is also eligible. The thesis work corresponds to 20 points.

15.1.1 Fields of study and degree subjects

The programme is designed – whether it is a licentiate degree or a PhD – with one of two main emphases:

Computer science field including the degree subjects computer science and computer systems, comprising areas of study such as computer architecture, VLSI, programming languages, artificial intelligence and data bases. Studies in this field has an engineering emphasis and a technology orientation. Students are assumed to have a strong background in mathematics and mathematical logic.

Information science field including the degree subjects information science and computational linguistics, comprising areas of study such as system analysis and system analysis tools, specification of demands on and effects of computer technology, cognitive science, natural language processing, and information resource management.

The division between these fields is, however, not clear-cut, and many courses may be followed in both fields. Each field is profiled by a number of courses which are specific and compulsory. Each student chooses courses after consultation with his or her supervisor.

The following degree subjects are presently offered in the department:

Computer Science
(*Prof. Erik Sandewall, Prof. Sture Hägglund, Prof. Jan Maluszynski.)*

Computer Systems (*Acting Prof. Krzysztof Kuchcinski)*

Computational Linguistics (*Associate Prof. Lars Ahrenberg)*

Economic Information Systems (*Prof. Birger Rapp)*

Library and Information Science (*Prof. Erik Sandewall)*

15.1.2 Admission to graduate studies

For the computer science field, a degree is required from one of the following areas:

- M Sc (or corresponding course of study at another institute of technology)
- system analysis, majoring in software
- mathematics with emphasis on computer science

for the information science field, a degree is required from one of the following areas:

- system analysis
- a similar approved qualification, such as a degree in economics or librarianship, supplemented by at least 40 points in computer studies.

A person with a different academic background may be admitted after supplementary studies and/or may be granted an exemption.

Applications for admission to graduate studies are invited twice a year. A decision will be taken by the department's committee for graduate studies. Each person who is admitted as a graduate student is assigned a supervisor, whose duty it is to give instructions and advice concerning the further organization of studies. Normally an advisory committee, including two assisting advisors is formed during the first year of graduate studies. If a student later chooses a field of study or speciality different from the one represented by the supervisor, he or she may change to another supervisor.

The task for the advisory committee is thus to

1. follow the progress of the graduate student and at least once a year assemble for a meeting where the student present his plans and results. The presentation should be followed by a discussion, where the advisors comment upon the thesis project. The student is responsible for scheduling the meeting and inviting the committee;
2. support the main advisor and contribute to an equal treatment and judgement of graduate students in the department.Advisors in the committee may or may not engage in a more frequent and active guidance of the student, depending on individual considerations.

The main supervisor for a graduate student has the full and formal responsibility for approving the thesis subject, courses and other issues in connection with the studies. The advisory committees are thus consultative.

15.1.3 Teaching faculty

Teaching staff, consisting of those teachers at the department who are course leaders, supervisors and project leaders for graduate students, are responsible for the curriculum of the graduate studies programme. They have, of course, completed graduate education themselves, and the majority of them devote most of their time and attention to research of their own, graduate education and research supervision.

The teaching staff are also complemented by other external teachers, who are employed by another department or equivalent, and who also lead courses or projects, or who participate in the planning of the graduate studies programme in computer science.

Teachers responsible for areas of study

Area	*Teacher responsible*
Administrative Data Processing and System Analysis	*Göran Goldkuhl*
Algorithm Analysis and Complexity Theory	*Per-Olof Fjällström*
Artificial Intelligence	*Erik Sandewall*
Databases	*Sture Hägglund*
Decision Theory and Support Systems	*Dimiter Driankov*
Natural Language Processing	*Lars Ahrenberg*
The Mathematical Foundations of Computer Science	*Jan Maluszynski, Douglas Busch*
Computer Architechture/VLSI	*Krzysztof Kuchcinski*
Expert Systems	*Sture Hägglund*
Research Methodology	*Erik Sandewall*
Information Retrieval	*Roland Hjerppe*
Programming Environments	*Bengt Lennartsson, Peter Fritzson*
Compiler Technology	*Peter Fritzson*
Logic	*Erik Sandewall, Douglas Busch*
Logic Programming	*Jan Maluszynski*
Distributed Systems	*Johan Fagerström*
Programming Languages	*Anders Haraldsson*
Software Engineering	*Bengt Lennartsson*
Real time Operating Systems	*Krzysztof Kuchcinski*
Human-Computer Interaction	*Lars Ahrenberg, Sture Hägglund*

Recent courses and seminars are listed in Appendix B.

15.2 (Re)organization of the undergraduate education at IDA

Undergraduate education accounts for roughly 35% of the total budget of the department, with an additional 5% for continuing education and technology transfer activities. (Close to 19 MSEK together.)

During the spring of 1990 we started to reorganize the undergraduate teaching activities within the department, and the new structure is effective from July on.

15.2.1 Previous organization of the undergraduate education

Under the department board, there was a Undergraduate Teaching Committee (IDUN – IDA's undervisningsnämnd), headed by Anders Haraldsson, responsible for the contents of courses given by the department and the assignment of teachers to courses. There are representatives from the student unions in this committee.

The department is responsible for the subject areas computer science (datalogi) and computer systems (datorsystem) in the School of Engineering, and administrative data processing (administrativ databehandling) in the School of Arts and Sciences. The external structure of the university has also been reflected in the internal organization. Some teachers were tied to the School of Engineering and others to the School of Arts and Sciences.

There has been a responsible director of undergraduate studies (studierektor) for each subject area. The computer science area was further divided into subareas. The directors were:

Anders Haraldsson, computer science (School of Engineering)

Lise-Lotte Raunio, administrative data processing (School of Arts and Sciences)

Mikael Patel, computer systems, systems programming
Arne Jönsson, artificial intelligence
Mats Wirén, natural language processing

Appendix C lists the courses given during the academic year 1989/90 and teaching staff for undergraduate education.

15.2.2 The new model

As Anders Haraldsson left his positions as director of studies and as chairman in the Undergraduate Teaching Committee for the department chairmanship, the organization of the undergraduate education in general was revised. We still have the Undergraduate Teaching Committee, but under this committee there are five executive directors of studies, and there is one secretary, Britt-Marie Ahlenbäck, for undergraduate education. The 'subject areas' for the five executive directors of studies have been composed from the course content point of view. The distinction between the university's School of Engineering on the one hand, and it's School of Humanities and Sciences and on the other hand, is now invisible inside the department. The new organization is presented in Figure 15-2.

Most of the teachers and lecturers have an association both to a research lab and to one of the five areas for undergraduate education. The subject area Economic Information Systems has been transferred to IDA during the autumn of 1990. This group, involved in both research and undergraduate education, has been moved almost unchanged into the IDA framework.

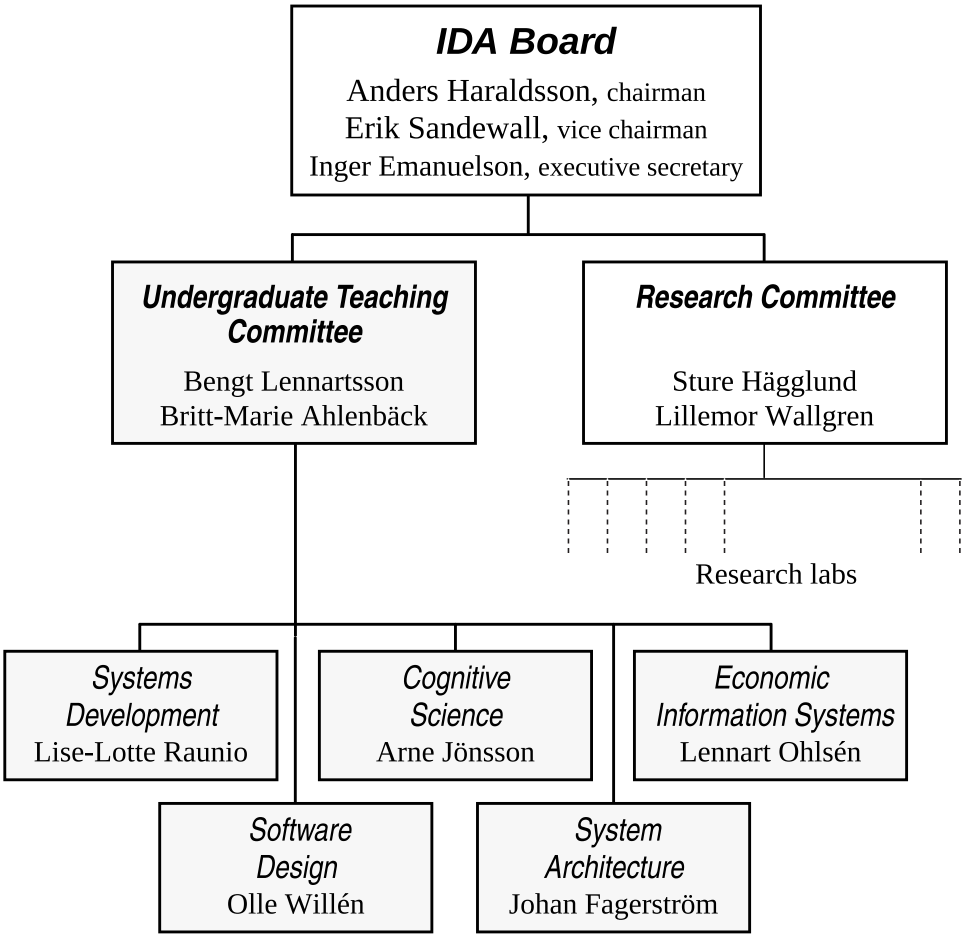


Figure 15-2. *The strategic and executive responsibilities for undergraduate education at IDA 1990/91.*

15.3 Undergraduate curricula

Since 1975 Linköping University has had a strong position in undergraduate curricula and teaching in computer science. Linköping is today the only Swedish university offering the three main 3–4.5 years undergraduate study programs in the area of computer science and systems analysis. The volume of other educational activities, such as a continuing education programme in computer science for industry, is increasing.

As the first institute of technology in Sweden we started the D study line (Computer Engineering – ’Datatekniklinjen’) in 1975 as a four-year (now converted to 4.5 years) programme leading to a Master of Engineering. It was the first full and specialized programme in computer science, specialized on software and hardware. The programme was introduced in 1982 at all other Swedish institutes of technology. Linköping University was also the first to introduce the I-line (Industrial and Management Engineering – ’Industriell ekonomilinjen’) in 1969. This programme has also been copied by other institutes of technology, more than a decade and a half after it was introduced in Linköping, however.

About 25% of the courses in the D-line are given by IDA. The expansion of staff and graduate students at IDA during the period 1980–1985 is to a large extent a result from recruiting students graduated from the D-line. The number of students accepted annually to the line has grown from 30 students the first year to 120 students.

A new computer science programme, the C-line (Computer Science – ’Datavetenskapliga linjen’) was started in 1982. It is a four-year programme leading to a Master of Science degree. The number of students accepted annually is 30. The programme is also given at Uppsala and Umeå Universities. At Linköping this programme is in the School of Engineering, but differs from ordinary engineering curricula (such as electrical engineering, or mechanical engineering) in some significant ways:

- significantly more discrete mathematics and logics, partly gained by reduction of the calculus courses
- LISP as the first programming language
- relevant humanities, such as psychology and linguistics, are significant parts of the curriculum, and are introduced as basic courses during the first years
- courses in theoretical branches of computer science
- courses in AI and AI-oriented subjects
- less emphasis on physics and physics-related subjects

The major part of the C-line was developed by persons from IDA and most courses are given by us. It is quite clear that these students develop a different ’culture’, and in particular a more solid basis for graduate research in computer science, than students in our other lines do. While certainly our other lines will continue to be of very high importance, the computer science line has provided a significant addition.

The D-line has also passed through changes. The new computer science base with discrete mathematics, logics and programming beginning with LISP instead of Pascal is introduced there. An advantage is that the students from the C- and D-lines get the same basis and we expect a large number of students from the D-line to be better prepared to specialize their studies in both more theoretical computer science areas and in artificial intelligence. In the D-line there are possibilities for specialization during the last years. In particular a new set of courses prepare students for software engineering assignments, i.e. the process of developing and maintaining large software systems in an industrial environment.

The set of courses that are available in the other programmes has been extended, and many of the courses have been improved. Technically, this has often been done by making new courses from the computer science curriculum available to other lines as well.

Industrial and Management Engineering, the I-line, was introduced in 1969 on an initiative from industry and business. It has steadily grown and adapted to meet new demands ever since. The transfer to IDA of the subject area Economic Information Systems including a full professorship, Birger Rapp, and a dozen researchers, teachers, and graduate students, will strengthen the links between IDA and the I-line. Today the students can specialize in either electrical or mechanical engineering. A new branch towards economic control, the controller function, and information management, would be welcome. The subject area Economic Information Systems is a natural bridge between computer science and its applications in business, industry, and organizations.

In the mechanical engineering programme there is a specialization that combines mechanical and computer engineering. We believe that especially research in artificial intelligence will be significant within that specialization.

The SVL programme, (Systems analysis – ’Systemvetenskapliga linjen’), has recently been modernized and ranges over three years of full-time studies. It aims at professional activities of design, evaluation and implementation of computer-based information systems. ADP-systems analysis dominates the program but nevertheless great importance has been attached to other subjects in order to give the program the necessary breadth and also to ensure that the students will become aware of the complexity of the community where computers can be used.

The first two years of the SVL programme constitute a common core of basic studies for all students. Within the subject of ADP-systems analysis there are courses in systems development and systems theory as well as courses in programming and computer science. The courses about systems development and systems theory deal with formal methods and prototyping. For the programming courses Pascal has been chosen as the main language but, other languages are taught as well. Within the field of computer science the students take courses in database design, development of interactive systems, communication, evaluation of computer systems, programming methodology, etc. Other subjects given within the common core of basic studies within the SVL programme are:

- business economics and management, to get basic knowledge about the organization of corporations and public services and their ’commonday’ routines.
- human factors, industrial and social psychology, including ergonomics, work environment, co-determination and participative management, group dynamics etc.

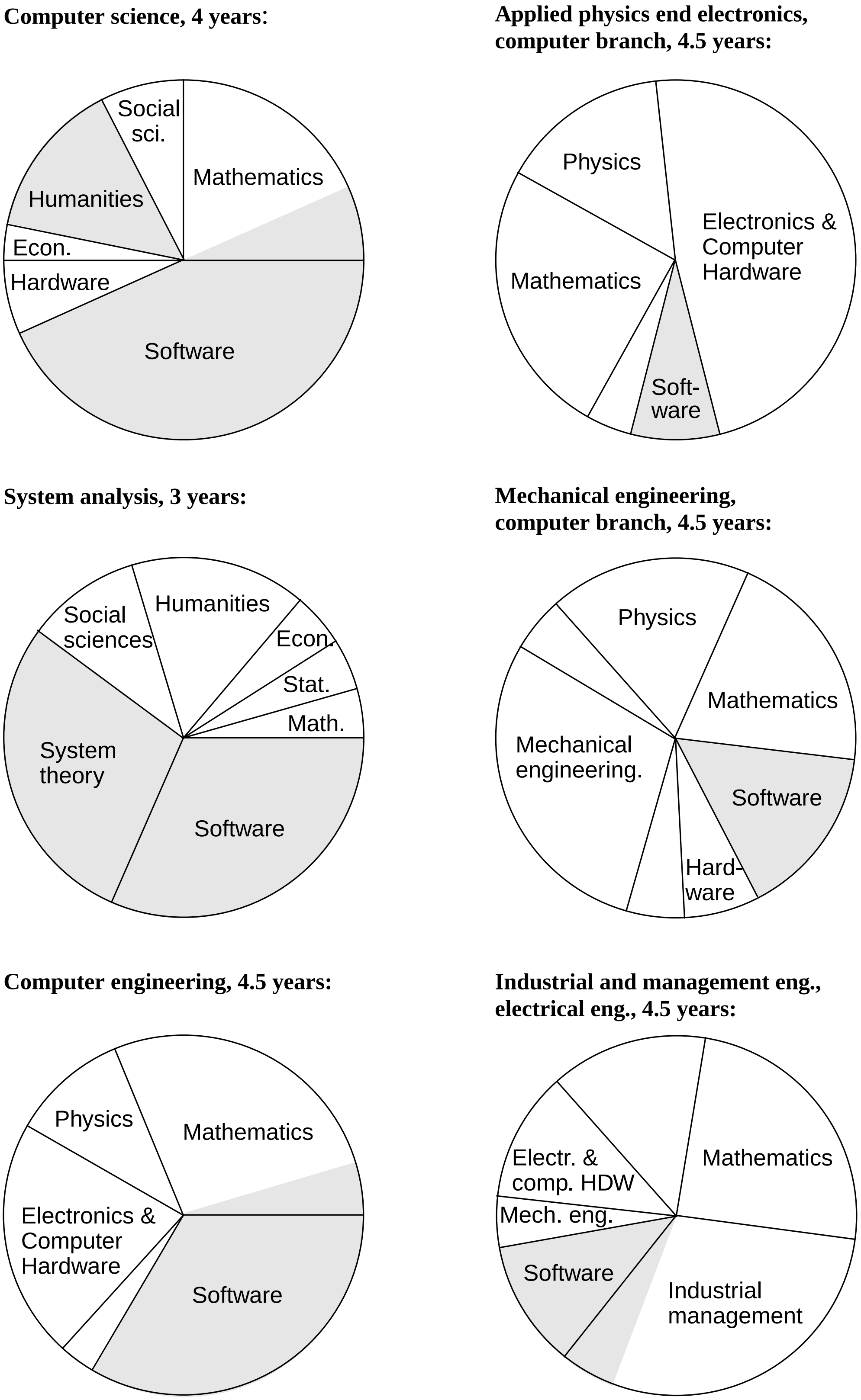
There are also courses in practical Swedish language for professional use, social science, mathematics and statistics. The second year ends with about five months of on-the-job training.

During the last year the SVL students can choose one of the following two specializations:

- Development of computer programs and program systems (program development) aimed at program development, methodology, and technology.
- Development of information systems (systemering), aimed at methodology for design and evaluation of information systems.

Both SVL specializations end with a term-paper reporting the development and implementation of an individual project.

The transfer of the subject area Economic Information Systems to IDA offers new opportunities for the development also of the SVL line.



Computer science, 4 years:

Applied physics end electronics, computer branch, 4.5 years:

System analysis, 3 years:

Mechanical engineering, computer branch, 4.5 years:

Computer engineering, 4.5 years:

Industrial and management eng., electrical eng., 4.5 years:

Figure 15-3. *The department share (shadowed) of the different study programmes.*

15.4 Continuing education for Swedish Industry

During several years we have had a number of ’continuing education’ programmes in computer science, primarily for engineers.

The courses are given as academic courses and give academic credits after normal examination. They are organized for half-times studies and are given in such a way that the participants are free for studies 2 days a week with one full-day teaching and one day for reading and exercises of their own.

The last year we have given a 25 points programme in the area of AI/expert systems in the form of a *knowledge engineering training programme*, which covers the theoretical basis needed in that area. The programme was developed in 1986 in connection with the knowledge transfer program, KTP, (where also practical issues in knowledge engineering are covered). The programme was given during 1988 for ASEA Brown Boveri in Västerås and a new programme started spring 1989 for Volvo in Gothenburg.

The programme consists of the following courses:

- Discrete mathematics
- Logics
- AI programming languages (LISP Prolog)
- AI cognitive structures
- AI knowledge representation
- Expert systems
- Project work with an expert system tool.

Among other educational activities we can mention programmes in Process Programming and Operating Systems for Ellemtel and a Programming Methodology Block (including Data Structures, Computer Architectures, Process Programming and a project, etc.) for LM Ericsson and a number of Ada courses given for local companies in Linköping.

Appendix A
Department organization

The Department of Computer and Information Science (IDA) at Linköping University covers now four teaching subjects (computer science, economic information systems, telecommunication and computer systems, and administrative data processing).

The Department was formed in 1983, bringing together groups previously in the Mathematics and the Electrical Engineering departments. In September 1990 another group, Economic Information Systems, came from the Department of Production Economics.

A considerable flexibility was allowed when the internal organization and routines were to be decided. The basic idea was to build research within the department upon vital, autonomous, and cooperating research groups, each with a distinct leader and about five to ten more teachers, researchers, and employed graduate students. From the beginning there were four such groups or laboratories. Today there are twelve.

Formally all significant administrative decisions, such as the annual budget are taken by the Department Board. Important and general issues regarding research or undergraduate studies are treated by the Research Committee or the Undergraduate Teaching Committee respectively. Running economy and personnel issues are handled by Inger Emanuelson, who is also the leader for the group providing administrative services.

The lab leader is responsible for supervision and guidance of the work in his/her group, and also for writing grant proposals and reports to funders. Each lab also takes responsibility for maintaining competence in its area of research and some related areas, and to make it available to the rest of IDA in graduate courses and seminars, as well as in the undergraduate course program. The set of labs is designed to provide a sufficiently wide basis for a vital computer science department and also to give the necessary spectrum required for the undergraduate courses given by the department. At the same time it is important that research is sufficiently focused and that a group can achieve critical size in its area of specialization.

The department budget for the fiscal year 1990/91 balances at 52 MSEK. (One SEK is at present approximately 0.15 USD.) The resources for undergraduate education supplied by the university amount to 18,7 MSEK, and corresponding resources for research and graduate education are 13,1 MSEK. The research activities are thus heavily dependent on external sources, where the Swedish Board for Technical Development, STU, is the main contributor (90/91 10,7 MSEK). Additional funds are provided by the Swedish Council for Planning and Coordination of Research, FRN, (90/91 1 MSEK) and the Swedish Council for Research in the Humanities and Social Sciences, HSFR (90/91 1,2 MSEK). During 1990/91 the Swedish National Information Technology Program has supplied about 2,8 MSEK. Through European cooperation between ESPRI and NFR, we get research resources (1990/91 1 MSEK). Occasional sources, such as contributions from companies participating in the knowledge transfer programme and shorter projects supported by e.g. The Trade Association of the Swedish Mechanical and Electrical Engineering Industry (Sveriges Mekanförbund) are in the order of 2,4 MSEK. Commissioned education programmes for industry are budgeted at about 1,1 MSEK 1990/91. Costs for office space and investment in equipment are not included in the above figures.

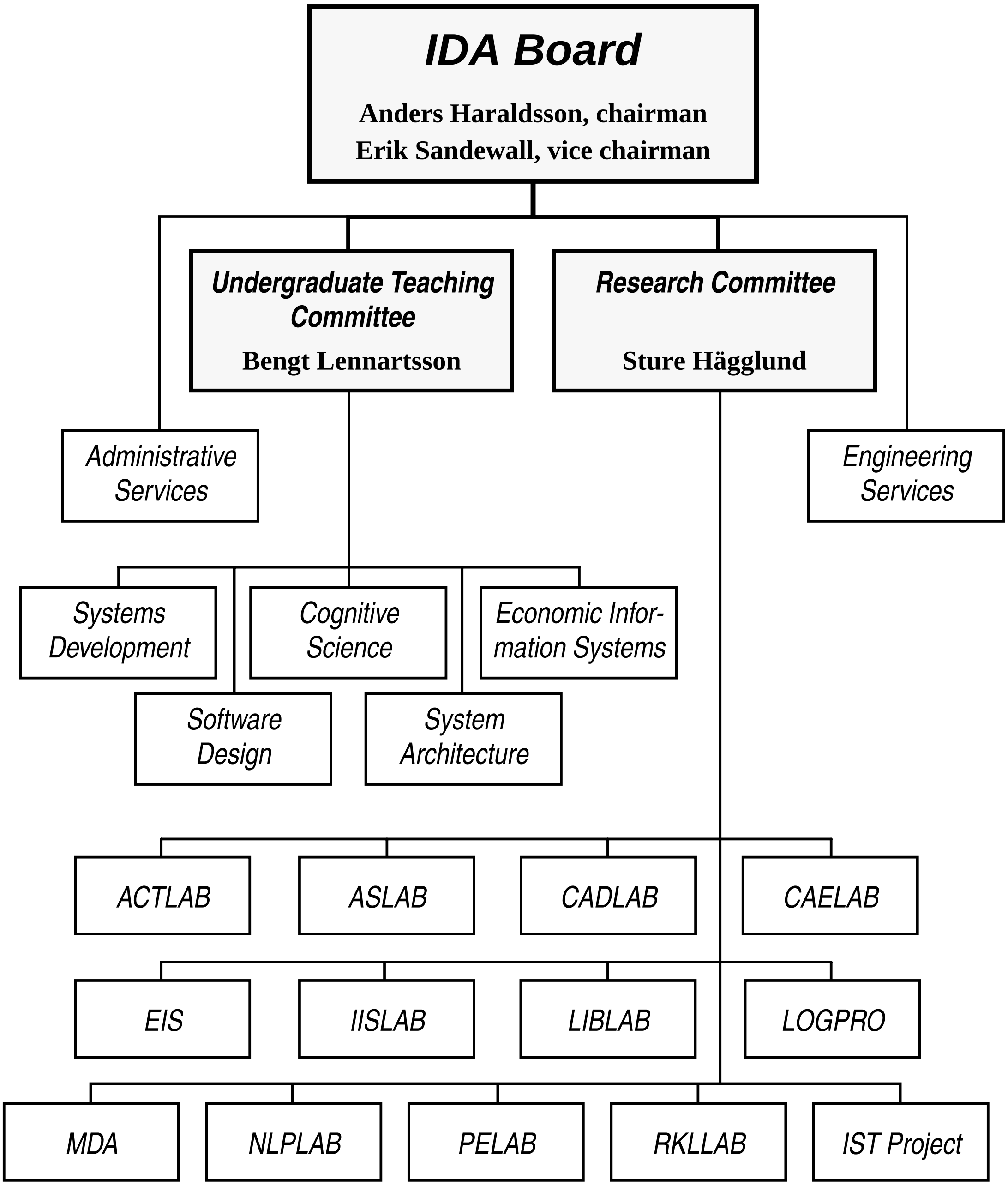
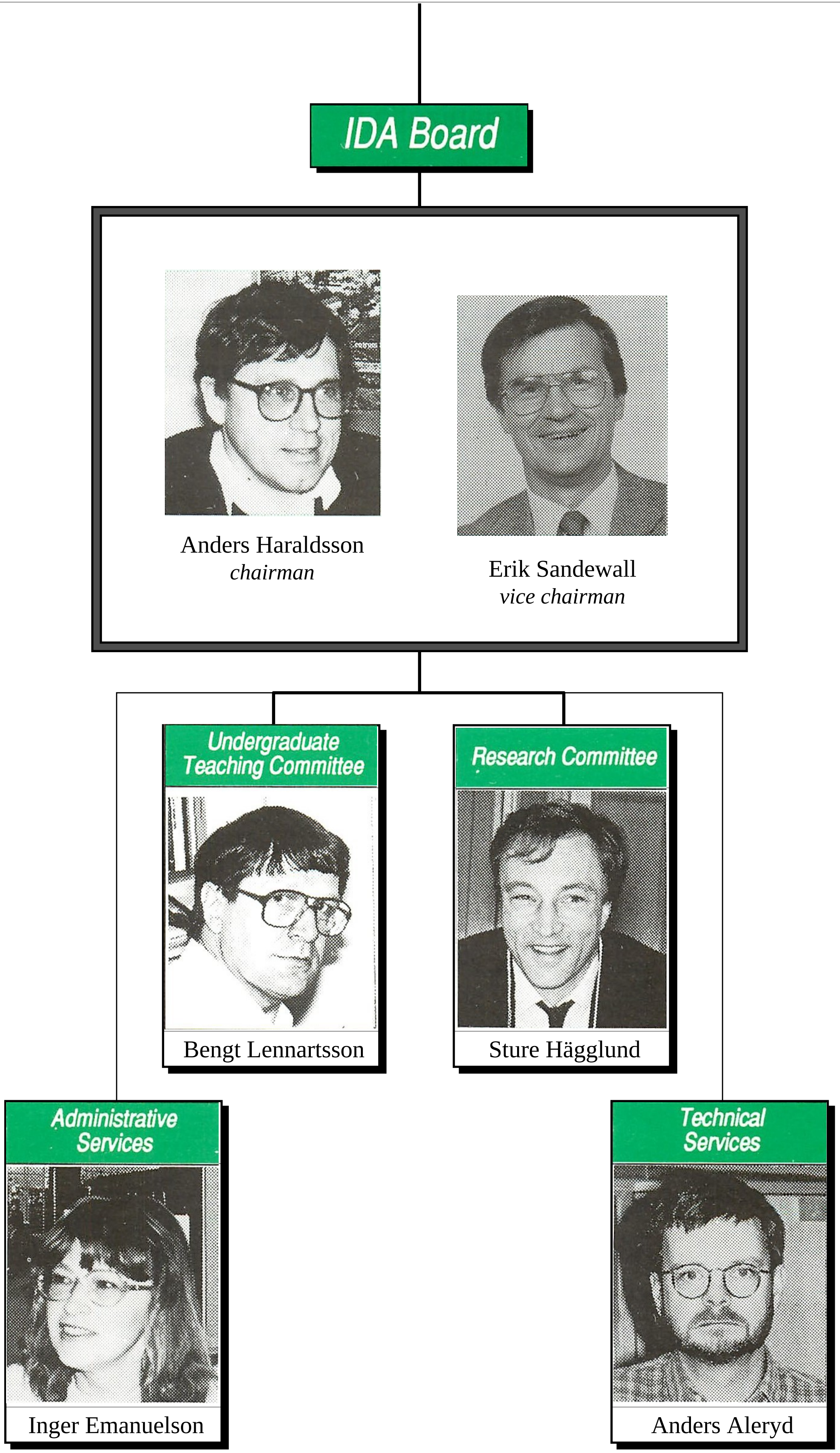


Figure A-1. *Organization of the department 1990/91.*

A.1 The Department Board

The Department Board is chaired by Anders Haraldsson, with Birgitta Franzén as secretary and the items are mostly prepared and submitted by Inger Emanuelson. Annually the board delegates to the two committees all issues about research and graduate studies, and about undergraduate education, respectively. The board also handles items related to both committees, normally by approving their coordinated proposals.

Figure A-2. *The departmental board.*



IDA Board

Anders Haraldsson
chairman

Erik Sandewall
vice chairman

Undergraduate Teaching Committee

Bengt Lennartsson

Research Committee

Sture Hägglund

Administrative Services

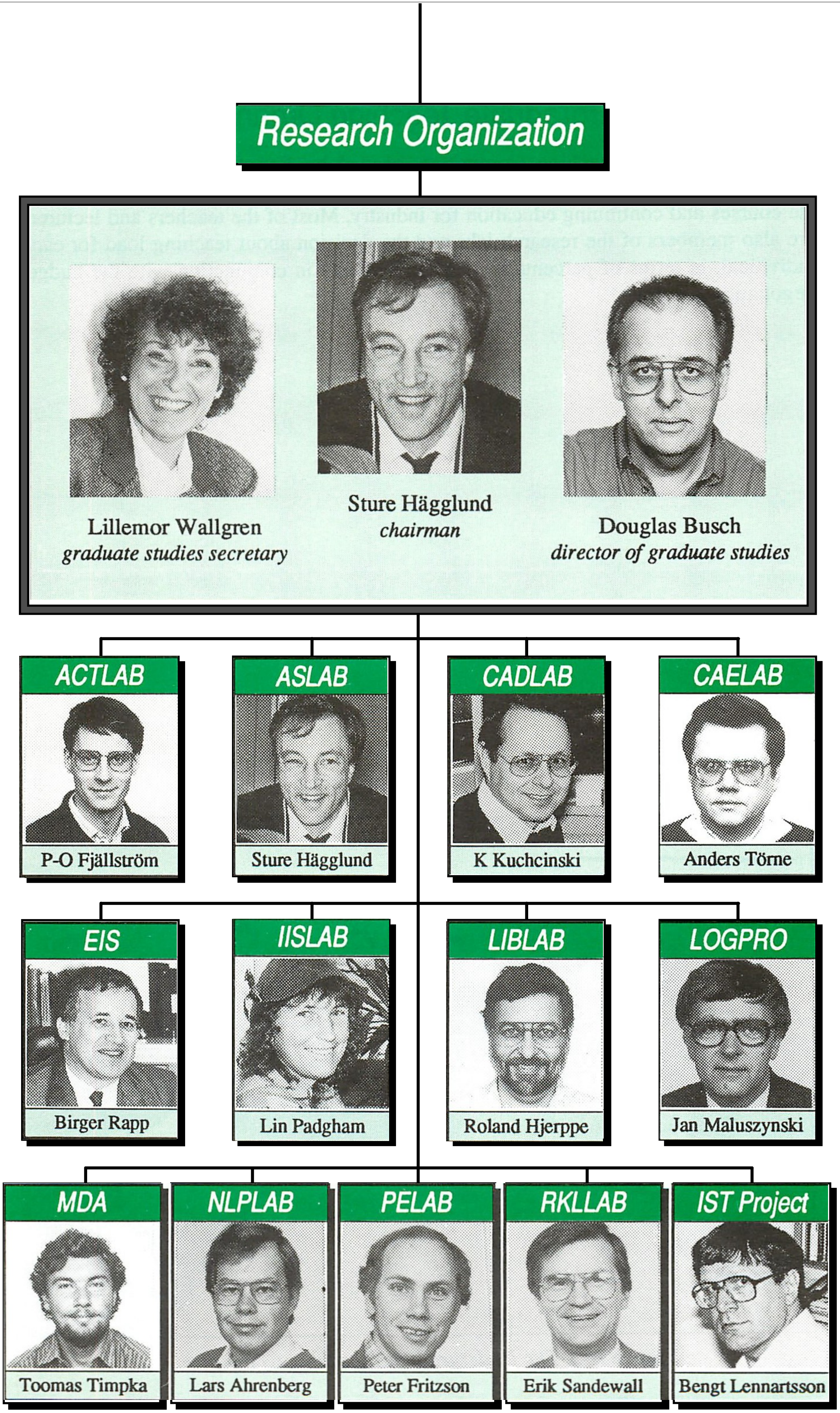
Inger Emanuelson

Technical Services

Anders Aleryd

A.2 The Research Organization

The Research Committee, headed by Sture Hägglund and with Lillemor Wallgren as secretary, handles research activities and graduate education. The committee suggests the annual budget for each research lab, based on the grant situation, and can also modify the lab structure by merging, splitting, creating or deleting labs and appointing lab leaders. Admission of doctorate students and master students has to be confirmed by the Research Committee. The committee also discusses and takes appropriate actions on research and equipment strategy in general, and coordinates the lab-based activities. The philosophy, however, is to support and assist rather than to control and supervise the labs.



Research Organization

Lillemor Wallgren
graduate studies secretary

Sture Hägglund
chairman

Douglas Busch
director of graduate studies

ACTLAB
P-O Fjällström

ASLAB
Sture Hägglund

CADLAB
K Kuchcinski

CAELAB
Anders Törne

EIS
Birger Rapp

IISLAB
Lin Padgham

LIBLAB
Roland Hjerppe

LOGPRO
Jan Maluszynski

MDA
Toomas Timpka

NLPLAB
Lars Ahrenberg

PELAB
Peter Fritzson

RKLLAB
Erik Sandewall

IST Project
Bengt Lennartsson

A.3 The Undergraduate Teaching Organization

The Undergraduate Teaching Committee, headed by Bengt Lennartsson and with Britt-Marie Ahlenbäck as secretary, is responsible for the organization of undergraduate courses and continuing education for industry. Most of the teachers and lecturers are also members of the research labs and the decision about teaching load for each individual, in terms of percentage, is taken annually in conjunction with the budget negotiation process.

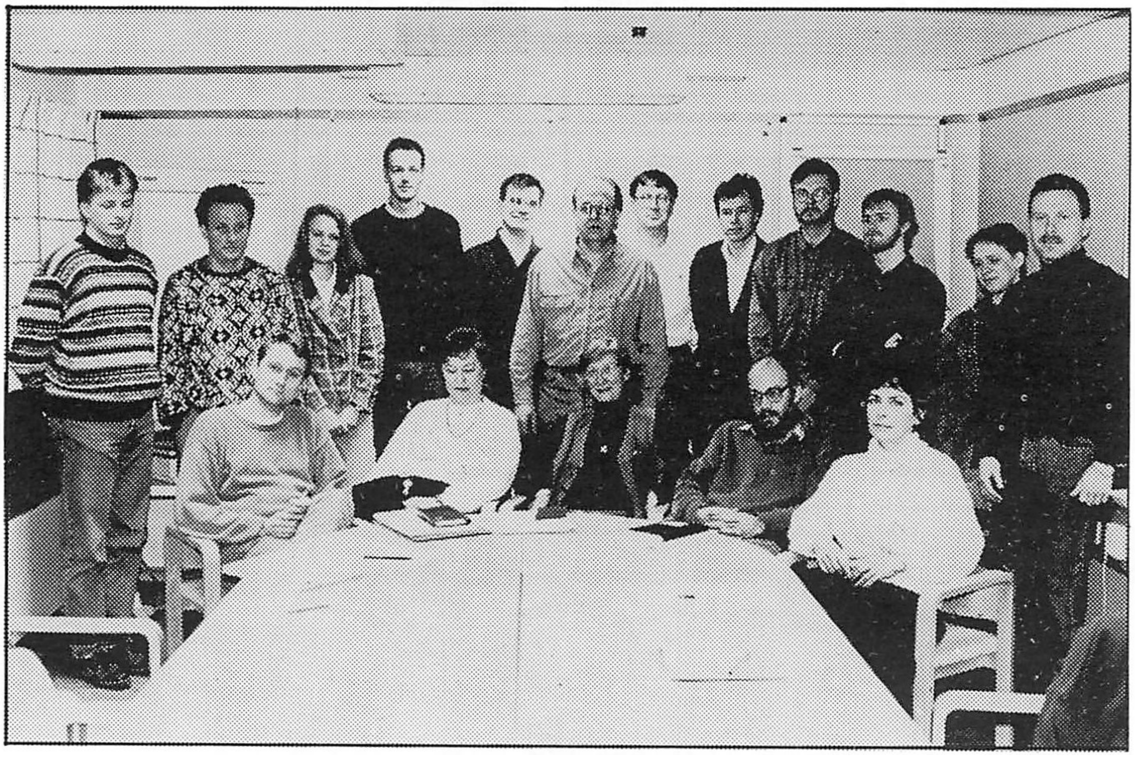


Figure A-3. *The Undergraduate Teaching Committee.*

Undergraduate Teaching Organization

Bengt Lennartsson
chairman

Britt-Marie Ahlenbäck
undergraduate studies secretary

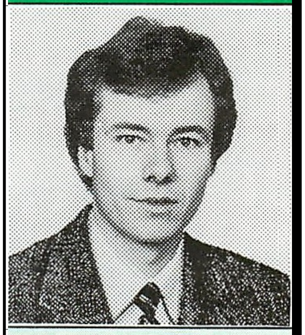
System Development

Lise-Lotte Raunio

Cognitive Science

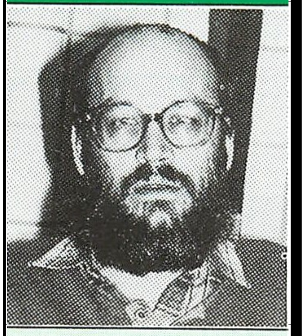
Arne Jönsson

Economic Information Systems



Lennart Ohlsén

Program Design



Olle Willén

System Architecture

Johan Fagerström

A.3.1 Subject areas

The organization of the undergraduate education was, as mentioned in chapter 15.2, revised during 1990. Under the Undergraduate Teaching Committee there are five subject areas, each with an executive director of studies, and there is one secretary, Britt-Marie Ahlenbäck, for undergraduate education.

The executive responsibility for undergraduate studies is taken by the directors of studies. We have five directors, Johan Fagerström (Computer Architecture), Arne Jönsson (Cognitive Science), Lennart Ohlsén (Economic Information Systems), Lise-Lotte Raunio (Systems Development) and Olle Willén (Software Design). They are all responsible for the study programs within their area of knowledge.

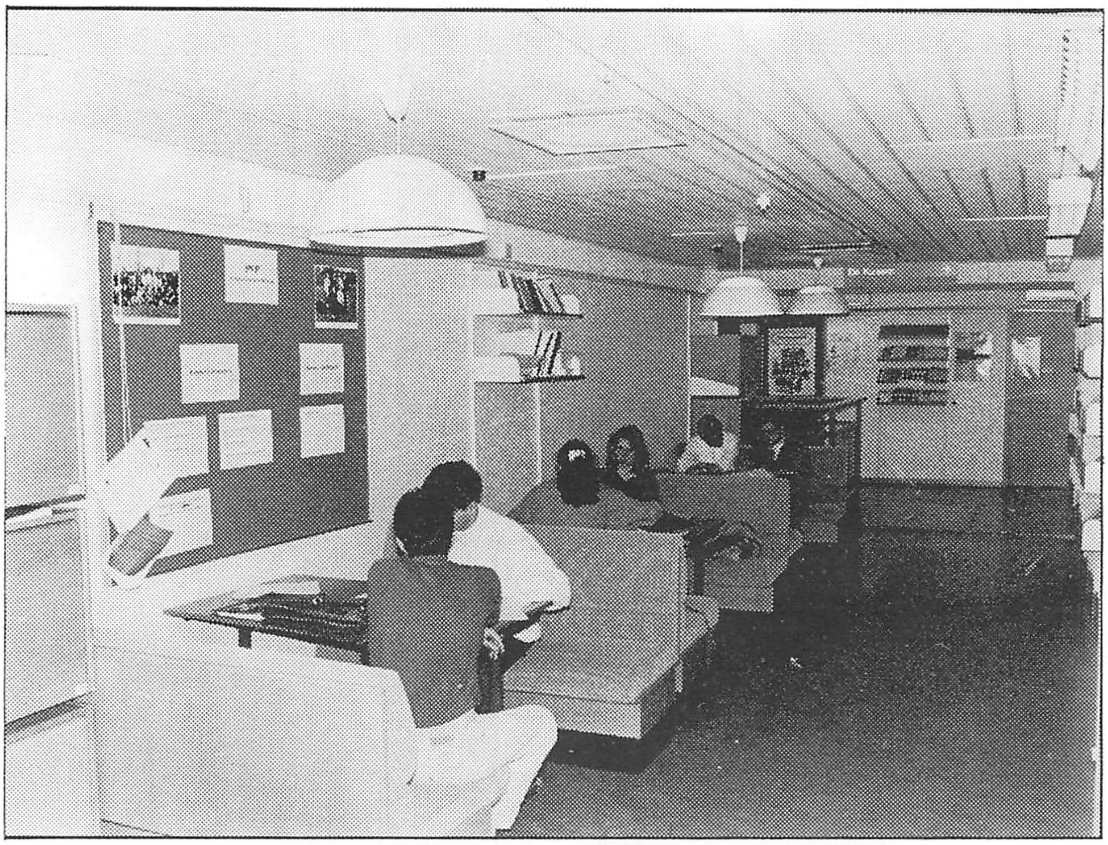
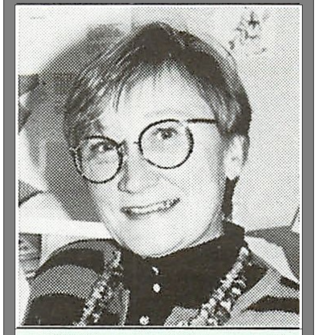


Figure A-4. *Students' working area.*

Systems Development

Subject area:

Systems Development



Lise-Lotte Raunio
director

Back row: Rolf Nilsson, Siv Söderlund, Hans Holmgren, Lise-Lotte Raunio, Carita Lilja, Annie Röstlinger.
Front row: Christian Krysander, Eva-Chris Svensson, Ivan Nilsson, Stefan Cronholm.
Missing: Göran Goldkuhl, Mikael Johansson.

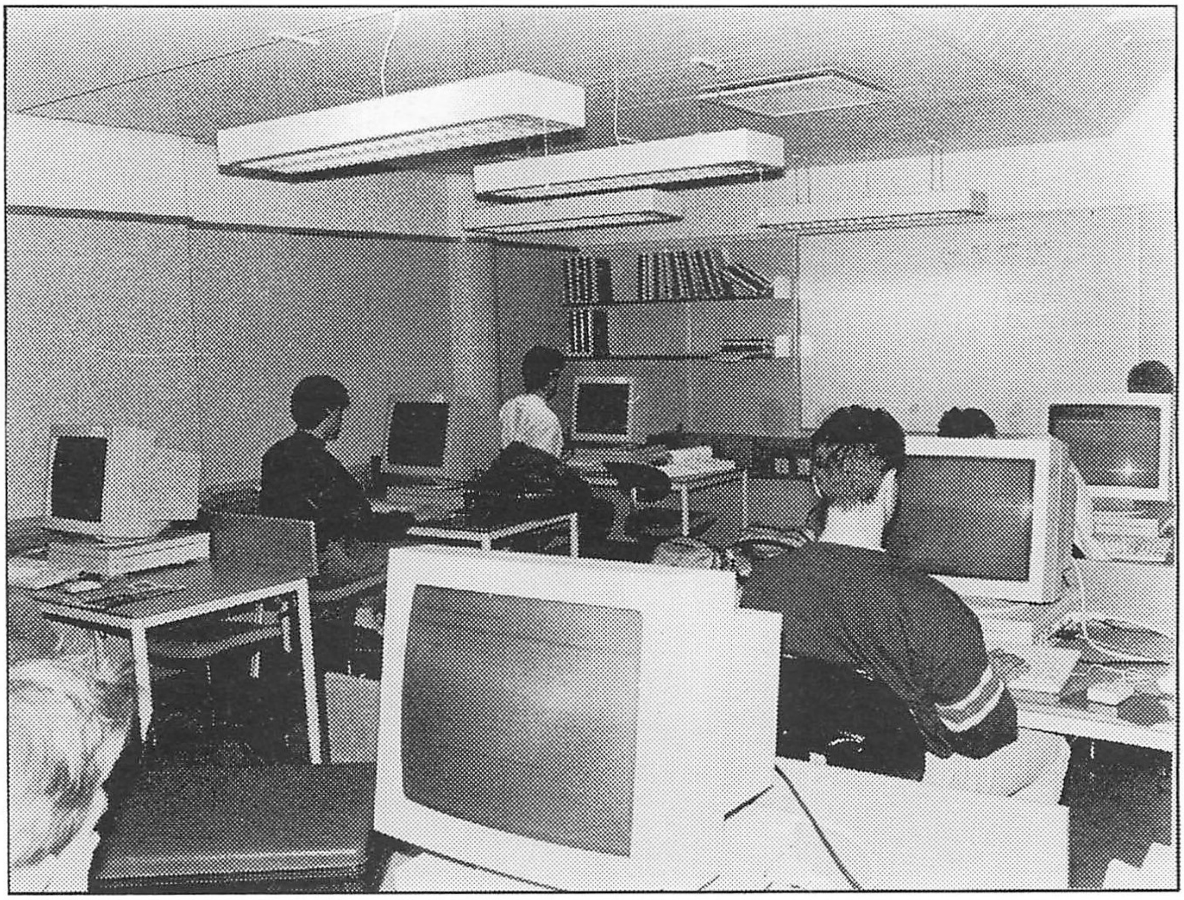
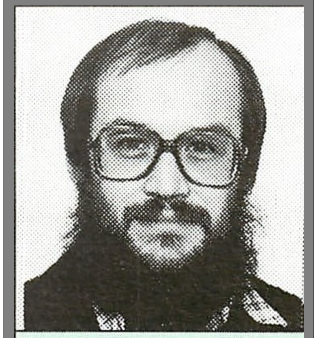


Figure A-5. *Sun computer lab.*

Software Design

Subject area:

Software Design



Olle Willén
director

Back row: Michael Jansson, Mikael Patel, Mariam Kamkar, Åke Thurée, Lars Viklund.
Middle row: Jan Alexandersson, Per-Olof Fjällström, Peter Fritzson, Mats Persson, Jan Petersson, Rober Bilos, Olle Willén, David Andersson, Jonas Löwgren, Christian Krysander, Erik Stoy, Stefan Cronholm.
Front row: Rego Granlund, Tommy Olsson, Åsa Wretland, Katarina Sunnerud, Eva Lundberg, Peter Loborg, Torbjörn Jonsson, Patrik Krajina.
Missing: Barbara Ekman, Anders Haraldsson, Lotta Månsbacka, Ulf Nilsson, Kerstin Olsson.

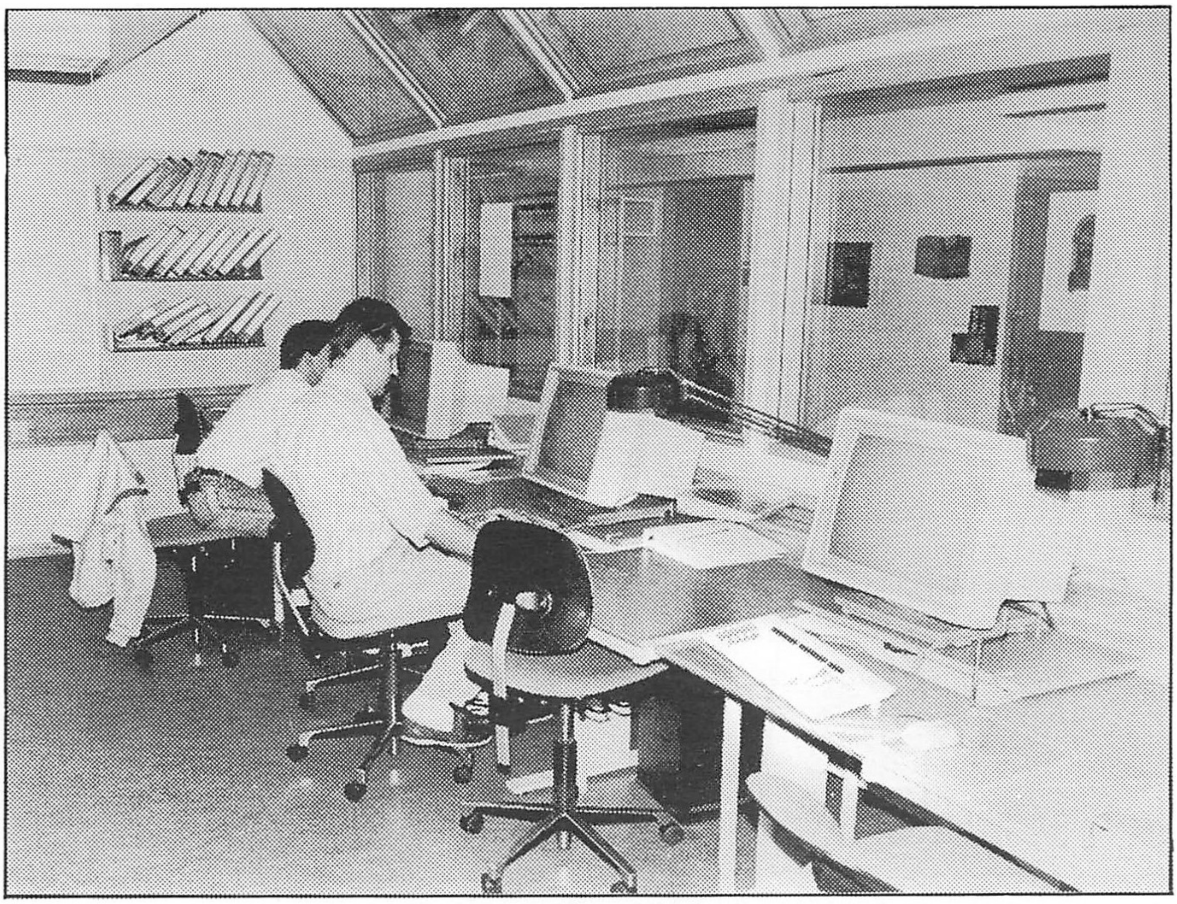
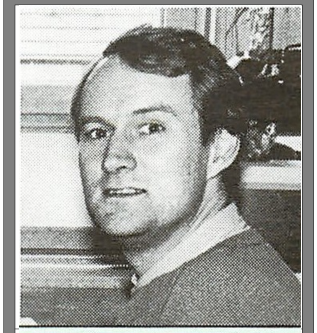


Figure A-6. *Xerox computer lab.*

Cognitive Science

Subject area:
Cognitive Science



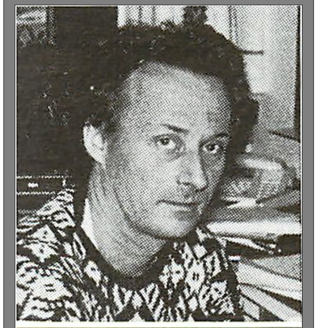
Arne Jönsson
director

Back row: Richard Hirsch, Ivan Rankin, Lars Ahrenberg, Dimiter Driankov, Nils Dahlbäck, Christian Krysander.
Front row: Arne Jönsson, Åke Thurée, Rolf Nilsson, Lisbeth Linge, Mats Wirén.
Missing: Jan Alexandersson, Douglas Busch, Jonni Harrius, Hans Holmgren, Sture Hägglund, Jalal Maleki, Erik Sandewall.

System
Architecture

Subject area:

System Architecture



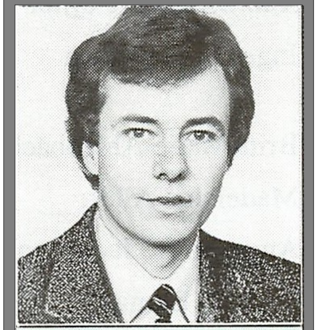
Johan Fagerström
director

Back row: Lars Strömberg, Johan Fagerström, Anders Törne, Björn Fjellborg, Staffan Bonnier.
Front row: Gunilla Lingenhult, Jan Maluszynski, Krzysztof Kuchcinski.
Missing: Rober Bilos, Peter Fritzson, Peter Johannesson, Mariam Kamkar, Bengt Lennartsson, Zebo Peng, Nahid Shahmehri, Katarina Sunnerud.

Economic Information Systems

Subject area:

Economic Information System



Lennart Ohlsén
director

Back row: Carl-Gunnar Hansson, Lars Mattson, Torbjörn Näslund, Eva Elfinger, Birger Rapp, (Maria Olsson, Camilla Sjöström), Merhan Noghabai, (Anette Stolt).
Front row: (Peter Carlsson), Jonas Lind, Björn Helander, Jörgen Andersson, Anna Moberg, Jaime Villegas.
Missing: Lennart Ohlsén., Thomas Seld, Rickard Wendel.

Persons named within parentheses are not members of the teaching staff.

A.4 Administrative services

Administrative office personnel

Inger Emanuelson administrative manager.

Britt-Marie Ahlenbäck general educational secretary.

Madeleine Allert office assistant (on leave).

Anne-Marie Jacobson secretary of the Technical Support Group.

Barbara Ekman secretary of the subject area Software Design, and The Laboratory for Library and Information Science.

Eva Elfinger secretary of the subject area Economic Information Systems, and the group for Economic Information Systems.

Birgitta Franzén secretary of the Laboratory for Applications Systems.

Ewa Johansson on leave.

Carita Lilja secretary of the subject area System Development.

Lisbeth Linge secretary of the subject area Cognitive Sciences, and the Laboratory for Natural Language Processing.

Gunilla Lingenhult secretary of the subject area Computer Architecture, and the laboratories of Computer Assistance in Engineering, Logic Programming, Programming Environments, and the project for Industrial Software Technology.

Bodil Mattsson Kihlström secretary of laboratories for Complexity of Algorithms, Computer Aided Design, and Intelligent Information Systems.

Nils Nilsson department porter.

Ingrid Sandblom secretary of the laboratory for Representation of Knowledge in Logic.

Siv Söderlund Studie Councelor, and secretary of the subject area System Development.

Lillemor Wallgren general research secretary

Lena Wigh secretary of the group for People, Computers and Work, and the department registrar.

Administrative services

Administrative Services

Inger Emanuelson
administrative manager

B-M Ahlenbäck

Madeleine Allert

Anna Bladh

Barbara Ekman

Eva Elfinger

Birgitta Franzén

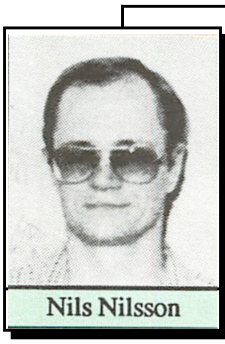
AM Jacobson

Carita Lilja

Lisbeth Linge

Gunilla Lingenhult

Bodil Mattsson


Nils Nilsson

Ingrid Sandblom

Siv Söderlund

Lillemor Wallgren

Lena Wigh

A.5 Technical services

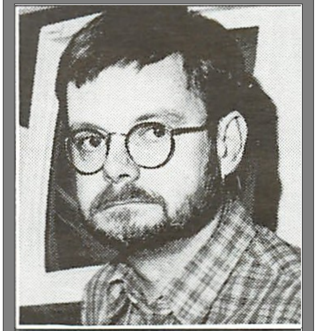
The system support group under Anders Aleryd and Mats S Andersson is responsible for computer systems and services, as well as for other kinds of equipment at the department. Computer resources and other equipment are normally not reserved for a specific group or project, but shared as far as possible and supported at the department level. This allows a good economy for support costs and effective use of the facilities, although projects needing exclusive access to a particular piece of equipment of course can be granted that right for a specific period of time.

Technical services personnel

Anders Aleryd	managing engineer
Anne-Marie Jacobson	secretary
Mats S Andersson,	senior research engineer, system manager for research facilities
Ulf Dahlén	research engineer, system manager for Xerox Lisp systems
Leif Finmo	research engineer, system manager for Sun system, hardware facilities
Dimitrios Fotiadis	research engineer, system manager for Sun system and networks
Arne Fäldt	senior research engineer, system manager for Sun system and Editing and Publishing program environment
Bernt Nilsson	research engineer, system manager for Sun system and Xerox Lisp systems
Björn Nilsson	senior research engineer, system manager for Sun system, administration environment
Peter Nilsson	research engineer, system manager for Sun system, educational systems
Rolf Nilsson	lecturer, system manager for PC systems
Göran Sedvall	engineer, manager of computer networks and hardware
Katarina Sunnerud	research engineer

Technical Services

Technical Services



Anders Aleryd
managing engineer

Mats S Andersson

Ulf Dahlén

Leif Finmo

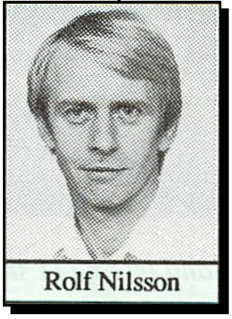
Dimitrios Fotiadis

Arne Fäldt

Bernt Nilsson

Björn Nilsson

Peter J Nilsson



Rolf Nilsson

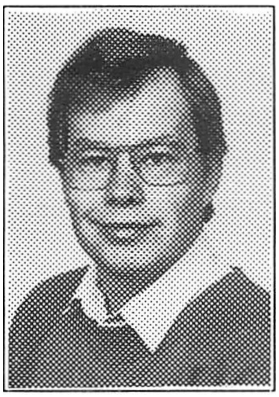
Katarina Sunnerud

Göran Sedvall

Figure A-7. *Administrative and technical staff.*

Appendix B Graduate Study Program

Faculty presently engaged in graduate study program.

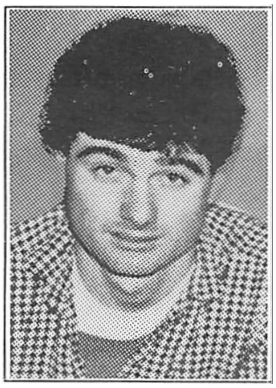


Lars Ahrenberg, PhD, Uppsala 1987. Associate professor (*docent, högskolelektor*) of computational linguistics. Group leader, NLPLAB. Previous affiliations Uppsala and Göteborg.

Syntax, semantics and pragmatics of natural language; natural language understanding; natural language interfaces; text generation.

Douglas Busch, PhD, Rockefeller 1973. Associate professor (*högskolelektor*) of logic and theoretical computer science. Previous affiliation Mcquarie Univ., Sydney, Australia. Director of the graduate study programme.

Application of theories from formal logic to problems in theoretical computer science and artificial intelligence; semantics of logic programs; philosophical questions in artificial intelligence.



Keith Downing, PhD, Univ. of Oregon 1990. Guest researcher.

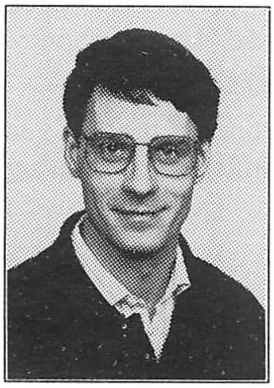
Model-based diagnosis, qualitative physics, reason maintenance, artificial intelligence in medicine. Applications of model-based simulation and diagnosis to physiological domains

Dimiter Driankov, PhD, Linköping 1989. Assistant professor *(högskolelektor)*, logic and AI.

Reasoning under uncertainty, many-valued logics, knowledge-based plan-recognition, decision support systems.

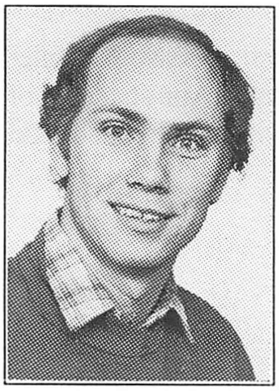
Johan Fagerström, PhD, Linköping 1988. Assistant professor (*högskolelektor*), computer science. Director of undergraduate studies in System Architecture.

Distributed systems, parallel systems, operating systems.



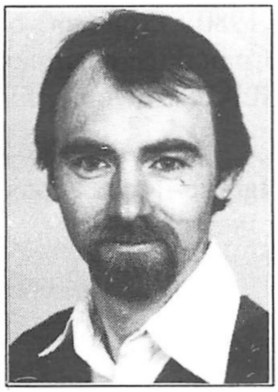
Per-Olof Fjällström, PhD, Stockholm 1985. Associate professor *(högskolelektor)*, theoretical computer science. Previous affiliation KTH and IBM. Group leader, ACT-LAB.

Computational geometry, analysis of algorithms, data structures.



Peter Fritzson, PhD, Linköping 1984. Associate professor (*högskolelektor*), computer science. Group leader, PELAB. Previous affiliations Sun MicroSystems USA 1985-86, Linköping Univ. Hospital 1976-79.

Programming environments, scientific computing, debugging tools, incremental compilation technology, compiler generation, compilers for parallel hardware.



Göran Goldkuhl, PhD, Stockholm 1980. Associate professor (*docent, högskolelektor*), administrative data processing. Group leader in systems development research. Previous affiliation Göteborg.

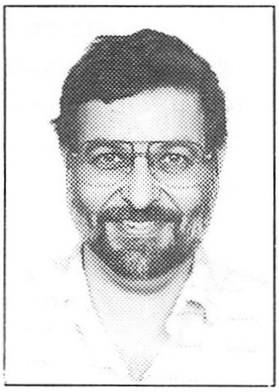
Information requirement analysis, behavioral aspects of information systems, research methodologies, information systems and quality of working life.

Anders Haraldsson, PhD, Linköping 1977. Associate professor (*högskolelektor*), computer science. Department chairman (from autumn 1990) and Undergraduate Committée chairman (until autumn 1990). Previous affiliation Uppsala.

Programming languages and systems, programming methodology, program manipulation, partial evaluation.

Richard Hirsch, PhD, Gothenburg 1989. Lecturer (*högskoleadjunkt*) of general and computational linguistics. Thesis supervision in the laboratory for natural language processing. Previous affiliation Gothenburg.

Syntax, semantics, and pragmatics of natural languages; discourse analysis; argumentation theory; philosophy of language.



Roland Hjerppe. Researcher. Group leader, LIBLAB. Previous affiliation KTH, DFI and expert mission Tanzania. Visiting Distinguished Scholar at Office of Research, OCLC Inc. in Columbus, Ohio, 1988-89.

Library science and systems, hypertext and -media, knowledge organization and information retrieval, citation analysis and bibliometrics, computer support for personal and cooperative activities, virtual environments.

Sture Hägglund, PhD, Linköping 1980. Professor of knowledge-based systems. Research Committee chairman (from autumn 1990). Group leader, ASLAB. Previous affiliation Uppsala.

Expert systems and artificial intelligence applications, database technology, human-computer interaction.

Feliks Kluzniak, PhD, Warsaw University 1987. Guest researcher. On leave from the Institute of Informatics, Warsaw University.

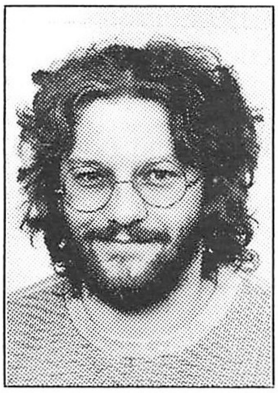
Logic programming, compilers, programming methodology.

Krzysztof Kuchcinski, PhD, Gdansk 1984. Acting professor of computer systems. Group leader, CADLAB. Previous affiliation Technical Univ. of Gdansk, Poland.

Computer architecture, computer-aided design of digital systems, VLSI, test generation methods.

Bengt Lennartsson, PhD, Göteborg 1974. Associate professor (*högskolelektor*), software engineering. Previous affiliation Luleå. Group leader, PELAB until 1988. Department chairman 1983-1990. Undergraduate Committee chairman (autumn 1990).

Real-time systems, industrial software technology, large scale software development.



Jacek Malec, PhD, Wroclaw 1987. Assistant professor. Previous affiliation Technical Univ. of Wroclaw, Poland.

Artificial Intelligence: knowledge representation, planning, reactive systems, autonomous systems architecture, dynamic scene description.

Jan Maluszynski, PhD, Warszawa 1973. Professor of programming theory. Several previous affiliations. Group leader, LOGPRO.

Logic programming, formal language theory, amalgamation of programming paradigms.

Lin Padgham, PhD Linköping 1989. Assistant professor *(högskolelektor),* computer science. Previous affiliation Univ. of Oregon, USA, and Tektronix. Group leader, IISLAB.

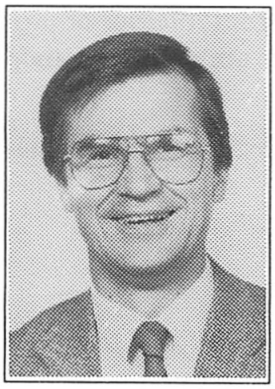
Inheritance, default reasoning, taxonomical reasoning, object-oriented systems.

Zebo Peng, PhD, Linköping 1987. Assistant professor (*högskolelektor*), computer architecture.

Automated synthesis of digital systems, formal description of hardware, VLSI, computer-aided design, computer architecture.

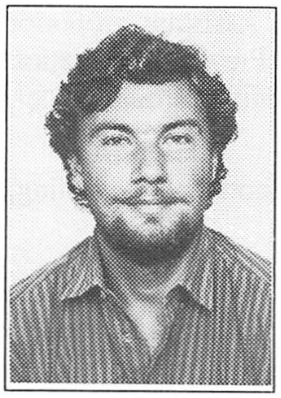
Birger Rapp, Econ. Dr., Stockholm 1974, Professor of Economic Information Systems. Several previous affiliations.

Accounting systems, economic control, IT and organisation, production, economics.



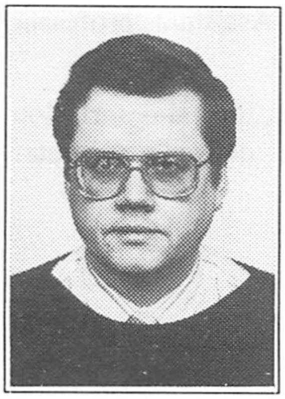
Erik Sandewall, PhD, Uppsala 1969. Professor of computer science. Research Committee chairman (1983-1990). Group leader, RKLLAB. Several previous affiliations.

Representation of knowledge with logic, autonomous agents, knowledge-based planning.



Toomas Timpka, MD., Stockholm 1983, PhD, Linköping 1989. Assistant professor (*högskolelektor*), computer and information science. Group leader, MDA.

Hypermedia, computers and society, human-computer interaction, systems development.



Anders Törne PhD, Uppsala 1980. Associate professor (*högskolelektor)*, computer support in automation. Group leader, CAELAB. Previously at ABB Corporate Research, Västerås.

Computer support for generation, transformation, and use of information in manufacturing processes. Architectures for processing control and supervision. Robot programming.

B.1 Graduate Study Course Program 1989–90

Basic and Occasional Graduate Courses:

Distributed Operating Systems (Johan Fagerström)
Temporal Logics (Douglas Busch)
Programming Environments (Bengt Lennartsson)
Distributed Databases (Johan Fagerström)
CIM: Data and Process Models (Anders Törne)
Combinators and Lambda Calculus (Jan Maluszynski)
Formal Hardware Description and Verification (Tony Larsson)
Principles of Database and Knowledge-Base Systems (Sture Hägglund, Jerker Wilander)
Hypermedia: History, concepts and applications (Toomas Timpka)
Functional and Logic Programming Languages and their Implementation (Gary Lindstrom)
Data Structures and Graph Algorithms (Per-Olof Fjällström)
Petri Nets and their applications (Krzysztof Kuchcinski, Zebo Peng)
Grammars and Logic Programming (Jan Maluszynski)
Truth Maintenance (Michael Reinfrank)
Machine Learning (John H. Holland, et al.)
Semantics of Logic Programs and Deductive Databases (Teodor Przymusinski)
Formalization of Non-Monotonic Reasoning with the Emphasis on the Autoepistemic Approach (Halina Przymusinska)
Test Generation Methods for Digital System (Raimund Ubar)
Inductive Learning (Nada Lavrac)

Research-Related Courses and Seminars:

Knowledge Engineering Methodology (Sture Hägglund)
Theories of Grammar (Lars Ahrenberg)

B.2 Graduate Study Course Program 1990–91

Basic and Occasional Graduate Courses:

Advanced Logic (Douglas Busch)
Artificial Intelligence Paradigms (Dimiter Driankov, Patrick Doherty)
Compiler Construction – Advanced Course (Peter Fritzson)
Compiling for Parallelism (Peter Fritzson, Krzysztof Kuchcinski, Johan Fagerström)
Computational Geometry (Per-Olof Fjällström, Jyrki Katajainen)
Construction of Correct Programs (Feliks Kluzniak)
Document Description and Architecture – Theories and Standards (Roland Hjerppe)
Empirical Research Methods (Toomas Timpka)
Epistemic Logics (Douglas Busch)
Hardware Description Methods and the VHDL Language (Krzysztof Kuchcinski, Zebo Peng)
Hypermedia: History, Concepts and Applications (Toomas Timpka)
Knowledge Based Planning (Jalal Maleki, Dimiter Driankov)
Methodology of Research in Computer Science (Sture Hägglund, Roland Hjerppe)
Modal Logic (Douglas Busch)
Object-Oriented Programming and Design Methodology (Lin Padgham)
Qualitative Reasoning (Keith Downing)
Reason Maintenance Systems (Michael Reinfrank)
Type and Proofs (Jan Maluszynski)

B.3 Seminars

Seminars Autumn 1989

Aug. 15 Dr. Tore Risch, Stanford Univ. “Monitoring Database Objects”.

29 Prof. Jon Doyle, MIT, Cambridge Mass. “Stochastic Analyses of Qualitative Dynamics”.

Sept. 1 Keith Downing, Univ. of Oregon. “The use of teleology in the qualitative explanation and evaluation of circulatory models”.

7 Dr. Elaine Frankowski, Honeywell, Minnesota. “Current trends in UIMS research”.

Dr. Elaine Frankowski, Honeywell, Minnesota. “Presentation of UIMS project at Honeywell”.

Dr. Elaine Frankowski, Honeywell, Minesota. “UIMS – State of the Art”.

7 Dr Harrison Schmitt, astronaut, senator etc. “Information Management for a Trip to the Moon”.

11 Ivan Rankin, IDA, Linköping Univ. Licentiate seminar. “The Deep Generation of Text in Expert Critiquing Systems”.

14 Henrik Eriksson, IDA, Linköping Univ. Licentiate seminar. “A Study in Domain-Oriented Tool Support for Knowledge Acquisition”.

25 Vincenzo Ambriola, Pisa. “Software Process Modelling in Object Oriented Style”.

27 Dr. Harvey Abramson, Univ. of Bristol. “A Concise formalism for Specifying Machine Translation: First steps to CAT3”.

Oct. 2 Simin Nadjm-Tehrani, IDA, Linköping Univ. Licentiate seminar. “Contributions to the Declarative Approach to Debugging Prolog Programs”.

16 Tony Larsson, IDA, Linköping Univ. Disputation.“A Formal Hardware Description and Verification Method”.

16 Dr. George Milne, Univ. of Strathclyde, Glasgow. “Design for verifiability Future trends for formal hardware verification”.

16 Dr. George Milne. Univ. of Strathclyde, Glasgow. “Hierarchical Structure in Cognitive Maps”.

Nov. 2 Magnus Merkel, IDA, Linköping Univ. Licentiate seminar. “Temporal Information in Natural Language”.

17 Prof. David A. Thomas. ACTRA – “A Smalltalk Based Multi-processor OBJECT-ORIENTED Development System”.

23 Dr. Edward Tsang, Univ. of Essex, Dept. of Computer Science. “Constraint Propagation”.

23 Dr. Hilding Elmqvist, SATT Control, Lund. “Visual Programming of Control Systems”.

24 Dr. Edward Tsang, Univ. of Essex, Dept. of Computer Science. “What is planning anyway?”.

30 Dr. Carl-Martin Allwood, Dept. for Psychology, Gothenburg Univ. “On beginners debugging of own Pascal programs, and how to improve it”.

Dec. 5 Dr. Jacek Malec, Institute of Technical Cybernetics, Technical Univ. of Wroclaw. “Describing Dynamic Scene of a Robot”.

7 Dr. Witold , Institute of Informatics, Warsaw Univ. “Abnormality Logic”.

11 Ulf Nilsson, IDA, Linköping Univ. Licentiate seminar. “A Systematic Approach to Abstract Interpretation of Logic Programs”.

12 Catuscia Palamidessi, Univ. of Pisa. “Semantics of K-leaf”.

13 Dr. Seif Haridi, SICS, Kista. “ANDORRA Prolog”. This is a large international project based on a new idea of parallel implementation of Prolog.

18 Dr. Lars Eldén, MAI, Linköping Univ. “Parallelism in Linear Algebraic Computations”.

21 Dr. Kent Petersson, Chalmers Institute of Technology, Gothenburg. “Concrete Syntax for Data Objects in Functional Languages”.

21 Staffan Bonnier, IDA, Linköping Univ. Licentiate seminar. “Horn Clause Logic with External Procedures: Towards a Theoretical Framework”.

Seminars Spring 1990

Jan. 15 Dave Fawcett, Ford Motor Co: “AI and expert systems at Ford”.

23 Prof. Joao Martins, Instituto Superior Tecnico, Lisbon. “Computational Issues in Belief Revision”.

25 Prof. Joerg Siekmann, Univ. of Kaiserslautern. “Unificaton Theory”.

31 Prof. Peter Tancig, Jozef Stefan Institute, Ljubljana. “Computational Linguistics in Slovenia and Yugoslavia”.

Feb. 19 Christer Hansson, IDA, Linköping Univ. Licentiate seminar. “A Prototype System for Logical Reasoning about Time and Action”.

March 6 Prof. Bud Lawson, Förlag och Konsult AB, Lidingö. “Philosophies for Engineering Real-time Systems”.

15 Bill Graves.”Contested Constructions: Divergent Uses and Understandings of a Central Informational Resource.”

23 Björn Fjellborg, IDA, Linköping Univ. Licentiate seminar. “An Approach to Extraction of Pipeline Structures for VLSE High-Level Synthesis”.

26 Erik Tiden, Siemens, Munich. “Boolean unification in Prolog and its application”.

April 4 Prof. Jan Komorowski, Åbo Akademi. “Towards systematic design of logic programs via partial deduction”.

20 Dr. Alan Borning, Dept. of Computer Science and Engineering, Univ. of Washington and Xerox EuroPARC, Cambridge, England. “Hierarchical Constraint Logic Programming: Recent Development”.

27 Dr. Per Svensson, FOA, Stockholm. “Query Languages for Spatial Databases – An Overview”.

May 4 Bo Kähler, Dr. Oddvar Risnes. “Distributed database management and database machines”.

11 Dr. Wojciech Rytter, Warsaw Univ. “Parallel expression evaluation”.

18 Dr. John Molgaard, ElektronikCentralen, Hœrsholm. “DESIGN/CPN, a Tool Package Supporting the Use of Coloured Petri Nets”.

21 Dr. Tore Risch, Stanford Univ. “Object-oriented Databases”.

23 Dr. Chris Marlin, The Dept. of Computer Science, The Univ. Of Adelaide, Australia. “1. MultiView: An Integrated Programming Environment Based on Multiple Concurrent Views”, 2. Making Progress in the Design of Programming Languages”.

28 Dr. Feliks Kluzniak, Dept. of Computer Science, Univ. of Bristol. “Developing applications for Aurora”.

June 5 Prof. Daniel Tabak, George Mason Univ., Fairfax, USA. “1. The Intel 80860 system, 2. The IBM RS/6000 POWER (Performance Optimized With Enhanced RISC) system, 3. The Motorola M 88000 system, 4. The AMD 29000 system, 5. The Sun SPARC, 6. The MIPS Computer Systems R3000 and R6000, 7. Overview of applications in workstations, multiprocessors and real time”.

13 Prof. Gul Agha, Dept. of Computer Science, Univ. of Illinois, Urbana. “Linguistic Abstractions for Concurrent Computing: A Model and Its Applications”.

14 Patrick Doherty, IDA, Linköping Univ. Licentiate seminar. “A Three-Valued Approach to Non-Monotonic Reasoning”.

Aug. 6 Dr. Jörg-Rudiger Sack, Carleton Univ. “Computing the configuration space for a robot on a mesh-of-processors.”

Appendix C
Courses and teachers in the undergraduate education 1989/90

IDA is responsible for computer science courses in the School of Engineering as well as in the School of Arts and Sciences. There are close to 100 such courses with a total of almost 5000 students.

C.1 Undergraduate courses in the School of Engineering

The annual intakes to the study programmes in the School of Engineering are:

Computer Science (C), 30 students per year
Computer Engineering (D),120 students per year
Industrial and Management Engineering (I), 180 students per year
Mechanical Engineering (M),120 students per year
Applied Physics and Electrical Engineering (Y),180 students per year

These study programs run over 4–4.5 years and lead to a Master of Engineering or (for the C-program) a Master of Science degree.There are also single-subject courses given as part-time and evening courses, and external courses given directly to companies and organizations.

Course name in Swedish (study progr. &year)	**Course name in English**	**Teacher**
Databaser (I4, M4, Y4)	Databases	Christian Krysander
Orientering datateknik och datorutrustn. (C1, D1)	Introd. CS and computer equipment	Katarina Sunnerud
Programmering i Ada(C4, I4, M4, Y4, SVL)	Programming in Ada	Olle Willén
Ada och programspråk (C3, C4, D3, D4)	Ada and programming languages	Tommy Olsson
Systemutveckling, teori och tillämpn. (C4, D4)	System development	Christian Krysander
Artificiell intelligens D (D3)	Artificial intelligence	Jonni Harrius
AI-programmering (C4)	AI programming	Jonni Harrius
Logik, grundkurs (C1, D2)	Logic, introductory course	Erik Sandewall
AI-kunskapsrepresentation (C4)	Representation of knowledge in AI	Douglas Busch
Psykologisk grundkurs (C1, D4, Y4)	Psychology, introductory course	Nils Dahlbäck
Databehandling av naturligt språk (C4)	Natural-language processing	Mats Wirén
Processprogrammering (C3, D3)	Concurrent programming	Johan Fagerström
Artificiell intelligens och LISP (I4, M4, Y4)	Artificial intelligence and LISP	Jalal Maleki
Programmering Y, grundkurs (Y2)	Introd. to CS and programming	Peter Loborg
Kompilatorer och interpr. (D4, I4, Y4, SVL)	Compilers and interpreters	Nahid Shahmehri
Datastrukturer (M4, Y4)	Data structures	Tommy Olsson
Programmeringsteori II (C4)	Programming theory II	Staffan Bonnier
Verktyg för programvaruproduktion (C4, D4)	Tools for production of programs	Peter Fritzson
Konstruktion och analys av algoritmer (C4, D5)	Design and analysis of algorithms	Per-Olof Fjällström
Objektorienterad programmering (C4, D4)	Object-oriented programming	Mikael Patel
Kompilatorkonstruktion (C3, D4)	Compiler construction	Rober Bilos
Databasteknik (C3, D4)	Database technology	Christian Krysander
Logikprogrammering (C3, D4)	Logic programming	Ulf Nilsson
Datorer – verktyg och tillämpningar (I1)	Computers – tools and applications	Lotta Månsbacka

Programmeringsteori (C3)	Programming theory	Peter Fritzson
Realtids- och processprogr. (I4, M4, Y4, SVL)	Real time and concurrent progr.	Bengt Lennartsson
Programmering i Pascal (C1)	Programming in Pascal, C	Kerstin Olsson
Programmering i Pascal (D2)	Programming in Pascal, D	Kerstin Olsson
Projektarbete i programmering (D2)	Program developm. proj.t in Pascal	Åke Thurée
Datastrukturer och algoritmer (C2, D3)	Data structures and algorithms	Mariam Kamkar
Artificiell intelligens C (C2, C3, C4)	Artificial intelligence	Arne Jönsson
Programutv.metodik o progr.proj.D (D4)	Meth. of prog. dev. and project	Christian Krysander
Programutvecklingsmetodik M (M3)	Methodology of program devel.	Olle Willén
Programutv.metodik o progr.proj.C (C3)	Meth. of prog. dev. and project	Christian Krysander
Expertsyst. – metodik och verktyg (C4, D4, I4)	Expert systems – methods and tools	Sture Hägglund
Distribuerade system(C4, D4)	Distributed systems	Johan Fagerström
Data- och programstrukturer (C2, D4)	Data and program structures	Patrick Doherty
Datastrukturer och programutv.metodik (I2)	Data structures and prog. dev. meth.	Lotta Månsbacka
Datalingvistik (C2)	Computational linguistics	Lars Ahrenberg
Formella språk och automatateori (C2, D4)	Formal lang. and automata theory	Torbjörn Näslund
Datorsystem och programmering (M2)	Computer systems and programming	Lotta Månsbacka
Programmering i inkrementellt system (C1)	Programming in an incr. system	Anders Haraldsson
Programmering i inkrementellt syst. (D1, SVL)	Programming in an incr. system	Anders Haraldsson
Lingvistik grundkurs (C1)	Introduction to linguistics	Magnus Merkel
Kommunikationspsykologi (C3)	Psychology of communications	Ivan Rankin
Diskret simuleringsteknik (D4, I4, Y4)	Discrete simulation	Zebo Peng
Datornät (D4, I4, Y4, SVL)	Computer networks	Björn Fjellborg
Datorarkitektur (C4, D4, Y4)	Advanced computer architecture	Krzysztof Kuchcinski
Datorstödd elektronikkonstruktion (D4, I4, Y4)	Computer aided design of electronics	Tony Larsson
Datalogi 2 – programmeringsprinciper	CS 2 – programming principles	Tommy Olsson

C.2 Undergraduate courses in the School of Arts and Science

The program for systems analysis (systemvetenskapliga linjen, SVL) ranges over three years of full-time studies. It aims at professional activities of design, evaluation and implementation of computer-based information systems. ADP-systems analysis dominates the program but nevertheless great importance has been attached to other subjects in order to give the program the necessary breadth and also to ensure that the students will become aware of the complexity of the community where computers can be used. 30 students are accepted annually.

In the list below “ADB” means single subject course in administrative data processing, and ’ADB-N’ indicates that such a course is given in Norrköping. ’FL’ means the study programme for Science of public administration (förvaltningslinjen), and ’SL’ the study programme for Statistics (statistikerlinjen).

Course name in Swedish (study progr.)	**Course name in English**	**Teacher**
Systemvetenskaplig grundkurs (SVL)	Introduction to systems science	Mikael Johansson
Utv. och förändring av informationssyst. (SVL)	Systems development	Annie Röstlinger
Inledande programmering och datatekn. (SVL)	Programming and CS, basic course	Torbjörn Näslund
Systemutvecklingsprojekt I (SVL)	Systems development, project I	Pia Arendell
Lagringsstrukturer (SVL)	Elementary data structures	Mariam Kamkar
Datorteknik	Elementary computer architecture	Katarina Sunnerud
Systemutredningsmetoder (SVL)	Methods for systems development	Eva-Chris Svensson
Informationshanteringssystem (SVL)	Fourth generation languages	Mikael Johansson
Systemutvecklingsprojekt II (SVL)	Systems development, project II	Eva-Chris Svensson
Förändringsanalys (SVL)	Change analysis	Annie Röstlinger
Praktik (SVL)	Practical work	Lise-Lotte Raunio

Samhällsvetenskaplig metodik (SVL)	Methods in social science	Göran Goldkuhl
Programutvecklingsmetodik (SVL)	Methodology of program developm.	Olle Willén
Utvärdering av datasystem (SVL)	Evaluation of EDP systems	Eva-Chris Svensson
Interaktiva system och databaser (SVL)	Interactive systems and databases	Hans Holmgren
Ekon./Samhällsvet. utvärd. av inf.syst. (SVL)	Social and econ. eval. of inf. syst.	Annie Röstlinger
Kompilatorer/Interpretatorer (SVL)	Compilers and interpreters	Nahid Shahmehri
Systemering, projekt (SVL)	Systemeering, project	Eva-Chris Svensson
Programmering, projekt (SVL)	Programming, project	Hans Holmgren
Introd. till systemutveckling (ADB)	Introduction to system developm.	Mikael Johansson
Inledande datateknik (ADB)	Computer science, basic course	Rolf Nilsson
Datorisering av administration (ADB)	Computers in office	Mikael Johansson
Interaktiva syst. och utvärd. av datasyst. (ADB)	Interactive syst. and syst. evaluation	Eva-Chris Svensson/ Hans Holmgren
Programmering i administrativ miljö (ADB-N)	Programming in an office environm.	Stefan Cronholm
Datorisering av administration (ADB-N)	Office automation	Lise-Lotte Raunio
Systemutvecklingsprojekt (ADB-N)	Systems development project	Stefan Cronholm
Administrativt utvecklingsarbete (FL)	Organizational development	Lise-Lotte Raunio
Grundkurs i programmering (SL)	Elementary programming	Katarina Sunnerud
Databasmetodik (SL)	Data base methodology	Rolf Nilsson
4G-språk (SL)	Fourth generation languages	A. Papageorgiu

C.3 Courses in Economic information systems

(Not at IDA 1989/90.)

Course name in Swedish (study progr.& year)	**Course name in English**	**Teacher**
Tekn. och ekon. utvärd. av datorsyst. (I4)	Techn.-econ. eval. of comp. syst.	Birger Rapp
Redovisning och budgetering (I2)	Accounting and budgeting	Janerik Lundquist
Industriell ekonomi (M4)	Ind. economics and management	Lennart Ohlsén
Kontorsinformationssystem (I4)	Office information systems	John Andersson
Datorer – verktyg och tillämpningar	Computers – tools and applications	Birger Rapp

C.4 Continuing education

Program development, continued, 2p. LM Ericsson, 14 participants.
Operating systems, 2.5p. Ellemtel, 22 participants.
Computer systems and program development, 15p. LM Ericsson, 19 participants.
AI and Expert systems, 25p. Volvo, 11 participants.
Programming in Ada. Saab-Scania, 10 participants.
Programmering i Ada, repeated. Saab-Scania, 10 participants

C.5 Teaching staff

Director of undergraduate studies in the School of Engineering:

Anders Haraldsson, PhD, associate professor in computer science,
Secretaries: Britt-Marie Ahlenbäck and Barbara Ekman.

Director of undergraduate studies in the School of Arts and Sciences:

Lise-Lotte Raunio, MSc, lecturer
Secretaries: Carita Lilja and Siv Söderlund.

The following persons from IDA have been responsible for one or more courses 1989/90:

Lars Ahrenberg, PhD.
Pia Arendell, BSc.
Rober Bilos, Lic.
Staffan Bonnier, Lic.
Douglas Busch, PhD.
Stefan Cronholm, BSc.
Nils Dahlbäck, B.A.
Patrick Doherty, Lic.
Johan Fagerström, PhD.
Björn Fjellborg, Lic.
Per-Olof Fjällström, PhD.
Peter Fritzson, PhD.
Göran Goldkuhl, PhD.
Anders Haraldsson, PhD.
Jonni Harrius, MSc.
Hans Holmgren, MScE.
Sture Hägglund, PhD.
Mikael Johansson, BSc.
Arne Jönsson, Lic.
Mariam Kamkar, Lic.
Krzysztof Kuchcinski, PhD.
Christian Krysander, MSc.
Tony Larsson, PhD.
Bengt Lennartsson, PhD.
Peter Loborg, MSc.
Jalal Maleki, Lic.
Magnus Merkel, B.A.
Lotta Månsbacka, MSc.
Rolf Nilsson, BSc.
Ulf Nilsson, Lic.
Torbjörn Näslund, MSc.
Torbjörn Näslund, BSc.
Kerstin Olsson, MSc.
Tommy Olsson, MSc.
Mikael Patel, Lic.
Zebo Peng, PhD.
Ivan Rankin, Lic.
Lise-Lotte Raunio, MSc.
Annie Röstlinger, BSc.
Erik Sandewall, PhD.
Nahid Shahmehri, Lic.
Katarina Sunnerud, MSc.
Eva-Chris Svensson, MSc.
Åke Thurée, MSc.
Olle Willén, BSc.
Mats Wirén, Lic.

Appendix D
Computer Facilities

The department has a policy of giving high priority to the supply of appropriate computing resources for research and education. We have also during the years been able to modernize and keep in pace with the rapid development in the area, e.g. regarding the emergence of powerful workstations with high-resolution graphics and high-performance CPU. Our orientation towards experimental computer science makes such a policy especially important and we believe that adequate computer equipment is essential for the quality of research and education.

Our main computer resources are 200 Sun SPARCstations with 9 Sun-4 fileservers, 23 Sun-3 workstations with 3 Sun-3 file servers and 24 Xerox 1186 Lisp Machines with file server.

Economic Information Systems group, who joined the department Oct. 1st 1990, have an IBM 9370 and a number of IBM PC's which has been made available by a grant from IBM.

In addition there are lots of smaller computers (MicroVax, PDP-11s, XEROX Viewpoint systems, Macintoshes and other PC:s of various kinds) and around 20 laser printers.

There is also special purpose equipment for specific research projects.

The schematic pictures on the following pages shows the local network and the accessible computer systems.

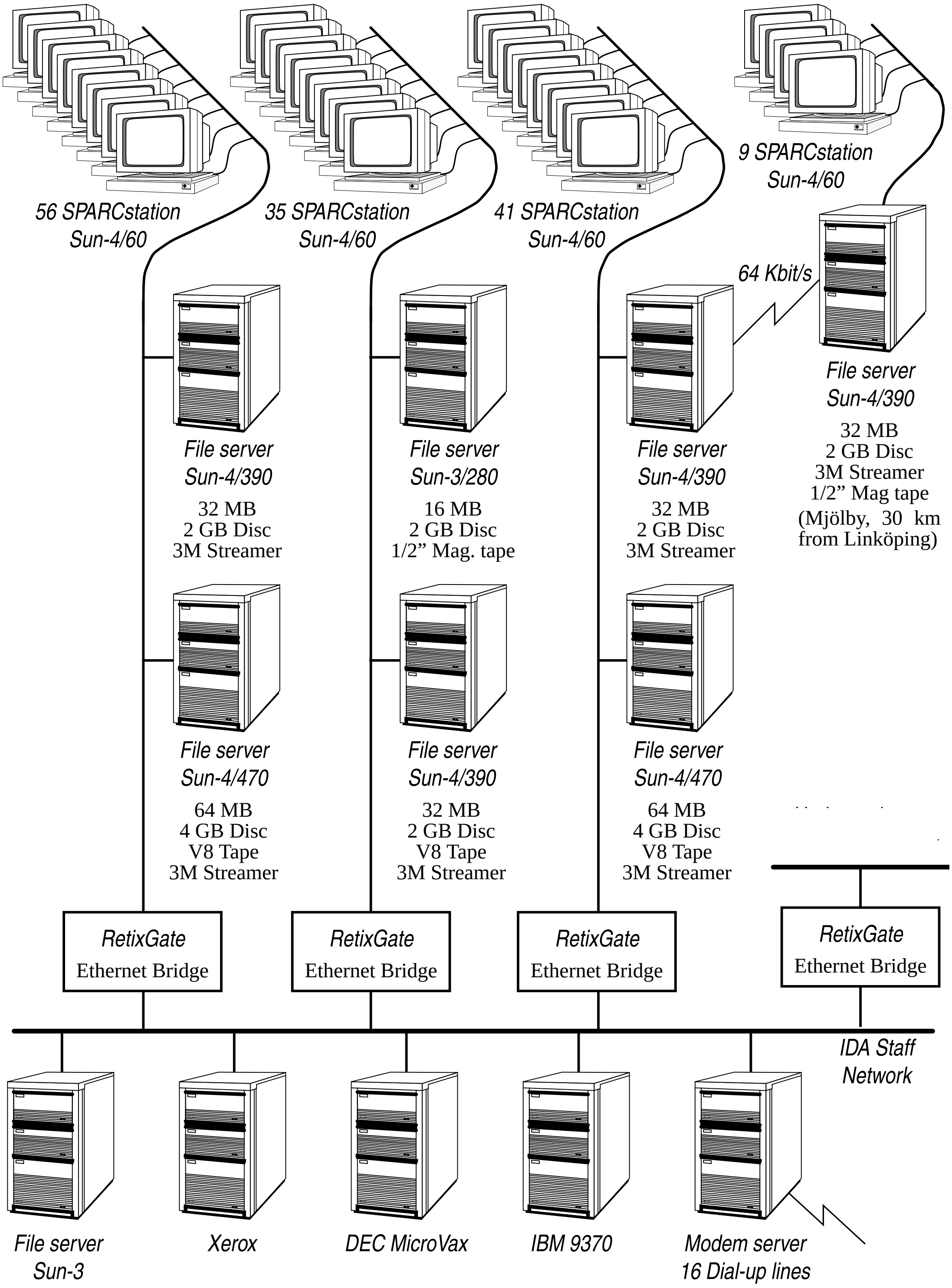


Figure D-1. *Main computer resources for the department staff.*

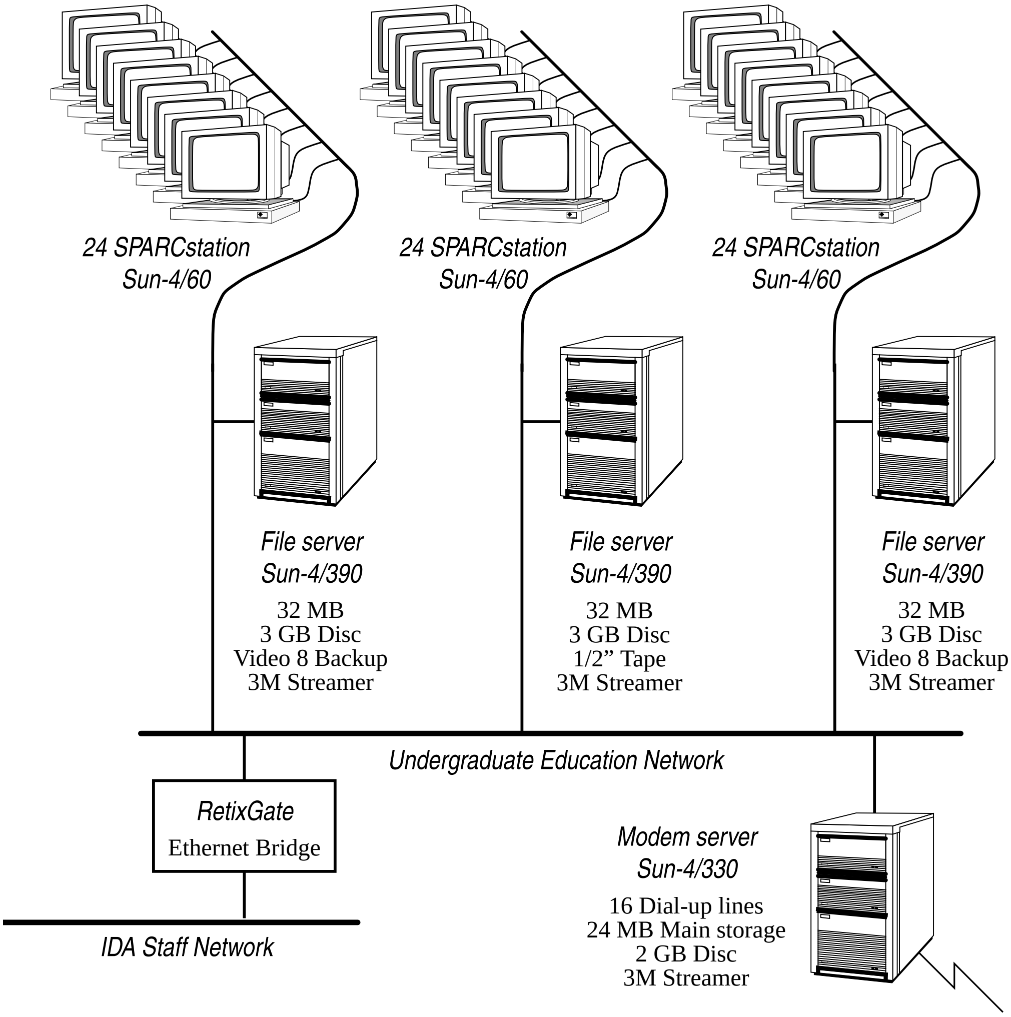


Figure D-2. *Main computer resources for the undergraduate education.*

Figure D-3. *Central computer parc.*

Appendix E
Publications

E.1 Dissertations

Linköping Studies in Science and Technology. Dissertations.

No 14 **Anders Haraldsson:** A Program Manipulation System Based on Partial Evaluation, 1977.

No 17 **Bengt Magnhagen:** Probability Based Verification of Time Margins in Digital Designs, 1977.

No 18 **Mats Cedwall:** Semantisk analys av processbeskrivningar i naturligt språk, 1977.

No 22 **Jaak Urmi:** A Machine Independent LISP Compiler and its Implications for Ideal Hardware, 1978.

No 33 **Tore Risch:** Compilation of Multiple File Queries in a Meta-Database System, 1978.

No 51 **Erland Jungert:** Synthesizing Database Structures from a User Oriented Data Model, 1980.

No 54 **Sture Hägglund:** Contributions to the Development of Methods and Tools for Interactive Design of Applications Software, 1980.

No 55 **Pär Emanuelson:** Performance Enhancement in a Well-Structured Pattern Matcher through Partial Evaluation, 1980.

No 58 **Bengt Johnsson, Bertil Andersson:** The Human-Computer Interface in Commercial Systems, 1981.

No 69 **H. Jan Komorowski:** A Specification of an Abstract Prolog Machine and its Application to Partial Evaluation, 1981.

No 71 **René Reboh:** Knowledge Engineering Techniques and Tools for Expert Systems, 1981.

No 77 **Östen Oskarsson:** Mechanisms of Modifiability in Large Software Systems, 1982.

No 94 **Hans Lunell:** Code Generator Writing Systems, 1983.

No 97 **Andrzej Lingas:** Advances in Minimum Weight Triangulation, 1983.

No 109 **Peter Fritzson:** Towards a Distributed Programming Environment based on Incremental Compilation, 1984.

No 111 **Erik Tengvald:** The Design of Expert Planning Systems. An Experimental Operations Planning System for Turning, 1984.

No 155 **Christos Levcopoulos:** Heuristics for Minimum Decompositions of Polygons, 1987.

No 165 **James W. Goodwin:** A Theory and System for Non-Monotonic Reasoning, 1987.

No 170 **Zebo Peng:** A Formal Methodology for Automated Synthesis of VLSI Systems, 1987.

No 174 **Johan Fagerström:** A Paradigm and system for design of distributed systems, 1988.

No 192 **Dimiter Driankov:** Towards a Many-Valued Logic of Quantified Belief, 1988.

No 213 **Lin Padgham:** Non-Monotonic Inheritance for An Object-Oriented Knowledge-Base, 1989.

No 214 **Tony Larsson:** A Formal Hardware Description and Verification Method, 1989.

No 221 **Michael Reinfrank:** Fundamentals and Logical Foundations of Truth Maintenance, 1989.

No 239 **Jonas Löwgren:** Knowledge-Based Design Support and Discourse Management in User Interface Management Systems, 1991.

No 244 **Henrik Eriksson:** Meta-Tool Support for Knowledge Acquisition, 1991.

Dissertation by IDA member published elsewhere.

Lars Ahrenberg: Interrogative Structures of Swedish: Aspects of the Relation between Grammar and Speech Acts. (Reports from Uppsala University Department of Linguistics No. 15, 1987).

E.2 Licentiate theses

Linköping Studies in Science and Technology. Theses.

No 17 **Vojin Plavsic:** Interleaved Processing of Non-Numerical Data Stored on a Cyclic Memory. 1983.

No 28 **Arne Jönsson, Mikael Patel:** An Interactive Flowcharting Technique for Communicating and Realizing Algorithms. 1984.

No 29 **Johnny Eckerland:** Retargeting of an Incremental Code Generator. 1984.

No 48 **Henrik Nordin:** On the Use of Typical Cases for Knowledge-Based Consultation and Teaching. 1985.

No 52 **Zebo Peng:** Steps towards the Formalization of Designing VLSI Systems. 1985.

No 60 **Johan Fagerström:** Simulation and Evaluation of an Architecture based on Asynchronous Processes. 1986.

No 71 **Jalal Maleki:** ICONStraint, A Dependency Directed Constraint Maintenance System. 1987.

No 72 **Tony Larsson:** On the Specification and Verification of VLSI Systems. 1986.

No 73 **Ola Strömfors:** A Structure Editor for Documents and Programs. 1986.

No 74 **Christos Levcopoulos:** New Results about the Approximation Behaviour of the Greedy Triangulation. 1986.

No 104 **Shamsul I. Chowdhury:** Statistical Expert Systems – a special application area for knowledge-based computer methodology. 1987.

No 108 **Rober Bilos:** Incremental Scanning and Token-based Editing. 1987.

No 111 **Hans Block:** Sport Sort – Sorting algorithms and sport tournaments. 1987.

No 113 **Ralph Rönnquist:** Network and Lattice Based Approaches to the Representation of Knowledge. 1987.

No 118 **Mariam Kamkar and Nahid Shahmehri:** Affect-Chaining in Program Flow Analysis Applied to Queries of Programs. 1987.

No 126 **Dan Strömberg:** Transfer and Distribution of Application Programs. 1987.

No 127 **Kristian Sandahl:** Case Studies in Knowledge Acquisition, Migration and Use Acceptance of Expert Systems. 1987.

No 139 **Christer Bäckström:** Reasoning about Interdependent Actions. 1988.

No 140 **Mats Wirén:** On Control Strategies and Incrementality in Unification-Based Chart Parsing. 1988.

No 146 **Johan Hultman:** A Software System for Defining and Controlling Actions in a Mechanical System. 1988.

No 150 **Tim Hansen:** Diagnosing Faults using Knowledge about Malfunctioning Behavior. 1988.

No 165 **Jonas Löwgren:** Supporting Design and Management of Expert System User Interfaces. 1989.

No 166 **Ola Petersson:** On Adaptive Sorting in Sequential and Parallel Models. 1989.

No 174 **Yngve Larsson:** Dynamic Configuration in a Distributed Environment. 1989.

No 177 **Peter Åberg:** Design of a Multiple View Presentation and Interaction Manager. 1989.

No 181 **Henrik Eriksson:** A Study in Domain-Oriented Tool Support for Knowledge Acquisition. 1989.

No 184 **Ivan Rankin:** The Deep Generation of Text in Expert Critiquing Systems. 1989.

No 187 **Simin Nadjm-Tehrani**: Contributions to the Declarative Approach to Debugging Prolog Programs. 1989.

No 189 **Magnus Merkel:** Temporal Information in Natural Language. 1989

No 196 **Ulf Nilsson:** A Systematic Approach to Abstract Interpretation of Logic Programs. 1989.

No 197 **Staffan Bonnier:** Horn Clause Logic with External Procedures: Towards a Theoretical Framework. 1989.

No 203 **Christer Hansson:** A Prototype System for Logical Reasoning about Time and Action. 1990.

No 212 **Björn Fjellborg:** An Approach to Extraction of Pipeline Structures for VLSI High-Level Synthesis. 1990.

No 230 **Patrick Doherty:** A Three-Valued Approach to Non-Monotonic Reasoning. 1990.

No 237 **Tomas Sokolnicki:** Coaching Partial Plans: An Approach to Knowledge-Based Tutoring. 1990.

No 250 **Lars Strömberg:** Postmortem Debugging of Distributed Systems. 1990.

No 253 **Torbjörn Näslund:** SLDFA-Resolution – Computing Answers for Negative Queries. 1990.

No 260 **Peter D. Holmes**: Using Connectivity Graphs to Support Map-Related Reasonong. 1990.

E.3 External publications since 1988

Refereed papers published in books, journals or international conference proceedings.

L. Ahrenberg: Functional Constraints in Knowledge-Based Natural Language Understanding. In *Proc. of the 12th International Conf. on Computational Linguistics*, in Budapest, August 22–26, 1988.

L. Ahrenberg: On the Integration of Linguistic Knowledge and World Knowledge in Natural Language Understanding. In *Proc. of the Nordic Conf. on Text Comprehension in Man and Machine,* Stockholm, October, 1988.

L. Ahrenberg: The Formalization of Field Grammars. In *Proc. of the 7th Int. Conf. of Nordic and General Linguistics.* Thorshavn, 7–11 August, 1989.

L. Ahrenberg: A Grammar Combining Phrase Structure and Field Structure. In *Papers presented to the 13th International Conference on Computational Linguistics*, Helsinki 20–25 August, 1990, Vol. 2 pp. 1–6, 1990.

L. Ahrenberg: Topological Fields and Word Order Constraints. In E. Engdahl, M. Reape, M. Mellor and R. Cooper (eds.) Parametric Variation in Germanic and Romance: *Proceedings from a DYANA Workshop,* September 1989. Edinburgh Working Papers in Cognitive Science, Vol. 6, pp. 1–19.

L. Ahrenberg, A. Jönsson, N. Dahlbäck: Discourse Representation and Discourse Management for a Natural Language Dialogue System. In *Papers from the 2nd Nordic Conf. on Text Comprehension in Man and Machine*, Täby April 25–27, 1990. (to appear).

E. Arborelius, T. Timpka: Study of the Practitioner's Knowledge Need and Use during Health Care Consultations. Part 2: The dilemma spectrum of the GP. In *Proc. of MedInfo '89,* North-Holland, Amsterdam, 1989.

E. Arborelius, T. Timpka: Physicians Comment on Video-Recorded Consultations: What can we learn from the comments to understand the process in the patient-physician relationship. *Family Practice,* 1989;7: 84–90.

E. Arborelius, S. Bremberg, T. Timpka: What is going on when the general practitioner doesn't grasp the consultation? *Family Practice* 1990, in press.

E. Arborelius, T. Timpka: In what way may video-tapes be used to get significant information about the patient-physician relationship. *Medical Teacher* 1990;12: 194–208.

C. Bäckström: A Representation of Coordinated Actions. *Proc. of the 1st Scandinavian Conf. on Artificial Intelligence,* Tromsö, Norway, March 9–11, 1988.

C. Bäckström: Logical Modelling of Simplified Geometrical Objects and Mechanical Assembly Processes. In *Proc. of the Workshop on Spatial Reasoning and Multi-Sensor Fusion,* St. Charles, Ill., USA, 1987. Extended version to appear in the *Advances in Spatial Reasoning* book series, ABLEX, 1990.

C. Bäckström: A Representation of Coordinated Actions Characterized by Interval Valued Conditions. In *Proc. of ISMIS'88,* Torino, Italy, October 12–15, 1988.

C. Bäckström: Action Structures with Implicit Coordination: In *Proc. AIMSA'88,* Varna, Bulgaria, September 20–23, 1988.

C. Bäckström, I. Klein: Planning in polynomial time. In *Proc. International Workshop on Expert Systems in Engineering,* Vienna, 1990.

T. Buur, T. Timpka, M. Lundberg: Urea kinetics and the clinical evaluation of the hemodialysis patient. Nephrology, *Dialysis and Transplantation* 1990;5: 347–51.

T. Buur, T. Timpka: The potential improvement of decisions regarding hemodialysis patients by the use of urea kinetic modeling. *Medical Decision Making* 1990;10: 274–82.

R. Bilos, P. Fritzson: Experience from a token sequence representation of programs, documents, and their deltas. In *International Workshop on Software Version and Configuration Control,* Grassau, FRG, January 27–29, 1988.

L. Björklund: HYPERCATalog – the Researcher's Tool for Information Management. In *Proc. of the Information Technology and The Research Process,* Cranfield, England, July 18–21, 1989.

L. Björklund: HYPERCLASS. Four hypertext applications of a classification scheme. In: *Tools for Knowledge Organization and the Human Interface: In Proc. of the First International ISKO-Conference,* Darmstadt, 14–17 Aug., 1990. R. Fugman, Ed. INDEKS Verlag. Frankfurt/Main FRG. 1990 pp.190–198.

L. Björklund, B. Olander: The Mystery Cat – The Modern Magician. *Proc of the 7. Nordiske IoD,* Denmark, August 28–30, 1989.

L. Björklund, B. Olander, L. Smith: The Personal HYPERCATalog. *Proc of the 1989 Annual Meeting of the American Society for Information Science*, Washington DC, October, 1989.

L. Björklund, L. Kristiansson: Problems of Knowledge Organization in an Archival Environment – KAM. In: *Tools for Knowledge Organization and the Human Interface: Proceedings of the First International ISKO-Conference,* Darmstadt, 14–17 Aug., 1990. *R. Fugman, Ed. INDEKS Verlag*. Frankfurt/Main FRG. 1990 pp. 199–205.

S. Bonnier, J. Maluszynski: Towards a clean amalgamation of logic programs with external procedures. In *Proc. of the 5th Int. Conf/Symp. on Logic Programming,* Seattle, MIT Press, 1988. In *Proc. of PLILP'88,* LNCS 348, Springer-Verlag, 1989.

N. Dahlbäck: Mental Models and Text Coherence. In *Proc. of the Nordic Conf. on Text Comprehension in Man and Machine,* Stockholm, October, 1988.

N. Dahlbäck, A. Jönsson: Empirical Studies of Discourse Representations for Natural Language Interfaces.In *Proc of the 4th Conf. of the European Chapter of the Association for Computational Linguistics,* Manchester, pp. 291–298, 1989.

N. Dahlbäck, A. Jönsson: Analyzing Human-Computer Dialogues in Natural Language. In Ranta (ed.) *Analysis, Design and Evaluation of Man-Machine Systems,* Pergamon Press, pp. 405–408, Oxford, 1988.

N. Dahlbäck: A Symbol Is Not A Symbol. In *Proc. of the 11th International Joint Conference on Artificial Intelligence (IJCAI 89)*, pp. 8–14, Detroit, Michigan, USA, August 20–25, 1989.

P. Deransart, J. Maluszynski: A Grammatical View of Logic Programming. In *Proc. of PLILP'88, L*NCS 348, Springer-Verlag. Orleans, France, May 16–18, 1989.

P. Deransart, J. Maluszynski: What kind of Grammars are Logic Programs. In *P. Saint-Dizier and S. Szpakowicz, editors, Logic and Logic Grammars for Language Processing*. Ellis Horwood, 1990.

H. Djidjev, A. Lingas, J. R Sack: An O(nlog n) Algorithm for Computing a Link Center in a Simple Polygon. In *Proc. of the 6th Symposium on Theoretical Aspects of Computer Science,* Paderborn, February 16–18, 1989.

P. Doherty: A Semantics for Inheritance Hierachies with Exceptions using a Logic of Preferential Entailment. In *Proc. SCAI'89*, Tampere, 1989.

P. Doherty: A Correspondence Between Inheritance Hierarchies and a Logic of Preferential Entailment, *IV, Z. Raz et.al. Methodologies for Intelligent Systems,. ISMIS 90*, 1989.

P. Doherty: Preliminary Report: NM3 – A Three-Valued Non-Monotonic Formalism, *V, Z. Raz et. al. Methodologies for Intelligent Systems,. ISMIS* 90, 1990.

P. Doherty: A Three-Valued Cumulative Non-Monotonic Formalism, In *Proc. of 1990 European Workshop on Logics in AI, JELIA90*. To be published in Artificial Intelligence Lecture Notes Series, Springer-Verlag, 1990.

P. Doherty, D. Driankov: A non-monotonic fuzzy logic. Accepted at IFSA'91 – The World Congress on Fuzzy Logic.

W. Drabent, J. Maluszynski: Inductive Assertion Method for Logic Programs. In *Theoretical Computer Science 59,* North-Holland, pp. 133–155, 1988.

W. Drabent, S. Nadjm-Tehrani, J. Maluszynski: Algorithmic Debugging with Assertions. In *H. Abramson and M. Rogers, editors, Meta-Programing in Logic Programming*, pp. 501–522. *MIT Press*, 1989. Also in *Proc. of the Workshop on Meta-Programming in Logic Programming, META '88,* University of Bristol, June 1988, pp. 365–378 J. Lloyd (ed.).

W. Drabent, S. Nadjm-Tehrani, J. Maluszynski: The Use of Assertions in Algorithmic Debugging. In *Proc. of the Int. Conf. on Fifth Generation Computer Systems,* Tokyo, 1988.

W. Drabent, M. Martelli: Strict Completion of Logic Programs. *New Generation Computing*, 1990. To appear.

D. Driankov: Inference with single fuzzy conditional proposition. *Int. Journal of Fuzzy Sets and Systems* vol. 24, No 1 pp. 51–63.

D. Driankov: A calculus for belief-intervals- representation of uncertainty. In *Proc of the Int. Conf. on Information Processing and Management of Uncertainty*, Paris, June 30 – July 4, 1986, pp. 235–239. (Extended abstract). In Lecture Notes on *Computer Science,* B. Bouchon and R.R. Yager (eds) vol. 286, pp. 205–216.

D. Driankov: Inference with consistent probabilities in expert systems. In *Proc. Int. Joint Conf. on A.I,* pp. 899–901 vol. 2, Milano, August,1987, and in *Int. Journal of Intelligent Systems, vol. 4*, no 1, pp. 1–22, 1989.

D. Driankov: Non-truthfunctional Aspects of Belief/Disbelief Pairs.In R Trappl (ed.) *Cybernetics and Systems,* Kluwer Acad. Publ., pp. 635–642, 1988.

D. Driankov: A Many Valued Logic of Belief: Detachment operators. In *Uncertainty and Intelligent Systems.* B. Bouchon, Lorenzo Saitta, R.R Yager (eds), *Lecture Notes in Computer Science, vol. 313,* Springer-Verlag, pp. 265–272, 1988.

D. Driankov: Towards a Many-Valued Logic of Belief: Detachment operators with an exception condition. In *Methodologies for Intelligent Systems 3,* Z.W Ras, Lorenzo Saitta (eds), North-Holland, pp 418–427, 1988.

D. Driankov: Towards a Many-Valued Logic of Quantified Belief: The information lattice. In *Proc of the 3rd IFSA Congress,* Seattle, Washington, USA, August 6–11, 1989.

D. Driankov, et al.: Plan Recognition for a Fighter Aircraft. In *Proc. of 9th Int. Conf. on Expert Systems of Applications,* Vol. 3: Defense and AI, pp 277–286, Avignon, June, 1989.

P. Eklund, D. Driankov, A Study of Induction Learning and Statistical Methods for Attribute Prediction, to appear in *Proc. of the 3rd International Workshop on AI and Statistics,* Fort Lauderdale, FL, 1991.

P. Eklund, J Kellett: Prospects for Conceptual Graphs in Engineering Interfaces. In *Proc. of the 2nd Int. Conf. on Industrial and Engineering Applications of AI and ES,* Tullahoma, Tennessee, June, 1989.

P. Eklund: Using Path Algebras and Conceptual Graphs to Encode Knowledge Bases. In P*roc. of the 2nd Scandinavian Conf. SCAI,* Tampere, June, 1989.

P. Eklund: Prospects for Conceptual Graphs in Knowledge Acquisition Interfaces. In P*roc. of the 3rd European Knowledge Acquisition Workshop*, Paris, June 3–7, 1989.

P. Eklund: Using Path Algebras to Encode Directed Graph Knowledge-bases. *Proc of the 4th Annual Workshop on Conceptual Graphs,* Detroit, 1989.

P. Eklund: Negotiating Conceptual Structures. *Proc of the 5th Annual Workshop on Conceptual Graphs,* Stockholm 1990.

H. Eriksson, K. Sandahl: Knowledge-based Planning of Experiments in a Biochemical Domain – Membrane-Protein Purification. In *Proc. of the 2nd Scandinavian Conf. on AI*, 1989.

H. Eriksson: Meta-Tool Support for Customized Domain-Oriented Knowledge Acquisition. Accepted for the *Fifth Banff Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff, Nov. 1990.

H. Eriksson: Exploring the Meta-Level – A Supportive Environment for Implementation of Domain-Oriented Knowledge Acquisition Tools. In *Proceedings of AAAI-90 Workshop on Knowledge Acquisition: Practical Tools and Techniques,* Boston, MA, 1990.

J. Fagerström, L. Strömberg: A Paradigm and System for Design and Test of Distributed Applications. In *Proc. of IEEE Compcon Spring 1988. S*an Francisco, CA, February, 1988.

J. Fagerström: Design and test of Distributed Applications. *10th Int. Conf. on Software Engineering,* Raffles City, Singapore, April 11–15, 1988.

J. Fagerström: Enabling Structured Debugging of Distributed Systems. In the *Proc. of the Workshop on Parallel and Distributed Debugging.* Madison, WI, May, 1988.

J. Fagerström, Y. Larsson: Two Contributions on Debugging Distributed Systems. In *Proc. of ACM SIGPLAN/SIGOPS, Workshop on Parallel and Distributed Debugging,* Madison, Wisconsin, May 5–6, 1988.

B. Fjellborg: A General Framework for Extraction of VLSI Pipeline Structures. In *Proc. of the 15th Symposium on Microprocessing and Microprogramming, EUROMICRO'89,* Cologne, September 4–8, 1989.

B. Fjellborg, K. Kuchcinski, M. Patel, Z. Peng: High-Level Synthesis Approach for Architectural Trade-Offs. *Presented at the Second Swedish Workshop on Computer System Architecture, DSA-89,* Bålsta, August 21–23, 1989.

B. Fjellborg: A General Strategy for Pipeline Extraction from High-Level Design Descriptions, *poster-presentation at European Design Automation Conference (EDAC'90),* Glasgow, Scotland, 12–15 March 1990.

B. Fjellborg: A High-Level Synthesis Tool for Exploiting Pipelines in Special-Purpose Systems, *IFIP Working Conference on Logic and Architecture Synthesis,* Paris, 30 May – 1 June 1990.

B. Fjellborg: A Petri Net Model for Pipeline Scheduling, *Sixteenth Symposium on Microprocessing and Microprogramming, EUROMICRO'90,* Short Notes Session, Amsterdam, August 27–30, 1990.

B. Fjellborg: A High-Level Synthesis Tool for Pipeline Extraction. 3rd Swedish Workshop on Computer Systems Architecture, Aspenäs, 6–8 June 1990.

P.-O. Fjällström, J. Katajainen, C. Levcopoulos, O. Petersson: A Sublogarithmic Convex Hull Algorithm, *BIT 30:3,* pp. 378–384, 1990.

P. Fritzson: Incremental Symbol Processing. Presented at the workshop on Compiler Compilers and High Speed Compilation, Berlin, October 1988. In *Lecture Notes in Computer Science, LNCS-371,* Springer Verlag.

P. Fritzson, T. Gyimothy, M. Kamkar, N Shahmehri: Generalized Algorithmic Debugging and Testing. In *SIGPLAN'91, Conference on Programming Language Design and Implementation, Toronto, Ontario, Canada June 24–28, 1991.*

S. Hägglund: Emerging Systems for Computer-Based Knowledge Processing in Office Work. In *Office: Technology and People, 4,* pp 135–145, 1989.

S. Hägglund: The Linköping Approach to Technology Transfer in Knowledge Engineering. In *The Knowledge Engineering Review,* vol. 2, No 3, Cambridge University Press, 1988.

S. Hägglund: Iterative Development and Adaptive Maintenance of Knowledge-Based Systems. In *Office Information Systems: The Design Process*, B. Pernici and A.A Verrijn-Stuart (Eds.), North-Holland, 1989.

S. Hägglund: The impact of intelligent systems on office procedures and knowledge management. Invited paper, the *Int. Conf. on Opportunities and Risks of Artificial Intelligence Systems, ORAIS'89,* Hamburg, 1989.

S. Hägglund, I. Rankin: Investigating the Usability of Expert Critiquing in Knowledge-Based Consultation Systems. In *Proc. of the ECAI Conf.* Muenchen, July, 1988.

T. Hansen: Diagnosing Multiple Fault Using Knowledge about Malfunctioning Behaviour. In *Proc. of the 1st Int. Conf. on Industrial and Engineering Applications of AI and Expert Systems,* Tullahoma, 1988.

J. Harrius: Text Generation in Expert Critiquing Systems Using Rhetorical Structure Theory and Conceptual Graps. *Proc. of the 5th Annual Workshop on Conceptual Graphs,* Stockholm 1990.

R. Hjerppe: HYPERCAT at LIBLAB in Sweden: a progress report. In *The Online Catalogue: Developments and Directions,* Charles R. Hildreth, Ed. The Library Association, 1989, pp. 177–209.

R. Hjerppe: Bibliographic Control and Document Architecture in Hypermedia Databases. In *Annual Review of OCLC Research*, July 1988–June 1989, OCLC, 1989. pp. 6–8.

R. Hjerppe: A Framework for Characterizing systems for Knowledge Organization: a first basis for comparizons and evaluation. In: *Tools for Knowledge Organization and the Human Interface: In Proc. of the First International ISKO-Conference,* Darmstadt, 14–17 Aug., 1990. R. Fugman, Ed. INDEKS Verlag. Frankfurt/Main FRG. 1990 pp. 21–34.

R. Hjerppe: The Role of Classification and Indexes in Hypertext. Issues in implementing Roget's Thesaurus as a Hypertext. In *Tools for Knowledge Organization and the Human Interface: In proc.of the First International ISKO-Conference,* Darmstadt, 14–17 Aug., 1990. R. Fugman, Ed. INDEKS Verlag. Frankfurt/Main FRG. 1990 pp. 206–215.

R. Hjerppe, B. Olander: Cataloging and Expert Systems: AACR2 as a Knowledge Base. In *Journal of the American Society for Information Science 40:27–44*, January 1989.

H. Holmgren, T. Timpka: System development in health care as action research. *Proc. of Medical Informatics in Europe ‘90,* Springer Verlag, Berlin, 1990.

J. Hultman, A. Nyberg, M. Svensson: A Software Architecture for Autonomous Systems. In Proc. of the *6th Int. Symposium on unmanned, untethered submensible Technology.* Ellicott City, Maryland, June 12–14, 1989.

A. Jönsson, N. Dahlbäck: Talking to a Computer is not like Talking to Your Best Friend. In *Proc. of the SCAI,* Tromsö, Norway, 1988.

A. Jönsson, L. Ahrenberg: Extensions of a Descriptor-Based Tagging System into a Tool for the Generation of Unification-Based Grammars.In *Readings in Humanities Computing, vol. 1.*

A. Jönsson: Application-Dependent Discourse Management for Natural Language Interfaces: An Empirical Investigation. J. Pind and E.Rögnvaldsen (eds.) *Papers from the 7th Scand. Conf. of Computational Linguistics,* pp 297–307.

A. Jönsson: A Dialogue Manager Using Initiative Response Units and Distributed Control. *5th European ACL Conference*, Berlin 10–12 April.

M. Kamkar, N. Shahmehri, P. Fritzson: Affect-Chaining and Dependency Oriented Flow Analysis Applied to Queries of Program. In *Proceedings of the ACM Symposium on Personal and Small Computers.* Cannes, France, May 4–6,1988.

M. Kamkar, N. Shahmehri, P. Fritzson: Bug Localization by Algorithmic Debugging and Program Slicing. In *Proc. of the Int. Conf. on Programming Language Implementation and Logic Programming,* 1990, LNCS 456, Springer-Verlag, 1990.

R. G. Karlsson: Greedy matching on a grid. In *BIT,* 1988.

R. G. Karlsson, M. Overmars: Normalized Divide-and-Conquer: A Scaling Technique for Solving Multi-dimensional Problems. In *Information Processing Letters,* 26, pp. 307–312, 1988.

R. G Karlsson, M. Overmars: Scanline Algorithms on a Grid. In *BIT,* 1988.

J. Katajainen, C. Levcopoulos, O. Petersson: Local Insertion Sort Revisited. In *Proc. of the Int. Symposium on Optimal Algorithms,* Varna, Bulgaria, LNCS, Springer-Verlag, 1989.

P. Kroha, P. Fritzson: A Compiler with Scheduling for a Specialized Synchronous Multiprocessor System. In *Proc. of the 3rd International Workshop on Compiler Compilers,* Schwerin, GDR, October 22–26 1990. Proc in the *LNCS* series by Springer-Verlag, 1991.

K. Kuchcinski, B. Wiszniewski: Path Analysis of Distributed Programs. In *Proc. of the 16th ACM Computer Science Conf.,* Atlanta, Georgia, February, 23–25 1988.

K. Kuchcinski, Z. Peng: Parallelism Extraction from Sequential Programs for VLSI Applications. In *Euromicro Journal, Microprocessing and Microprogramming, Vol.23, Nos 1–5,* pp 87–92, 1988.

K. Kuchcinski, Z. Peng: Testability Analysis in a VLSI High-Level Synthesis System. In *Proc. of the 15th Symposium on Microprocessing and Microprogramming, EUROMICRO’89,* Cologne, September 4–8, 1989.

K. Kuchcinski: High-Level Design Methodology for Complex VLSI Systems, *Proc. NORSILC/NORCHIP seminar’89,* October 23–24, 1989.

K. Kuchcinski, Z. Peng, B. Fjellborg: Parallelism Exploration for Automatic Design of VLSI Systems. *1st Nordic Workshop on Parallel Algorithms and Architectures in Vision and Image Processing*, Linköping, March 7–9, 1990.

K. Kuchcinski, Z. Peng: Towards Design for Testability in VLSI High-Level Synthesis Systems, *NORSILC-2/EKF seminar on VLSI testing,* Oslo, 19–20 March, 1990.

P. Kuczora, P. Eklund: A Conceptual Graph Implementation for Knowledge Engineering. In: *“Human and Organisational Issues in Expert Systems”,* A. Hart and D. Berry (Eds), Chapman and Hall, 1990.

M. Larsson: Mixed Static and Dynamic Reasoning in a Formal System, *Sixteenth Symposium on Microprocessing and Microprogramming, EUROMICRO’90,* Short Notes Session, Amsterdam, August 27–30, 1990.

T. Larsson: Hardware Verification based on Algebraic Manipulation and Partial Evaluation. In *Proc of IFIP WG 10.2 International Working Conference on The Fusion of Hardware Design and Verification,* Scotland, July 3–6, 1988.

T. Larsson: Mixed Functional and Temporal Hardware Verification. In *Proc. of the IFIP International Workshop on Applied Formal Methods For Correct VLSI Design,* Leuven, Belgium, 1989.

T. Larsson, Z. Peng: A Relational Approach to VLSI Design Databases. In *Proc. of the 36th ISMM Int. Conf. on Mini and Microcomputers and Their Applications,* Sant Feliu de Guixols, Spain, June 27–30, 1988.

Y. Larsson: A Testbed Environment for Debugging Distributed Systems. To appear in the *Proc. of the Workshop on Parallel and Distributed Debugging.* Madison, WI, May, 1988.

T. Larsson: The Symbolic Manipulation of Event Traces. To appear in *Proc. of the ACM/IFIP 1991 International Workshop on Formal Methods in VLSI Design,* Miami, 9–11 Jan, 1991.

B. Lennartsson: Multi-Level Software Architectures – Industrial Experiences and Research Directions. In *Advance Working Papers of The Second Int. Workshop on Computer-Aided Software Engineering.* Cambridge, Massachusetts, USA, July 12–15, 1988.

J. Leszczylowski, S. Bonnier, J. Maluszynski: Logic Programming with External Procedures: Introducing S-Unification. In *Information Processing Letters 28, N*orth Holland, 1988.

C. Levcopoulos, A. Lingas, J. Sack: Heuristics for Optimal Search Trees with Geometric Generalizations. In *Theoretical Computer Science 66,* 1989, pp. 181–203. (A preliminary version in Proc. *of 14th Colloquium on Automata, Languages and Programming, Karlsruhe (1987), Lecture Notes in Computer Science 267,* Springer-Verlag).

C. Levcopoulos, M. H Overmars: A Balanced Search Tree with 0(1) Worst-Case Update Time. To appear in *Acta Informatica 26,* 1988.

C. Levcopoulos, O. Petersson: An Optimal Parallel Algorithm for Sorting Presorted Files. In *Proc. of the 8th Conf. on Foundations of Software Technology and Theoretical Computer Science*, Pune, India, December 21–23, 1988.

C. Levcopoulos, O. Petersson: Heapsort-Adapted for Presorted Files. In *Proc. of 1989 workshop on Data Structures and Algorithms,* Ottawa, August 17–19, 1989, LNC, Springer-Verlag.

C. Levcopoulos, J. Katajainen, A. Lingas: An Optimal Expected-Time Parallel Algorithm for Voronoi Diagrams. In Proc. *1st Scandinavian Workshop on Algorithm Theory,* Halmstad, Sweden, July 1988, Lecture Notes in Computer Scienxce 318, pp. 190–198, Springer-Verlag

C. Levcopoulos, A. Lingas: There are Planar Graphs Almost as Good as the Complete Graphs and as Short as Minimum Spanning Trees. In Proc. *Int. Symposium on Optimal Algorithms,* Varna, Bulgaria, 1989. Lecture Notes in Computer Science 401, pp. 9–13, Springer-Verlag

A. Lingas, M. Syslo: A Polynomial Algorithm for Subgraph Isomorphism of Two-Connected Series-Parallel Graphs. In *Proc. 15th Colloquium on Automata Languages and Programming,* Tampere (1988). Lecture Notes in Computer Science 317, pp. 394–409, Springer Verlag.

A. Lingas: Voronoi Diagrams with Barries and the Shortest Diagonal Problem. In *Information Processing Letters 32(1989)*, pp. 191–198.

A. Lingas, M. Karpinski: Subtree Isomorphism is NC reducible to Bipartite Perfect Matching. Accepted for publication in *Information Processing Letters*, May, 1988.

A. Lingas: Subgraph Isomorphism for Connected Graphs of Bounded Valence and Bounded Separator is in NC. In *Proc. of the 1988 Int. Conf. on Parallel Processing,* Chicago, USA, August, 1988.

A. Lingas: Greedy Triangulation Can Be Efficiently Implemented in the Average Case. In *Proc. of the 14th Int. Workshop on Graph-Theoretic Concepts in Computer Science,* Amsterdam, June, 1988. Lecture Notes in Computer Science 344, pp. 253–261, Springer Verlag.

J. Löwgren: History, State and Future of User Interface Management Systems. In *SIGCHI Bulletin, vol. 20, No 1,* 1988.

J. Löwgren: An Architecture for Expert System User Interface Design and Management. In *Proc of the Symposium On User Interface Software and Technologies. W*illiamsburg, Virginia, November 13–15, 1989.

J. Löwgren, T. Nordqvist: A Knowledge-Based Tool for User Interface Evaluation and its Integration in a UIMS. In *Proceedings of Human-Computer Interaction – Interact'90,* Cambridge, UK, August 27–31 1990. Amsterdam: Elsevier.

J. Malec: Dynamic Scene Description Systems – ODYS, In *Proc. of the 11th IFAC World Congress*, Tallin, Estonia, Aug. 13–17, 1990.

J. Maleki: Towards a system for temporal reasoning. In *Proc. of the Scand Conf. on A.I*, Tampere, Finland, 1989.

J. Maluszynski, T. Näslund: Fail Substitutions for Negation as Failure. In *E. Lusk and R. Overbeek, editors, Proc. of the North American Conference on Logic Programming,* pages 461–476. MIT Press, 1989.

N. Mat, L. Padgham, P. Åberg, R. Rönnquist, M. Jansson, T. Teitenberg: User Interface and Information Management for an Object-Oriented Database, *Techno-Data '89*, Berlin 1989.

M. Merkel: Temporal Structure in Discourse. In *Proc. of the 7th Int. Conf. of Nordic and General Linguistics.*Thorshavn, August 7–11, 1989.

M. Merkel: A Novel Analysis of Temporal Frame Adverbials. In P*roc. of the 12th International Conf. on Computational Linguistics* in Budapest, August 22–26, 1988.

S. Moen: Drawing Dynamic Trees. In *IEEE Software, vol. 7, No 4*, July 1990.

M. Morin, E. Sandewall: A Software Architecture supporting Integration of Sub-systems. In *Prometheus: Proc. of the fourth Workshop*, 1990.

S. Nadjm-Tehrani: Debugging Prolog Programs Declaratively. In *M. Bruynooghe, editor, Proc. of the Second Workshop on Meta-Programming in Logic*, pages 137–155, April 1990.

S. Nadjm-Tehrani: Driver interactions with a signal support system. In *Prometheus: Proc of the Third Workshop,* 1990.

U. Nilsson: Towards A Framework for The Abstract Interpretation of Logic Programs. In *Proc. of PLILP'88,* LNCS 348, Springer-Verlag. Orleans, France, May 16–18, 1989.

U. Nilsson: Systematic Semantic Approximations of Logic Programs. In *P. Deransart and J. Maluszynski, editors, Programming Language Implementation and Logic Programming, International Workshop PLILP '90,* pages 293–306. Springer-Verlag LNCS 456, 1990.

U. Nilsson, J. Maluszynski: Logic, *Programming and Prolog*, John Wiley & Sons, 1990.

L. Padgham: A Model and Representation for Type Information and its Use in Reasoning with Defaults, In *Proc. AAAI-88*, St. Paul, USA, 1988.

L. Padgham: NODE: A Database for Use by Intelligent Systems. In *Proc. of ISMIS '88*, Torino, Italy, October 12–15, 1988.

L. Padgham: A Lattice Based Model for Inheritance Reasoning. In *Proc. of the Workshop on Inheritance Hierarchies in Knowledge Representation on Programming Languages.* Viareggio, Italy, February 6–8, 1989. Revised version in *Inheritance Hierarchies in Knowledge Representation and Programming Languages*, John Wiley and Sons, England.

L. Padgham: Negative Reasoning Using Inheritance, *Proc. of IJCAI '89,* Aug. 20–25, 1989, Detroit, Michigan, USA. vol.2 pp.. 1086.

L. Padgham: Defeasible Inheritance: A Lattice Based Approach, *Computers and Mathematics with Applications,* special issue on Semantic Nets.1991.

M. R.K. Patel: What should a Programming Environment for Threaded Interpretive Languages provide? *In Proc. of the 8th Rochester 4th Conf. on Programming Environments*, Rochester, New York, USA, June, 1988.

M.R.K Patel: TAO: A Hierarchical Design Representation for High-Level Synthesis of Hardware Systems. In *Proc. of the 15th Symposium on Microprocessing and Microprogramming, EUROMICRO'89,* Cologne, September 4–8, 1989.

M.R.K Patel: Selecting Design Representation for High level Synthesis of Hardware Systems. In *Proc. of the 4th Int. Workshop on High-Level Synthesis,* Kennebunkport, Maine, USA, October 15–18, 1989

M.R.K Patel: A Design Representation for High Level Synthesis, *Proc. European Design Automation Conference (EDAC'90),* Glasgow, Scotland, 12–15 March 1990.

M.R.K Patel: Random Logic Circuit Implementation of Extended Timed Petri Nets, *Microprocessing and Microprogramming, The Euromicro Journal*, vol. 30, no. 1–5, August 1990, pp. 313–319.

Z. Peng: Semantics of a Parallel Computation Model and its Applications in Digital Hardware Design.In *Proc. of the 1988 Int. Conf. on Parallel Processing,* Pennsylvania State University, August 15–19, 1988.

Z. Peng: Let's Design Asynchronous VLSI Systems.In *Microprocessing and Microprogramming, the Euromicro Journal, Vol. 24, 1988.*

Z. Peng: A Horizontal Optimization Algorithm for Data Path/Control Synthesis. In *Proc. of the 1988 IEEE Int. Symposium on Circuits and Systems,* Helsinki Univ. of Technology, Finland, June 7–9, 1988.

Z. Peng, K. Kuchcinski: A Multi-Level Optimization Strategy for Behavior-Entry Synthesis. In *Proc. of the Fourth Int. Workshop on High-Level Synthesis,* Kennebunkport, Maine, USA, October 15–18, 1989

Z. Peng, K. Kuchcinski, B. Lyles: CAMAD: A Unified Data Path/Control Synthesis Environment. In *Proc. of the IFIP TC-10 Conf. on Design Methodologies for VLSI and Computer Architecture*, Pisa, Italy, September 19–23, 1988. Also in D.A Edwards (Editor), *Design Methodologies for VLSI and Computer Architecture,* North-Holland, pp 53–67, 1989.

Z. Peng: Design of Clocking Schemes in High-Level Synthesis, *Sixteenth Symposium on Microprocessing and Microprogramming, EUROMICRO'90,* Short Notes Session, Amsterdam, August 27–30, 1990.

T. Persson, L. Staflin: A causation theory for a logic of continuous change. In *Proc. European Conf. on AI. (ECAI),* 1990.

M. Pettersson: Generating Efficient Code from Continuation Semantics. In *Proc. of the 3rd International Workshop on Compiler Compilers,* Schwerin, GDR, October 22–26 1990. To appear in LNCS, Springer-Verlag, 1991.

I. Rankin: Deep Generation of a Critique. Extended abstract in *Proc of the Second European Workshop on Natural Language Generation,* Edinburgh, April, 1989.

I. Rankin, S. Hägglund, Y. Waern: Generating User-Centered Explanations in a Critiquing Context. In *Proc. of the AAAI-88 Workshop on Explanation.* St. Paul, Minnesota, August, 1988.

I. Rankin: Rhetorical Structure Theory in Text Generation. In *Proceedings of the Second Nordic Conf on Text Comprehension in Man and Machine* Täby, Sweden, April 1990.

M. Reinfrank, O. Dressler, G. Brewka: On the Relation between Truth Maintenance and Autoepistemic Logic. In Proc. of the *11th Int. Joint Conf. on Artificial Intelligence*, Detroit, Michigan, USA, August 20–25, 1989.

L. Reshagen, J-O Hildén, B Lindström, K. Sandahl, O Wigertz: An Evaluation of an Antibody Analysis Advisor in a Laboratory Environment. The EXODUS Project. In *Proc. of the MIE 88,* August 17–20, 1988.

R. Rönnquist: Reasoning about Control of Dynamic Systems. In *Proc.* of the *International Symposium on Methodologies for Intelligent Systems (ISMIS*), Charlotte, USA, 1989.

R. Rönnquist: A Logic for Propagation Based Characterisation of Process Behaviour. In *Proc.* of the *International Symposium on Methodologies for Intelligent Systems (ISMIS*), October 24–27, 1990.

K. Sandahl, H. Eriksson, T. Padron-McCarthy, T. Sokolnicki, B. Österlund: Meeting the Requirements of Knowledge Management. Accepted for publication in *Expert Systems with Applications.*

K. Sandahl, A. Brogren, L. Reshagen: Transferring Knowledge from an Active Expert to an End-User Environment. *Accepted for the Fifth Banff Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff, Nov., 1990.

E. Sandewall: The semantics of Non-Monotonic Etailment Defined Using Partial Interpretations. In Michael Reinfrank et al., editor, *Non-Monotonic Reasoning,* pp. 27–41, Springer-Verlag, 1988.

E. Sandewall: An Approach to Non-Monotonic Entailment. In *Methodologies for Intelligent Systems,* vol. 3, *Proc. of ISMIS 1988 conf.,* Torino, Italy, 1988, Z.W. Ras and L. Saitta (eds), *North Holland*, pp. 391–397, 1988.

E. Sandewall: Future Developments in Artificial Intelligence. In *Proc. of European Conf. on Artificial Intelligence (ECAI),* Munich, August, 1988. (Invited paper)

E. Sandewall: Artificial Intelligence Techniques for Complex Information Processing in Vehicles. In *Proc. XXXVI Convegno Internazionale delle Comunicazioni,* Genoa, pp. 431–437, 1988.

E. Sandewall: Combining Logic and Differential Equations for Describing Real-World Systems. In *Proc. of the Int. Conf. on Knowledge Representation,* Toronto, Canada, 1989.

E. Sandewall: Filter Preferential Entailment for the Logic of Action in Almost Continuous Words. In *Proc. of the Int. Joint Conf. on Artificial Intelligence,* Detroit, USA, 1989.

E. Sandewall: A Decision Procedure for a Theory of Actions and Plan. In Z. Ras et al, editor, *Methodologies for Intelligent Systems,* IV, 1989.

E. Sandewall: LISP as a Very High Level Implementation Language. In *Proc. First European Conf. on the Practical Applications of LISP (EuroPAL),* 1990.

E. Sandewall: Towards a logic of dynamic frames. In Ken Ford and Patrick Hayes, editors, *The Frame Problem in Artificial Intelligence,* JAI Press, 1990

E. Sandewall: Reasoning about the world as perceived by an agent. In *Proc. European Conf. on A.I.*(ECAI), 1990.

E. Sandewall, Hamm, de Charentenay, Höfflinger, Reister, E. Sandewall, Williams: Improved driver information and driver assistance by PROMETHEUS Systems. In *Proc. FISITA Congress*, Torino, 1990.

N. Shahmehri, P. Fritzson: Algorithmic Debugging for Imperative Languages with Side-effects. In P*roc. of the 3rd International Workshop on Compiler Compilers,* Schwerin, Germany, October 22–24, 1991. The proceedings will be published by Springer-Verlag in the LNCS series.

N. Shahmehri, M. Kamkar, P. Fritzson: Semi-Automatic Bug Localization in Software Maintenance, Accepted to the *IEEE Conf. on Software Maintenance,* San Diego, Nov.,1990.

N. Shahmehri, M. Kamkar, P Fritzson: Semi-Automatic Bug Localizationin Software Maintenance. In *Proc. of the IEEE Conf. on Software Maintenance,* San Diego Nov. 1990.

H. Shu: An Expert System Mechanical Engineering Design System. In *Proc. of the FEMCAD-88 Conf.* Paris,1988.

H. Shu: Planning with reactivity. In *Proc. Pacific Rim International Conf. on Artificial Intelligence*, 1990.

H. Shu: The Architecture of a Reactive Car Support System. In *Proc. of the 4th PROMETHEUS workshop,* Compiègue, Oct. 1990.

H. Shu: Hierarchical planning using action taxonomies. In Z. Ras, editor, *Methodologies for Intelligent Systems, V.* 1990.

L. Smith: 'Wholly New Forms of Encyclopedias': Electronic Knowledge in the Form of Hypertext. In: *Information * Knowledge * Evolution: Proceedings of the forty-fourth FID Congress,* Helsinki, Finland, 28 August – 1 September 1988, Sinikka Koskiala and Ritva Launo, Eds. North Holland, 1989, pp. 245–250.

T. Sokolnicki: Integrating Tutoring in Knowledge-Based Systems. In *Proc. of the SCAI 89, Workshop on AI Solutions in Education,* Tampere, June, 1989.

T. Sokolnicki: Towards Knowledge-Based Tutors: A Survey and Appraisal of Intelligent Tutoring Systems. *Journal Knowledge Engineering Review,* vol. 6.2 or 6.3.

E. Tengvald: Advances in Intelligent Robotics Systems. In *Proc. of the SPIE Cambridge Symposium on Optical and Optoelectronic Engineering,* Cambridge, USA, November, 1988.

E. Tengvald, E. Jungert: Problem-Dependent Transformations of 3rd Terrain Elevation Maps into Cost Precursor Maps. In *Proc. of the SPIE Conf. on Advances in Intelligence Robotic Systems,* Cambridge, MA, November 6–11, 1988.

T. Timpka, P. Bjurulf: The Semantics of Diagnosis and Management of Genitourinary Infections: A cross-specialty study. To appear in *Family Practice* Vol. 6:1989.

T. Timpka, E. Arborelius: Study of the Practitioner's Knowledge Need and Use During Health Care Consultations. Part 1: Theory and method. In *Proc. of MedInfo' 89,* North-Holland, Amsterdam, 1989.

T. Timpka, M. Ekström, P. Bjurulf: Information Needs and Information Seeking Behaviour in Primary Health Care. *Scandinavian Journal of Primary Health Care* 1989;7: 105–9.

T. Timpka: Introducing Hypertext in Primary Health Care: A Study on the Teasibility of Decision Support for Practitioner. *Computer Methods and Programs in Biomedicine* 1989;29:1–13.

T. Timpka, R. Hjerppe, J. Zimmer, M. Ekström: Introducing Hypertext in Primary Care: Supporting the Doctor-Patient Relationship. In *Proc. of RIAO-88,* Cambridge, 1988.

T. Timpka, L. Padgham, P Hedblom, S Wallin, G Tibblin: A Hypertext Knowledge Base for Primary Care: LIMEDS in LINCKS. In *Proc. of the SIGIR'89*, Boston, ACM Publ., New York, 1989.

T. Timpka, P. Bjurulf, T. Buur: Audit of Decision-Making Regarding Female Genitourinary Infections in Out-Patient Practice. *Scandinavian Journal of Infectious Disease 1989*;22: 49–57.

T. Timpka, E. Arborelius: The GP's Dilemmas: A Study of Knowledge Need and Use During Health Care Consultations. *Methods of Information in Medicine* 1990;29:23–29.

T. Timpka, T. Buur: Inter-observer variation in decision-making regarding patients on chronic hemodialysis. *Methods of Information in Medicine* 1990;29: 153–7.

T. Timpka, P. Hedblom, H. Holmgren: Using an object-oriented environment for group process development of medical software. *Proc. of Software Engineering in Medical Informatics ‘90*, Rotterdam, 1990.

T. Timpka, P. Hedblom, G. Tibblin: A hypermedia document collection for primary health care: Why, What, and How. *Artificial Intelligence in Medicine* 1990;2: 179–92.

T. Timpka, E. Arborelius: The primary health care nurse's dilemmas: a study of knowledge use and need during telephone consultations. *Journal of Advanced Nursing* 1990;15: 1457–65.

R. Tjärnström, T Larsson: Automatic generation of timing specifications for CMOS VLSI designs. Accepted for presentation at the IFIP 10.2–10.5 workshop on Applied formal Methods for correct VLSI designs, 13–16 November 1989, Leuven, Belgium.

A. Törne: The Instruction and Control of Multi-Machine Environments. In *Proc. Applications of Artificial Intelligence in Engineering V,* Springer-Verlag, Boston, 1990.

R. Ubar, K. Kuchcinski, Z. Peng: Test Generation for Digital Systems at Functional Level. In *Proc. European Test Conference*, Munich, April 10–12, 1991.

A. Vainio-Larsson: Hypermedia and Human-Computer Interaction. In *Proc. of the 3d IFAC/IEA Conf. on Man-Machine Systems Analysis, Design and Evaluation,* Oulo, Finland, June -88.

A. Vainio-Larsson, R. Orring: Evaluating the Usability of User Interfaces: Research in Practice, November 1989, 18p. Presented at INTERACT 90, The 3rd IFIP Conference on Human-Computer Interaction, Cambridge, England, August 1990.

Y. Waern, S. Hägglund, I. Rankin: Expert Critiquing Systems – psychological and computational aspects of generating comments. In P*reprints of ECCE Conf.* Cambridge, England, September, 1988.

Y. Waern, L. Bannon, T. Timpka, W. Schneider: User participation in HCI research Effects on processes and results. *Proc. of Interact ‘90,* Elsevier, Amsterdam, 1990.

M. Wedlin: Interface between UNIX and Hypercard. Accepted for the EUUG Spring Conference 90, Munich, FRG, April 23–27, 1990.

M. Wirén: A Control-Strategy-Independent Implementation of PATR. *In Proc. of the 1st Scandinavian Conf. on Artificial Intelligence*, Tromsö, Norway, March 9–11, 1988.

M. Wirén: Incremental Parsing and Reason Maintenance. In *Papers presented to the 13th International Conference on Computational Linguistics,* Helsinki 20–25 August, 1990, Vol. 3, pp. 287–293.

M. Wirén: Interactive Incremental Chart Parsing. In *Proc of the 4th Conf. of the European Chapter of the Association for Computational Linguistics, pp 241–248,* Manchester, April 10–12, 1989.

T. Zhang, L. Padgham: A Diagnosis System Using Classification. In *Proc of the 5th International Symposium on Methodologies for Intelligent Systems*, Knoxville, Tennessee, 1990.

P. Österling: A description of traffic scenes – definition and realization. In *Prometheus: Proc. of the Third Workshop*, 1990.

E.4 Departmental Reports 1989–90

Research Reports

LiTH-IDA-R-89-01 **Rolf G. Karlsson:** Traversing a Maze with a Robot Arm.

LiTH-IDA-R-89-02 **Mariam Kamkar, Nahid Shahmehri, Peter Fritzson:** Affect-Chaining and Dependency Oriented Flow Analysis Applied to Queries of Programs.

LiTH-IDA-R-89-03 **Jan Maluszynski, Torbjörn Näslund:** Fail Substitutions for Negation as Failure. Also in *Proc. of the North American Conference on Logic Programming 89*, Cleveland, Ohio, USA.

LiTH-IDA-R-89-04 **Andrzej Lingas:** Voronoi Diagrams with Barriers and the Shortest Diagonal Problem.

LiTH-IDA-R-89-05 **Andrzej Lingas, Maciej M. Syslo:** A Polynomial-Time Algorithm for Subgraph Isomorphism of Two-Connected Series-Parallel Graphs. Also in *Proc. of the 15th International Qolloquium on Automata, Languages and Programming,* Tampere, Finland, July 1988.

LiTH-IDA-R-89-06 **Andrzej Lingas:** Greedy Triangulation can be Efficiently Implemented in the Average Case. Also in *Proc. of the 14th International Workshop on Graph-Theoretic Concepts in Computer Science,* Amsterdam, The Netherlands, June 1988.

LiTH-IDA-R-89-07 **Andrzej Lingas:** Subgraph Isomorphism for Connected Graphs of Bounded Valence and Bounded Separator is in NC. Also in *Proc. of 1988 International Confernece on Parallel Processing,* Chicago, USA, August 1988.

LiTH-IDA-R-89-08 **Christos Levcopoulos, Jyrki Katajainen, Andrzej Lingas:** An Optimal Expected-Time Parallel Algorithm for Voronoi Diagrams. Also in *Proc. of the 1st Scandinavian Workshop on Algorithm Theory,* Halmstad, Sweden, July 1988.

LiTH-IDA-R-89-09 **Hristo N. Djidjev, Andrzej Lingas, Jörg-R. Sack:** An O(n log n) Algorithm for Computing a Link Center in a Simple Polygon. Also in *Proc. of the 6th Symposium on Theoretical Aspects of Computer Science,* Paderborn, Germany, February 1989.

LiTH-IDA-R-89-10 **Elias Dahlhaus, Marek Karpinski, Andrzej Lingas:** A Parallel Algorithm for Maximum Matching in Planar Graphs.

LiTH-IDA-R-89-11 **Andrzej Lingas:** An Efficient Parallel Algorithm for Planar Directed Reachability.

LiTH-IDA-R-89-12 **Rober Bilos, Peter Fritzson:** Experience from a Token Sequence Representation of Programs, Documents, and their Deltas. Also presented at the *International Workshop on Software Version and Configuration Control*, Grassau, FRG, January 27-29, 1988

LiTH-IDA-R-89-13 **Øystein Santi:** Retargeting of an Incremental Code Generator to MC68020.

LiTH-IDA-R-89-16 **Christos Levcopoulos, Ola Petersson:** An Optimal Parallel Algorithm for Sorting Presorted Files. A previous version presented at *the 8th Conference on Foundations of Software Technology and Theoretical Computer Science,* Poona, India, Dec. 1988.

LiTH-IDA-R-89-17 **Christos Levcopoulos, Ola Petersson:** A Note on Adaptive Parallel Sorting.

LiTH-IDA-R-89-18 **Rober Bilos:** A Survey of Visibility Control Models.

LiTH-IDA-R-89-20 **Jyrki Katajainen, Christos Levcopoulos, Ola Petersson:** Local Insertion Sort Revisited. Also presented at the *International Symposium on Optimal Algorithms*, Varna, Bulgaria, May 29 – June 2, 1989.

LiTH-IDA-R-89-21 **Christos Levcopoulos, Ola Petersson:** Heapsort – Adapted for Presorted Files. Also presented at the *1989 Workshop on Data Structures and Algorithms,* Ottawa, Canada, May 17-19, 1989.

LiTH-IDA-R-89-22 **Lars Ahrenberg:** A Constraint-Based Model for Natural-Language Understanding and a Pilot Implementation.

LiTH-IDA-R-89-23 **Lars Ahrenberg:** On the Integration of Linguistic Knowledge and World Knowledge in Natural Language Understanding. Also in *Proc. of the Nordic Conference of Text Comprehension in Man and Machine*, Sigtuna, Sweden, October 27-28, 1989.

LiTH-IDA-R-89-24 **Mats Wirén:** Interactive Incremental Chart Parsing. Also in *Proc. of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, England, April 10-12, 1989.

LiTH-IDA-R-89-25 **Petr Kroha:** An Extension of the Single Instruction Machine Idea.

LiTH-IDA-R-89-26 **Petr Kroha, Peter Fritzson:** A Compiler with Scheduling for a Specialized Synchronous Multiprocessor System.

LiTH-IDA-R-89-27 **Peter Fritzson, Petr Kroha:** An Object-Oriented Database Approach to the Symbol Processing in an Incremental Compiler.

LiTH-IDA-R-89-28 **Wlodzimierz Drabent, Maurizio Martelli:** Strict Completion of Logic Programs.

LiTH-IDA-R-89-29 **Per-Olof Fjällström, Jyrki Katajainen, Christos Levcopoulos, Ola Petersson:** A Sublogarithmic Parallel Algorithm for Finding the Convex Hull of Sorted Points. Also in *BIT 30(1990), 378-384.*

LiTH-IDA-R-89-30 **Per-Olof Fjällström:** Approximation of Polygonal Lines.

LiTH-IDA-R-89-32 **Jonas Löwgren, Tommy Nordqvist, Staffan Löf:** Knowledge-Based Support for User Interface Evaluation in User Interface Management Systems.

LiTH-IDA-R-89-33 **Nils Dahlbäck, Arne Jönsson:** Empirical Studies of Discourse Representations for Natural Language Interfaces. Also in *Proc. of the 4th Conf. of the European Chapter of the Association for Computational Linguistics*, Manchester, April, 1989.

LiTH-IDA-R-89-34 **Arne Jönsson:** Application-Dependent Discourse Management for Natural Language Interfaces: An Empirical Investigation. Also in *Papers from the Seventh Scandinavian Conference of Computational Linguistics* (*Nordiska Datalingvistikdagarna*), Reykjavik, Iceland, April, 1989.

LiTH-IDA-R-89-35 **Arne Jönsson, Lars Ahrenberg:** Extensions of a Descriptor-Based Tagging System into a Tool for the Generation of Unification-Based Grammars. Paper presented at the *16th International ALLC Conference and 9th ICCH Conference,* Toronto, Canada, June, 1989.

LiTH-IDA-R-89-36 **Björn Fjellborg:** A General Framework for Extraction of VLSI Pipeline Structures. Also in *Proc. of the 15th Symposium on Microprocessing and Microprogramming, EUROMICRO'89, Short Notes Session*, Cologne, West Germany, September 4-8, 1989.

LiTH-IDA-R-89-37 **Erik Sandewall:** Filter Preferential Entailment for the Logic of Action in Almost Continuous Worlds. Also in *Proc. of the International Joint Conference on Artificial Intelligence*, Detroit, USA, 1989.

LiTH-IDA-R-89-38 **Erik Sandewall:** Combining Logic and Differential Equations for Describing Real-World Systems. Also in *Proc. of the Conf. on Representation and Reasoning about Knowledge,* Toronto, Canada, 1989.

LiTH-IDA-R-89-39 **Erik Sandewall:** A Decision Procedure for A Theory of Actions and Plans. Also published in *Z. Ras et al. (eds), Methodologies for Intelligent Systems*, 1989.

LiTH-IDA-R-89-40 **Johan Hultman, Anders Nyberg, Mikael Svensson:** A Software Architecture for Autonomous Systems. Also presented at the *First PROMETHEUS Scientific Workshop*, Wolfsburg, 1989 and the *Sixth International Symposium on Unmanned Untethered Submersible Technology,* Ellicott City, Maryland, USA, 1989.

LiTH-IDA-R-89-41 **Zebo Peng, Krzysztof Kuchcinski:** A Multi-Level Optimization Strategy for Behavior-Entry Synthesis. Also presented at the *Fourth International Workshop on High-Level Synthesis,* Kennebunkport, Maine, USA, October 15-18, 1989.

LiTH-IDA-R-89-42 **Krzysztof Kuchcinski, Zebo Peng:** Testability Analysis in a VLSI High-Level Synthesis System. Also presented at the*15th Symposium on Microprocessing and Microprogramming, EUROMICRO'89*, Cologne, West Germany, September 4-8, 1989 and will be published in *Microprocessing and Microprogramming, The EUROMICRO Journal*, 1990.

LiTH-IDA-R-89-44 **Olof Johansson:** An Experiment with a Neural Network for Handwritten Character Recognition.

LiTH-IDA-R-89-45 **Erik Sandewall et al.:** Department of Computer and Information Science Annual Research Report 1988.

LiTH-IDA-R-89-46 **Lars Ahrenberg:** A Formal Field Grammar.

LiTH-IDA-R-89-47 **Jyrki Katajainen, Erkki Mäkinen:** Tree Compression and Optimization with Applications.

LiTH-IDA-R-89-48 **Krister Joas:** Interprocess Communication Primitives for a Distributed Debugger under Unix.

LiTH-IDA-R-89-49 **Nahid Shahmehri, Peter Fritzson:** Algorithmic Debugging for Imperatives for Languages with Side-effects. Also presented at *The 3rd International Workshop on Compiler Compilers,* Schwerin, Germany. October 22-24, 1990. The proceedings will be published by Springer Verlag in the LNCS series.

LiTH-IDA-R-89-50 **Jonas Löwgren:** An Architecture for Expert System User Interface Design and Management. Aslo in *Proc. of ACM SIGGRAPH Symposium on User Interface Software Technology (VIST'89),* Williamsburg, Virginia, November 13-15, 1989.

LiTH-IDA-R-89-52 **Mikael Pettersson:** Generating Interpreters from Denotational Definitions using C++ as a Meta-language.

LiTH-IDA-R-89-53 **Tony Larsson:** Mixed Functional and Temporal Hardware Verification. Also in *Proc. of Applied FormalMethods For Correct VLSI Design*, 13-16 November 1989, Leuven, Belgium.

LiTH-IDA-R-90-01 **Nils Dahlbäck:** A Symbol Is Not a Symbol. Also in *Proc. of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89),* Detroit, Michigan, USA, August 20-25, 1989.

LiTH-IDA-R-90-03 **Mikael Wedlin:** Interface between UNIX and Hypercard. Also presented at *EUUG Spring Conference '90*, Munich, FRG, April 23-27, 1990.

LiTH-IDA-R-90-04 **Mariam Kamkar, Peter Fritzson:** Focusing the Bug Localization Process in Algorithmic Debugging through Program Slicing. Also in *Proc. of PLILP'90*, Linköping, Augusti 1990, *LNCS 456*, Springer-Verlag 1990.

LiTH-IDA-R-90-05 **Mikael Pettersson:** Generating Efficient Code from Continuation Semantics. Presented at the *3rd International Workshop on Compiler Compilers,* Schwerin, GDR, October 22-26, 1990. Proc. will appear in the *LNCS* series by Springer Verlag 1991.

LiTH-IDA-R-90-06 **Raimund Ubar, Krzysztof Kuchcinski, Zebo Peng:** Test Generation for Digital Systems at Functional Level. Also presented at the *European Test Conference,* Munich, April 10-12, 1991.

LiTH-IDA-R-90-07 **Anders Törne:** The Instruction and Control of Multi-Machine Environments.Also in *Proc. of 5th Int'l Conf on Applications of Artificial Intelligence in Engineering, vol 2,* Springer Verlag.

LiTH-IDA-R-90-08 **Göran Goldkuhl:** Förändringsanalys på vårdcentral – erfarenheter från en organisationsutveckling.

LiTH-IDA-R-90-09 **Patrick Doherty:** A Correspondence between Inheritance Hierarchies and a Logic of Preferential Entailment. Also in *Proc. of 4th International Symposium on Methodologies for Intelligent Systems*, Charlotte, North Carolina, October 1989. Also published in *Z. Ras et al. (eds), Methodologies for Intelligent Systems*, 1989.

LiTH-IDA-R-90-10 **Petr Jirku:** Defeasible Reasoning.

LiTH-IDA-R-90-11 **Staffan Bonnier:** Unification in Classes of Equational Theories: A Case Study.

LiTH-IDA-R-90-12 **Ulf Nilsson:** Towards a Methodology for the Design of Abstract Machines of Logic Programming Languages.

LiTH-IDA-R-90-13 **Tingting Zhang, Lin Padgham:** A Diagnosis System Using Classification. Also presented at the *5th International Symposium on Methodologies for Intelligent Systems, IMIS'90,* Knowville, Tennessee, USA, October 24-26.

LiTH-IDA-R-90-14 **Björn Fjellborg:** A High-Level Synthesis Tool for Exploiting Pipelines in Special-Purpose Systems. Also in *Proc. of the IFIP WG 10.5 Working Conference on Logic and Architecture Synthesis,* Paris, France, May 30 – June 1, 1990.

LiTH-IDA-R-90-15 **Jonas Löwgren, Tommy Nordqvist:** A Knowledge-Based Tool for User Interface Evaluation and its Integration in a UIMS. Also in *Proc. of Interact'90,* Cambridge, UK, August 27–31, 1990.

LiTH-IDA-R-90-16 **Christer Bäckström, Inger Klein:** Planning in Polynomial Time: The SAS-PUBS Class. A short version of this report is also in *Proc. of the International Workshop on Expert Systems in Engineering,* Vienna, September 24-27, 1990. Springer lecture notes in AI.

LiTH-IDA-R-90-17 **Yvonne Waern, Sture Hägglund, Jonas Löwgren, Ivan Rankin, Tomas Sokolnicki, Anne Steinemann:** Communication Knowledge for Knowledge Communication.

LiTH-IDA-R-90-18 **Tommy Persson, Lennart Staflin:** A Causation Theory for a Logic of Continuous Change. Also in *Proc. of ECAI'90, European Conference on Artificial Intelligence,* Stockholm, Sweden, August 6–10, 1990.

LiTH-IDA-R-90-19 **Björn Fjellborg:** A Petri Net Model for Pipeline Scheduling. Also published in *Microprocessing and Microprogramming – the EUROMICRO Journal, Special Volume on EUROMICRO 90 Short Notes Programme,* 1991.

LiTH-IDA-R-90-21 **Peter Eklund:** Negotiating Conceptual Structures: Inheritance Hierarchies. Also in *Proc. of 5th Annual Workshop on Conseptual Structures*, Boston and Stockholm, August 1990.

LiTH-IDA-R-90-22 **Peter Eklund:** Using Path-algebras to Encode Directed Graph Knowledge-bases. Also in *Proc. of 4th Annual Workshop on Conseptual Structures*, Detroit, August 1989.

LiTH-IDA-R-90-23 **Peter Eklund, John Kellett:** Prospects for Conceptual Graphs in Acquisition Interfaces. Also in *Proc. of 3rd European Workshop on Knowledge Acquisition*, Paris, July 1989.

LiTH-IDA-R-90-24 **Petr Jirku:** On Consequence and Inference Operations.

LiTH-IDA-R-90-25 **Lin Padgham:** Defeasible Inheritance: A Lattice Based Approach. Also published in *Mathematics and Computers with Applications International Journal special issue on Semantic Nets in Artificial Intelligence.*

LiTH-IDA-R-90-26 **Mats Wirén:** Incremental Parsing and Reason Maintenance. Also in *Proc. of 3rd International Conference on computational Linguistics,* August 20-25, 1990, Helsinki, Finland, Vol 3, 287-292.

LiTH-IDA-R-90-27 **Ralph Rönnquist:** A Logic for Propagation Based Characterisation of Process Behaviour. Also in *Proc.* of the *Interntional Symposium on Methodologies for Intelligent Systems (ISMIS*), October 24.27, 1990.

LiTH-IDA-R-90-28 **Tony Larsson:** The Symbolic Manipulation of Event Traces. Also in *Proc. of the ACM/IFIP 1991 International Workshop on Formal Methods in VLSI Design,* Miami, 9-11 Jan, 1991.

LiTH-IDA-R-90-29 **Toomas Timpka, Per Bjurulf, Tom Buur:** Audit of Decision-making Regarding Female Genitourinary Infections in Outpatient Practice. Also published in *Scand J Inf Dis 1990;22:49–57.*

LiTH-IDA-R-90-30 **Toomas Timpka, Elisabeth Arborenius:** The GP's Dilemmas: A Study of Knowledge Need and Use During Health Care Consultations. Also published in *Methods of Information in Medicine 1990;29:23–29.*

LiTH-IDA-R-90-31 **Toomas Timpka, Elisabeth Arborelius:** The Primary Care Nurse's Dilemmas: A Study of Knowledge Use and Need DuringTelephone Consultations. Also published in *Journal of Advanced Nursing 1990.*

LiTH-IDA-R-90-32 **Toomas Timpka, Per Hedblom, Hans Holmgren:** Action Design: Using an Object Oriented Environment for Group Process Development of Medical Software. To be published in *Software Engineering in Medical Informatics, Elsevier, Amsterdam, 1991.*

LiTH-IDA-R-90-33 **Toomas Timpka, Lin Padgham, Per Hedblom, Stefan Wallin, Gösta Tibblin:** A Hypertext Knowledge Base for Primary Care – LIMEDS in LINCKS. Also published in *Proc. of SIGIR ‘89, ACM Press, New York, 1989.*

LiTH-IDA-R-90-34 **Toomas Timpka, Marie Ekström, Per Bjurulf:** Information Needs and Information Seeking Behaviour in Primary Health Care. Also published in *Scand J Prim Health Care 1989;7:105-9.*

LiTH-IDA-R-90-35 **Toomas Timpka:** Introducing Hypertext in Primary Health Care: – A Study on the Feasibility of Decision Support for Practitioners. Also published in *Comp Prog Biomed 1989;29:1-13.*

LiTH-IDA-R-90-36 **Toomas Timpka, Tom Buur:** Inter-observer Variation in Decision-making Regarding Patients on Chronic Hemodialysis: A Study Using the Kappa Index. Also published in *Methods of Information in Medicine 1990;29:153-7.*

LiTH-IDA-R-90-37 **Toomas Timpka, Per Bjurulf:** The Semantics of Diagnosis and Management of Genitourinary Infections – A Cross-specialty Study. Also published in *Family Practice 1989;6:279–85.*

LiTH-IDA-R-90-38 **Toomas Timpka, Per Hedblom, Gösta Tibblin:** A Hypermedia Document Collection For Primary Care: Why, What, and How. Also published in *Artificial Intelligence in Medicine 1990;2:179–92.*

LiTH-IDA-R-90-39 **Hans Holmgren, Toomas Timpka:** Systems Development in Health Care as Action Research. Also published in *Medical Informatics in Europe ‘90,* Springer Verlag, Berlin, 1990.

LiTH-IDA-R-90-40 **Elisabeth Arborelius, Toomas Timpka:** In what way may Videotapes be used to get Significant Information about the Patient-Physician Relationship? Also published in *Medical Teacher 1990;12:194-208.*

LiTH-IDA-R-90-41 **Johan Ringström:** PREDULA – A Multi-Paradigm Parallel Programming Environment.

LiTH-IDA-R-90-42 **Peter Fritzson, Tibor Gyimothy, Mariam Kamkar, Nahid Shahmehri:** Generalized Algorithmic Debugging and Testing. Also accepted to *SIGPLAN’91 Conference on Programming Language Design and Implementation.*

LiTH-IDA-R-90-43 **Mikael Pettersson:** DML – a Meta-Language for Compiler Generation from Denotational Specifications

LiTH-IDA-R-90-44 **Patrick Doherty:** Preliminary Report: NM3 – A Three-Valued Non-Monotonic Formalism. Also in *Proc. of 5th International Symposium on Methodologies for Intelligent Systems,* 1990**.**

LiTH-IDA-R-90-45 **Tommy Persson, Lennart Staflin:** Cause as an Operator in a Logic with Real-valued Fluents and Continuous Time.

LiTH-IDA-R-90-46 **Zebo Peng:** Design of Clocking Schemes in High-Level Synthesis.This report will be published in *Microprocessing and Microprogramming the EUROMICRO Journal, Vol. 1,* 1991.

LiTH-IDA-R-91-01 **Staffan Bonnier, Ulf Nilsson, Torbjörn Näslund:** A Simple Fixed Point Characterization of Three-valued Stable Model Semantics.

LiTH-IDA-R-91-02 **Ulf Nilsson:** Abstract Interpretation: A Kind of Magic.

LiTH-IDA-R-91-03 **Arne Jönsson:** A Dialogue Manager Using Initiative-Response Units and Distributed Control. Also presented at the *5th Conference of the European Chapter of the ACL,* Berlin, Germany, April 9–11, 1991.

LiTH-IDA-R-91-04 **Peter Fritzson, Dag Fritzson:** The Need for High-Level Programming Support in Scientific Computing Applied to Mechanical Analysis.

LiTH-IDA-R-91-05 **Dag Fritzson, Peter Fritzson:** Equational Modeling of Machine Elements — Applied to Rolling Bearings.

LiTH-IDA-R-91-06 **Peter Fritzson:** ObjectMath – Object Oriented Mathematical Modeling in Scientific Computing, Applied to Machine Elements Analysis

LiTH-IDA-R-91-07 **Simin Nadjm-Tehrani:** Analysis of the Overtaking Scenario: Specification of an Autonomous Car and a Driver Support System.

Proceeding

P. Eklund and L Gerholz (eds.): Proceedings of the Fifth Annual Workshop On Conceptual Structures.

Appendix F
Mailing Addresses, Phone and Electronic Mail

The departmental mailing address is:

Dept. of Computer and Information Science
Linköping University
S-581 83 Linköping, Sweden

The phone number to the university switchboard is +46 13 281000. Direct phone number to each staff member is given below. Telex and telefax may be sent to the department:

Telex	8155076
Telefax	+46 13 142231

Electronic mail may be sent using one of the following net addresses:

Internet	*usename*`@ida.liu.se`
UUCP	`enea!liuida!`*username*
BITNET	*username*`@seliuida`

Each person's *username* is listed in the column *E-mail* in the staff list below.

IDA staff

Below follows an alphabetically ordered list of the IDA staff members, as by May 1991, with phone numbers (including area code), room numbers and user names on the local computer network. Room numbers are composed by a building name (B, E or E++), floor, and room number (a leading B, F or G in a room number means corridor B, F or G).

Name	*Phone*	*Room no.*	*E-mail*
Ahlenbäck, Britt-Marie	013-28 1318	E, 1st fl., F.458	bma
Ahrenberg, Lars	013-28 2422	E, 1st fl., F.488	lah
Aleryd, Anders	013-28 1707	E, gd. fl., G.434	aoa
Alexandersson, Jan	013-28 2679	E, 1st fl., G.485	janal
Allert, Madeleine (on leave)			madal
Andersson, David	013-28 2698	E++, gd. fl., 117	dan
Andersson, John	013-28 2544	E++, gd. fl., 102	
Andersson, Jonas	013-28 1977	E, gd. fl., G.442	jonan
Andersson, Jörgen	013-28 2537	E++, gd. fl., 106	
Andersson, Lena	013-28 2676	E, 1st fl., F.478	lan
Andersson, Mats S	013-28 1425	E, gd. fl., G.436	msa
Arendell, Pia (on leave)			piaar

Bilos, Rober	013-28 2164	E++, 1st fl., 215	rob
Björklind, Andreas	013-28 1969	E, gd. fl., F.448	abj
Björklund, Lisbeth	013-28 2420	E, gd. fl., F.452	lbj
Bladh, Anna	013-28 1652	E, 1st fl., F.477B	annbl
Bonnier, Staffan	013-28 2416	E++, 1st fl., 203	sbo
Boye, Johan	013-28 2694	E++, 1st fl., 201	johbo
Busch, Douglas	013-28 2403	E, gd. fl., F.466	dbu
Bäckström, Christer	013-28 2429	E, gd. fl., F.488	cba
Carlsson, Peter	013-28 2537	E++, gd. fl., 106	petca
Cederling, Ulf	013-28 2859	B, 1st fl., B.480	ulfce
Cronholm, Stefan	013-28 1638	E, 1st fl., F.453	scr
Dahlbäck, Nils	013-28 1664	E, 1st fl., F.472	nda
Dahlén, Ulf	013-28 2414	E, gd. fl., G.454	uda
Degerstedt, Lars	013-28 2168	E, gd. fl., G.478	larde
Doherty, Patrick	013-28 2426	E, gd. fl., G.460 A	pdy
Downing, Keith	013-28 1891	E, gd. fl., G.484	kdo
Drabent, Wlodzimiertz	013-28 2660	E++, 1st fl., 202	wdr
Driankov, Dimiter	013-28 1716	E, gd. fl., F.470	ddr
Eklund, Peter	013-28 1950	E, 1st fl., G.464	pwe
Ekman, Barbara	013-28 1410	E, 1st fl., F.464	bek
Elfinger, Eva	013-28 1524	E++, gd. fl., 103	evael
Emanuelson, Inger	013-28 2180	E, 1st fl., 1:151	ime
Eriksson, Henrik	013-28 1935	E, 1st fl., G.436	her
Eriksson, Ulrik	013-28 1465	E++, gd. fl., 127	ulrer
Fagerström, Johan	013-28 1364	E++, 1st fl., 214	jfa
Fahl, Gustav	013-28 2465	E, 1st fl., G.435	gusfa
Finmo, Leif	013-28 1443	E, gd. fl., G.488	lbf
Fjellborg, Björn	013-28 1819	E++, 1st fl., 230	bfj
Fjällström, Per-Olof	013-28 2412	E++, gd. fl., 119	pof
Fotiadis, Dimitrios	013-28 1463	E, gd. fl., G.440	dif
Franzén, Birgitta	013-28 2692	E, 1st fl., G.446	bfr
Fritzson, Peter	013-28 1484	E++, 1st fl., A 216	paf
Fäldt, Arne	013-28 1962	E, gd. fl., G.452	arf
Goldkuhl, Göran	013-28 1452	E, 1st fl., G.488	ggo
Granlund, Rego	013-28 2679	E, 1st fl., G.485	reg
Gu, Xinli	013-28 1970	E++, 1st fl., 205	xgu
Hansson, Carl-Gunnar	013-28 2549	E++, gd. fl., 112	carha
Haraldsson, Anders	013-28 1403	E, 1st fl., 29:149	aha
Harrius, Jonni	013-28 2493	E, 1st fl., G.454	jha
Hedblom, Per	013-28 2413	E, 1st fl., F.434	phe
Helander, Björn	013-28 1521	E++, gd. fl., 105	bjohe
Herber, Johan	013-28 2670	E++, 1st fl., 231	johhe
Hirsch, Richard	013-28 1937	E, 1st fl., F.482	richi
Hjerppe, Roland	013-28 1965	E, gd. fl., F.454	rhj
Holmgren, Hans	013-28 1966	E, 1st fl., G.470	heh
Hägglund, Sture	013-28 1431	E, 1st fl., G.442	sth
Jacobson, Anne-Marie	013-28 1496	E, gd. fl., G.458	amj

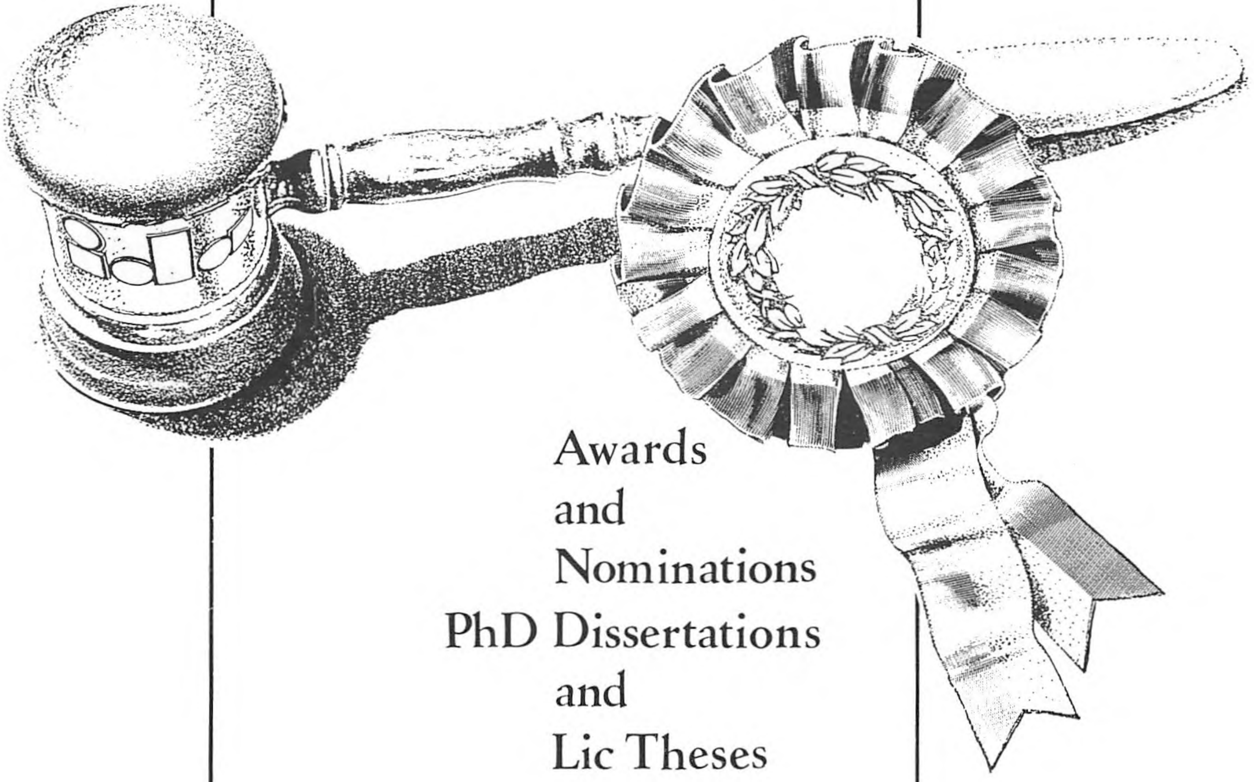
Jansson, Michael	013-28 1470	E++, gd. fl., 124	mij
Johannesson, Peter	013-20 7440	Berzeliusskolan	petjo
Johansson, Mikael	013-28 1479	E, 1st fl., G.472	mjo
Johansson, Olof	013-28 2390	E++, 1st fl., 224	ojo
Jonsson, Torbjörn	013-28 2467	E, 1st fl., F.460	tjo
Jönsson, Arne	013-28 1717	E, 1st fl., F.484	arj
Kamkar, Mariam	013-28 1949	E++, 1st fl., 211	mak
Kluzniak, Feliks	013-28 2601	E++, 1st fl., 207	fkl
Krajina, Patrik	013-28 2669	E, 1st fl., G.485	pak
Krysander, Christian	013-28 1411	E, 1st fl., F.448	ckr
Kuchcinski, Krzysztof	013-28 1883	E++, 1st fl., 228	kku
Kågedal, Andreas	013-28 2419	E++, 1st fl., 206	aka
Lambric, Patric	013-28 2605	E++, gd. fl., 120	patla
Larsson, Mats	013-28 1844	E++, 1st fl., 229	mla
Larsson, Tony	013-28 1832		tla
Lennartsson, Bengt	013-28 1427	E, gd. fl., 29:149	bln
Lilja, Carita	013-28 1458	E, 1st fl., G.484	cli
Lind, Jonas	013-28 1526	E++, gd. fl., 114	jonli
Lindblom, Patrik	013-28 2594	E, 1st fl., F.440	patli
Linge, Lisbeth	013-28 1472	E, 1st fl., F.490	lli
Lingenhult, Gunilla	013-28 2297	E++, 1st fl., 212	glt
Loborg, Peter	013-28 2494	E++, 1st fl., 227	plo
Lundberg, Eva	013-28 1947	E, 1st fl., F.453	elu
Löwgren, Jonas	013-28 1482	E, 1st fl., G.452	jlo
Malec, Jacek	013-28 2362	E, gd. fl., F.490	jam
Maleki, Jalal	013-28 1963	E, 1st fl., F.470	jma
Maluszynski, Jan	013-28 1483	E++, 1st fl., 210	jmz
Mattsson Kihlström, Bodil	013-28 2169	E++, 1st floor118	bmk
Mattsson, Lars	013-28 1088	E++, gd. fl., 110	larma
Moberg, Anna	013-28 2542	E++, gd. fl., 107	annmo
Morin, Magnus	013-28 2425	E, gd. fl., G.490	mmo
Månsbacka, Lotta (on leave)			lotma
Nadjm-Tehrani, Simin	013-28 2411	E, gd. fl., G.482	snt
Nilsson, Bernt	013-28 1975	E, gd. fl., G.454	bkn
Nilsson, Björn	013-28 1476	E, gd. fl., G.446	ban
Nilsson, Ivan	013-28 2423	E, 1st fl., G.466	ivani
Nilsson, Nils	013-28 2418	E, gd. fl., G.460	nni
Nilsson, Peter J	013-28 1354	E, gd. fl., G.448	pjn
Nilsson, Rolf	013-28 2405	E, 1st fl., G.478	ron
Nilsson, Ulf	013-28 2410	E++, 1st fl., 208	urn
Noghabai, Mehran	013-28 1486	E++, gd. fl., 109	mehno
Näslund, Torbjörn	013-28 1957	E, 1st fl., G.448	tor
Ohlsén, Lennart	013-28 1555	E++, gd. fl., 108	lenoh
Ohlsson, Mikael	013-28 2608	E++, gd. fl., 125	mikoh
Olander, Birgitta (on leave)			
Olnäs, Patrick	013-28 2609	E++, gd. fl., 120	patol
Olsson, Kerstin (on leave)			kerol

Olsson, Maria	013-28 2538	E++, gd. fl., 104	
Olsson, Tommy	013-28 1954	E, 1st fl., F.454	tao
Padgham, Lin	013-28 2066	E++, gd. fl., 122	lin
Padron-McCarthy, Thomas	013-28 2677	E, 1st fl., G.434	tpm
Patel, Mikael	013-28 1821	E++, 1st fl., 226	mip
Peng, Zebo	013-28 2067	E++, 1st fl., 235	zpe
Peolsson, Mikael			micpe
Persson, Mats	013-28 1948	E, 1st fl., G.467	map
Persson, Tommy	013-28 2428	E, gd. fl., G.474	tpe
Petersson, Jan	013-28 1936	E++, gd. fl., 116	jpe
Pettersson, Mikael	013-28 2683	E++, 1st fl., 213	mpe
Rankin, Ivan	013-28 2424	E, 1st fl., G.458	ira
Rapp, Birger	013-28 1525	E++, gd. fl., 101	birra
Rauch, Ewa	013-28 1938	E, 1st fl., F.440	era
Raunio, Lise-Lotte	013-28 2430	E, 1st fl., G.482	llr
Reshagen, Lars	013-28 1973	E, 1st fl., F.440	lre
Ringström, Johan	013-28 2406	E++, 1st fl., 217	jri
Rönnquist, Ralph	013-28 1979	E++, gd. fl., 121	rrq
Röstlinger, Annie	013-28 2407	E, 1st fl., G.490	aro
Sandahl, Kristian	013-28 1467	E, 1st fl., G.440	krs
Sandblom, Ingrid	013-28 2360	E, gd. fl., G.466	ingsa
Sandewall, Erik	013-28 1408	E, gd. fl., G.470	ejs
Sedvall, Göran	013-28 1977	E, gd. fl., G.442	gse
Seld, Thomas			
Shahmehri, Nahid	013-28 1464	E++, 1st fl., 218	nsh
Shu, Hua	013-28 1703	E, gd. fl., F.482	hsh
Shu, Ying	013-28 2427	E, 1st fl., G.453	yinsh
Sjöberg, Cecilia	013-28 2596	E, 1st fl., F.446	cecsj
Sjöström, Camilla	013-28 2538	E++, gd. fl., 104	
Sköld, Martin	013-28 2698	E++, 1st fl., 222	marsk
Staflin, Lennart	013-28 1314	E, gd. fl., G.474	les
Stolt, Anette	013-28 2538	E++, gd. fl., 104	
Stoy, Erik	013-28 2691	E++, 1st fl., 233	est
Strömberg, Lars	013-28 1976	E++, 1st fl., 223	lst
Sunnerud, Katarina	013-28 1481	E, 1st fl., F.466	ksd
Svenberg, Stefan	013-28 2690	E, 1st fl., F.489	ssv
Svensson, Birgitta	013-28 2595	E, 1st fl., F.446	birsv
Svensson, Eva-Chris	013-28 1457	E, 1st fl., G.476	ecs
Söderlund, Siv	013-28 1426	E, 1st fl., G.489	sso
Söderman, Ulf	013-28 2681	E, gd. fl., F.484	uso
Thurée, Åke	013-28 2492	E, 1st fl., F.478	ath
Timpka, Toomas	013-28 1471	E, 1st fl., F.436	tti
Törne, Anders	013-28 2365	E++, 1st fl., 225	ato
Vainio-Larsson, Arja	013-28 1477	E, gd. fl., 29: 151	ava
Viklund, Lars	013-28 2686	E++, 1st fl., 219	lvi
Wallgren, Jonas	013-28 2682	E++, 1st fl., 204	jwc
Wallgren, Lillemor	013-28 1480	E, 1st fl., G.460	lew

Wendel, Rickard			
Wigh, Lena	013-28 1706	E, 1st fl., F.435	`len`
Wilander, Jerker	013-28 2694	E++	`jwi`
Willén, Olle	013-28 1413	E, 1st fl., F.452	`owl`
Wirén, Mats	013-28 1466	E, 1st fl., F.476	`mgw`
Wretland, Åsa	013-28 2668	E, 1st fl., G.467	`asw`
Zhang, Tingting	013-28 1430	E++, gd. fl., 129	`tzh`
Ökvist, Christian	013-28 2609	E++, gd. fl., 123	`chrok`
Österling, Per	013-28 2415	E, gd. fl., G.478	`pos`

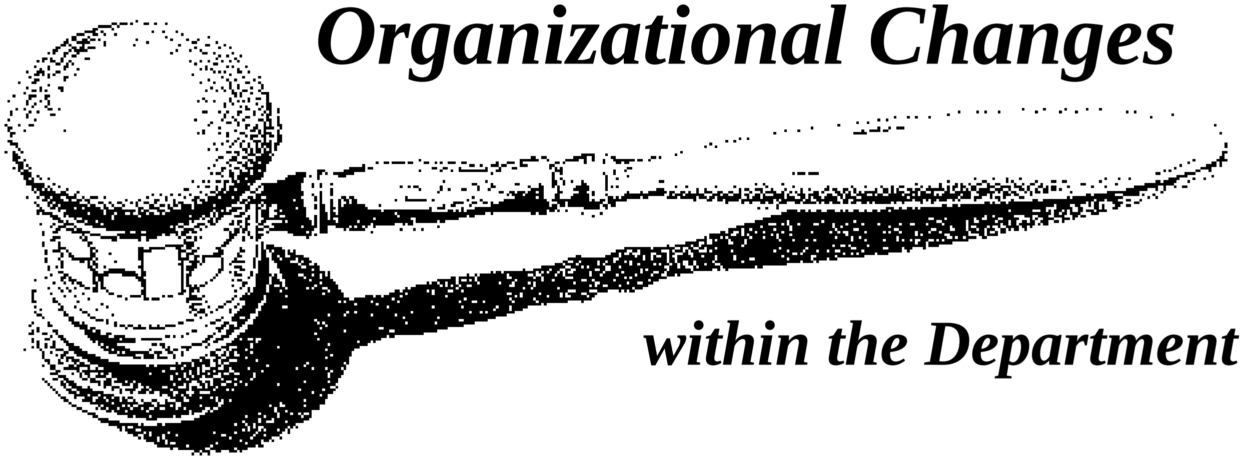
Events of the Year

Organizational
Changes
Special
Assignments



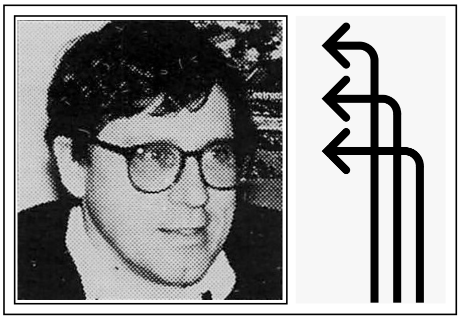
Awards
and
Nominations
PhD Dissertations
and
Lic Theses

Organizational Changes



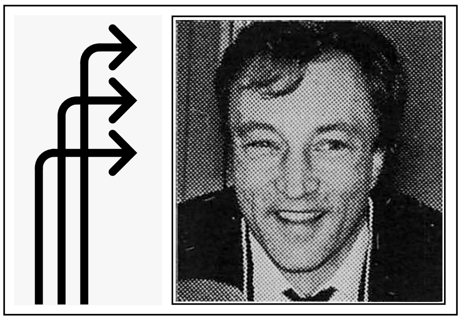
within the Department

On July 1st 1990 there were changes to the offices of Chairman of the Department, two committee Chairmen and Director of Graduate Studies.

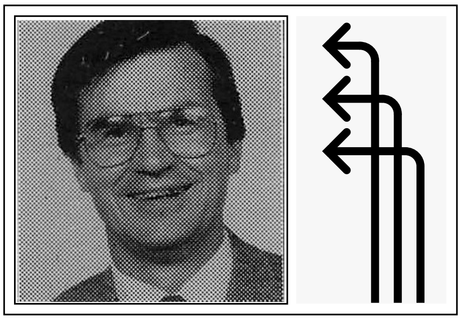


Anders Haraldsson replaced Bengt Lennartsson as Chairman of the Department.

Bengt Lennartsson replaced Anders Haraldsson as Chairman of the Undergraduate Teaching Committee (IDUN).

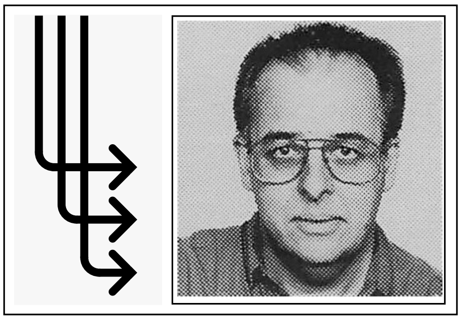


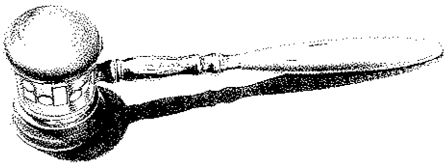
Sture Hägglund replaced Erik Sandewall as Chairman of the Research Committee (FND).



Erik Sandewall is Vice-Chairman of the Department.

Douglas Busch replaced Johan Fagerström as Director of Graduate Studies.





New Organization for Undergraduate Education

Undergraduate education was organised into areas of study, each with a Director of Studies:

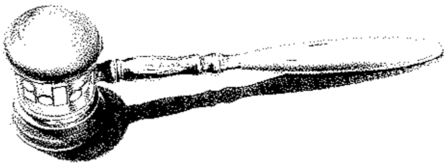
Arne Jönsson became Director of Studies for Cognitive Science

Olle Willén became Director of Studies for Software Design

Johan Fagerström became Director of Studies for Computer Architecture

Lise-Lotte Raunio became Director of Studies for Systems Development

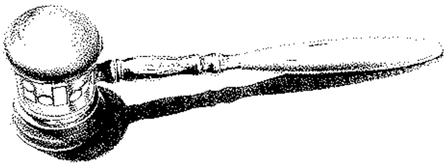
Lennart Ohlsén became Director of Studies for Economic Information Systems.



Economic Information Systems Joins IDA

On September 1st 1990 the section **Economic Information Systems** transferred all their activities to the Department of Computer and Information Science. The person in charge of this subject area is **Birger Rapp**.

Planned research and a graduate program in Systems Development will be conducted within the area of Economic Information Systems, in collaboration with **Göran Goldkuhl**.



New Graduate Program: Computational Linguistics

In December 1990 a graduate program in **Computational Linguistics** was initiated, under the charge of **Lars Ahrenberg**.

On December 7th Lars Ahrenberg delivered his inaugural lecture as docent in Computational Linguistics, entitled “Technology and Theory of Language”.

Awards and Nominations

Mats Larsson and Prince Bertil

Prince Bertil Scholarship Award

Mats Larsson has been awarded "The Prince Bertil Scholarship 1991/92". The award is co-sponsored by Tetra-Pak and the British Foreign and Commonwealth Office and it is intended to make it possible for a graduate student in the broad field of technology to spend one year at an academic institution in Great Britain. Mats will use the £10,000 award to finance a year at the University of Cambridge.

Since this was the first year the scholarship was announced, Prince Bertil himself participated in a presentation at the British ambassadors residence on 6 March, 1991.

The Lawson Award

The 1990 Lawson Award was presented to **Ulf Nilsson** in recognition of his achievement as the principal author of the textbook *Logic, Programming and Prolog,* coauthored by Professor **Jan Maluszynski** and published by John Wiley & Sons, Ltd, 1990.

The price was founded in 1989 and is awarded to a graduate student at the department who has attained ‘well-deserved international recognition‘.

Sweden-America Foundation Award

A SEK80,000 grant has been awarded to **Ulf Nilsson** by the Sweden-America Foundation. The grants are given to facilitate graduate and post-doctorate studies in the United States or Canada. The award will be used to partially finance a one-year stay at the State University of New York at Stony Brook where Ulf Nilsson will be joining the logic programming group headed by Professor David S. Warren.

Hosei University Foreign Scholars Fellowship Award

ABB Automation AB and SAAB Missiles AB Int. Exchange Scholarship

Peter Eklund has been awarded a scholarship of SEK70.000 which partially funds an eight month post-doctoral study at Hosei University in Tokyo, Japan. Peter will be investigating the “State of Non-monotonic and nonstandard logics in Japan”, and in particular concentrating on applications of fuzzy logic and to control applications. During his stay Peter will be working with the Fuzzy Control Group at the Department of Industrial and Systems Engineering under Professors U. Omori and K. Hirota.

An additional scholarship of SEK 60.000 for international exchange involving post-doctoral studies of fuzzy control has been created by donations to our department from ABB Automation AB and SAAB Missiles AB.

The Micro-Electronics Centre Award

Each year the Micro-Electronics Centre (MicroElektronikCentrum) bestows an award on a teacher of distinction within the area of its activities, namely computing and electronics.

In 1991 the recipient for teaching in computer science was **Anders Haraldsson**, who was awarded a travel stipend of SEK10,000, and a diploma ‘‘For his work in developing the computer science program at Linköping University, in close collaboration with the Curriculum Committee (linjenämnd) and students’’.

Previously **Arne Jönsson** received the same award ‘‘For his greatly appreciated teaching, and for the up-to-date quality of the courses he developed in the area of Artificial Intelligence’’.

The 2nd IDA Graduate Students’ Conference

The 2nd Annual IDA Conference on Computer and Information Science, where graduate students in the department are invited to present their work, was held May 3-4.

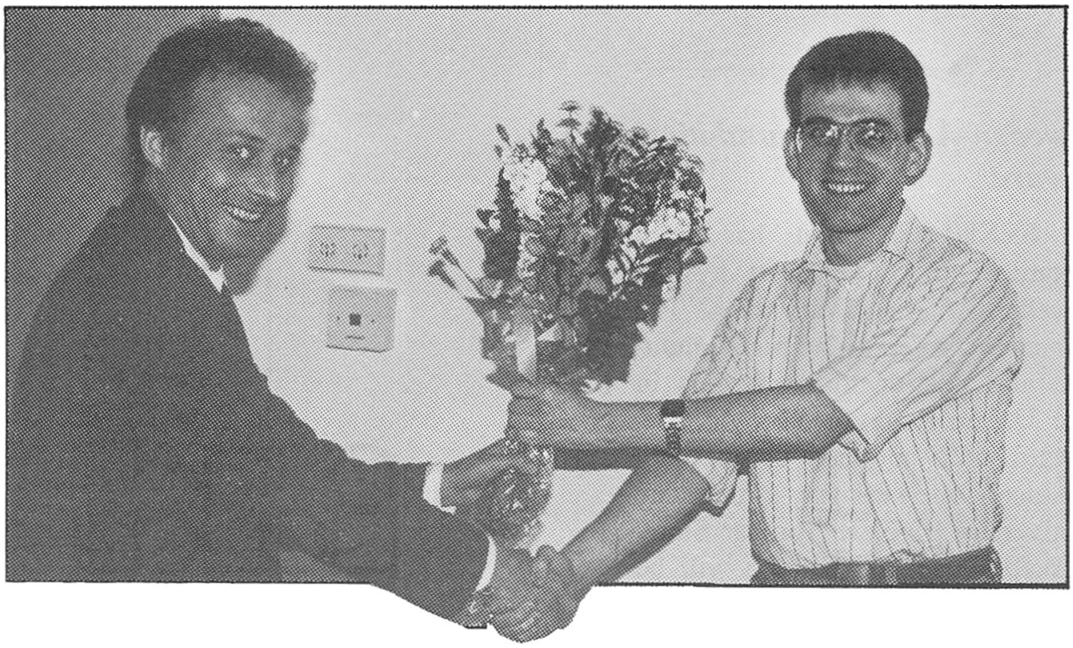
Mikael Petterson’s paper “Generating Efficient Code from Continuation Semantics” was selected for the best paper award.

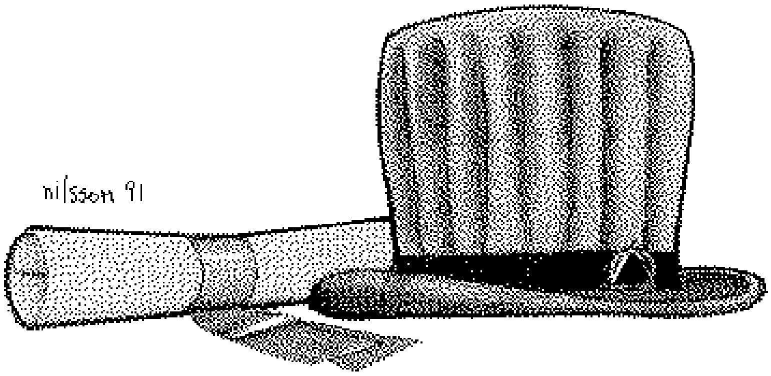
The “Golden Carrot” Award

Annual nomination by the student body for the ‘‘Golden Carrot’’ award for best teacher of the year.

The 1990 nominations for teacher of the year within C-line and D-line were respectively **Nils Dahlbäck** and **Johan Fagerström**.

In 1991 **Douglas Busch** was nominated teacher of the year within C-line.





PhD Dissertations and Lic Theses

The Dissertations of the period October 1989 to April 1991

Tony Larsson

A Formal Hardware Description and Verification Method. (No 214)

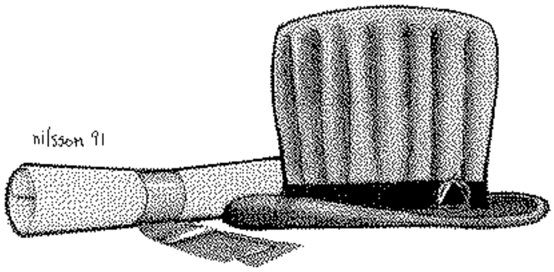
Design of correctly working hardware systems involves the description of functional, structural and temporal aspects at different levels of abstraction and the verification of the requested equivalence between these descriptions. This process is usually very time-consuming and its simplification is a desirable aim.

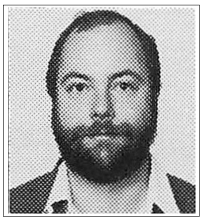
To provide for this it is important that the description language and the verification method can be used at as many abstraction levels as possible and that previous results can be reused. Further in order to support formal reasoning about hardware circuits and their correctness, it is preferable if the description method is based on a well-founded formalism.

As one goal of this thesis we will illustrate that by the extension of predicate logic with a temporal reference operator, it is possible to specify both functional, temporal and structural properties of hardware circuits. The temporal reference operator makes it possible to describe and reason about relationships between the streams of values observable at the ports of a hardware circuit. We specify the intended meaning of this temporal operator by a set of axioms and by giving it an interpretation vis-a-vis a temporal model. Based on these axioms it is possible to formally reason about temporal relationships.

This leads to the major goal, i.e. to provide support for a further mechanized verification of hardware based on transformation of boolean, arithmetic, relational and temporal constructs expressed in the description language.

Important contributions of the thesis are methods for multi-level hardware description and methods for mechanized verification including functional, structural and temporal aspects that can be used as a complement to existing theorem proving systems. A prototype implementation of the description language based on the generalized (untyped) predicate logic presented and an implementation of a verification system has been part of the research underlying this thesis.





Michael Reinfrank

Fundamentals and Logical Foundations of Truth Maintenance (No. 221)

Despite their importance in AI problem solving, nonmonotonic truth maintenance systems (TMSs) still lack sufficiently well-understood logical foundations. In this thesis, I present a rigorous logical theory of TMSs. I pursue a two-step, bottom-up approach. First, I specify a direct, but implementation-independent, theory of truth maintenance. This theory, then, is used to draw a connection between TMSs and Autoepistemic Logic, thus closing a gap between theory and implementation in Nonmonotonic Reasoning, provide a correctness proof for an encoding of nonmonotonic justifications in an essentially monotonic assumption-based TMS, design a uniform framework for truth maintenance and nonmonotonic inference based on the concept of justification-schemata, discuss a model theory of TMSs in terms of stable, maximally preferred model sets.

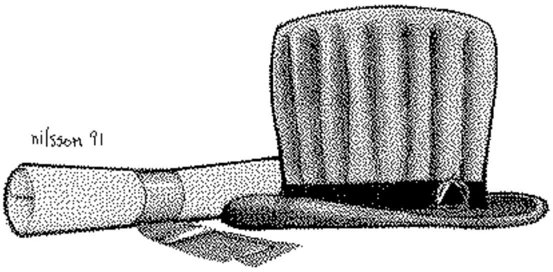
At the time of writing, no comprehensive introductory readings on truth maintenance are available. Therefore, the present thesis begins with a set of lecture notes which provide the necessary background information for the subsequent formal treatment of foundational issues.

Jonas Löwgren

Knowledge-Based Design Support and Discourse Management in User Interface Management Systems. (No. 239)

This dissertation is about User Interface Management Systems (UIMSs), and more specifically about new ways to extend the scope and the functionality of these systems.

I define a UIMS as an interactive tool or set of tools intended to facilitate the design, development and delivery of user interfaces. The assumption underlying the application of UIMS techniques to software development is that the user interface can to some extent be separated from the underlying functionality of the application. Current UIMS technology is, however, not capable of coping with this separation in the case of conversational expert systems. In the first part of the dissertation, I present a new UIMS architecture, based on planning techniques and a representation of the beliefs of the user and the system, and show by means of an example that dialogue independence can be achieved for a task-oriented expert system by using this new architecture.



The second part is concerned with support for the user of the UIMS – the user interface designer. The approach I advocate is to enhance the design and development environment with knowledge of user interface design, knowledge which is used to generate comments on the user interface designer's work. A prototype expert critiquing system was built to test the feasibility of knowledge-based evaluation of user interface designs. The results were encouraging and also demonstrated that the level of user interface representation is crucial for the quality of the evaluation. I propose an architecture where this kind of knowledge-based support is integrated with a UIMS and argue that the requirements on a high-level user interface representation can be relaxed if the system also analyses data from empirical tests of the user interface prototype.

Henrik Eriksson

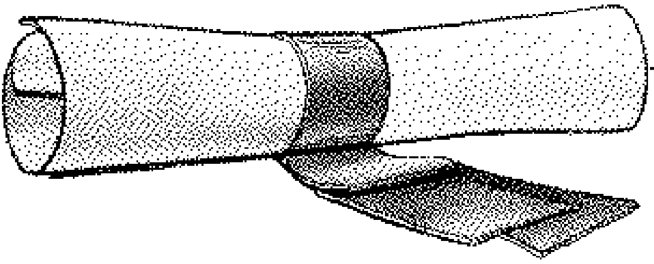
Meta-Tool Support for Knowledge Acquisition. (No. 244)

Knowledge acquisition is a major bottleneck in expert system development. Specialized, or domain-oriented, knowledge acquisition tools can provide efficient support in restricted domains. However, the principal drawback with specialized knowledge acquisition tools is that the tool cost per expert system developed is typically high.

Meta-level environments is an approach to support knowledge engineers in developing such knowledge acquisition tools. Meta-tools, i.e. tools for creating know-ledge acquisition tools, can be used to specify and automatically generate knowledge acquisition tools for single domains and even single applications.

This thesis presents an abstract architecture approach to the specification of knowledge acquisition tools. In this framework knowledge acquisition tools can be specified according to an abstract model of the target tool architecture. DOTS is a meta-tool that supports the abstract-architecture specification scheme. Knowledge engineers can use DOTS to specify and generate domain-oriented knowledge acquisition tools that can be used by domain experts directly.

Two implementations of knowledge acquisition tools for different domains are presented in this thesis. These tools are representatives of knowledge acquisition tools that are desirable to generate from meta-tools. One of them was handcrafted and specialized to the domain of protein purification planning. The other emerged from an evaluation of DOTS by developing a knowledge acquisition tool in a different domain (troubleshooting laboratory equipment). Results from this evaluation are also reported.



...and Lic. Theses

from November 1989 to April 1991

Magnus Merkel

Temporal Information in Natural Language.
(No 189, 1989.)

Ulf Nilsson

A Systematic Approach to Abstract Interpretation of Logic Programs. (No 196, 1989.)

Staffan Bonnier

Horn Clause Logic with External Procedures: Towards a Theoretical Framework. (No 197, 1989.)

Christer Hansson

A Prototype System for Logical Reasoning about Time and Action. (No 203, 1990.)

Björn Fjellborg

An Approach to Extraction of Pipeline Structures for VLSI High-Level Synthesis. (No 212, 1990.)

Patrick Doherty

A Three-Valued Approach to Non-Monotonic Reasoning.
(No 230, 1990.)

Tomas Sokolnicki

Coaching Partial Plans: An Approach to Knowledge-Based Tutoring. (No 237, 1990.)

Lars Strömberg

Postmortem Debugging of Distributed Systems.
(No 250, 1990.)

Torbjörn Näslund

SLDFA-Resolution – Computing Answers for Negative Queries. (No 253, 1990.)

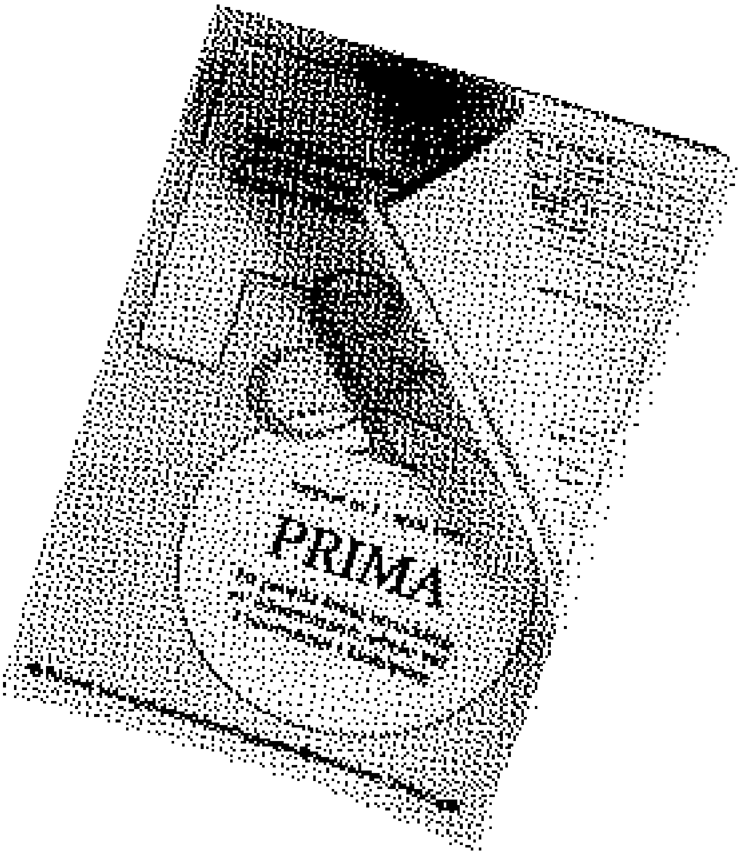
Special Assignments

within the University

From April 15th 1991 **Bengt Lennartsson** is substituting part-time for the University's Head of Personnel, Curt Carlsson, with respect to some of his duties. He will be responsible for co-ordinated planning of education and research, investment and personnel, and organizational development.

Erik Sandewall sits on the Board for the Swedish Research Council for Engineering Sciences (TFR, which replaces STUF).

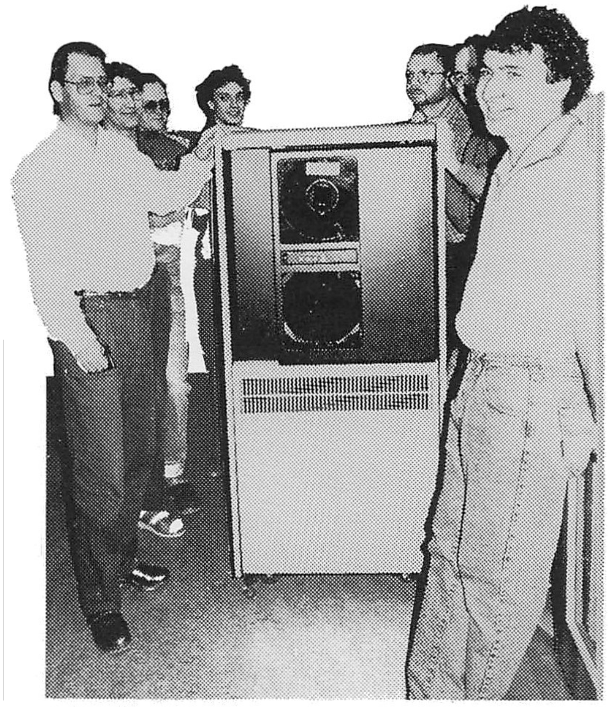
Erik Sandewall is also a member of the University Board.



Inger Emanuelson has had a partial leave of absence in order to devote time to a project concerned with the development of administrative work within the university.

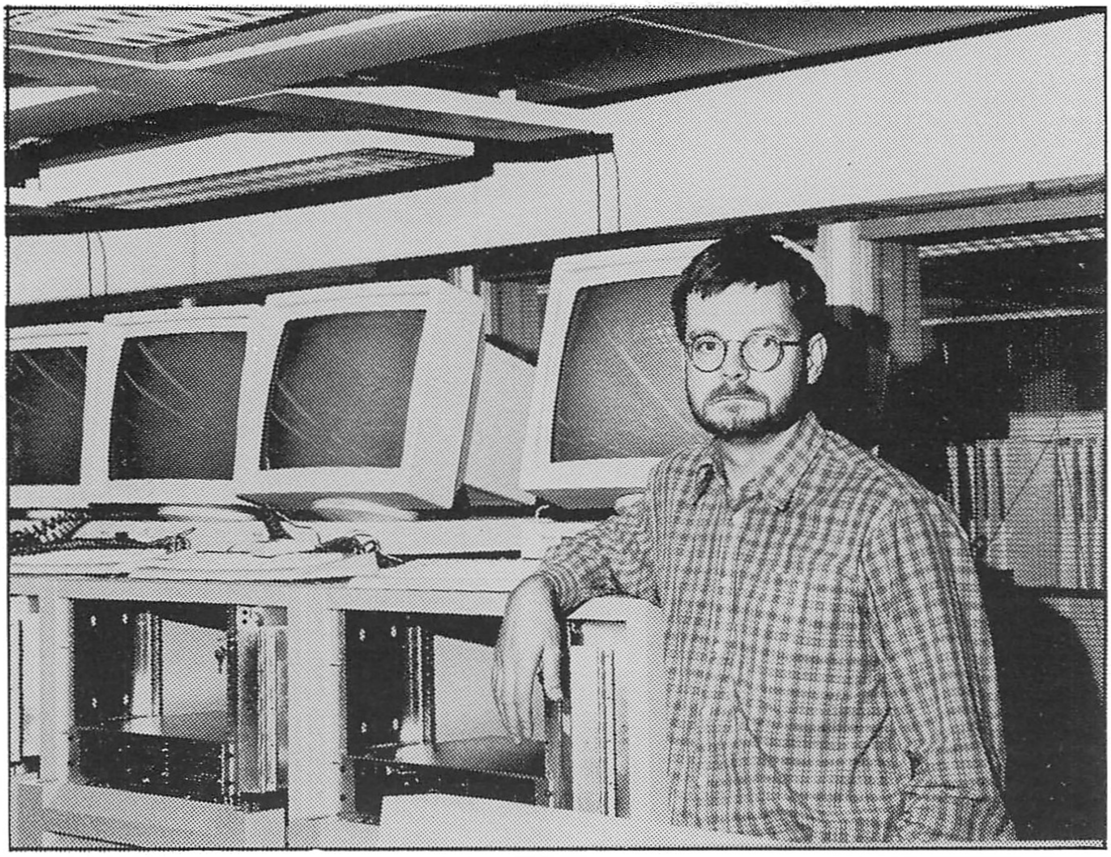
Computer Facilities

In the summer of 1990, the DEC-system 2060, named Lisbet, was replaced by about 200 networked Sun workstations.



The departure of Lisbet…

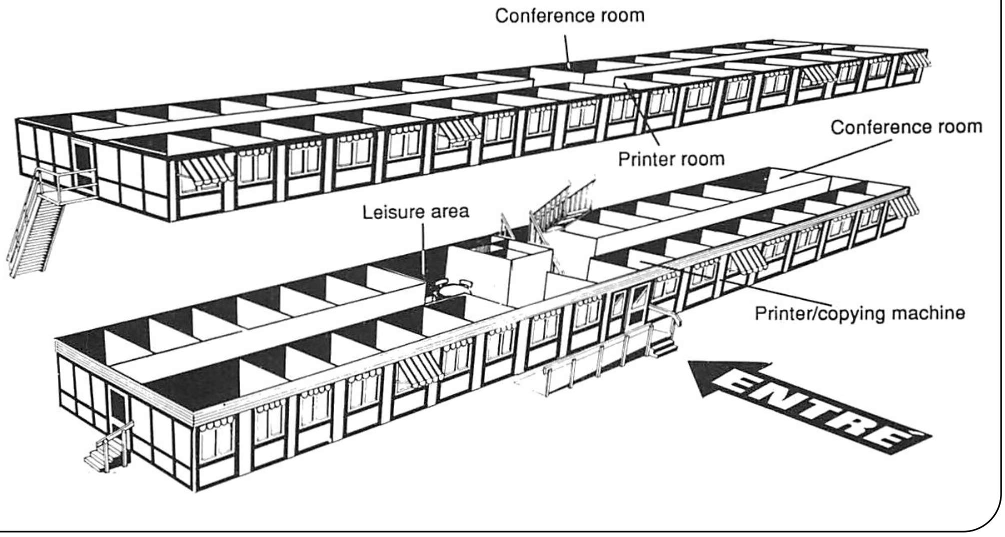
…and the arrival of Sun



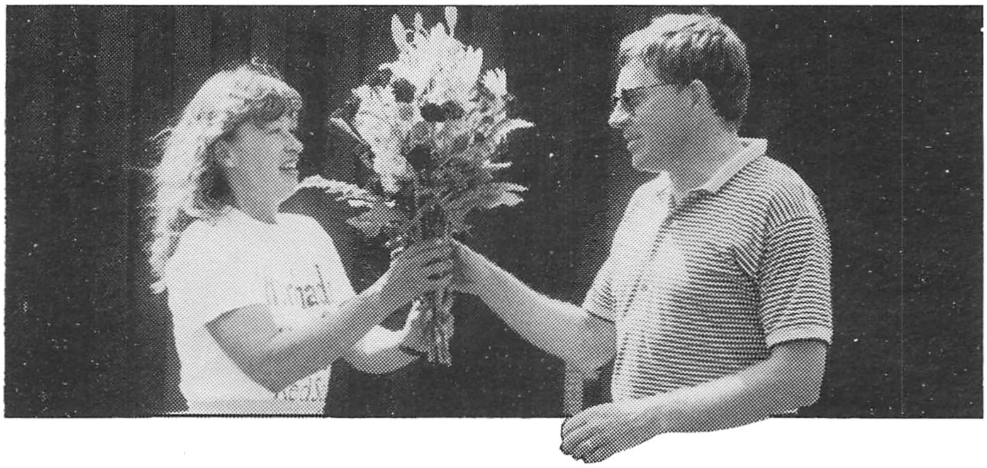
New Building

The arrival of E++

In October 1990 the department rented an annex to E-building, which was named E++ by popular vote.

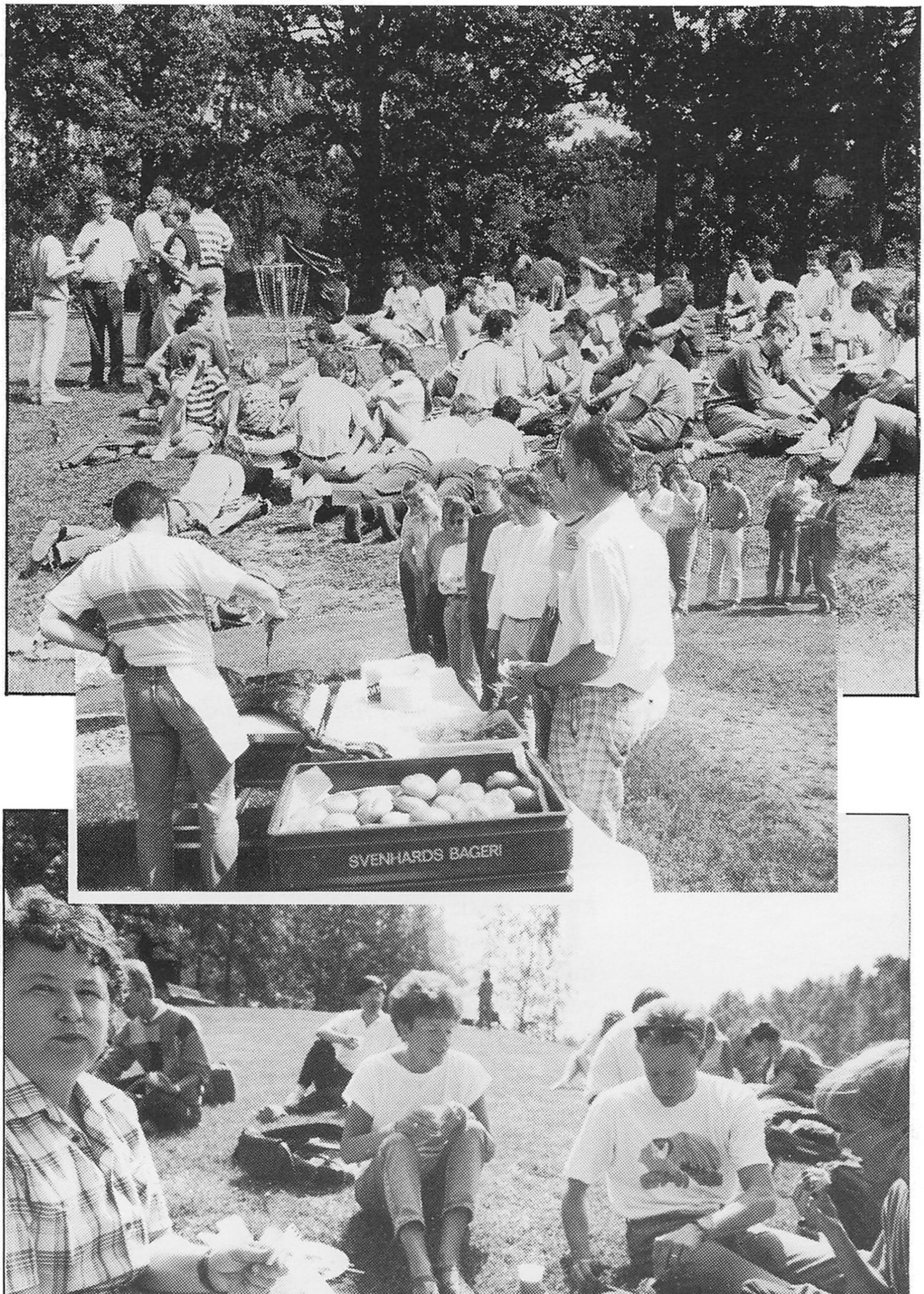


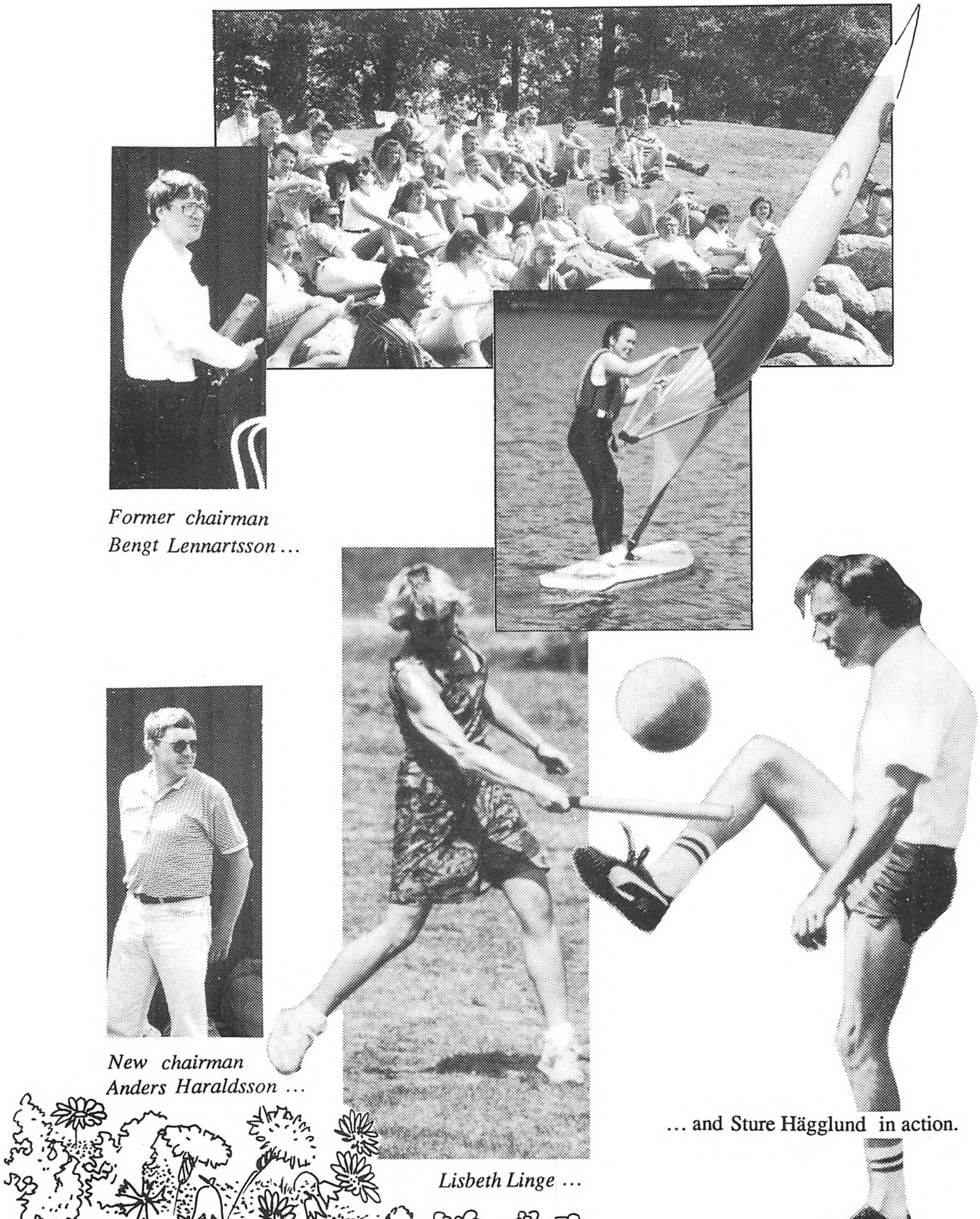
Change of Department Chairman



Inger Emanuelson welcomes Anders Haraldsson

To mark the change of Department Chairman, an informal day devoted to diverse activities for the whole department took place on May 30th 1990.


SVENHARDS BAGERI



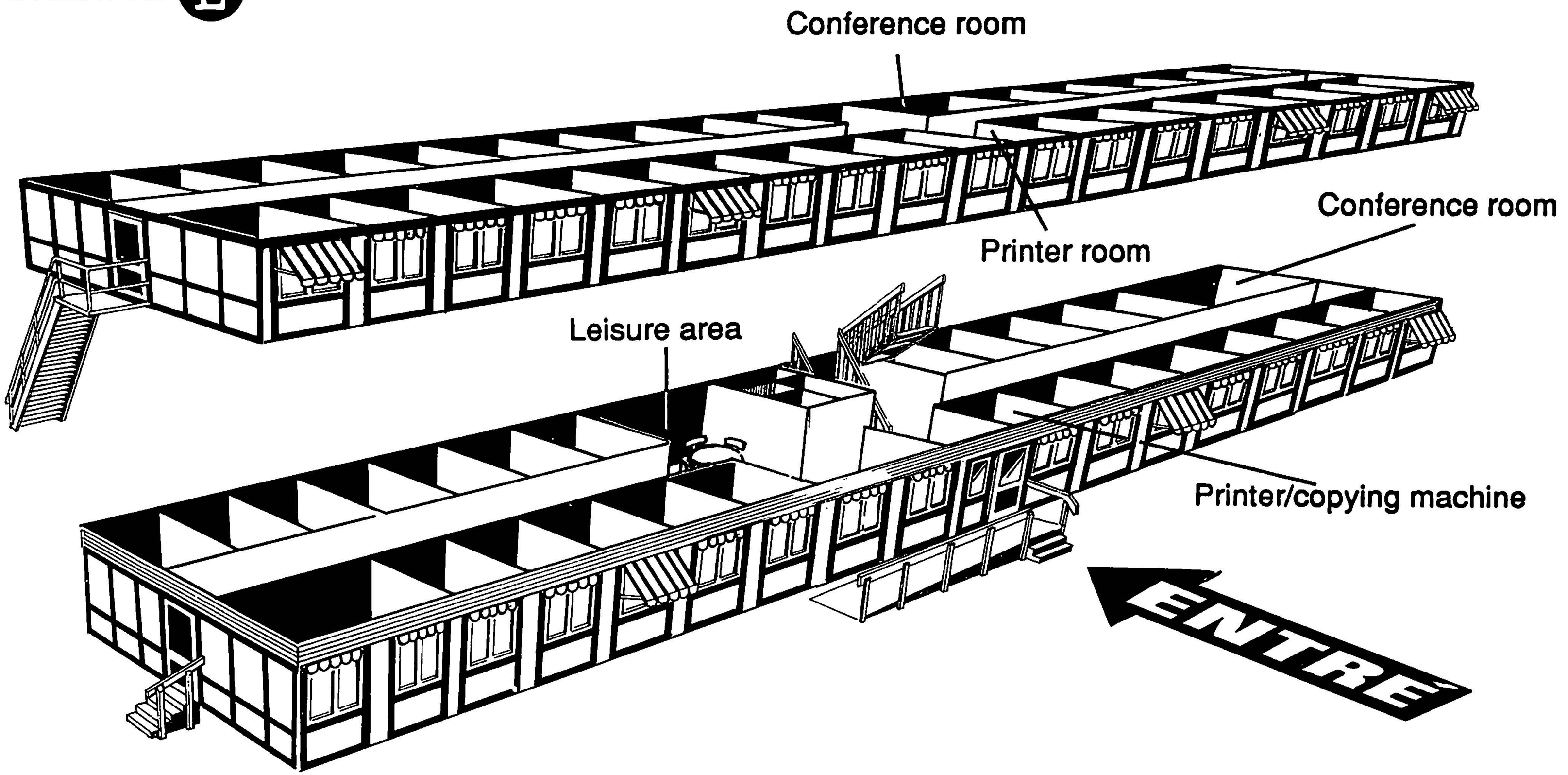
Former chairman Bengt Lennartsson ...

New chairman Anders Haraldsson ...

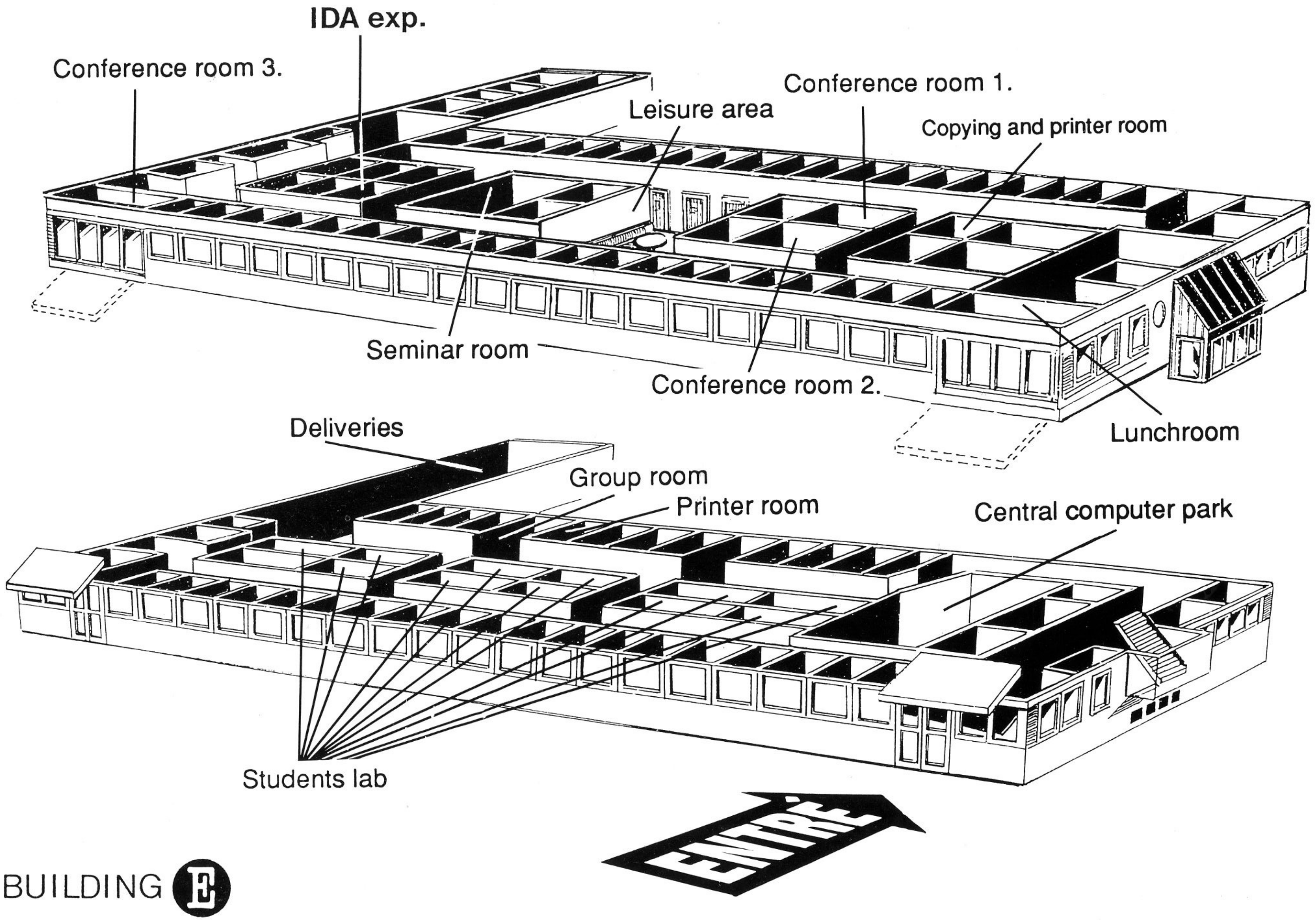
Lisbeth Linge ...

... and Sture Hägglund in action.

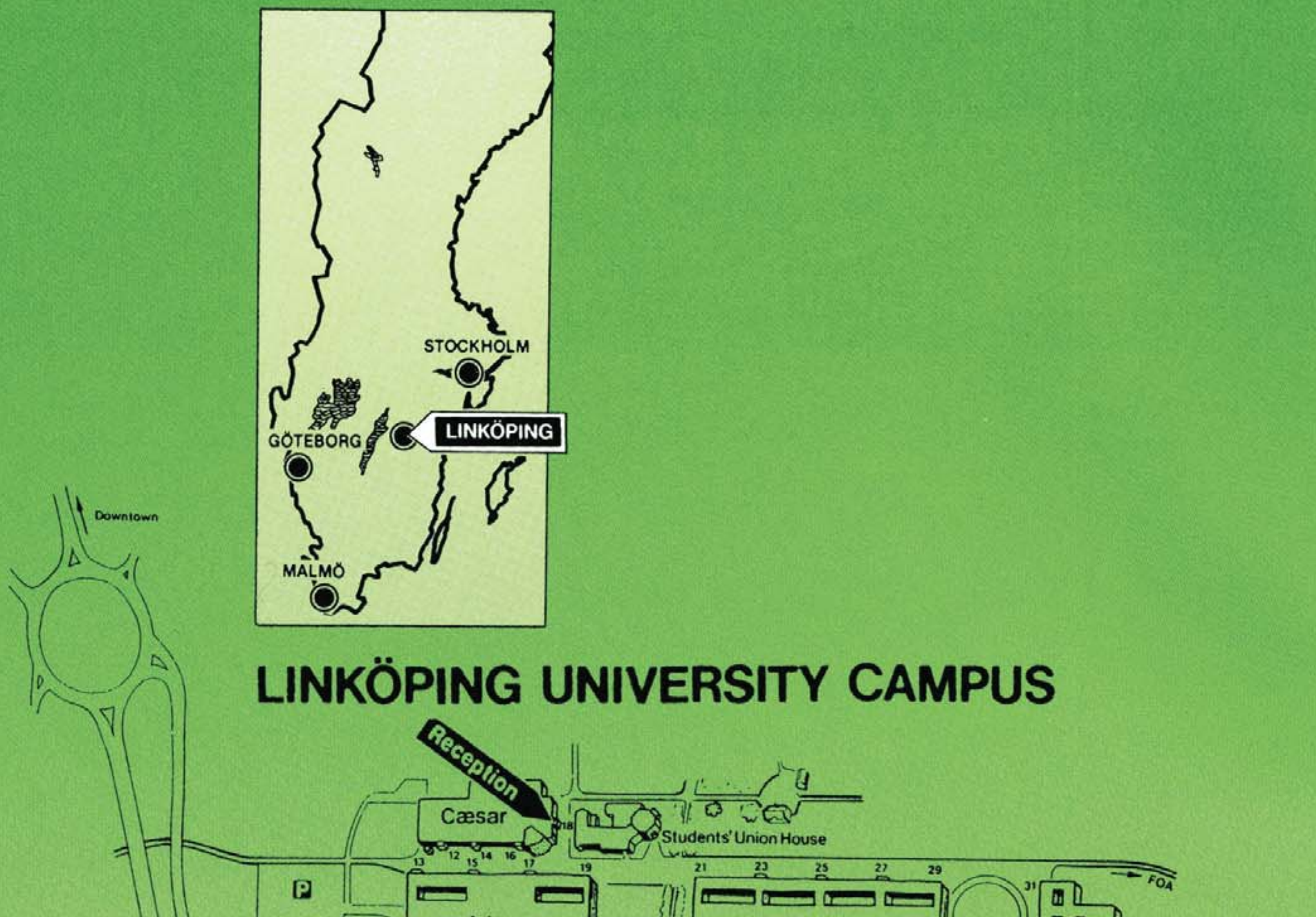
BUILDING E ++



Department of COMPUTER and INFORMATION SCIENCE


IDA exp.
Conference room 3.
Conference room 1.
Leisure area
Copying and printer room
Seminar room
Conference room 2.
Lunchroom
Deliveries
Group room
Printer room
Central computer park
Students lab
ENTRÉ

BUILDING E



LINKÖPING UNIVERSITY CAMPUS

UNIVERSITETET I LINKÖPING

LINKÖPING UNIVERSITY