Dept of COMPUTER AND INFORMATION SCIENCE

Annual research report

88/89

LINKÖPING
University and
Institute of technology

LiTH-IDA-R-89-45
ISSN-0281-4250

Fall 1989

The Department of Computer and Information Science

Linköping University

Annual Research Report 1988/89

This report describes research on software technology and related areas within the Department of Computer and Information Science at Linköping University and Institute of Technology. Main areas of current research are programming environments, artificial intelligence, natural language processing, application systems, computer-aided design of digital systems, representation of knowledge in logic, complexity of algorithms, logic programming, library and information science, and administrative data processing. The report also presents other activities in the department, e.g. undergraduate education, the graduate study programme, knowledge transfer to industry, etc.

Mailing address:
Dept. of Computer and Information Science
Linköping University
S-581 83 Linköping
Sweden
Tel: int + 46 13 28 10 00
Telex: 8155076 LIUIDA S
Telefax: int + 46 13 14 22 31

Postadress:
Inst. för datavetenskap
Universitetet och
Tekniska Högskolan i Linköping
581 83 Linköping
Tel: 013 - 28 10 00
Telex: 8155076 LIUIDA S
Telefax: 013 - 14 22 31

CONTENTS

GRADUATE AND UNDERGRADUATE PROGRAMMES STAFF. Secretaries names are given in italics.

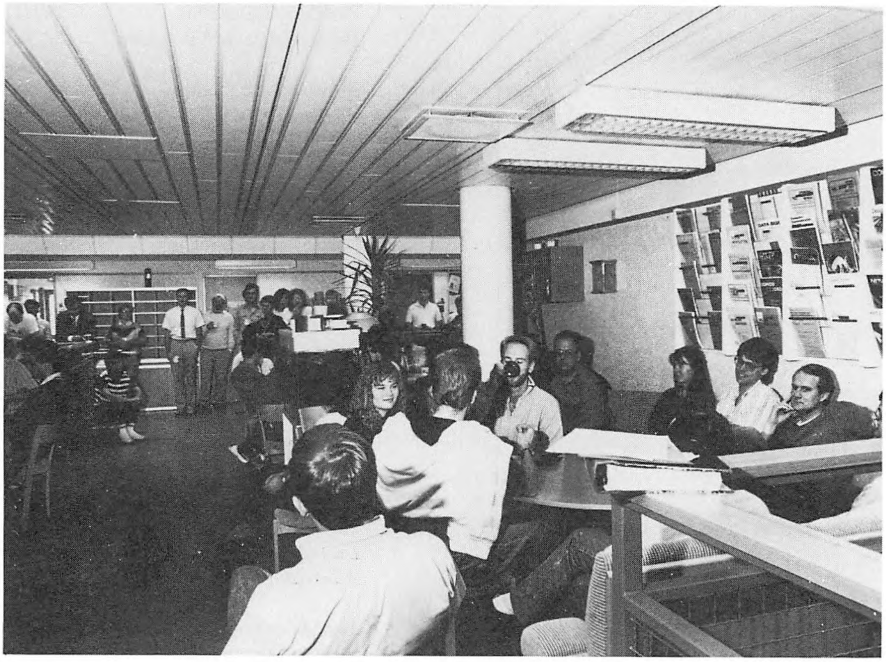
Graduate Studies Programme, general	Erik Sandewall	281408
	Johan Fagerström	281364
	Lillemor Wallgren	281480
	Lena Wigh	281706
Administration, graduate studies	Lillemor Wallgren	281480
Admittance of foreign graduate students	Johan Fagerström	281364
	Lillemor Wallgren	281480
Undergraduate Studies Programme, Engineering	Anders Haraldsson	281403
	Barbara Ekman	281410
	Madeleine Allert	281652
Undergraduate Studies, Humanities	Lise-Lotte Raunio	282430
	Carita Lilja	281458
	Siv Söderlund	281426
Undergraduate Studies Programme, general	Carina Björkman	282404

Telefax +46 13 142231

Telex 8155076 LiUIDA S

LABORATORIES:

Unit	*Leaders*	
Administrative Data Processing-Systems Analysis	Göran Goldkuhl	281452
	Carita Lilja	281458
Application Systems	Sture Hägglund	281431
	Britt-Marie Ahlenbäck	281318
Complexity of Algorithms	Per-Olof Fjällström	282412
	Bodil Mattsson Kihlström	282169
Computer Aided Design	Krzysztof Kuchcinski	281883
	Bodil Mattsson Kihlström	282169
Computer Assistance in Engineering	Anders Törne	282365
	Gunilla Lingenhult	282297
Industrial Software Technology Project	Bengt Lennartsson	281427
	Gunilla Lingenhult	282297
Knowledge Representation in Logic	Erik Sandewall	281408
	Anne-Marie Jacobson	281496
	Lillemor Wallgren	281480
	Lena Wigh	281706
Library and Information Science	Roland Hjerppe	281965
	Barbara Ekman	281410
Intelligent Information Systems (LINCKS)	Lin Padgham	282066
	Lillemor Wallgren	281480
	Anne-Marie Jacobson	281496
Logic Programming	Jan Maluszynski	281483
	Gunilla Lingenhult	282297
Natural Language Processing	Lars Ahrenberg	282422
	Lisbeth Linge	281472
People, Computers and Work	Toomas Timpka	281471
	Lena Wigh	281706
Programming Environments	Peter Fritzson	281484
	Gunilla Lingenhult	282297



1.

Introduction and overview

This is the July, 1989 issue of the overview report of research done at the Department of Computer and Information Science (IDA) of Linköping University. The report period this time is January 1988 - June 1989 as we are converting from calender year to academic year presentation.

IDA has now existed for six years. During this period several educational programmes with a large number of courses have been developed for the undergraduate education, new research laboratories have been established, and there is an effective infra structure in terms of computer systems, system support staff, and organization in general. Today IDA has about 150 employees (about 20 with a PhD), an annual budget of 39 MSEK (5.7 mill. $), about 100 undergraduate courses for 1200 students, and about 60 fully active graduate students. About 35% of the money and efforts 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 nine research labs and two research groups. A new activity normally starts as a project, promoted by one or several of the existing labs. When it has matured and gathered useful knowledge and experience within its area a group is created. A lab is characterized by its long term committment to development and maintenance of knowledge within a defined area, and by its long term responsibility for graduate students. We have, however, applied create and delete operations to our lab structure, as well as split and join.

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
LIBLAB	(Hjerppe)	for library and information sciences.
LOGPRO	(Maluszynski)	for logic programming,
NLPLAB	(Ahrenberg)	for natural language processing,
PELAB	(Fritzson)	for programming environments,

RKLLAB	(Sandewall)	for representation of knowledge in logic,
LINCKS	(Padgham)	for intelligent information systems
MDA	(Timpka)	for people, computers and work
IPT	(Lennartsson)	industrial software technology project

During that last years there have been some changes in the research organization. The former Artificial Intelligence Environments Laboratory was merged with the Laboratoray for Representation of Knowledge in Logic, which in its turn recently split off an independent group (LINCKS) in the area of intelligent information systems. Another group (MDA) has been created for interdiciplinary studies of people, computers and work, in particular human-computer consultation situations in health care (joint studies with Department of Community Medicine).

There have also been some changes in lab leadership. Andrzej Lingas was appointed professor of computer science at Lund University 1988 and Per-Olof Fjällström took over as new leader in the Laboratory for complexity of algorithms from January 1989. Bengt Lennartsson is starting a new independent software engineering project and Peter Fritzson is new leader for the Programming Environments Laboratory from the summer 1989. Professor Harold W. Lawson, who has been on leave for some years, formally resigned from his professorship in telecommunications and computer systems at the end of 1988.

Within the Center for Industrial Information Technology, CENIIT, our department has expanded its activities. A new laboratory for Computer Assistance in Engineering has been created, with Anders Törne, formerly at ABB Corporate Research, as lab leader. Lennartsson's project in the area of industrial software technology is also funded by CENIIT. Engineering databases, computational geometry, natural language applications, user interface management systems and plan-guided systems are other areas of study in CENIIT projects.

External cooperation has been intensified during the last years. IDA has taken a leading in 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. Within the national information technology programme (IT4), IDA is also 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.

The successful Knowledge Tranfer Programme (KTP/KÖP), which has been in operation since 1984, includes recent affiliations by Pharmacia, Volvo and TeleLOGIC Programsystem. Commissioned educational activites 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 1988-89.

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 aministrative 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 in the 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 lead by a department board (institutionsstyrelse), whose chairman ("prefekt") is Bengt Lennartsson. The two main areas of activity are reflected in the two subordinate committees:

- the committee for undergraduate teaching, whose chairman is Anders Haraldsson;
- the research committee, whose chairman is Erik Sandewall.

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 lab leaders;
- two undergraduate teaching groups, one for the teaching in the School of Engineering ("tekniska högskolan"), and another for the teaching in the School of Arts and Sciences ("filosofiska fakulteten"). The teaching groups are each headed by a "studierektor", namely Anders Haraldsson and Lise-Lotte Raunio. 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 de-centralize 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

We are 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 those 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 tend to lag behind. The situation for IDA illustrates this situation to a rather extreme degree. Only about 25% of the resources for research and graduate education are 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 and the applied goal should be achieved reasonably well. We also recognize the importance of continuous interaction between basic and applied research in our field.

Besides the external factors, 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 been complemented with research also in areas such as

mathematical logic for knowledge representation
logic programming
complexity theory
library science
information systems
human-computer interaction
industrial information technology

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 described in chapters 2-13, attempts to make a reasonable trade-off between concentration and breadth. Our main sponsor, STU, established a "ramprogram" (literally, "frame program") for research in information processing during the period 1980 to 1985. That program was 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 five year long period. A new five year (2+3) frame program in our area started January 1988.

The STU program described above is part of the national information technology program, which also contains a substantial part aiming at industrial R&D projects. A special effort in that part is joint activities between our university and the Defense Research Institute, where IDA participate in cooperative research on AI. 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 nine research laboratories and two smaller groups in the department. Each lab 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 "home". 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, headed by Erik Sandewall. 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 which 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 does 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 (for instance of a robot arm in a bounded space).

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 Kuchinski

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 is heavily engaged in cooperative efforts with for instance the mechanical engineering department. Close links to ASLAB are also maintained.

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 catalogs 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.

LINCKS Group - Intelligent Information Systems,
Lin Padgham

which studies theory and methods for distributed information management, including object-orientation and inheritance strategies. In a first step an object-oriented database, with a distributed architecture, support for history maintenance and parallel editing, has been developed. The group is a spinoff from the RKL laboratory, with which it still is closely related.

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 Group - People, Computers and Work
Toomas Timpka

which studies the use of interactive information systems in the context of the working environment and in particular the consultant-client relationship. The MDA project is specifically aimed at investigating computer support for pairs of users, one of whom is giving professional advice to the other. The project is multi-disciplinary and carried out in cooperation with the Department of Community Medicine and the LINCKS group among others.

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 major role in the AI part of the European Prometheus (future road trafic) project.

IPT - Industrial Software Technology Project
Bengt Lennartsson

with the objective to study the programming-in-the-many and the programming-in-the-large aspects of software development and maintenance. The research emphasizes close contacts with industry and experiences from industrial-size projects. The project is affiliated with PELAB, but has its main funding from CENIIT.

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 possibilites to utilize 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 were pursued as our department's contribution to CENIIT during 1988/89:

- *Geometrical algorithms.* The purpose of this activity is to develop competence in the area "efficient algorithms for three-dimensional geometrical problems and the planning of movements", in an active communication with applied projects in other departments. (Rolf Karlsson, Per-Olof Fjällström)
- *Engineering databases.* This area covers issues related to database technology, with a special emphasis on databases to be used in support of design, development and maintenance of (large) technical systems. Activities involve several laboratories in IDA. (Area coordinator: Sture Hägglund)
- *User Interface Management Systems.* This area covers certain aspects of human-computer interaction, with a special emphasis on tools for design and management of user-computer dialogues. (Area coordinator: Sture Hägglund)

- *Plan-guided systems for autonomous vehicles and computer-integrated manufacturing.* This area is for the time being covered by existing projects in IDA's laboratories without direct CENITT funding. (Erik Sandewall)

Additional CENIIT projects in IDA starting 1989 are:

- *Computer support for automation*, which is concerned with advanced techniques in using computers to support automation and control of manufacturing processes. (Anders Törne)
- *Industrial software technology*, which studies industrial-scale aspects and experiences of software development and maintenance. (Bengt Lennartsson)
- *Conceptual text representation for generation and translation*, where an intermediate representation language suitable for instance for translation between different natural languages is developed. (Lars Ahrenberg)

CENIIT projects in IDA are described in greater detail in chapter 13 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 14 of this report, with additional details given in appendices B and 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 4.5 year programme with an annual admittance of 120 students leading to a masters degree (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 are prepared 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, economical and legal aspects.

The computer science programme ("Datavetenskapliga linjen") started in 1982. It is a four-year programme focussing 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 linguistis, and less physics-related subjects. Discrete mathematcis, 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 contribute 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 that specialization.

The School of humanities and sciences offers since 1977 a three-year programme in Systems Anlysis (Computer Science and Business Administration). Usually 60 students are accepted annually. Since the courses are now being revised, only 30 students are admitted at present. This programme aims at professional activities of design, evaluation and implementation of computer-based information systems. Because of that, 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 courses in this curriculum.

IDA's work in the area of administrative data processing and system analysis has been plagued for many years with a number of problems. The undergraduate study-line in Systems Analysis ("systemvetenskapliga linjen") is badly under-funded. On the research side, there are organizational problems along the administrative borderline between 'engineering' and 'social sciences'. IDA brought out these problems for concrete discussion in 1986, and there has been a fairly lengthy debate about how to proceed for the future. Some temporary solutions have been created, but the main problems still remain to be solved.

1.4.2 Graduate education

The department has given high priority to the task of organizing a comprehensive graduate study programme, with coordinated set courses given in addition to support for thesis work in the research groups. This year a new model for supervision has been adopted, were each graduate student is assigned a supervisor committee consisting of one main and two assisting advisors.

It will naturally take a number of years before improvments in the educational activities show up in the form of increased production of PhDs. During 1988/89 three PhDs and six licentiate exams were awarded in the department. This represents a significant share of the higher examina in computer science awarded in Sweden (7 PhD on software-related subjects in 1988/89).

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

1.4.3 Continuing education

A separate programme for continuing education activities is organized in the department. The most prominent part is suitably adapted academic courses given for people from industry. Typically such courses are organized for half-times studies and given in such 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.

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 LM 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. Thus it was given during 1988 for ASEA Brown Boveri in Västerås and a new programme started in the spring 1989 in Gothenburg, with participants from Volvo and Chalmers.

Linköping University is actively participating in the European PACE programme for continuing education. EuroPACE offers courses and seminars distributed via satellite transmission. IDA is participating in the planning of courses in the AI area.

The department is also arranging shorter courses and seminars for the purpose of technology transfer. Last years activities include for instance a one-week course on software engineering with professor David Parnas.

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.

Continuing education activities were described in section 1.4.2. Other activites are presented below.

1.5.1 Knowledge Transfer Program to industry.

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 to restricted in order 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 six companies have joined the programme, namely in the order of the time for their affiliation: S-E-Banken, Ericsson, ASEA, Alfa-Laval, Philips and Sandvik Coromant. All these are large multi-national manufacturing companies, with the exception of S-E-Banken, which is a major Swedish bank.

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.

The major area of interest for the companies participating in the knowledge transfer programme has been AI and expert systems, but software engineering and human-computer interfaces are other examples of areas covered by the programme.

In recent years, we have extended the knowledge transfer programme with new forms for industrial cooperation. Thus we offer affiliated membership for such companies, with which we have established a working infra structure of personal contacts and a common background. In these cases customized agreements are defining the contents of the cooperation. Such agreements on joint efforts have been arranged with e.g. Volvo, Pharmacia LKB Biotechnology, Ericsson Telecom and TeleLogic Programsystem.

1.5.2 R&D consortia

IDA is strongly committed to the conviction that effective means of transfering 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.

LAIC - Linköping AI Consortium

An activity initiated within the national information technology programme (IT4) is the establishment of LAIC, Linköping AI Consortium, which incorporates the cooperative research programme in AI with the Defence Research Institute and participation in the AI subproject of the Prometheus Eureka project together with the Swedish car manufacturers Volvo and Saab, as mentioned above.

One major theme for R&D efforts within LAIC is *plan-guided systems*, as described in chapter 10 of this report, with applications for autonomous vehicles and mission planning. Examples of other areas to be pursued within LAIC are geographical information systems and systems supporting strategical decision making.

FOKUS - Knowledge-based and user interface systems

Since 1988 a joint project in the area of human-computer interaction in intelligent systems has been pursued together with the Defense Research Institute (Foa) in Linköping (Hans Marmolin, Tommy Nordkvist) with partial funding from the national information technology programme (IT4). Presently this project has been reorganised as a part of a larger effort in the form of a R&D consortium for knowledge systems and human-computer interaction, FOKUS. In addition to Foa, the former spinoff company Epitec is actively participating in FOKUS projects.

1.5.3 Spinoff Companies.

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 spinoff companies, and we may also encourage IDA employees to form new spin-off companies in order to catch an opportunity.

The significance of university spinoff companies for industrial growth is well known. One part of the previous artificial intelligence laboratory split off a few years ago and formed Epitec AB. The company has presently about 25 employees and is active in development of tools for knowledge systems, in development of applications for clients in industry, in consulting and in training of knowledge engineers.

As an even earlier spinoff, Jerker Wilander and Kenth Ericson founded the company Softlab AB in Linköping in 1981. Softlab is working in the areas of compiler design, certain aspects of database technology and advanced tools for software development The company is growing steadily and is presently investing in a new building in the Mjärdevi Science Park area close to the university.

There are also a number of software companies founded by former students, such as e.g. TeleLOGIC Programsystem AB (programming support environments, user interface management systems, datacom), having close contacts with the department and active in transferring software originating from research projects into commercial products.

The rapidly growing Science Park contains a number of other spinoffs, or other software and harware 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.

1.6 International Cooperation.

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

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

- the COST-13 project, 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). This project is now being completed, but other COST project participation is under negotiation.

- the PROMETHEUS project, which is part of the European EUREKA programme, and which has entered its operational phase during 1988. 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.

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

1.7 Research Facilities.

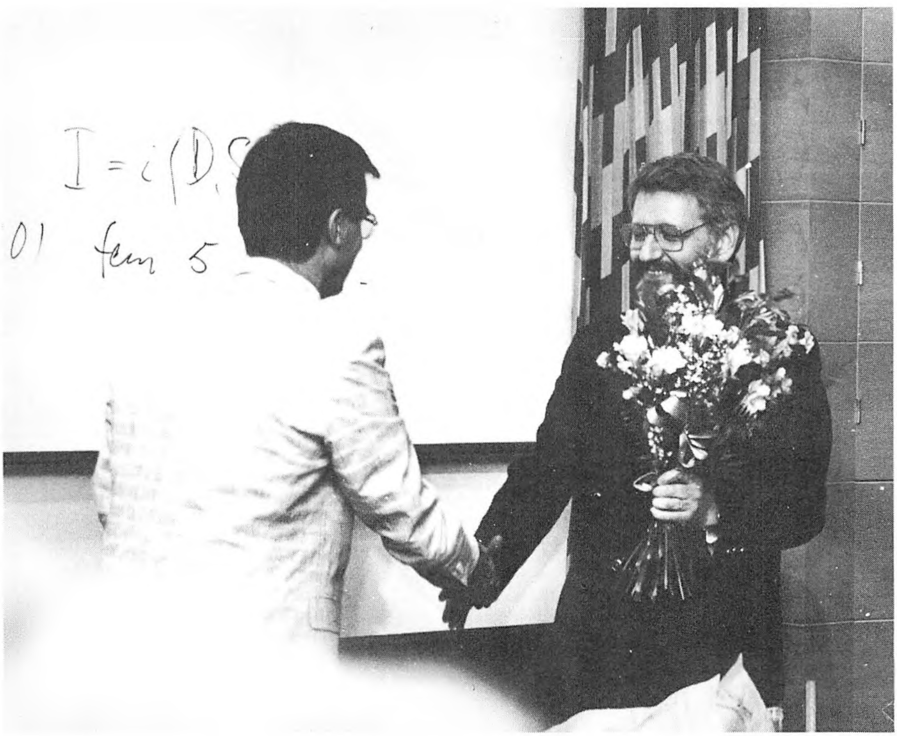
The main part of IDA is situated in the university's E building. About one third of the department's personnel is located to the adjacent B building. The premises are in general well planned and very nicely designed, by unfortunately at present heavily overcrowded. The inability of central authorities to provide new office and laboratory space for expanding activities in a flexible way represents a serious source of inefficiency for the university's research and education undertakings.

With respect to equipment, IDA is presently in a transition phase where the DEC-system 2060s, which have been used as the backbone of computing services for research and undergraduate education, are replaced by networked workstations. In addition to the present facilities of about 30 Xerox AI workstations and 30 SUN Unix workstations, 145 new SUN Sparc workstations have been ordered to replace the 2060s. Further, there are also a large number of Macintoshes, PCs and some VaxStations in the department, as well as some odd or special-purpose computers.

More details on equipment are given in appendix D.

1.8 Miscellaneous

Professor Harold W. Lawson resigned from his professorship in Telecommunication and Computer Systems by the end of 1988. Lawson has been on leave for some years and among other things worked with the establishing of a Swedish International University. From 1989 on he will work as an independent consultant in Stockholm.



Professor Harold W. Lawson resigning from the professorship in telecommunication and computer systems.

In honour of Lawson's services to the department a special fund was established in 1989, with an initial donation from the local branch of the Swedish Society for Information Processing (SSI Östergötland). The annual yield of the fund will be used for a stipend, to be given to a graduate student in computer science for outstanding achievements.

The first stipend was awarded Lin Padgham, who later this year finished her PhD as the first female in computer science (datalogi) in Sweden.

In the 1st Annual IDA Conference on Computer and Information Science, where graduate students in the department are invited to present their work, Björn Fjellborg's paper "Extraction of VLSI Pipeline Structures from Arbitrary Algorithms" was selected for the best paper award. Patrick Doherty was awarded the 1989 travel stipend of the Swedish AI Society, for his paper "A semantics for Inheritance Hierachies with Exceptions using a logic of Preferential Entailment".

2. ACTLAB The Laboratory for Complexity of Algorithms

Andrzej Lingas
Per-Olof Fjällström

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 arising in computer science 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.

The laboratory is a continuation of the so called Group for Complexity of Algorithms which in turn originated from a part of the former Group for Theoretical Computer Science in the spring of 1985. Since then, the members of the laboratory have published more than thirty papers in international journals and conference proceedings on computer science.

In 1988, ACTLAB coorganized the 1st Scandinavian Workshop on Algorithm Theory (SWAT'88) in Halmstad. Also, this year Andrzej Lingas was granted a professorship at the Department of Computer Science and Numerical Analysis at Lund University. Almost all members of ACTLAB'88 will very likely move to Lund in 1989. The exception here is Sven Moen who moved to PELAB. The task of reconstructing ACTLAB at IDA in 1989 was given to Dr. Per-Olof Fjällström. Per-Olof has previously done research at KTH on solid modeling,

---

The work in the Laboratory for Complexity of Algorithms is mainly supported by STU, The Swedish Board for Technical Development.

and received his Ph.D. degree 1985.

The last one and half year of the laboratory (group) research was mainly spent on a project called Efficient Algorithms and Data Structures for Geometric and Graph Problems funded by STUF and STU. The objectives of the project fall into three mutually interrelated categories of data structures, computational geometry and graph algorithms. The realization of the project in 1988 and the first six months of 1989 involved cooperation with the following long terms visitors: Dr. J. Katajainen sponsored by Finnish Academy of Science (14 months), Dr. C. Djidjev from Bulgarian Academy of Sciences (2 months), Dr. S. Gosh from TATA Institute of Fundamental Research, Bombay (2 months), and Dr. J. Sack of Carleton University, Ottawa (2 months). The results of the research on the project in 1988 and the first six months of 1989 can be structured as follows:

Computational Geometry:

Computational Geometry on a Grid
Theoretical Aspects of Robotics: Traversing a Maze with a Robot Arm
Voronoi Diagrams with Barriers and Minimum Weight Triangulation
Parallel Algorithms for Constructing Voronoi Diagrams
Constructing the Link Center in a Simple Polygon
Approximating the Complete Euclidean Graph

Data Structures and Sorting:

Heuristics for Optimal Search Trees
Adaptive Algorithms for Sorting

Graph Algorithms:

Fast Parallel and Sequential Algorithms for Subgraph Isomorphism
Parallel Algorithms for Planar Graph Searching and Matching

The considered problems have applications in graphics, robotics, numerical analysis (in particular, terrain interpolation), chemistry and optimization.
In 1988 and the first six months of 1989, several new problems within the STU-STUF project have been attacked by the laboratory members. Here we mention only the most important: computing the link center in a simple polygon, approximating the complete Euclidean graph (see Section 2.3.1) efficient parallel and sequential algorithms for sorting presorted files (see Section 2.3.2), and efficient parallel algorithms for directed reachability and matching in planar graphs (see Section 2.3.3).
The results on sorting presorted files have been obtained by Ola Petersson together with his supervisor, Dr. Christos Levcopoulos, and partly Dr. Katajainen. They comprise Ola Petersson's licentiate thesis, which was successfully defended in April 1989.

In addition to STU and STUF support, research at ACTLAB is also funded by CENIIT, that is, Centrum för Industriell Informationsteknologi. Since July 1988, CENIIT supports a number of application oriented research projects at the University in Linköping. Rolf Karlsson was responsible for one of these projects. The topic of this project is efficient algorithms for three-dimensional geometric problems, and motion planning. In January 1989, when Rolf moved to Lund, the responsibility was transferred to Per-Olof Fjällström.

The first six months of the project resulted in two research reports. Also, Dr. J. Sack's visit was mainly funded by CENIIT. In 1989, the work in this project has mainly focused on two problems: polyhedral approximation of points in space, and a parallel algorithm for the convex hull of points in the plane. These problems are further described in Section 2.3.1.

Besides the research, the laboratory has undertaken important consulting and educational tasks on aspects of algorithm analysis and complexity theory within the department, and has been open to cooperate with other laboratories and departments.

2.2 Laboratory Members

Laboratory leadership :

Andrzej Lingas, Ph.D. (until 31.12.1988)

Per-Olof Fjällström, Ph.D. (since 1.01.1989)

Secretary: Bodil Mattsson-Kihlström

Supervisors:

Rolf Karlsson, Ph.D.

Christos Levcopoulos, Ph.D.

Graduate Students:

Sven Moen

Ola Petersson

Jan Petersson (starting 1989)

2.3 Current Research

2.3.1 Computational Geometry

Polyhedral Approximation of Points in Space (Per-Olof Fjällström)

In terrain modeling 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 of the terrain 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. 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 is necessary.
In this project the following problem is investigated. Let T(S) be a 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.

Finding the Convex Hull of Sorted Points in Parallel
(Per-Olof Fjällström, Christos Levcopoulos, Ola Petersson)

Given a set S of points in the plane, the problem is to determine which points that lie on the perimeter of the smalles convex region containing S. In the special case we study, the points in S are assumed to be sorted by increasing x-coordinate. Several sequential and parallel algorithms have been proposed for the general problem. For the case of sorted points, Goodrich has presented an optimal parallel algorithm for the so-called CREW PRAM computational model. In this model it is assumed that several processors can simultaneously read, but not write, in the same memory cell. Together with Dr. Jyrki Katajainen, we are developing an optimal algorithm for the so-called CRCW PRAM model, where both simultaneous read and write operations are allowed.

Computational Geometry on a Grid (Rolf Karlsson)

Computational geometry studies the computational complexity of finite geometric problems. This research focuses on problems where geometric objects are defined by edges between points taken from multi-dimensional grids. Typical problems we consider are: finding closest points, determining connected components, and computing all line segment intersections. The solutions are based on an efficient point location algorithm or use a new data structure, the interval trie, when sweeping the plane with a line. For problems in higher dimensions, we use a divide-and-conquer technique until the dimension is reduced to 2 (or 3) where the sweep-line algorithms apply. The efficient methods we present should be useful within computer graphics and VLSI. For instance, when implementing geometry routines in computer graphics the domain is a moderate sized raster. Our attention is concentrated to orthogonal objects (the edges are parallel to one of the coordinate axes). VLSI technology, for example, often uses only a fixed number of orientations for the object boundaries and wires. Much of this research is joint work with Dr. Mark Overmars of Utrecht University. Some of the results have been published and some are submitted for publication.

Theoretical Aspects of Robotics (Rolf Karlsson)

Motion planning, which tries to move an object from one position to another, while avoiding obstacles, has attracted much interest in recent years. In this preliminary investigation we consider the problem of efficiently propagating a linkage through an orthogonal maze. The linkage moves strictly in two dimensions, no crossing is allowed. Given the layout of a 2-dimensional maze with corridors of width w, we express the time complexity of moving a robot arm (linkage) of k equal-length links through the maze using a basic 2-joint motion. We give an algorithm which pushes a linkage with link length *l= 2sqrt(2) w - epsilon* around one corner in *O(log(w epsilon))* time, combine this with the complexity of moving along the corridors, and generalize it to apply for maze routes with corridors of different widths. The derived traversal complexity we can use to efficiently find a shortest route.

Voronoi diagrams with barriers, the shortest diagonal problem and minimum weight triangulation (Andrzej Lingas)

Planar figures which are called planar straight-line graphs (PSLG for short) are considered. A PSLG G is a pair (V,E) such that V is a set of points in the plane and E is a set of non-intersecting, open straight-line segments whose endpoints are in V. The points in V are called vertices of G, whereas the segments in E are called edges of G. If G is a simple cycle, it is just a (simple) polygon. If G has no edges, it is a planar point set. A diagonal of G is an open straight-line segment that neither intersects any edge of G nor includes any vertex of G and that has its endpoints in V. A triangulation of G is a maximal set of non-intersecting diagonals of G. The greedy triangulation of G is the result of iterating the following step: insert a shortest diagonal of G that does not intersect those already in the plane.
To solve the problem of finding a shortest diagonal of a PSLG, a new generalization of Voronoi diagrams of planar finite point sets to include diagrams of PSLG's has been used. The Voronoi diagram with barriers of G (Vorb(G) for short) is a net that together with G partition the plane into the regions P(v), v in V, such that a point p is inside the region P(v) if and only if (p,v) is the shortest straight-line segment connecting p with a vertex in V that does not intersect any edge in E. If G has no edges then Vorb(G) is just the standard Voronoi diagram of the set V of points in the plane. Wang and Shubert independently considered such diagrams and showed that they can be constructed in time O(n log n)
A simple characterization of shortest diagonals of G in terms of Vorb(G) has been provided. This combined with the algorithm of Wang and Shubert yields O(n log n) time solution to the shortest diagonal problem.
Our method of finding a shortest diagonal of a PSLG implies that the greedy triangulation of a planar point set or any PSLG can be computed in time O(n**2 log n) and space O(n). The most efficient known algorithm for computing the greedy triangulation of a planar point set which is due to Gilbert takes O(n**2 log n)-time and O(n**2) space. Instead of computing the Voronoi diagram with barriers of the current PSLG from scratch every time after inserting a shortest diagonal, the current diagram can be adequately

updated. We have shown that the latter method applied to a set of n points uniformly distributed in a unit square takes O(n (log n)**1.5) expected-time. The best previously known upper-bound on the expected-time performance of an algorithm for the greedy triangulation was O(n**2).
Further important applications of Voronoi diagrams with barriers can be found in planar nearest visible neighbor, shortest-path, minimum spanning tree and matching problems in the presence of barriers.
The obtained results have been already published or accepted for publication.

Parallel Algorithms for Constructing Voronoi Diagrams (Christos Levcopoulos, Andrzej Lingas)

A considerable amount of work has been done on parallel computational geometry. The efficiency of parallel algorithms depends crucially on the computational model assumed. The fastest known algorithm for constructing the Voronoi diagram of a set of n points in the plane (see the previous section) takes O((log n)**2) (worst-case) time using a Concurrent Read exclusive Write Parallel Random Access Machine with O(n) processors.
We have shown how to construct the Voronoi diagram of a set of n points in the plane in O(log n) expected-time using a Concurrent Read Concurrent Write Parallel Random Access Machine with O(n log n) processors. Our algorithm cannot be seen as a direct parallelization of any known sequential algorithm for Voronoi diagrams. When deriving the expected-time bound we assume that sites are independently drawn from a uniform distribution in a given square. Actually, our algorithm does not use concurrent writes but for sorting of integers in the range 1 through n. Thus, if it is possible to sort such integers in O(log n) time, with O(n log n) processors without concurrent writes, our algorithm can be also implemented without concurrent writes.
The above results are joint work with Dr. J. Katajainen, and they have been already published.

Computing the Link Center in a Simple Polygon (Andrzej Lingas)

A link path p between a pair of points v and w in a simple polygon P is a polygonal line inside P connecting v and w. The link distance between v and w is the minimum number of segments required to connect v to w, over all link paths between v and w. The link center is the set of all points p in P for which the maximum link distance between c and any point in P is minimized.
The problem of finding the link center of a simple n-vertex polygon P occurs in motion planning. It had previously been solved in quadratic time. It was posed as an open problem as to whether a faster algorithm exists for determining at least one point inside the link center. This question has been answered affirmatively. We have designed an algorithm that determines, in O(n log n) time, either a triangle inside the link center or the entire link center. As a consequence, we have also obtained an O(n log n) time solution to the problem of determining the link radius of P. Both results are improvements over the O(n**2) bound previously established. They are joint work with Dr. H. Djidjev from Bulgarian Academy of Sciences, and Prof. J.R. Sack of Carleton

University, already published in an international conference proceedings.

Approximating the Complete Euclidean Graph

(Christos Levcopoulos, Andrzej Lingas)

Consider a set S of n points in the plane. We would like to design a planar network between the points in S that approximates the complete Euclidean graph on S in the following sense: there exists a constant c such that for any pair of points in S there is a path in the network that connects the points and is of length bounded by c times the straight-line distance between them. Planar networks approximating the complete Euclidean graph can be applied in the design of route networks and transmission networks, and in the design of geometric and combinatorial algorithms.

In the above design, both the goodness of approximating the complete Euclidean graph and the cost of the resulting network are important. In the simplest case, the cost of a planar network is proportional to the total length of its edges. Clearly, if the network is connected, in particular, if it approximates the complete Euclidean graph on S, it has length not smaller than that of a minimum Euclidean spanning tree of S. Therefore, it is of interest to ask whether there exists a planar straight-line graph on S that (1) approximates the complete graph on S and (2) has length within a constant factor from the length of a minimum Euclidean spanning tree of S.

Among others, Dobkin et al. showed that the Delaunay triangulation of S in the Euclidean plane gives an approximation of the complete graph on S within a constant factor. However, it is not difficult to construct examples of sets S where the Delaunay triangulation of S does not satisfy the second requirement above.

We construct a planar straight-line graph that satisfies both requirements. We do it by pruning the Delaunay triangulation of S. Given the Delaunay triangulation of S, the graph can be constructed in linear time.

The above result has been already accepted for publication.

2.3.2 Data Structures

Optimal Search Trees and Optimal Partitions of Polygons

(Christos Levcopoulos, Andrzej Lingas)

A journal, refined version of the paper on optimal binary search trees with zero key access probabilities (with applications eg. in point location) presented at ICALP'87, has been finished. In the paper, it has been shown that for an arbitrarily small positive constant *e* there exists a linear-time heuristic for such search trees, producing solutions within the factor of *1+e* from the optimum. Also, by using an interesting amortization argument, a simple and practical, linear-time implementation of a known greedy heuristic for such trees has been given. The above results have been obtained in a more general setting, namely in the context of minimum length triangulations of so-called semi-circular polygons. They have been carried over to binary search trees by proving a duality between minimum weight partitions of infinitely-flat semi-circular polygons into *m*-gons and optimal *(m-1)*-way search trees. This duality has also helped to obtain better heuristics or algorithms for minimum length partitions of polygons using known algorithms for optimal search trees. The above results have been obtained in cooperation with Dr. J. Sack, they the journal version has been already accepted for a publication.

Adaptive Sorting Algorithms (Christos Levcopoulos, Ola Petersson)

Sorting is probably the most well-studied problem in computer science. In many applications, the sequences to be sorted do not consist of randomly distributed elements, but are already nearly ordered (*presorted*). Most O(n log n) time sorting algorithms (for example, mergesort and heapsort) do not take the existing order in their input into account.
A *measure of presortedness* is an integer function on a permutation p of a totally ordered set that reflects how much p differs from the total order. The term presortedness was coined by Mehlhorn, who used the number of inversions (pairs of elements out of order) as a measure. Since then, numerous other measures of presortedness have been proposed in the literature. These include the number of runs (maximal ascending consecutive subsequences), and the minimun number of elements whose removal leaves a sorted sequence. Mannila formalized the concept of presortedness.
Mehlhorn gave an algorithm, A-Sort, which is efficient with respect to the number of inversions within the input sequence. Intuitively, a sorting algorithm is efficient with respect to a measure of presortedness if it sorts all sequences, but performs particularly well on those having a high degree of presortedness. Such algorithms are thus *adaptive* to the existing order within the input. The concept of optimality of a sorting algorithm, with respect to a measure of presortedness, was defined by Mannila. Other algorithms which are efficient, or optimal, with respect to some measure(s) of presortedness are natural merge sort, smoothsort, local insertion sort, try-to-merge sort, and melsort. Cook and Kim, and Mannila have studied the problem of designing adaptive sorting algorithms empirically.
1) Local Insertion Sort Revisited: Two measures of presortedness, motivated by a geometric interpretation of the input sequence, are introduced. These measures help us to characterize the performance of the local insertion sort algorithm (LIS) of Mannila. Guided by this characterization, we propose some modifications of the basic algorithm.
LIS sorts by inserting the elements, one by one, into a sorted sequence formed by the already inserted elements. The already inserted elements are implemented by a finger (search) tree, i.e., a level-linked (2-4)-tree with pointers to some of the leaves, the fingers. Finger search trees support very fast searching in the vicinity of fingers. We use one insertion finger, pointing at the

latest inserted element. The modifications of LIS are based on the following ideas.
a) Use many finger trees instead of only one, and maintain one finger in each tree.
b) Maintain many fingers in one single finger tree. The number of fingers can either be fixed beforehand or vary during the sorting process.
The paper explores different strategies for moving the fingers. Further, it provides a method for dynamizing the algorithm: during the sorting process, new fingers are allocated such that the algorithm sorts in optimal time.
This result, which is joint work with Dr. Jyrki Katajainen, University of Turku, has been accepted for publication.
2) Heapsort-Adapted for Presorted Files: A new sorting algorithm, *maxtree-sort* , which is optimal with respect to several measures of presortedness, is presented. The algorithm is motivated by a new measure of presortedness, which measures the oscillation within the input sequence. The measure is proved superior to several well-known measures of presortedness.
Maxtree-sort is a variant of heapsort. Unlike heapsort, however, it is adaptive to existing order among the input. Recall that heapsort keeps all the elements in a heap, and then repeatedly extracts the maximum element. The cost of extracting an element depends on the number of elements currently in the heap. In maxtree-sort this number depends on the oscillation in the input. If the oscillation is high, maxtree-sort degenerates to heapsort, and sorts in time Theta(n log n). If it is low, the sorting will be completed considerably faster than it would in heapsort. This result has already been accepted for publication.
3) An Optimal Parallel Algorithm for Sorting Presorted Files: A cost-optimal parallel sorting algorithm for presorted files is presented. The measure of presortedness considered is the number of inversions, and the model of computation is the exclusive read exclusive write parallel random access machine (PRAM for short), i.e., the weakest of the PRAM's. This is the first PRAM sorting algorithm which is optimal with respect to the number of inversions.
The algorithm first divides the input sequence into subsequences of equal length, and sorts each subsequence in parallel. The subsequences are then permuted, and elements causing inversions are removed. The removed elements are sorted in parallel, and merged with the non-removed elements, yielding the sorted output. The obtained result has already been presented at a conference.

2.3.3 Parallel and Sequential Graph Algorithms

(Andrzej Lingas)

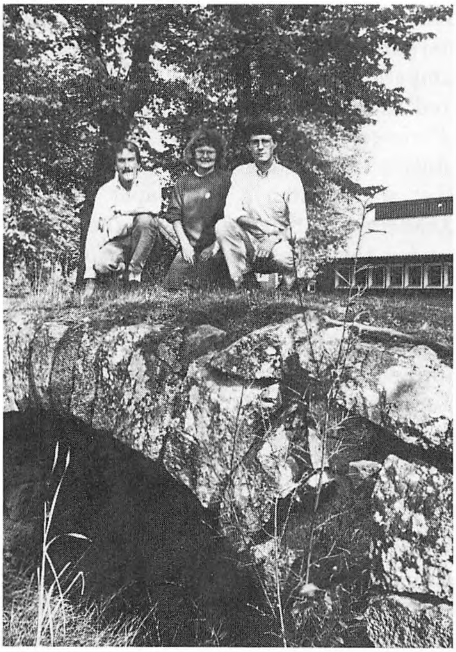
Subgraph Isomorphism

The problem of subgraph isomorphism belongs to the most general NP-complete graph problems and has many applications in computer science. It consists in determining whether a graph is isomorphic to a subgraph of another graph. The first known polynomial-time algorithm for subgraph isomorphism restricted to two-connected series-parallel graphs has been designed in a cooperation with Prof. M. Syslo of Wroclaw University. Series-parallel graphs are a popular model of electrical circuits. Thus, the important problem of deciding whether a circuit contains a given pattern can be modeled as the subgraph isomorphism problem for series-parallel graphs. It has been also shown that the subgraph isomorphism problem for two-connected series-parallel graphs can be solved by a random parallel algorithm running in poly-log time and using a polynomial number of processors. In addition, it has been proved that the problem of subgraph isomorphism restricted to two-connected graphs of constantly bounded separator and valence is in the class NC, i.e. can be solved by parallel algorithms running in poly-log time using a polynomial number of processors. The discussed results have been already published (among others in ICALP'88 proceedings).

Searching and Matching in Planar Graphs

Almost all known parallel implementations of basic problems for directed graphs (digraphs for short) , e.g. reachability, breadth-first search (BFS for short), shortest paths use matrix multiplication. Therefore the processor-time costs of these algorithmic solutions are trivially at least the number M(n) of operations required to multiply two n*n Boolean matrices (the best known result is presently O(n**2.376)). This situation is especially annoying in the case of sparse digraphs for which we can sequentially solve for instance the problem of reachability in time proportional to the number of edges and vertices by using depth-first search (DFS for short).

The directed reachability problem can be defined as follows: given a directed graph and its two distinguished vertices s and t, decide whether there exists a directed path from s to t in the graph. Clearly, the directed reachability problem can be easily solved by DFS or BFS in the input digraph. As no parallel algorithm for DFS or BFS of a digraph or even a planar digraph that runs in polylog-time and uses a sub-quadratic number of processors is presently known, no parallel algorithm for the directed reachability problem for planar digraphs running in polylog-time and using a sub-quadratic number of processors can be found in the literature. We develop an efficient parallel method for searching planar digraphs different from DFS and BFS. The main idea is to augment the input graph G by some edges appropriately such that any directed path in G can be simulated by a path of logarithmic length, and then to apply logarithmically shallow BFS in the augmented digraph. Given the augmented digraph, we can solve all the reachability problems for a distinguished vertex v of G (i.e. to report all vertices reachable from v by directed paths in G) in time O((log n)**2) using an exclusive read exclusive write parallel random access machine (PRAM for short) with O(n) processors. The augmented digraph can be build in time O((log n)**6) using a concurrent read concurrent write PRAM with O(n**(1.5)(log n)**4) processors. Using the above method of preprocessed searching in planar digraphs, the transitive closure of a planar digraph can be found in time O(log **6 n) using a concurrent read concurrent write PRAM with O(n**2 log**5 n) processors. The presented method of preprocessed searching of planar digraphs can find several further applications in the design of parallel algorithms for planar digraphs and planar graphs with small processors-time costs. For instance, it has been already successfully applied in a processor-time efficient parallel algorithm for maximum cardinality matching in planar bipartite graphs presented in a joint paper with Dr. E. Dahlhaus and Prof. M. Karpinski of Bonn University. The above results have been partially presented at international conferences.



ACTLAB personnel: Jan Petersson, Bodil Mattson-Kihlström and Per-Olof Fjällström

2.4 External contacts

Per-Olof Fjällström: Participated in the *ALCOM Summer School on Computational Geometry*, and the *ACM Symposium on Computational Geometry*, both held in Saarbrucken, Germany, June 1989.

Rolf Karlsson: Coorganized the *1st Scandinavian Workshop on Algorithm Theory (SWAT'88)* together with Andrzej Lingas, Bengt Aspvall (Bergen), and Svante Carlsson (Lund). Joint editor, with Andrzej Lingas of the SWAT proceedings published by Springer Verlag. Participated in the *29th Annual IEEE Symposium on Foundations of Computer Science,* New York, in October. Given seminar at National University of Singapore. Participated in the *ACM Symposium on Computational Geometry,* both held in Saarbrucken, Germany, June 1989.

Christos Levcopoulos: Presented his joint paper with Andrzej Lingas at the *Symposium on Optimal Algorithms,* Varna, Bulgaria, June 1989.

Andrzej Lingas: Visited the Department of Computer Science at Bonn University where started writing a joint paper with Dr. E. Dahlhaus and Prof. M. Karpinski. Presented papers at the *Workshop on Graph-Theoretic Concepts in Computer Science,* Amsterdam, in June 1988, and the *International Conference on Parallel Processing,* Chicago, in August 1988. Coorganized the 1st Scandinavian Workshop on Algorithms and Data Structures. Co-authored a paper presented at this workshop. Participated in the *15th International Colloquium on Automata, Languages and Programming,* Tampere, Finland, July 1988, where Prof. Syslo of Wroclaw University presented their common paper. Also in Tampere, became elected to the *Council of European Association for Theoretical Computer Science* as the first representative of Sweden. Finished a paper with Prof. J. Sack of Carleton University, and written a new paper with him and Dr. C. Djidjev from Bulgarian Academy of Science during their visit at IDA in July and August 1988. Attended the *6th Annual Symposium on Theoretical Aspects of Computer Science* , Paderborn, February 1989, where the above paper was presented by Prof. Sack. In April 1989, took part in the work of the program committee for the *1st Canadian Workshop on Algorithms and Data Structures* which will be held in August 1989, in Ottawa. Presented paper at the *British Coloquium on Theoretical Computer Science,* London, April 1989. In May 1989, gave a talk at the *International Institute of Computer Science,* Berkley, and attended the *ACM Symposium on Theoretical Computer Science* in Seattle.

Ola Petersson: Presented his joint paper with Dr. Christos Levcopoulos at the *8th Conference on Foundations of Software Technology and Theoretical Computer Science,* Pune, India, in December. Participated in the *3rd Aegean Workshop on Computing, AWOC 88,* Greece, and SWAT'88. Presented his joint paper with Dr. J. Katajainen and Dr. Christos Levcopoulos at the

Symposium on Optimal Algorithms, Varna, Bulgaria, June 1989. Participated in the *ALCOM Summer School on Computational Geometry*, and the *ACM Symposium on Computational Geometry,* both held in Saarbrucken, Germany, June 1989.

Courses for Graduate Students

An important task of the group has been to spread the knowledge of algorithm analysis and complexity theory among graduate students within the department. The following graduate courses has been offered for the academic year 88-89:

Computer Algebra

Survey of Parallel Algorithms for Memory-Shared Machines

Previously, the following courses were given by the group members:

Algorithm Analysis and Complexity Theory (83,84-85,88)

Mathematical Aspects of VLSI (84)

Search Structures (85)

Analysis and Complexity of Parallel Algorithms (86,88)

Amortized Computational Complexity (87)

Computational Geometry (87)

Lower Bounds Techniques (88)

3.

ASLAB
The Application Systems Laboratory

Sture Hägglund
Professor of knowledge-based systems

The research program in the Applications 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 major activity in the lab is the five-year research programme on *engineering environments for generic knowledge systems*, funded by the Swedish board for technical development, STU. Current project activities are summarized below.

Our research on support for generic knowledge systems has during the last year been carried out with a concentrated effort in an application domain defined together with Pharmacia LKB Biotechnology. The K-Linker project, led by Kristian Sandahl and also engaging Henrik Eriksson, Thomas Padron-McCarthy and Tomas Sokolnicki, aims at developing a tool-supported methodology for knowledge acquisition in restricted domains with a high degree of re-usability of represented knowledge. Our approach is based on a two-phase iterative development methodology and assumes that domain experts should be as active and as much in control of the knowledge

The work in ASLAB is mainly supported by STU, The Swedish Board for Technical Development. Additional funding comes from CENIIT, HSFR, Foa/IT4 and others.

acquisition process as possible. Important subthemes are knowledge-based experiment planning, intelligent tutoring, knowledge management strategies and an "active expert" development methodology.

Several prototype systems have been developed concerning planning of protein purification experiments. Experiences as reported in Eriksson's licentiate thesis illustrate the possibilities to customize development tools with respect to a conceptual model of the application domain.

Closely related to the K-Linker project is the PhD thesis work by Henrik Nordin, now continued at Information Science Institute, University of Southern California. Nordin is working in the context of Swartout's *EES (explainable expert systems)* approach, where he is interested in an explicit representation of constraints imposed by design decisions in knowledge bases, resulting in improvements in re-usability and maintainability of the knowledge. In a licentiate thesis by Tim Hansen, a similar model-based approach was applied for the purpose of showing how the explicit representation of malfunction behaviour of components in a technical system could be used to improve the diagnosis of faults, as exemplified in a separator trouble shooting expert system.

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 support dialogue interfaces to expert systems is being developed by Jonas Löwgren, with preliminary results reported in a licentiate thesis 1989. Löwgrens approach identifies two different structures in an expert system user interface; the *surface dialogue* and the *session discourse.* His proposal for mangaging 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. Rankin's work in this area, as presented in his recent licentiate thesis, is concerned with generalized methods for text generation in expert critiquing systems. 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 included and organizing the various parts of the text to form a cohesive unit. His approach uses speech act theory as a means of expressing the goals, a user model to influence the level of detail and Mann's rhetorical structure theory (RST) for the organization of the text. A system for RST generation has been implemented together with a number of rhetorical schemata and is reported in a master's thesis by Harrius.

Anne Steinemann, a PhD student at Stanford University who visited our group during 1988-89, has investigated the potential of expert critiquing for dealing with problems of multiple experts and conflicting expertise. Related research done by Peter Eklund on knowledge acquisition interfaces, partly based on Sowa's conceptual graphs, bridges the gap between critiquing and intelligent user interfaces in general. Eklund is originally from Australia, with a master's degree from Brighton in England, and he started as a graduate student in our group during 1988-89.

The bordering area between knowledge systems, database technology and decision support methodologies plays an increasingly important role. Our research covers several aspects of this interesting field. Initial investigations of object-oriented approaches common for knowledge bases and databases with application in engineering (Johansson) with CENIIT support are continued within the new CAELAB. Cooperation with the LINCKS group (Padgham) on related subjects is also focussing on decision support by encoding classification knowledge in type hierarchies with inheritance (Zhang). Another line of research concerns information management in the context of crisis management and emergency decision making, where we cooperate with the Defense Research Institute in studies of decision support in command systems (Näslund).

During 1988-89 ASLAB has intensified the external cooperation activities. Knowledge transfer activities with companies such as Pharmacia, Volvo, ABB, and TeleLogic Programsystem provide us with a direct contact with problems and experiences from a real-world environment. Within the framework of our participation in the University / Defense Research Institute (FOA) applied research programme managed by IT4 (the Swedish national information technology programme) we have formed a R&D consortium with Foa and Epitec in the area of knowledge-based systems and user interfaces, with a special emphasis on evaluation methodology. In connection with the planning of contacts with the ESPRIT programme, this consortium developed a demonstration prototype for a knowledge-based system for dialogue design review (KRI), based on the critiquing approach. Continued efforts in this area include the integration with UIMS technology.

Another new cooperative effort is the CAFKA project together with cognitive psychologists in Stockholm (Yvonne Waern), which undertakes a study of the communication that goes on in the development and use of expert systems. ASLAB also has taken an active part in starting up CENIIT projects in *engineering databases* (Olof Johansson), *Computer support for automation* (Anders Törne) and *UIMS*. These projects are now continuing within the new CAELAB, which for the time being is closely linked with ASLAB.

Core issues and central aspects of our research are discussed in greater detail later in this chapter.

3.2 ASLAB personnel

The following list presents persons contributing to project activities in ASLAB during 1988-89.

Project leadership/thesis supervision:

Sture Hägglund, PhD, professor
Kevin Ryan, PhD, guest researcher, (CENIIT, 1988-89)
Anders Törne, PhD (from spring 1989, now heading CAELAB)

Britt-Marie Ahlenbäck, secr. (from summer 1989)
Gunilla Lingenhult, secr. (until summer 1989)

Graduate students:

Henrik Eriksson, MSc, Tekn. Lic.
Peter Eklund, M.Phil.
Tim Hansen, MSc, Tekn. Lic. (until 1988)
Jonni Harrius, MSc, (starting summer 1989)
Olof Johansson, MSc (now in CAELAB)
Jonas Löwgren, MSc, Tekn. Lic.
Henrik Nordin, MSc, Tekn. Lic. (at ISI/USC from spring 89)
Torbjörn Näslund, BSc
Thomas Padron-McCarthy, MSc
Ivan Rankin, BA, Fil. Lic.
Kristian Sandahl, MSc, Tekn. Lic.
Tomas Sokolnicki, MSc
Tingting Zhang, MSc
Anne Steinemann, MSc, visiting PhD student, 1988/89

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.

Carl Ericsson, project assignment fall 1988 (now at Epitec)
Staffan Löf, Epitec AB
Hans Marmolin, Foa 5
Tommy Nordkvist, Foa 5
Yvonne Waern, PhD, Dept. of Psychology, Stockholm University

3.3 Review of major research themes

Knowledge-based approaches to systems development have gained considerable interest during recent years. Numerous tools for development of knowledge systems have been developed for research or commercial purposes. Still it appears that the full potential in the area will not be released until more effective techniques for delivery and integration of finished systems into the applications environment are presented.

We feel that this is not merely a question of improving the functionality and efficiency of developed expert systems, nor of modifying tools for a more easy integration with conventional hardware and software. Rather we believe that an important part of the solution to the problem of making knowledge-based system widely available in practical use, is

- to identify classes of generic applications and tasks in order to build customized support in these areas;
- to provide development support at a level which allow domain experts to take the main responsibility for creating knowledge-based applications in their specific domain;
- to prepare for mechanized or semi-automated transformation of knowledge bases from the development environment into existing delivery environments;
- to strengthen the emphasis on support for flexible dialogue management in the end user system, including explanations, critiquing, tutoring and other consultation models aiming at transferring knowledge and skills from the system and its knowledge base to the user.

With this approach we emphasize the support for

- creation of knowledge bases, which contain modular descriptions of entities and process knowledge encountered in a particular application domain. These modules should be reusable in such a way that they can be adapted and combined for use in new applications.
- knowledge acquisition and knowledge maintenance in the process of creating, customizing and maintaining new applications, typically within a predefined domain.
- a development methodology based on iterative design and rapid prototyping, concerning both the knowledge base structure, problem solving techniques and the user interface.
- delivery of applications into productions environments while preserving some knowledge acquisition and maintenance facilities. It is assumed that delivery environments may vary with respect to various interfaces (hardware, software, databases, manual routines, organizational setting, etc.)
- transparency of the knowledge base, in the sense that knowledge structures can be translated and presented in a suitable form for a human user to understand. In particular we want to support direct training of the user and also various forms of giving consultative advice including critiquing approaches.
- effective human-computer interaction, including task-oriented design methodologies with supporting architecture and tools.

Our work starts from the assumption that new systems are not developed from scratch but rather from an existing knowledge base, that application systems must be allowed to evolve and change over their life-time and that the development support environment should recognize characteristics of prospective delivery environments without introducing unnecessary restrictions during knowledge acquisition and application build-up.

3.3.1 Knowledge acquisition and maintenance

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 focusses 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.* This calls for a meta-level tool support strategy, which allows the knowledge engineer to create with minimal effort a customized environment for a specific genering application domain (i.e. a conceptual framework for a 'family' of knowledge systems).

One example of a special class of advisory systems of interest to us is *initial advice consultation systems*, i.e. knowledge-based systems for supporting a non-expert user to make decisions and solve problems. An example of what characterizes this area is the need to understand the limitiations of knowledge in the system, i.e. it is necessary to be able to decide when a case should be handed over to a real expert. We have identified and investigated such situations in several different applications, where we have been involved in development projects (references below are to licentiate theses):

- Sales support for advanced technical equipment, where we developed a system for welding robot configuration relative customer needs together with ASEA [Sandahl 87].
- Initial trouble shooting of process plants, where we have cooperated with Alfa-Laval on separator diagnosis systems [Hansen 88].
- Economical and legal advicing in a bank, where we worked together with S-E-Banken on a consultation system for back-office applications [Nordin 86].
- Advice on selection of the most appropriate testing techniques for blood analysis, where we cooperate with the medical informatics department [Sandahl 87].
- Planning of experiments in protein purification, where the knowledge bases are provided by experts from Pharmacia LKB Biotechnology [Eriksson 89].

The recurring pattern in these applications is the ambition to distribute expertise to less specilized people in order to support their ability to solve problems or identify the need for a consultation with a qualified specialist. The convenient management of this generic application problem calls for a solution which makes it easy for domain experts to feed information into the knowledge base and for end users to access and utilize that knowledge.

The availability of powerful techniques to represent and manipulate domain knowledge about objects, concepts and procedures, etc. will also in a decisive way improve the possibilities to employ methods in the tradition of the *rapid prototyping* approach to systems development. Experiences from previous projects in the area of office information systems led us to the formulation of *stepwise structuring* as a generalization of rapid prototyping, both being examples of methods for *iterative development* of software. Using a stepwise structuring approach essentially means that the degree of formalization (and thus the possibility for automated operations) of information is gradually increased during successive implementations of working prototypes or system generations allowing *adaptive maintenance* of software [Hägglund, North-Holland 89]. Nordin's work at ISI/USI illustrate a related approach to rapid prototyping [Swartout, Nordin, 89].

3.3.2 Knowledge-based experiment planning

In the K-Linker project and in applied experiments concerned with biotechnical experiment planning, we are now 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 design iterations where architectural issues are tested and experiences generalized into design revisions.

The selected domain is advisory systems for the planning of biotechnical experiments, with practical applications developed in cooperation with Pharmacia LKB Biotechnology AB. Experiments involve use of complicated laboratory equipment and domain experts are regularly consulted both for initial advice and when experiments fail. In addition to advisory systems for experiment planning, training and tutoring support is also of interest.

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. That includes customized support for knowledge acquistion 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.

An initial prototype for a knowledge-based experiment planner for purification of membrane proteins in solution was developed in cooperation with experts at

Pharmacia during 1988. In a continued effort a generalized design was employed, covering a family of similar systems in order to allow a more uniform treatment of planning strategies and to investigate the possibility of mechanizing the knowledge acquisition process for this type of systems. It was shown that with this generalized development system, P10, independent domain experts could produce a number of experiment planning applications with a minimal need for knowledge engineer support [Eriksson 89].

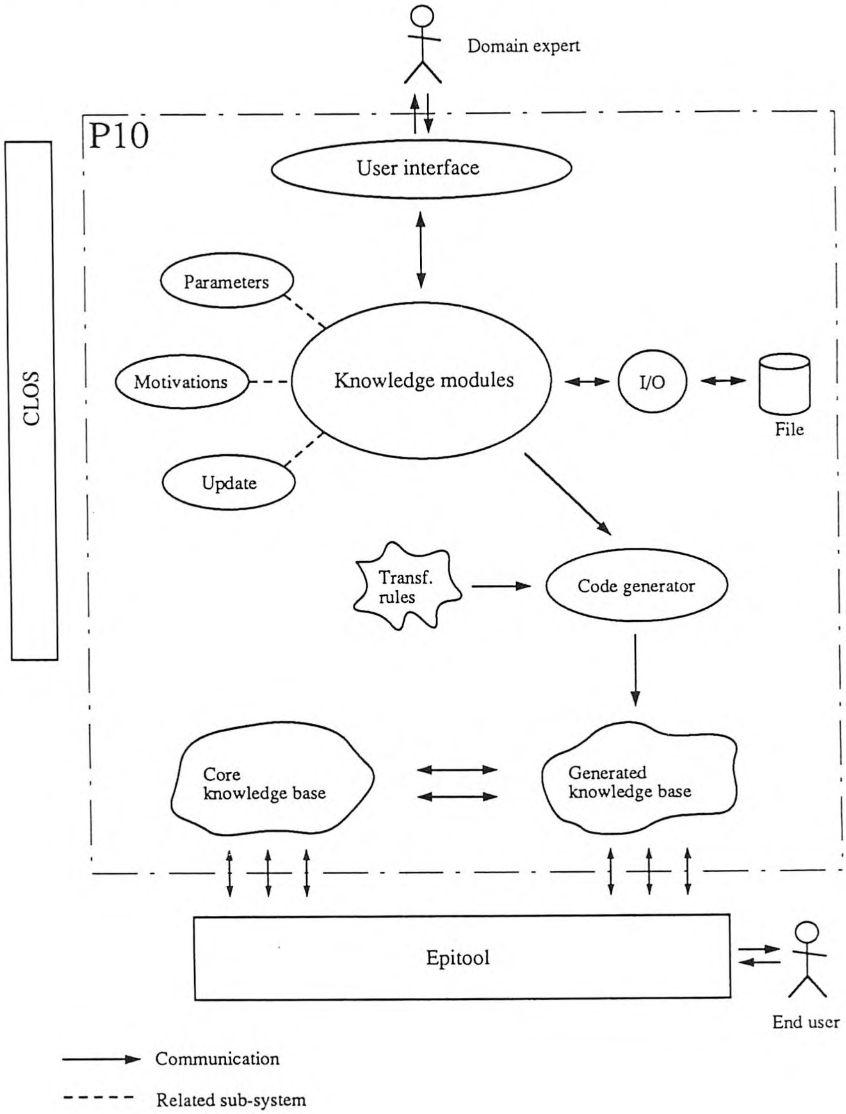


Figure 3.1. *Overall architecture of the P10 system*

P10 can be viewed as a domain-oriented knowledge acquisition tool. It generates target knowledge bases from information entered by the domain expert and expressed in a conceptual framework that is customized to the domain. Specialized knowledge editors are available for manipulating the knowledge base during development and iterative modifications. A code generator guided by transformation rules is used to transform the knowledge base into the selected target system for delivery. Figure 3.1 shows the overall structure of the P10 system. Continued work involve the study of meta-level tools to create systems like P10 for reasonably arbitrary domains.

3.3.3 User interface management

The dialogue style supported by the human-computer interface is an issue of prime importance for user acceptance in many use situations. Traditional transaction-oriented systems are often not appropriate for creative work processes. Neither is the inference-driven interrogation of the user in the first-generation expert systems.

Interactive knowledge-based systems, with their emphasis on cooperative problem solving and active user assistance, calls for an architecture with an independent *user interface management system* and supporting different styles of consultation. In our opinion, *expert critiquing* is an interesting and important complement to more direct advisory dialogues. This approach is discussed in more detail in the next subsection.

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. Investigations by Jonas Löwgren employ an architecture with a session discourse manager, which is mediating between the reasoning and the dialogue processes.

Of special interest is the possibility to develop knowledge-based support for dialogue design, giving critiquing 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 Epitec these and other related issues are studied [Löwgren et al 89]. The KRI user interface critic contains a rule base with evaluation knowledge, a database with interface design guidelines and a user interface evaluation taxonomi. The demonstration prototype developed so far allow 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.

3.3.4 Expert critiquing and text generation

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.

> 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 of a critique generated by the CRIME system in response to a suggested treatment plan for a diagnosed urinary tract infection.

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 power 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 a magnitude of order harder than creating an ordinary expert system.

However, there is also another side of the coin. We can thus 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 should be 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 be more dependent on the system's advice.

The work by Rankin demonstrate how text generation based on general-purpose linguistic principles can be acheived 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 structure. 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.

3.3.5 Training and tutoring

The primary motive for development of expert systems has typically been the desire to support or automate problem-solving processes in the domain of application. However the explicit representation of knowledge in a system can also serve the dual purpose of providing the basis also for a tutoring system, which can be used to train inexperienced personnel in decision making, especially for unfamiliar or extraordinary situations.

We believe the potential for re-use of knowledge in new applications or for different purposes, such as e.g. problem solving or training respectively, to be a core issue in expert systems technology. In previous experiments (giving advice on financial planning) we have demonstrated the possibility to invert the use of documented knowledge in such a way that it can be applied for training in decision making rather than only for problem solving. This means that the user, instead of feeding facts to the system and getting a solution in response, is expected to solve a problem generated by system. The system's task is then to coach and supervise the user and give hints and advice whenever needed.

Current work in our group [Sokolnicki 89] focusses on the coaching approach, relying on related work in the lab on expert critiquing for evaluating the actions and reactions of the user. It is an essential aspect of the work that intelligent tutoring should be provided as an additional service in the context of independently developed knowledge systems. Practical investigations are part of the K-Linker project with applications in biotechnology.

3.4 Joint projects and external cooperation.

ASLAB projects emphasize joint efforts with other groups and industry. The main current or recent involvements are listed below. More details are given elsewhere in this report.

3.4.1 CENIIT projects

ASLAB is actively participating in the interdisciplinary research program concerned with industrial information technology, CENIIT. This program emphasizes the use of information technology in industrial products and processes and is described in more detail in chapter 13 of this report.

Three areas of study have been initiated as ASLABs contribution to the CENIIT programme:

- *Engineering databases*, involving the study of database support for complexly structured data and design activities, including the relationship to knowledge systems.
- *User Interface Management Systems*, supporting dialogue design and application independence.
- *Computer-Support for Automation*, as described in greater detail in chapter 5 on CAELAB.

From July 1989 these CENIIT projects constitute the core of a new research group, CAELAB (*Laboratory for Computer Assistance in Engineering*). This group is presently working in close cooperation (and partly sharing personnel) with ASLAB. CAELAB is headed by Anders Törne with Sture Hägglund as assisting thesis supervisor. Current activities are described in the next chapter of this report.

3.4.2 Knowledge transfer activities

The following is a list of current involvements where ASLAB is actively cooperating with companies subscribing to the knowledge transfer programme at IDA.

1. Pharmacia, Uppsala. Cooperation on knowledge-based systems in support of experiment planning in bio-technology started during the spring, 1988. The K-Linker project, as described above, relies heavily on Pharmacia for its application domains.
2. Volvo PV, Gothenburg. Joint activities on expert systems started in December 1987 with a special emphasis on systems for fault diagnosis and technical maintenance.
3. TeleLOGIC Programsystem AB, Linköping. Cooperation with a special emphasis on human-computer interaction in general and UIMS in particular.
4. Volvo Data, Gothenburg. This cooperation is presently in a planning phase. ASLABs involvment is primarily in areas of AI planning and methodologies for knowledge acquisiton.

Experiences from technology transfer in the area of knowledge engineering are reviewed in [Hägglund 88].

Another major channel for industry contacts is *Sveriges Mekanförbund*, where we carry out commissioned study projects incooperation 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 contribute significantly to the funding of our group, in addition to providing an effective way of communication with people from industry.

3.4.3 Educational activites in knowledge systems

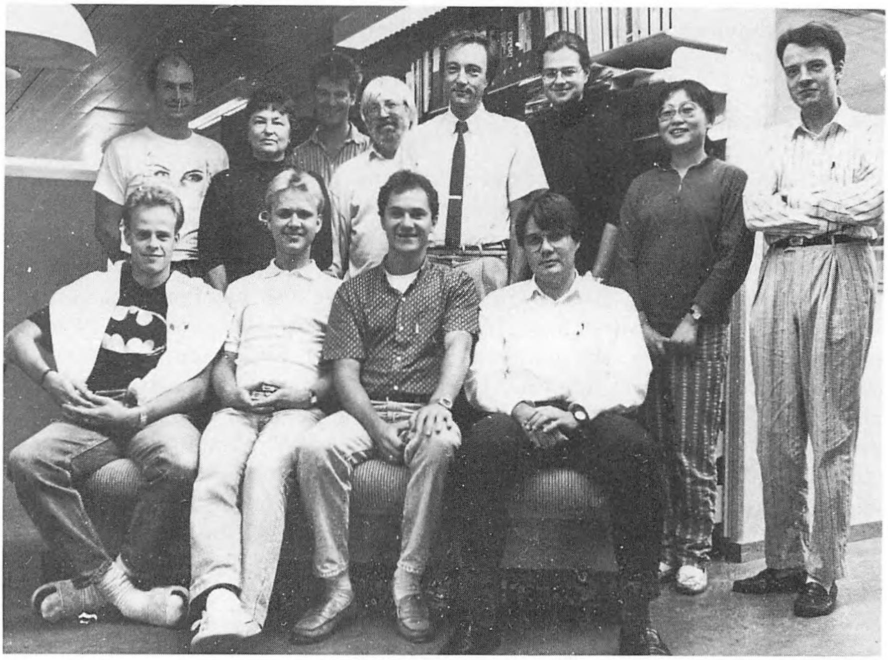
In addition to the normal undergraduate and graduate course activities, as described elsewhere in this report, ASLAB has actively engaged in the development of training programmes in *knowledge engineering*. Thus we are responsible for the expert systems courses in the continuing education programmes offered by IDA. During 1989 we have also together with ABB in Västerås developed a new model for an intermediate (between the 30 weeks programme offered by the university and the 1-2 weeks courses offered by companies) education in applied AI and expert systems. This new programme defines a *knowledge technician* level ("kunskapstekniker") and aims for a proficiency in developing systems with the use of established tools and techniques. We also cooperate with Epitec AB in the planning and production of industrial training programmes. Further we acknowledge a donation by IBM to LiTH of a 9370 system, which provide one part of the basis needed for development of project-oriented courses on expert systems.

3.4.4 CAFKA - knowledge communication

In a joint project with Department of Psychology, Stockholm University (Yvonne Waern), funded by HSFR, we study the "Communication, Application and Forwarding of Knowledge through Artificial Systems". The subject of studies are the processes which occur in knowledge-based expert systems when a domain expert is documenting his expertise and insights in a knowledge base and when the end user is consulting that knowledge base in the context of a problemn solving process.

3.4.5 FOKUS - AI and Human-Computer interaction

Since 1988 a joint project in the area of human-computer interaction in intelligent systems has been pursued together with the Defense Research Institute (Foa) in Linköping (Hans Marmolin, Tommy Nordkvist) with partial funding from the national information technology programme (IT4). Presently this project has been reorganised as a part of a larger effort in the form of a R&D consortium for knowledge systems and human-computer interaction, FOKUS. In addition to Foa, Epitec AB is actively participating in FOKUS projects.



ASLAB personnel. Front row: Jonas Löwgren, Torbjörn Näslund, Tomas Sokolnicki, Kristian Sandahl. Back row: Jonni Harrius, Britt-Marie Ahlenbäck, Peter Eklund, Ivan Rankin, Sture Hägglund, Thomas Padrón-McCarthy, Tingting Zhang and Henrik Eriksson.

3.5 Publications

For a more extensive listing of published papers, including departmental reports, see appendix E. Below a list of recent publications by lab members is given for an easy reference.

Licentiate theses 1988-89:

1. **Tim Hansen:** Diagnosing Faults using Knowledge about Malfunctioning Behavior, 1988.
2. **Henrik Eriksson:** A Study in Domain-oriented Tool Support for Knowledge Acquisition, 1989.
3. **Ivan Rankin:** A Generalized Approach to Text Generation for Expert Critiquing Systems, 1989.
4. **Jonas Löwgren:** Supporting Design and Management of Expert System User Interfaces, 1989.

International publications 1988-89:

1. **Peter Eklund:** Using Path Algebras and Conceptual Graphs to Encode Knowledge Bases, *Proc of the 2nd Scandinavian Conf. on AI, SCAI'89*, Tampere, 1989.
2. **Peter Eklund:** Prospects for Conceptual Graphs in Engineering Interfaces. *Proc. of the 2nd Int. Conf. on Industrial and Engineering Applications of AI and ES (IEA/AIE-89)*, Tullahoma, Tenessee, 1989.
3. **Henrik Eriksson, Kristian Sandahl:** Knowledge-based Planning of Experiments in a Biochemical Domain - Membrane-Protein Purification, in *Proc. of the 2nd Scandinavian Conf. on AI, SCAI'89*, Tampere, 1989.
4. **Tim 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.
5. **Sture Hägglund, Ivan Rankin:** Investigating the Usability of Expert Critiquing in Knowledge-Based Consultation Systems, in *Proc. of the ECAI Conf. Muenchen*, July 1988.
6. **Sture Hägglund:** Emerging Systems for Computer-Based Knowledge Processing in Office Work. In *Office: Technology and People*, 4, pp 135-145, 1989.
7. **Sture Hägglund:** The Linköping Approach to Technology Transfer in Knowledge Engineering. In *The Knowledge Engineering Review vol 2*, no 3, Cambridge University Press, 1988.
8. **Sture 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.
9. **Sture 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.*
10. **Paul Kuczora, Peter Eklund:** A Conceptual Graph Implementation for Knowledge Engineering. In *Human and Organisational Issues in Expert Systems* (edited by Anna Hart and Dianne Berry), Krogan-page, 1989.
11. **Jonas Löwgren:** History, State and Future of User Interface Management Systems. In *SIGCHI Bulletin, vol 20*, no 1, 1988. Also as LiTH-IDA-R-88-14, 1988.
12. **Jonas Löwgren:** An architecture for Expert System User Interface Design and Management. To appear in *Proc. UIST'89*, Williamsburgh, 1989.
13. **Ivan Rankin, Sture Hägglund, Yvonne Waern:** Generating User-Centered Explanations in a Critiquing Context. Position paper for *the AAAI Workshop on Explanations*, S:t Paul Minnesota, 1988.
14. **Ivan Rankin:** Deep Generation of a Critique. *Proceedings of Second European Workshop on Natural Language Generation,* Edinburgh, April 1989.
15. **L Reshagen, J-O Hilden, B Lindström, K Sandahl, O Wigertz:** An Evaluation of an Antibody Analysis Advisor in a Laboratory Environment. The EXODUS Project. In *Proc. of MIE 88*, Oslo, August 17-20 1988.
16. **Tomas Sokolnicki:** Integrating Tutoring in Knowledge-Based Systems, in *Proceedings of SCAI 89,* Tampere, 1989.
17. **Yvonne Waern, Sture Hägglund, Ivan Rankin:** Expert Critiquing Systems - psychological and computational aspects of generating comments. Presented at the *the 4th European Conf. on Cognitive Ergonomics*, Cambridge, 1988.

Additional reports 1988-89:

(Including papers by researchers visiting or on leave from our group.)

1. **Sture Hägglund, Peter Larsson**, Kunskapssystem vid produktframtagning: Möjligheter studerade i tio företag, rapport Sv. Mekanförbund, 1989.
2. **Sture Hägglund, et al.**, Dialogergonomi - Effektiv interaktion människa dator, report Sv. Mekanförbund, 1989.
3. **Jonni Harrius:** The Constructive Use of Rhetorical Structure Theory in Text Generation, Master's thesis, Linköping University, 1989.
4. **Olof Johansson:** A Perspective on Engineering Database Research

LiTH-IDA-R-89-14.
5. **Jonas Löwgren:** Metaphors and Information Systems: From Syracuse to Computer Use, LiTH-IDA-R-88-25, August 1988.
6. **Jonas Löwgren, Tommy Nordquist, Staffan Löf:** Knowledge-Based Support for User Interface Evaluation in User Interface Management Systems, Report LiTH-IDA-R-89-32, 1989.
7. **Lin Padgham, Peter Eklund, Ralph Rönnquist,** A Skeptical Reasoner Using Matrix Manipulations, submitted for publication, 1988.
8. **Tomas Sokolnicki:** Intelligent Tutoring Systems - Craft or Technology? Industriserien, ASLAB, September 1988. LiTH-IDA-R-88-33.
9. **Swartout, W, Nordin, H, Paris, C, Smoliar, S,** Toward a Rapid Prototyping Environment for Expert Systems, invited paper German Workshop on AI, Geseke, 1989. (Work carried out at ISI/USC.)

ASLAB Memo series 1988-89

88-01 **Löwgren, J.,** Outline of a Portable Display Manager for Direct-Manipulation User Interfaces to Knowledge-Based Systems.
88-02 **Hägglund, S., Rankin, I.,** Investigating the Usability of Expert Critiquing in Knowledge-Based Consultation Systems
88-03 **Rankin, I.,** Towards Effective Text Generation in Critiquing Expert Systems
88-04 **Rankin, I, Hägglund, S, Waern, Y,** Generating User-Centered Explanations in a Critiquing Context.
88-05 **Waern, Y, Hägglund, S, Rankin I,** Expert Critiquing Systems - psychological and computational aspects of generating comments.
88-06 **Zhang, T.,** An Intelligent Information System for Efficient Pharmacotherapeutic Management.
88-07 **Tingting Zhang:** Integrating Logic Programming, Expert System and Database Management System -- Expert Database System
88-08 **Hägglund, S.,** Towards Cooperative Engineering Environments for Generic Knowledge Systems.
88-09 **Andersson, M, Hansen, T.** Expertsystem i ekonomiska tillämpningar.

89-01 **Jonas Löwgren:** Phenomenological Philosophy and its Implications for AI
89-02 **Padron-McCarthy, T.,** IXEO - A System for Symbolic Representation and Interpretation of Ion Exchange Chromatograms
89-03/04/05
Natural Language Issues in Knowledge Systems:
Jonas Löwgren: Plan Recognition in Expert Systems Discourse Management
Kristian Sandahl: Connections between Computerized Understanding of Natural Language and Expert Systems
Tomas Sokolnicki: Language Issues in Knowledge-based Tutoring Systems
89-06 **Hägglund, S.,** A Note on the Communication Paradigm of Knowledge Systems
89-07 **Anne Steinemann,** Acquiring and Using Knowledge from Multiple Experts; Potential of Critiquing Systems.

4.

CAELAB The Laboratory for Computer Assistance in Engineering

Anders Törne

The laboratory for Computer Assistance in Engineering is a new laboratory at IDA and was formally started in July 1989. The research of the laboratory is interdiciplinary 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 will consist of applying results from Computer Science research to assist this process and from this experience initiate and try to influence research in Computer Science. The laboratory has, initially, two directions of research:

- Computer Support for Automation;
- Engineering databases.

The major funding 1989/90 is from CENIIT (Center for Industrial Information Technology), a special programme for interdisciplinary research within Linköping University.

The laboratory has strong connections with the Application Systems Laboratory, and cooperate closely on lab activities, seminars and graduate student supervision. The research theme of CAELAB is however more problem-oriented and focussing on manufacturing and engineering.

The work in CAELAB is mainly supported by CENIIT (The Center for Industrial Information Technology), Linköping University.

4.1 Laboratory Members

Anders Törne, Ph.D. Laboratory leader (formerly ABB Corporate Research)

Gunilla Lingenhult, secr.

Sture Hägglund, PhD. Supervisor, Engineering Databases

Kevin Ryan. PhD, guest researcher (20%) 1988/89, Engineering Databases

Jerker Wilander, MSc., adjunct researcher (20%) 1989/90, Eng. Databases

Olof Johansson, MSc. Engineering Databases

Peter Loborg, MSc.(Aug. 89) Computer Support for Automation

4.2 CENIIT project: Computer support in Automation

This project started during february 1989 and is concerned with advanced techniques in using computers to support automation and control of manufacturing processes.

Research area

The manufacturing (engineering) process includes product design and configuration, production, quality control and manufacturing support and management. In the context of computer science the process is divided into an ordered set of activities, each one generating information to subsequent activities by transforming information from earlier activities. During design or for order controlled production, during configuration, the functional and structural information is specified. This information is transformed by several stages of "planning" activities into detailed instructions for workers and programs for automatic manufacturing equipment.

The focus of the project is on the control of this information flow process in the manufacturing industry and specially for the "flexible" manufacturing area. With control is understood - 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? The goal is to "control" how, when and why information is not correct, and how corrections are to be made when the executing environment (the manufacturing environment) changes.

Particular emphasis will be put on computer support for product configuration, process and operations planning, factory scheduling, and control of the physical process at the cell level.

Method

The information flow in the manufacturing process is modelled as a set of cooperating autonomous agents. Each agent performes a task without direct interaction or intervention from the other agents. The needed interaction is, however, performed by negotiation between agents about tasks and goals. 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, focussing on sequential control and exception handling.

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 "continous" processes.

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 expert system technology and heuristic optimization.

Research in these issues is the first step to achieve understanding for how the general goal above can be achieved.

Activities

Activities Feb 1989 - June 1989:

- Presentation by Anders Törne at the FAMOS workshop on AI Technology as a Tool in Assembly 1989, titled "Planning for Assembly Operations".

- Structuring of the research area as outlined above. A presentation has been made to the research board of CENIIT (Center for Industrial Information Technology).

- Initialization of a master thesis work "Programming environment for sequential control of multimachine setups".

- Establishing contacts with other dept. and parties, for example the dept of mechanical engineering (IKP) at the University of Linköping and the Swedish Institute of Production Engineering Research (IVF) in Linköping.

- Participation in 21st Int. Seminar on Manufacturing Systems June 5-6, 1989, Stockholm, Sweden.

4.3 CENIIT project: Engineering Databases

This area covers such 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 modeling and knowledge engineering for design objects and related processes, and the relationship to administrative databases and information retrieval systems.

Current research focusses 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 the LINCKS group.

4.3.1 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 administratitive 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 implicitely 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 work in the area involves both fundamental studies of issues in engineering databases and applied as well as theoretical work (Padgham) in the context of the LINCKS project (see chapter 11 of this report).

4.3.2 Intelligent CAE frontends

Database management in an engineering environment refers to the task 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 adaption and integration of techniques from traditional database systems and knowledge-based systems respectively. An example of this situation is studied in a current cooperation with the department for mechanical engineering, where an integrated system for computer-assisted dimensioning with respect to the strength of materials and solid mechanics is being developed (Orsborn). It is our interest to view this problem as an instance of a generic application defined as an intelligent frontend for a CAE system. The possibilities for such generalization in the context of engineering database systems are studied by Olof Johansson.

4.3.3 User interface issues

This area covers research on methods for design of human-computer interfaces and techniques for effective implementation of these. Activities in this area, focussing on user interface management systems, started during 1988/89 with visiting researchers, course activities and preliminary investigations of core research topics.

At present the area is studied in the CENIIT context as a subarea of engineering databases. However, we believe that increased efforts on generalized UIMS approaches and supporting methodologies will be of prime importance for applied CENIIT projects in many areas. Thus the aim is to build a solid competence in the area, based on qualified research projects as well as engagement in applied projects.

Currently we are investigating the integration of knowledge-based approaches with UIMS technology (see chapter 3 on ASLAB in this report). This involves for instance management of cooperative dialogues, where the system as well as the user is actively generating and pursuing plans which initiate communicative actions. This problem is discussed in the licentiate thesis by Jonas Löwgren. ("Supporting Design and Management of Expert System User Interfaces", 1989.)

Another current line of related investigation is concerned with knowledge-based support for design of human-computer interfaces in general. Here we are participating in a study, where the aim is to utilize and presumable extend services provided by a UIMS (TeleUSE) as the basis for an evaluation of a user interface with respect to a knowledge base of dialogue design rules. A demonstration prototype (the KRI system) for such a system was developed within the joint research with the Defense Research Institute (Foa). (Löwgren, et al: "Knowledge-based Support for User Interface Evaluation in User Interface Management Systems", 1989.)

4.4 Other activities

Anders Törne is a member of the steering committee for the 5 year funding program for Computer Science in Sweden at the Swedish board for Technical Development.

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 administred by Mekanförbundet and financed by IT4. In the final report ten cases studied in participating companies were reported and discussed. (Sture Hägglund)

Current activities involve a seminar series on the problems and possibilities for intelligent, active support for design. As far as possible the approach would be generic to all areas of engineering and where specific domains are addressed (e.g. software design) efforts are made to assess the applicablility of the techniques in other engineering disciplines. The ambition was also to foster an exchange of ideas and approaches between designers in various fields of engineering and to specify some of the common requirements they have for any future, intelligent computer-based assistance.

Topics of seminars:

- The nature and characteristics of engineering design. Factors involved in design. What can AI do ? Some example problems.
- Existing support systems for engineering design. Examples from Civil, Mechanical and Electrical Engineering. Critical comparison of the systems.
- Designing Software - why and how is it different? Current approaches

and existing systems. Critical assessment.

- Prerequisites for intelligent support. Method, domain and target system knowledge. Access to large data/knowledge bases. Future prospects.
- Application problems from automated manufacturing, materials selection, robotics, design optimization, etc.

Course leaders: Kevin Ryan, Olof Johansson, Sture Hägglund

A joint study group on CIM (Computers in Manufacturing) has been organized by Anders Törne, who is also preparing a graduate course on CIM: Data and Process models.

CAELAB personnel: Olof Johansson, Peter Loborg and Anders Törne.

Publications:

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. Sv. Mekanförbund, 1989.

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

Joans Löwgren, et al: Knowledge-based Support for User Interface Evaluation in User Interface Management Systems, Report IDA, 1989.

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

5.

CADLAB
The Laboratory for Computer-Aided Design of Digital Systems

Krzysztof Kuchcinski

5.1 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 the register transfer level VLSI implementations. Such a transformation should be carried out so as to preserve 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, the intermediate representation of the design can be introduced to form a base for different optimization strategies. Finally, some stepwise refinement method can be utilized.

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

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 pipelines structures and design for testability. The support of the design environment is studied in the framework of the project also. The verification part is intended to create the methods for proving the correctness of the synthesized hardware against its specification.

5.2 Current Research

CADLAB is currently engaged in a research project called *Automatic Synthesis and Verification of VLSI Digital Systems.* The research topic of this project is focused on design automation methods and tools for VLSI digital systems. Particularly we are interested in synthesis and verification of embedded systems, sometimes called Application Specific Integrated Circuits (ASIC), and related topics. In the project we concentrate on the following problems:

1. Specification methods, target architecture strategy and influence of these issues on synthesis methods and results.

2. The development of algorithms and methods for hardware synthesis and verification; the evaluation of their usefulness and efficiency.

3. The integration of design automation tools into a single framework by the use of unified design representation and database technology to provide a common representational scheme for design specifications and implementations.

The research within the project is curried out in the three major areas as summarized below.

5.2.1 High-Level Synthesis

Automated synthesis of VLSI systems deals with the problem of automatically transforming VLSI systems from an abstract specification into a detailed implementation. Due to the complexity of VLSI systems, the synthesis process is usually partitioned into a sequence of smaller steps. The typical sub-processes focus on synthesises of algorithm, register-transfer structure, logic, and layout separately.

Algorithm synthesis deals with the synthesis of procedural behavior or control-flow/data-flow behavior from an abstract behavioral specification. The register-transfer synthesis process takes the procedural behavior or control-flow/data-flow behavior and synthesizes a register-transfer level structure. The logic synthesis procedure synthesizes then gate level logic (or physical implementations using a predefined cell or IC library) from the register-transfer level structure or a set of boolean equations. Finally, the layout synthesis algorithms will generate layout masks from, for example, the gate-level logic description.

The algorithm and register-transfer synthesis processes form the front end which is usually called *high-level* synthesis. The input to a high-level synthesis system usually consists of three components: (1) a high-level behavioral specification of the VLSI system to be designed; (2) a set of design constraints concerning, for example, cost, performance, power consumption, pin-count, testability, reliability, etc., and a cost function which specifies what are the criterion for optimization; and (3) 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 concerning, for example, geometrical placement and timing constraints of the function modules in the data-path structure. The additional information is used by the lower-level design tools.

The goal of a high-level synthesis algorithm is to generate a register-transfer structure that implements the specified behavior while satisfying the given design constraints and optimizing the cost function. To achieve this goal, a set of basic issues must be addressed by a high-level synthesis system.

The first issue is *event 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 the data-path which is able to perform the specified functions. Its output is a data path graph whose nodes represent components and whose edges represent communication links interconnecting the components. The control allocator, on the other hand, analyzes 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 domain of the design representations. For example, we can transform a design at the behavioral specification domain to make it easy for the synthesis tools to analyze and extract certain parallel operations. To substitute an ALU in the structural design by an adder followed by a shifter is another example. The basic objective of design transformations is to improve the design. In the extreme case, the design is transformed to an (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 *(de)composition* and partitioning. The former deals with the situation when there is no direct mapping between function 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. Our research activities at CADLAB have so far been concentrating on how to improve on the chance of getting the globally optimal solution. Two projects, the CAMAD system and the TAO approach will be discussed in the following sections. Other activities concerning synthesis of digital hardware, for example, the pipeline extraction, design for testability, will also be reported.

The CAMAD Synthesis System

The CAMAD approach to high-level synthesis is an attempt to achieve global optimization for several aspects of the design space. It makes use of a horizontal optimization algorithm which makes design decisions concerning operation scheduling, data path allocation, and control allocation simultaneously.

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 in different level of granularity. The arcs are used to model the dataflow between the operations.

The main feature of the ETPN design representation is its ability to capture the intermediate result of a design explicitly so as to allow the synthesis algorithms to make accurate design decisions. With the ETPN notation, CAMAD can represent and make trade-offs between the three elements of a VLSI system: control, communication and computational elements. Another important feature of the ETPN representation is that it allows us to "view" a high-level behavioral description as a *primitive* structural description which is of course very crude. That is, if we implement it directly, we get a very expensive design. However, once we have a structural description, we can make improvement on it to produce a better one. Further, this improvement can be done step by step until a satisfactory result is reached.

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 secondly heuristics concerning, for example, locality of computation, communication frequency, resource bound, etc.

Further, based on this design representation model, the problem of how to partition a VLSI system into several modules with well-defined interfaces can be formulated as a graph partitioning problem. CAMAD partitioning is provided both on the data part and on the control structure, which produces a set of pairs of corresponding data subparts and control subparts. As such, it allows potential asynchronous operation of the designed systems as well as physical distribution of the modules. Each of these modules can later be implemented independently, thus reducing the complexity of low level design to a manageable scale.

The formulation of the unified design representation model also provides a framework to incorporate 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 centralized 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 synchronization/asynchronization 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 during the last one and a half years.

The TAO Design Representation and Synthesis Approach

The TAO (Task, Algorithm, and Operation graphs) is a design representation for high level synthesis. Input algorithms are transformed into the TAO representation, which captures the different types of parallelism on separate levels so that the synthesis process is simplified. TAO is targeted to handled algorithm specification written in languages that allow declaration of concurrent task 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.

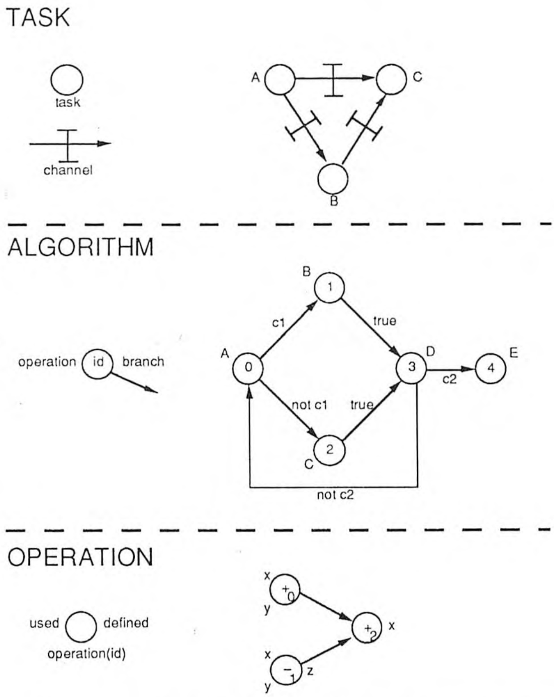


Figure 5.1 *TAO: Task, Algorithm, and Operation Graphs*

The design representation TAO is a hierarchal structure of graphs. 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 the a lower level. The next, and last level, is *the operation graph.* This graph expresses the partial ordering of primitive operations.

Compared to traditional compilers the task level corresponds to a global data-flow model, the algorithm level to a graph of basic blocks and the operation level to a local data-flow graph. On the highest level communication is viewed an asynchronous activity the two interacting tasks are not dependent on each others clocking rate etc. The only dependency is the style of interaction which can be, for instance, synchronous, like in languages as OCCAM, or rendezvous as in Ada.

The main motivation for "moving" an algorithm to hardware and VLSI is of course higher performance. To achieve this, higher levels of parallelism is needed. Studies of machines with multiple functional units, such as the Control Data CDC-6600, have shown that in normal sequential programs the amount of operations that can be performed in parallel is very low, only in the order of 2-5 operations. Thus program transformation is required. The main principle to achieve higher levels of this type of parallelism, functional, is to in some way grow the size of the operation graphs. This may be performed by merging nodes in the algorithm level. Typical transformation are distribution and loop unrolling of nodes.

The TAO representation takes a somewhat different approach to the synthesis problem than ETPN which, in principle, describes the system state, i.e., a graph of the composed tasks. This is mainly motivated by the reduction of the graph sizes when scheduling and transforming but also that the levels of parallelism should be separated in hierarchy. The TAO graph may be transformed into an ETPN and thus using CAMAD as a back-end.

The TAO synthesis approach combines methods and theories from computer architecture, compilers, and hardware design, to reduce the overall complexity of the synthesis problem. Computer architectures give valuable information about feasible hardware structures which exhibit high levels of parallelism. Compilers for these machines aid in finding methods for transforming algorithms to draw advantages of such structures. And last, basic principles of hardware design of sequential machines and interconnection technologies show good heuristic to searching for feasible solutions.

The main goal of TAO is the achieve an interactive and incremental high level synthesis environment where parallel programs may be entered together with design constraints and elaborated towards an optimal hardware structure. Currently the possible candidate languages for behavioral input are investigated but the experimental high level synthesizer is written in Common Lisp.

Synthesis of Pipeline Structures

Pipelining is a fundamental technique in high-speed computer design, but also a technique sensitive to the system environment and prone to design errors. This is due to the complex control required for handling the several simultaneous interacting and interdependent tasks that are active in a pipeline. For efficiency reasons it can be required to let the pipeline work internally in a way that is not consistent with a sequential model of execution; however, this is the preferred externally visible model in a case of, for instance, interrupt and state saving. Historically, several different techniques for handling these problems have been developed but not put into a framework for automatic design.

This project aims at constructing a tool that allows automatic extraction and synthesis of parts feasible for pipelining from a high-level behavioral design description. While previous work has concentrated on transforming loops (in a data flow graph) into pipelines, this project uses a more general approach where common substructures are mapped to shared hardware. The criteria for identifying such substructures include not only static information (can the system be described as a sequence of elementary operations), but also the dynamics of the system, i.e., whether there are enough task instances to defend a faster but more costly pipelined implementation. Several subtopics can be identified: the form of the design description and transformations on it to locate any potential pipeline structures; extraction of pipelines at different granularity levels; modeling of task dependencies, resource conflicts, and state preservation to allow expedient cost-performance estimates; selection criteria for different schemes and implementations for control.

The modeling of state can profit from a dual process/processor view on pipelining: either the computation stages are processes, exchanging data; or the computations are processes, moving over the stages/processors. The second view in particular connects each piece of state to the associated task. It can also be seen as higher-level description of the system that is independent of particular implementations. Those are then described by the first view. Formalisms for process definition in this context need be developed, as processes may interact and spawn into subprocesses.

In the context of the project, synthesizing pipelines can be interpreted as the dynamic run-time interconnection (by virtual circuits) of a given set of function modules into temporary pipeline configurations. Parallelisms can be drawn between this "dynamic pipelining" and the more conventional "static pipelining", but now the cost for run-time extraction/synthesis must be included.

Design for Testability

This subproject deals with the testability issue and the problem of test generation for a given design. This problem becomes more and more important as digital systems become more complex. This leads to a situation when the access to individual components is difficult or even impossible. A solution to this problem is to define and follow a set of design rules to make the final design testable. Such an approach is usually called design for testability. Two important concepts are introduced to understand the testability issue: observability and controllability. Observability refers to the degree by which internal states can be determined based on values available on the system's external ports. Controllability on the other hand refers to the simplicity by which a specific internal state can be produced. The proposed design rules should make possible the observability and controllability of a system for a chosen set of test patterns.

The subproject is on the stage of the definition and early results. The main proposed approach is to include the testability analysis on the early stage of the design process, namely on high-level synthesis level. This is justified because we can include requirements for testability on the early stage of the design process which makes its analysis less complex (than on the gate level).

5.2.2 Hardware Description and Verification

The increasing complexity of VLSI systems along with demands for high quality and fast development time has made it essential to improve system description, design and verification methods.

In this project the description and the verification of hardware is studied. The main goal of the project is to propose techniques that can support the *mechanized verification of hardware.* This goal has resulted in the proposal of a system where an equivalence hypothesis between a design's structural description and its behavioural description can be verified. The method suggested is to apply *algebraic transformations* on the equivalence hypothesis until the equivalence becomes apparent. More generally these transformations can be used for *canonization* of expressions and thus a theorem can be proved by being canonized to the truth value T.

Algebraic transformations of design descriptions can be used for several purposes, for example, as an important part of a synthesis system or a system for reuse and specialization of library components and generalized component descriptions. We concentrate on the use of algebraic transformations for verification purposes as a complement to other inference methods (such as those supported by natural deduction based theorem provers and resolution based theorem provers) and as a means for simplification of the hardware verification process.

These algebraic (or mathematical) transformations are to be carried out directly in the description language. Since most of these transformations can be based on well known mathematical concepts it is desirable that also the description language is based on a well known formalism that have been used for description of such mathematical concepts. As a consequence of this goal we want to illustrate that description of the functional, temporal and structural properties of a hardware circuit can be based on predicate logic if the temporal relations between a hardware circuits input and output ports are captured by the use of a temporal reference operator. We argue that the technique to use a temporal reference operator and the transformation techniques used for verification can be used both in conjunction with first order type constraints and in conjunction with higher order type constraints, the later to provide for more generalized descriptions. Thus the order of the logic is viewed as separated from more general syntactic constraints on the logic controlled by the chosen type constraints.

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 design process is usually very time-consuming and it is desirable if the descriptions and the verifications can be as general as possible and that previous results can be reused. It is also of importance that the description language and the verification tools can be used at as many (data, functional, temporal and structural) abstraction levels as possible. Further from a verification point of view it is preferable if the description method is based on a well founded formalism.

The use of a temporal reference operator allows us to associate a set of temporal canonization rules with this operator symbol. This leads us back to the main goal of this research project i.e. to provide some support for a further mechanized verification of hardware based on algebraic manipulation of boolean, arithmetic, relational, conditional and temporal constructs expressed in the logic. Important contributions that this research aims at are methods for multi-level hardware specification and methods for mechanized verification including functional, structural and temporal aspects that can be used as a complement to existing theorem proving systems. The implementation of a prototype (in LISP) of a (untyped) description language based on the presented view on a generalized logic and an implementation of a verification system is also part of this research.

5.2.3 Design Automation Environment

VLSI design environments require complex data management facilities. Traditional database processing is well adapted to managing large quantities of regular data, where most of the data is textual or numeric, and few links connect objects. Response time to a query is measured in seconds. The requirements of the design environment is quite unlike. As it is the framework for the integration of a set of tools, many different categories of objects need to exist concurrently, such as graphics, program data bases, and various kinds of design representations that capture the behavioral, structural and geometrical aspects of a design. Many of those categories are graph representations with high, and irregular interconnectivity, and explicit inter-category connections are desired, for instance, to provide debugging support. High-level synthesis tools require very fast access to the particular representations used by the tool, such as module libraries, INTERNAL, GRAPHICS

The basic idea of this project is to extend an existing programming language with a high-level data management facility (MARVIN), that support persistent objects. The approach of MARVIN is to support relations between any elementary value in the system. This is achieved through the well known concept of binary relations, which are allowed to relate values such as integers, strings, executable code in a network like structure. The major idea of the network proves to be a powerful tool to handle the requirements of a high-level synthesis environment.

MARVIN supports a data model that is free from basic restrictions and assumptions: Any two values can be related at any time, and the modeling of a value can at any time be extended to include a full network without effecting the application code that access the old part of the network. An important advantage of these characteristics is that a new tool can be added to, and integrated into, the environment without modifying the existing set of tools and data definitions.

Access to the network is achieved through a canonical interface that is independent of the underlying realization of the network. Each node with associated relations is realized independently. The particular realization may be changed during execution to adapt the realization to access patterns. The overhead of the access functions is in the order of a magnitude compared to statically compiled languages, like C, and about two to three magnitudes faster than typical database systems. The overhead can further be reduced by applying partial evaluation and compilation techniques to meet more than 100.000 data accesses/second.

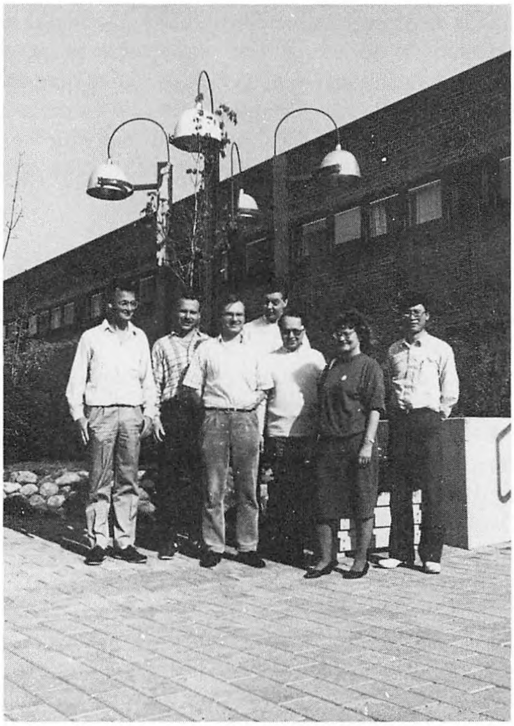
5.3 Related Activities and External Cooperation

CADLAB is involved in the graduate course program of IDA. During the reported time two graduate courses were given. During the spring 1988 the course "Advanced Computer Architecture" was given by Krzysztof Kuchcinski together with Björn Gudmundsson (Department of Electrical Engineering) and during the spring 1989 the course "Design Methodologies for VLSI Computer Architectures" was given by Zebo Peng and Krzysztof Kuchcinski. Additionally, a series of research seminaries on digital systems design automation were given every week by members of CADLAB.

We have established also contacts with other groups working in the area of digital systems design automation. In Sweden we cooperate with Dany Suk's group from Chalmers University and Adam Postula's group from Microelectronic Institute, Stockholm. The cooperation with the group from Microelectronic Institute concerns the transfer of our knowledge about high-level synthesis in order to design a high-level synthesis system for a subset of the VHDL language. We are also in the process of establishing contacts with Peter Denyer's group form University of Edinburgh and with Jouko Viitanen from Tampere University of Technology from Finland.

5.4 Personnel

Krzysztof Kuchcinski, Ph.D.	
Prof. Harold W. Lawson Jr., Ph.D.	(on leave from CADLAB)
Bryan Lyles, Ph.D.	(guest researcher March-June 1988)
Zebo Peng, Ph.D.	
Britt-Marie Ahlenbäck, secr. (until summer 1989)	
Bodil Mattsson-Kihlström, secr. (from summer 1989)	
Mikhail Ferapontous, Ph.D.	(guest researcher until July 1988)
Björn Fjellborg, MSE	
Mats Larsson, MSE	
Tony Larsson, Tech.Lic	
Fredrik Lindström, MSE	(until July 1988)
Mikael Patel, Tech.Lic.	
Göran Rydqvist, MSE	(from July 1988)



CADLAB personnel: Tony Larsson, Mats Larsson, Björn Fjellborg, Erik Stoy, Krzysztof Kuchinski, Bodil Mattsson-Kihlström and Zebo Peng.

5.5 Ph.D. Theses

Zebo Peng, A Formal Methodology for Automated Synthesis of VLSI Systems, December 1987.

5.6 References

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

1. **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.
2. **K. Kuchcinski, Z. Peng,** *Parallelism Extraction from Sequential Programs for VLSI Applications*, Euromicro Journal, Microprocessing and Microprogramming, vol.23, nrs 1-5, 1988.
3. **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.
4. **M. Patel,** *Selecting Design Representation for High Level Synthesis of Hardware Systems*, in Proc. of 4th Int. Workshop on High-Level Synthesis, Kennebunkport, Maine, USA, October 15-18, 1989.
5. **Z. Peng, K. Kuchcinski, B. Lyles,** *CAMAD: A Unified Data Path/Control Synthesis Environment*, preprints of IFIP TC-10 Conference: Design Methodologies for VLSI and Computer Architecture, Pisa, 19-23 September 1988.
6. **Z. Peng,** *A Horizontal Optimization Algorithm for Data Path/Control Synthesis*, Proc. 1988 IEEE International Symposium on Circuits and Systems, Helsinki University of Technology, Finland, June 7-9, 1988.
7. **G. Rydqvist,** *Extending the Data Modeling Support in Lisp*, Internal Report.

6.
LIBLAB
The Laboratory for Library and Information Science

Roland Hjerppe

6.1 Introduction

LIBLAB is the Laboratory for Library and Information Science. LIBLAB studies the possibilities of information technology and its effects on library and information work. We try to build up a broad competence in this subject area, with the objective of being 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 of primary interest to LIBLAB.

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)".

In 1988 all staff members formulated a new research program for LIBLAB, which replaced the original program from 1982. The new research program is directed toward two main themes:

Theme 1:
Users and information systems, especially bibliographic systems.

This theme is oriented toward the use and users of interactive information systems. The perspective is user-oriented. The user's central role as producer

---

The work in LIBLAB has been mainly supported by DFI, The Swedish Delegation for Scientific and Technical Information.

and consumer of information services is emphasized.

Within this theme we have identified two research areas:
a) User participation and user behavior, and
b) Orientation in databases: maps and other tools.

Two doctoral dissertations are under way in the first area. One is called "Personal Information Management in Computer Science Research". It is described in section 6.4.1. The other project called "Conceptual design and interactive multifunctional systems and their usability" is described in section 6.4.2.

The other research area deals with orientation in databases, which is a well-known problem in hypermedia research. A fundamental issue here is the user's ability to orient himself in abstract space. Therefore it is essential to study more closely different possibilities for graphically illustrating the database structure and content. Of particular interest are the generation and representation of maps and similar presentations, that can be updated dynamically in order to show both the "immediate environment" and the global view.

Classification systems and thesauri are descriptions of subject areas that can form a basis for illustrations. Co-citation graphs show the structures within particular literatures.

Theme 2:
Document description and representation

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

a) The convergence of hypertext and multimedia, Hypermedia,
b) HyperCatalogs, and
c) Formalisms for document description within documents and catalogs

Within the first area, Hypermedia, the interest is in the structure and components - the architecture - of hypermedia "documents" and the resulting problems encountered in describing and representing them at varying levels of detail.

The second area, Hypercatalogs, is concerned both with the application of hypertext approaches to catalogs of collections of traditional documents and the necessity for HyperCatalogs for hypermedia databases.

The third area, Formalisms, while studying formalisms such as SGML and their applications, is focused mainly on the potential for extracting "bibliographic" information, at varying degrees of detail and including structural information, from electronic documents and on the concomitant changes to access tools and hence access patterns.

Multimedia and hypertext, and their combination in hypermedia, are thus a major interest for LIBLAB. The interest is shared with many others. No one else, however, of all those engaged in these broad areas seems to be concerned with the problems of control of large hypermedia databases and finding and accessing (parts of) individual hypermedia "documents".

A discussion of the HYPERCATalog project is given in section 6.3 below and some specific reseach projects are described in section 6.4.

6.2 Laboratory Members

Roland Hjerppe, MSc.
Laboratory leader: planning, coordination and administration.
Research areas: the HYPERCATalog project; Hypermedia; SGML and other document description formalisms. For the period August 1988 - August 1989 he holds the position of Visiting Distinguished Scholar, Office of Research, OCLC Online Computer Library Center, Dublin, Ohio.

Birgitta Olander, BA, MLS.
Research assistant. Doctoral student at the Faculty of Library and Information Science, University of Toronto, Canada, since 1984.
Research areas: Personal information management; the HYPERCATalog project.

Arja Vainio-Larsson, MA.
Doctoral student since 1984.
Research areas: Conceptual system design and user interaction construction; hypertext/hypermedia and usability.

Lisbeth Björklund, BSc.
Doctoral student since 1985.
Research areas: Structures of systems for knowledge organization and representation; the HYPERCATalog project.

Manny Jägerfeld, BA.
Doctoral student since 1985.
Research areas: Data architecture; the HYPERCATalog project.

Siv Söderlund, BA. Administrative assistant since 1986.

Linda C. Smith, PhD.
Guest researcher and acting laboratory leader, January - June 1989 while on leave from her position as Associate Professor, Graduate School of Library and Information Science, University of Illinois at Urbana-Champaign.
Research areas: Hypertext and information retrieval; expert search intermediaries.

Paul Kahn, BA.
Guest researcher April - June 1989 while on leave from his position as Project Coordinator, IRIS (Institute for Research in Information and Scholarship), Brown University.
Research areas: Hypermedia; History of hypertext.

LIBLAB personnel: Arja Vainio-Larsson, Roland Hjerppe and Lisbeth Björklund.

6.3 The HYPERCATalog Project

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 of less implicit in document collections.

Experiences and insights from the work within the two research themes identified in section 6.1 above are combined in the HYPERCATalog project. The overall goal of the project is to design a document catalog that serves the information needs of the research community better than the traditional library catalogs we know today. Although some improvements have been made to the catalog, such as remote access to the catalog, local availability of extensions like bibliographic databases and better communication facilities there is still a long way to go. The catalog is still often a stand-alone service, demanding special techniques for access. 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.

The HYPERCATalog should give access to information in the same way, regardless of source. As a user you should be able to decide whether you would like to use the system as an ordinary OPAC (Online Public Access Catalog), containing only library material, an extended OPAC that gives access to external and internal databases, or as a personal information management tool allowing users to access and enter whatever information they want and make use of the information in different ways such as document creation and management. This choice should not affect the way the user interacts with the system. The same basic tools should be available, the information should be accessed and treated in the same way, independent of source.

In traditional online catalogs the structure and the content of the database are totally outside of user control, whereas the HYPERCATalog allows and encourages the user to exert substantial control over both. By displaying the logical organization of one or several collections to the user, searching is turned into navigation in a visible structure, which is contrary to traditional information retrieval.

Since we are talking about a hyperCATALOG, one of the primary goals is to represent documents and collections of documents. Document descriptions are represented in catalogs and bibliographic databases as static records, usually following some standard format. Collection descriptions are not included at all, and standards for collection descriptions do not exist. In the HYPERCATalog a description will be built by linking together basic elements of information. This gives the possibility to create a dynamic network for representation of both documents and collections.

Different representations of the same document can be obtained by applying different filters to the network - a filter is a set of object types that the user defines. In this way MARC-records, short entries or any other format can be obtained.

By using different modes of representation and selecting different linktypes as filters, documents can be brought together into virtual collections. More permanent virtual collections can be seen as different levels of the HYPERCATalog. A department can create a special virtual collection based on material from the library, complemented with special databases and in-house material. Groups in the department may take a piece of that collection to use for group-specific matters, such as co-operative writing, planning of courses etc. The individual user can derive a private level for all her personal information.

The primary mode of searching these collections is to follow links. Links may have been brought together to form paths. A path can be followed directly, or it can be used as a filter, creating a limited view of the network. This view can be traversed by navigation and browsing, which leaves a trail, showing the user's own way through the network. This trail, as well as the special view, might be saved for further use, in the user's private area.

While the HYPERCATalog serves as a testbed for theories and ideas generated within the research themes, it also functions as a generator for problems and research questions. Over the next few years, our research efforts will be focused on building small prototypes, HYPERKITtens, in order to gain experiences with various subproblems and to test different design options.

6.3.1 HYPERKITtens

Research activities in the HYPERCATalog project so far have focused on model construction, and the models are called HYPERKITtens. The HYPERKITtens are small prototypes - incomplete HYPERCATalogs - based on existing hypertext software. By creating HYPERKITtens it is possible to test ideas of the HYPERCATalog and get suggestions for modifications and improvements. So far, three HYPERKITtens have been developed:

HYPERLIB is a system demonstrating the use of geographical maps and floorplans for orientation and navigation in the library and its collection. Access points for the system are either geographical maps or a classification system. This is a typical physical location level application that illustrates navigation as a search method.

HYPERCLASS is a system that demonstrates how hypertext software can be used to introduce classification structures. In this system the classification scheme (in this case, the Swedish SAB) serves as the main access point to the library. This illustration of a part of a knowledge domain level is used to generate ideas about the support needed for presentation of knowledge organizing structures.

HYPERREF is a system where some 200 bibliographic references have been linked, in order to illustrate the notions of paths and trails. It is possible to follow subject-oriented paths and "cross over" where paths intersect. By amalgamating HYPERKITtens, a fuller HYPERCATalog prototype can be created, demonstrating many of the desired features listed above.

6.3.2 Implementation and Maintenance Problems

There are several areas where we may anticipate problems with the implementation of the HYPERCATalog concept. In addition, there is a class of problems that seems to be inherent in hypertext. The HYPERCATalog project is not yet advanced to a stage where it is possible or desirable to distinguish between the two sets of problems.

The HYPERCATalog is going to be a very large and complex database with many users working on different levels at any one time. This gives rise to problems with concurrency, version handling, user interaction, authority control and updating, to take a few examples. In order to support a large number of simultaneously working structure and text editors, rules for handling several versions, establishment of level priorities, and powerful processing are required.

Traditional OPACs often include authority control, which is essential for users as well as for catalogers. In a system that encourages dynamic growth and where bibliographic descriptions are enhanced by incorporation of new or alternative search terms, the issue of authority control is multidimensional. Who should perform the authority control? It is likely that structures created "privately" by single users are sometimes of common interest and deserve to be included in the public level. Who should be the arbiter in such cases? Would it be possible to leave some of the control to the system, and if so, which part?

Interaction privileges vary from level to level, and could also depend on user "status", naive or experienced. Who should assign such privileges? The tools for interacting with the system should give the users the full power of a hypertext information management system as well as dynamic access to an extremely rich database. Display and orientation devices must facilitate use and help the users to visualize linked structures. A powerful and sophisticated user interface is the key to some of these problems.

For LIBLAB the development of the HYPERCATalog is a process where design specification and problem identification are closely connected. The next section briefly describes some specific current research projects.

6.4 Current Research Projects

6.4.1 Personal Information Management at IDA

The project "Personal Information Management in Computer Science Research" has been carried out by Birgitta Olander as a series of nine case studies of scientists' information behavior at IDA. The subjects' collection and organization of scholarly information and use of information in their own writings have been the focus of the investigation, which will result in a doctoral dissertation in late 1989. The research questions of the study focus on the implications of personal and environmental characteristics for researchers' information handling. The overall objective of the project is to build a model of personal information management that may be used in the design of future information systems.

Information management is required whenever information is somehow processed. Information processing ranges from the very simple, such as taking a note, to the very elaborate, such as complicated calculations requiring the power of a super-computer. In either case the information management process originates in a perceived need of information. The satisfaction of information needs is related to the professional and personal social roles that the subjects play in their labs, and other groups they may belong to, and in the department as a whole.

Competition is fierce in the research community, and success is measured not just in how much you publish but also where you publish - "publish or perish" seems to always be at the back of most scientists' minds. The actors on the scene of scientific research change over time, sometimes slowly and sometimes quite rapidly. The new results may or may not be linked to new actors, but it is important for every researcher to know about these movements.

Findings

The most important findings of the study are related to the selection and use of information sources. Information behaviour characteristics of the group have been identified along with individual characteristics of the subjects of the study.

In the context of this study, the implicit information need of academic researchers may be concerned with maintaining the subject knowledge base. This includes being informed about new results and developments in the field as well as keeping track of people. The explicit information need may be manifested when the researcher writes a paper or prepares a presentation or lecture. Hunting for references or tracking down papers that others have referred to are both very explicit expressions of information need. The difference between implicit and explicit information need calls for different information management strategies.

The explicit information need often has to be satisfied under time pressure, and this is one reason why it is essential to have a reasonable collection of potentially useful sources close at hand. The implicit information need partially depends on a continuous flow of information - regularly scanning or browsing through the core journals and conference proceedings satisfies most of this - and partially is at the heart of any informal meeting of colleagues and friends working in the same field. All the subjects talk about the value and importance of off-the-record meetings at conferences. A combination of the implicit and explicit information needs influence the criteria for evaluating information.

The subjects try as far as possible to gather the source material they might need in their offices or the offices of the lab. If they succeed they rarely have to seek information that is external to their immediate working environment. This limitation of sources reduces the subjects' exposure to information and often seems to be self-imposed. This appears to be one way to make the information explosion manageable. Few, if any, of the subjects realize that the limitation of potentially valuable and useful sources to certain publications is to a large extent arbitrary.

One of the subjects is a gate-keeper in terms of scholarly information, and many of the others rely on him to provide information. This is especially obvious when it concerns a subject that is marginal in their area, yet important for the particular researcher. The role of the gate-keeper is to work as an extended information provider when the resources of the subject's own office or those of his lab are exhausted. By using a gate-keeper, information seeking is still kept internal within the department. Internal information seeking displays features that differ greatly from external information seeking.

Informal channels are of overwhelming importance for information provision. The subjects stress the importance of personal contacts for several reasons: to identify and provide useful information, for quality assessment, and for keeping updated in the field. They make very little or no use of the university library, which is a formal information channel. The expectation to find something useful there is very low: "Usually the library doesn't have what I'm interested in." It is used primarily for interlibrary loans.

When using formal information channels, such as printed documents, the user/subject himself has to search for, find, evaluate, and integrate the information into his existing knowledge structure. The integration process includes adjusting the content of the information, the message, to the environment of the receiver. The information is "re-packaged" to suit the situation at hand. The advantage of receiving information from trusted colleagues over finding it oneself, is that it is summarized, evaluated and adjusted to the user's situation.

The subjects of the study rely heavily on their own information resources or those of their immediate colleagues. Conference papers are the preferred source of information. The subjects are convinced that this is where the new results appear and where they can follow the developments in their field. They attend conferences to meet people and to learn about the most recent research developments. Conferences also provide the fastest way of publication. Proceedings are most easily acquired through personal participation in the conferences. This is one reason why the subjects see to it that they attend certain conferences themselves, or that a "lab representative" does. The choice of information sources to a certain extent depends on what the information will be used for. The subjects prefer journals as sources for knowledge maintenance and conference proceedings for knowledge development.

The subjects' views on "organizing principles" for information of interest to him or her are revealed by descriptions of the personal collection and the way it is accessed. They collect the proceedings of the important conferences in their field. These are usually physically separated from the rest of the personal collection, and a series of annual conferences, for example, are shelved in a row. The access method to conference proceedings is rather heuristic: "You tend to know approximately when something was presented and at what conference, so you flip through the proceedings around that year. If you don't remember that well, you just have to go through all of them."

They also collect reprints or copies of articles, and, quite often, working papers or reports from other research institutions. They may also collect a few core journals. The reprints and reports are usually organized by subject rather than by author. Individual subjects may have a card index to his collection, or organize the material in special categories used in his area of specialization.

In this study, most subjects have emphasized the timeliness aspect and indicated that one reason why conferences are mostly the preferred medium is because publication there is so much faster than other media. Even newsletters with a publication turnaround time of maybe as little as 4-6 weeks may seem slow to a scientist. The daily press would be a faster medium with most of the same features as the scientific media: printed record, attention, publication credit. But it lacks something that is fundamental in the research community, peer review, and is therefore unfit for serious science.

The study also investigated information use, especially as manifested in the use of bibliographic references. The subjects' use of references indicates that they tend to classify source material in terms of use (usefulness). They also do the reverse when they use the list of references in a paper as an aid to assess its relevance, and to identify the proper research context or tradition. It is not primarily a matter of satisfying cognitive needs when a scientist cites refers to a document, but rather a whole range of social processes that come into play. Those social processes are intrinsic in scientific research carried out according to the ethics of the scholarly community.

Quality or relevance assessment, finally, is an integral part of research. The way in which the subjects define "core" vs. "peripheral" in terms of information has to do with relevance assessments of the information acquired. The status of institutions, respect for names, awareness of ongoing research are all factors that influence the evaluation and subsequent selection and use of information. These factors constitute some of what the subjects usually refer to as "knowledge of the field", which has been identified as being of very great importance for the personal information management of the IDA researchers in this study.

6.4.2 Conceptual design

Every system usually starts off as a more or less well-formed concept. Often these concepts are brought together within the context of specific applications and/or techniques available. As the conceptual model for a system successively becomes more and more defined it also has to be generalized and hence clearly separated from a specific application domain as well as the implementation technique(s) used. In spite of this, the actual implementation of a system still becomes a question of anchoring and realizing such a model.

Interactive systems place great demands not only on a system's functionality (in terms of effectiveness as well as functionality), but also on its user interface. To some extent these demands vary due to the underlying application, and the means by which these demands are fulfilled within a system design process also vary. These variations often make explicit a conscious design choice and in this sense they can be said to express some kind of design paradigm. One such design paradigm is for example related to:

- the specification of a data- and interaction model, that is how data, relations between data and their interaction shall be modelled within the system?

- the choice of a presentation model; how to visualize the properties of a system's underlying data- and interaction model on the user interface level, what type(s) of user interface(s) should and can be supported by the system, what types of interaction techniques should be provided, etc?

Effective human-computer interaction presupposes that there in some sense 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. Appropriate user interface metaphors are said to effectively combine the system's underlying data- and interaction model; to logically link together actions by providing a domain (a context) out of which syntax as well as semantics easily can be inferred. Therefore, an important research area is to analyze on a general level the models that underlie the human-computer interaction part within different computer application domains. This should be done in addition to analyzing the use of metaphors as tools by which the complexity of user interfaces can be reduced and by which users' understanding and effective use of a computer system can be reinforced, as has been done within an earlier project by Arja Vainio-Larsson.

Interactive multifunctional systems and their usability

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 by Arja Vainio-Larsson is hence to study the interactivity and the usability of this kind of multifunctional tool. Hypertext/hypermedia is in this context seen as a more generic form of human-computer interaction and can be said to above all stress the usability of computer systems.

A promising property from a human factors perspective with these "hyper applications" is the fact that these systems seem to support a true separation between the physical implementation of a data model and its representation and presentation on the user interface level. Nonetheless, we know from our own and other research that the question of how to exploit the metaphorical quality of different abstract models in order to also (re)present complex information in a manner familiar to users becomes even more intricate in a multifunctional environment such as hypertext/hypermedia systems. Users often have to decide themselves how different representations of a task correspond to each other; how do we signal important properties such as equalities and differences between pieces of data and their relations without making this (re)presentation of information too complex for the users?

A common mixture of metaphors within the area of human-computer interaction is to combine the typewriting metaphor and the desktop metaphor with related metaphors as, for example, the writing and reading of documents. The 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. A popular information model is the graph represented on the screen as interlinked cards or documents (that is relative pieces, chunks of information). This kind of metaphorical representation of abstract information often causes users unforeseen problems. It is sometimes cognitively too difficult to correctly interpret the hidden properties contained within the metaphorical representation. The task seems either to be too abstract or too concrete for users to manage properly without much difficulty. Other related research questions focused on are:

- the separation of the interaction component and the functionality of the underlying application, that is the traditional aim within HCI to separate syntaxs from semantics.

- the problems of evaluation; how do we measure what constitutes a good user interface; how do we even separate a good user interface from a bad one when it comes to integrated, multifunctional environments?

- the need to make interaction more into a question of automation than as today a question of programming. Users, both programmer and nonprogrammers, need effective tools that in some way facilitate their specification of what and how to do specific tasks. How do we make general computer systems also into natural and integrated parts of users' daily work processes without turning them into programmers or technicians?

6.4.3 IRIS Intermedia

In April 1989 Paul Kahn installed IRIS Intermedia, the hypermedia system developed at the Institute for Research in Information and Scholarship at Brown University, on a Mac II running A/UX 1.1 for demonstration and evaluation purposes at LIBLAB. The installation includes the InterLex dictionary server and two Intermedia webs: Exploring the Moon, a collection of documents about the Apollo lunar missions, and Chinese poetry, a collection of documents about the translation of Tang dynasty Chinese poetry into English. Manny Jägerfeld and Paul Kahn have built some test data files on Intermedia, extracting records from the SOU (Statens Offentliga Utredningar) MARC records. These are being used to evaluate the suitability of Intermedia for storing and manipulating catalog records.

6.4.4 History of hypertext

Paul Kahn continued his research on the work of Vannevar Bush and the intellectual history of the hypertext concept, in cooperation with James Nyce of Brown University. Linda Smith began updating her 1980 study tracing the influence of Vannevar Bush on subsequent work in information retrieval.

6.5 Educational Activities

LIBLAB members contributed to courses, seminars, and workshops held at IDA.
Courses: Roland Hjerppe offered a course on Office Information Systems in spring 1988. Arja Vainio-Larsson offered, together with Sture Hägglund (ASLAB) and Lars Ahrenberg (NLPLAB), a course on "Human-Computer Interaction" in autumn 1987 and spring 1988.

Workshop
At the MDA-sponsored Workshop on Hypertext Applications in Working Life and Education, May 26, 1989, Paul Kahn presented a lecture and demonstration of Intermedia and Lisbeth Björklund, Birgitta Olander, and Arja Vainio-Larsson gave a presentation of the HYPERCAT project and related LIBLAB research.

Seminars:
Linda Smith presented an IDA seminar on Design, Development, and Evaluation of Expert Search Intermediary Systems, March 14, 1989.

Paul Kahn presented a Hypertext Seminar May 8-9, 1989, sponsored by UHÄ and attended by over 40 people from Swedish universities and industry. Lectures covered hypertext history, systems, and design issues. All members of LIBLAB assisted in demonstrations of several working hypertext systems including IRIS Intermedia, HyperCard, Guide, HyperTIES, KMS, NoteCards, and "Göstas Bok" running under LINCKS.

Two seminars were held at the University Library;
April 10, Linda Smith: Approaches to end user searching, and May 29, Paul Kahn: Linking reference books together using Intermedia.

6.6 External Contacts

Lisbeth Björklund: Participated in "Temadag om västsvensk biblioteksforskning", Bibliotekshögskolan i Borås, and in the Conference on Strategic Issues for Library Science Research, March 24, 1988 (together with B Olander). Attended the 11th International Conference on Research Development in Information Retrieval, Grenoble, France, June 13-15, 1988; the 44th FID Conference and Congress, Helsingfors, Finland, August 28 - September 1, 1988; the 12th International Online Information Meeting, December 6-8, and Optical Disc Forum III, December 9, 1988, London, England; and STIMDI, Konferens och föreningsmöte, April 17-18, 1989, Uppsala.

Roland Hjerppe: Served on the program committee and participated in RIAO 88: User-Oriented Content-Based Text and Image Handling, March 21-24, 1988, Cambridge, Mass. Participated in the American Society for Information Science Annual Meeting, October 23-27, 1988, Atlanta, Georgia.

Paul Kahn: Participated in STIMDI, Konferens och föreningsmöte, April 17-18, 1989, Uppsala and presented a lecture on the design of Intermedia. Participated in the Apple European University Consortium meeting on April 21 in Amsterdam at the invitation of Apple Europe Higher Education Marketing, giving a live demonstration and lecture on Intermedia. Presented invited lectures: April 24-27, educational uses of hypertext, Open University in Heerlen, The Netherlands; May 17, Hypertext and the design of Intermedia, Lund University and IDEON Research Park; and June 5-7, Intermedia and hypertext design, Department of Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm.

Birgitta Olander: Participated in Temadag om västsvensk biblioteksforskning, Bibliotekshögskolan i Borås, and in the Conference on Strategic Issues for Library Science Research, March 24, 1988 (together with L. Björklund); the IFLA 54th General Conference, August 28 - September 2, 1988, Sydney, Australia.

Linda Smith: Presented invited lectures: March 29, Research methodologies in library and information science, Bibliotekshögskolan i Borås with sponsorship by the Centrum för Biblioteksforskning (Göteborg); Attended the 17th Consultative Meeting of INIS (International Nuclear Information System) Liaison Officers, May 16-19 in Vienna, as one of six invited experts, to make a presentation on expert systems and participate in panel discussions on new developments in information technology.

Arja Vainio-Larsson: Participated in: "Psykologisk forskning i Linköping", a conference arranged by the Department of Education and Psychology, Linköping, October, 1987; the annual STIMDI conference, Stockholm, April 6-7, 1988, and presented a lecture on "Text som struktur - Hypertext"; the STIMDI workshop on UIMS, Stockholm, April 7, 1988; the Workshop on Hypertext, Lund, November 1-2, 1988, and presented a lecture on "Hypermedia, en generell metafor för människa-dator interaktion"; the HUFIT Design Seminar, Routhley Court, December 13-16, 1988; the Statskontorets seminars on European standards for Direct Manipulation, window managers and visual display units, 1988/1989; a joint project between IDA and the Swedish Telecom on "Methods for evaluating graphical user interfaces", 1988 - 1989; the Mekanproject on "Dialogergonomi: Effektiv interaktion människa-dator", 1988/1989. Presented invited lectures: Maj 20, 1988, on "Människa-dator interaktionsforskning vid LIBLAB och IDA", the HCI seminar series, Inst för informationsbehandling, Umeå University. Pursued study tours to: Finland visiting the Finish Telecom, VTT (Statens Tekniska Forskningsinst) and Tietotehdas (Helsingfors), July, 1987; W. Germany visiting Hamburg University, GMD (Darmstadt), Fraunhofer Institut f r Arbeitswirtschaft und Organisation (IAO-FhG, Stuttgart), IBM and ABB (Heidelberg), Universität des Saarlandes (Saarbr cken), Norsk Data (M lheim), GMD (San Augustin), Nixdorf (V. Berlin), November, 1988; Great Britain visiting EuroPARC (Cambridge) and HUSAT (Loughborough), December, 1988; and USA visiting IBM Watson Research Center and NYNEX (N.Y.), MIT and Wang Laboratories (Boston), CMU and Westinghouse (Pittsburgh), NASA, XeroxPARC, Stanford and Apple (Palo Alto), April, 1989.

6.7 Publications

Reports:

LiU-LIBLAB-R-88:1 Vainio-Larsson, A.: Hypermedia and Human-Computer Interaction. February 1988, 11p.

LiU-LIBLAB-R-88:2 Björklund, L.; Hjerppe, R.; Jägerfeld, M.; Olander, B.; Vainio-Larsson, A.: Forskningprogram för LIBLAB. February 1988, 6p. (also translated into English as Research Program for LIBLAB)

LiU-LIBLAB-R-89:1 Vainio-Larsson, A.: Hur enkla uppgifter blir till komplexa problem: om människa-dator interaktion. January 1989, 23p.

LiU-LIBLAB-R-89:2 Vainio-Larsson, A.: Hypermedia, a Vision for Human-Computer Interaction. January 1989, 13p.

LiU-LIBLAB-R-89-3 Vainio-Larsson A.: How to make easy tasks into complex problems: a report on human-computer interaction.

LiTH-IDA-R-88-20 Vainio-Larsson, A.; Orring, R.; Åberg, P.: Metoder för utvärdering av människa-maskin gränssnitt: En litteraturstudie. August 1988, 24p.

LiTH-IDA-R-89-15 Vainio-Larsson, A.: Hypertext/hypermedia från ide till praktisk tillämpning. (Extended report), April 1989, 13p.

LiTH-IDA-R-89-? Vainio-Larsson A., Åberg P.: Utvärdering av människa-maskin gränssnitt (delrapport 2).

Olander, B.: Strategi för stöd och vidareutveckling av datorbaserad informationsförsörjning för högre utbildning och forskning. Utkast.(Rapport till UHÄ). 7p. 1988.

Working paper:

LiU-LIBLAB-WP:41 Hjerppe, R.; Björklund, L.; Jägerfeld, M.; Olander, B.; Vainio-Larsson, A.: LIBLAB - Biblioteks- och informationsforskningslaboratoriet. Plan för verksamheten under 1988/89 och 1989/90. March 1988, 14p.

Conference papers:

Björklund, L.: HYPERCATalog - the Researcher's Tool for Information Management. Accepted for Information Technology and the Research Process, Cranfield, England, July 18-21, 1989.

Björklund, L.; Olander, B.: The Mystery Cat - the Modern Magician. Accepted for 7. Nordiske IoD, Århus, Danmark, August 28-30, 1989.

Björklund, L.; Olander, B.; Smith, L.: The Personal HYPERCATalog. Accepted for the Annual Meeting of the American Society for Information Science, Washington DC, October 29 - November 2, 1989.

Kahn, P.: Webs, Trees, and Stacks: How Hypermedia System Design Affects Hypermedia Content. Accepted for the Third International Conference on Human-Computer Interaction, Boston, Mass., September 18-22, 1989.

Externally published papers:

Björklund, L.; Olander, B.: LIBLAB - Nytt forskningsprogram. Svensk Biblioteksforskning, Meddelanden från Centrum för Biblioteksforskning, nr 1, 1988, pp. 23-25.

Hjerppe, R.; Olander, B.: Cataloging and Expert Systems: AACR2 as a Knowledge Base. Journal of the American Society for Information Science 40(January 1989):27-44.

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

Smith, L.: 'Wholly New Forms of Encyclopedias': Electronic Knowledge in the Form of Hypertext. Published 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.

Timpka, T.; Hjerppe, R.; Zimmer, J.L.; Ekström, M.: Introducing Hypertext in Primary Care: Supporting the Doctor-Patient Relationship. Published in RIAO 88: User Oriented Content-Based Text and Image Handling: Proceedings of the conference held in Cambridge, Mass., 21-24 March 1988. Volume 1, 1988, pp. 99-113.

Vainio-Larsson A.: Hypermedia and Human-Computer Interaction. In Proceedings of the 3d IFAC/IEA Conference on Man-Machine Systems Analysis, Design and Evaluation, Oulu, Finland, June '88.

Vainio-Larsson A.: Hypermedia, a Vision for Human-Computer Interaction. Submitted to Interacting with Computers, the Interdisciplinary Journal of Human-Computer Interaction.

Vainio-Larsson A.: Hypertext/hypermedia från idé till praktisk tillämpning. To appear in Dialogergonomi - effektiv interaktion människa-dator. Svenska Mekanförbundet, 1989.

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

Vainio-Larsson A.: Rapport från ännu ett hypertext/hypermedia seminarium. To appear in STIMMET, no 2, 1989.

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

7.
LOGPRO
The Logic Programming Laboratory

Jan Maluszynski
Professor of programming theory

7.1 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:

Jan Maluszynski , Ph.D. *group leader, professor*
Anders Haraldsson, Ph. D. *associate professor*
Wlodzimierz Drabent, Ph.D. *guest researcher*
Bodil Mattson-Kihlström, *secretary*
Staffan Bonnier, *graduate student*
Simin Nadjm-Tehrani, *graduate student*
Ulf Nilsson, *graduate student*
Torbjörn Näslund, *graduate student*
Roland Rehmnert, *graduate student*
Jonas Wallgren, *graduate student*

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.

Some of the research has been carried out in external cooperation with researchers at INRIA, France, Univ. 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.

7.2 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.

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.

7.3 The Research Topics

7.3.1 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 and APPLOG);
- 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 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 for 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 as 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.

7.3.2 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.

7.3.3 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 $P_1,...,P_n$ of source programs, partially evaluate the interpreter wrt. the programs. The result is a collection $O_1,...,O_n$ of programs in the same language as the interpreter. Next identify pieces of code in $O_1,...,O_n$ 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.

7.4 The Results

The following main results were obtained from January 1988 to June 1989:

7.4.1 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 [LBM88]. 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) An error message, indicating that S-unification is unable to solve the E-unification problem. A completeness check based on notions of abstract domains is also developed. The check provides a sufficient condition for the interpreter to be complete for a given program and a class of goals.

7.4.2 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 [DNM88a] 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 [DNM88a] 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. [DNM88c] reports on this implementation and the preliminary results obtained while debugging (almost) pure Prolog programs. In an augmented version of [DNM88a], we extended our debugging framework for programs using some Prolog features. In particular, we discussed extensions of the method for programs with cut, negation, and some Prolog built-in predicates [DNM88b].

7.4.3 A Method for Proving Run-Time Properties of Logic Programs

Certain properties of logic programs are inexpressible in terms of their declarative semantics. One example of such properties would be the actual form of procedure calls and successes which occur during computations of a program. They are often used by programmers in their informal reasoning.

We introduced and proved sound an inductive assertion method for proving partial correctness of logic programs. The method formalizes common ways of reasoning about logic programs and makes it possible to formulate and prove properties which are inexpressible in terms of the declarative semantics. An execution mechanism using the Prolog computation rule and arbitrary search strategy (eg. OR-parallelism or Prolog backtracking) is assumed. The method may be also used to specify the semantics of some extra-logical built-in procedures for which the declarative semantics is not applicable. [DM88] is an extended version of [DM87].

7.4.4 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.

7.4.5 Miscellaneous

During 1988 LOGPRO co-organized the *First International Workshop on Programming Language Implementation and Logic Programming* held in Orleans, France and during 1989 Jan Maluszynski was on the program committee of the *6:th International Conference on Logic Programming* held in

Lisbon, Portugal.

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.

As a result of previous research on the relationship between logic programs and grammars a survey of the area was presented in [DM89].

7.5 References

7.5.1 References to LOGPRO papers

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

[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.

[DM87] Drabent, W. and Maluszynski, J. Inductive Assertion Method for Logic Programs. In *Proc. of TAPSOFT'87*, Pisa, LNCS 250, Springer-Verlag (1987).

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

[DM89] Deransart, P. and Maluszynski, J. A Grammatical View of Logic Programming. In *Proc. of PLILP'88*, Orleans. LNCS 348, Springer-Verlag (1989).

[DNM88a] Drabent, W. and Nadjm-Tehrani, S. and Maluszynski, J. *Algorithmic Debugging with Assertions.* LiTH-IDA-R-88-04 (1988).

[DNM88b] Drabent, W. and Nadjm-Tehrani, S. and Maluszynski, J. Algorithmic Debugging with Assertions. In *Proc. of the Workshop on Meta Programming in Logic Programming, META88*, Bristol (1988). To appear in a book by MIT Press.

[DNM88c] Drabent, W. and Nadjm-Tehrani, S. and Maluszynski, J. The Use of Assertions in Algorithmic Debugging. In *Proc. Fifth Generation Computer Systems*, Tokyo (1988).

[LBM88] Leszczylowski, J. and Bonnier, S. and Maluszynski, J. Logic programming with external procedures: Introducing S-Unification. *Information Processing Letters* 27 (1988).

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

[Nil88] Nilsson, U. Towards a Framework for the Abstract Interpretation of Logic Programs. In *Proc. of PLILP '88*, Orleans, LNCS 348, Springer-Verlag (1989).

[NM90] Nilsson, U. and Maluszynski, J. *A Logical Perspective of Prolog* (provisional title). John Wiley & Sons Ltd. (to appear 1990)

7.5.2 Other References

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

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

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

[HT82] Hansson, Å. and 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).

[JLM84] Jaffar, J. and Lassez, J-L. and 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).

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

LOGPRO personnel: Xinli Gu, Anders Haraldsson, Gunilla Lingenhult, Andreas Kågedal, Torbjörn Näslund, Staffan Bonnier, Simin Nadjm-Tehrani, Ulf Nilsson, Jonas Wallgren and Jan Maluszynski

8. NLPLAB The Laboratory for Natural Language Processing

Lars Ahrenberg

8.1 Introduction

The interests of NLPLAB cover most aspects of the fields of Natural Language Processing and Computational Linguistics. Our theoretical research interests are primarily in the following areas: (i) parsing and interpretation within unification-based grammar formalisms; (ii) knowledge representation, including discourse representation, for natural language understanding, and (iii) the characteristics of man-machine interaction in natural language. One application area is presently of special interest to us, namely the construction and use of natural language interfaces (NLIs) to computer software. Research in this area is carried out within the project "Analysis and Generation of Natural Language Texts", financed by STU. The goal of this project is to develop a general-purpose NLI with ability to communicate in Swedish.

8.2 NLPLAB Personnel

Lars Ahrenberg, Ph.D., lab leader
Lisbeth Linge, secretary
Lena Andersson, B.A.
Nils Dahlbäck, B.A.
Arne Jönsson, Tech.Lic., B.A.
Magnus Merkel, B.A.

The work in the Laboratory for Natural Language Processing is mainly supported by STU, The Swedish Board for Technical Development. Support is also given from HSFR, *humanistisk-samhällsvetenskapliga forskningsrådet*.

Kerstin Olsson, M.Sc.
Stefan Svenberg, B.A.
Mats Wirén, Tech.Lic., B.A.

8.3 A Short Overview of Current Research

8.3.1 Parsing techniques for unification-based grammars

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. Our strategy is to use declarative, unification-based grammar formalisms and develop flexible and efficient parsers for such formalisms.

8.3.1.1 Incremental parsing

In a couple of previous papers (Wirén 1987, Wirén 1988a), Mats Wirén has studied parsers for unification-based grammar formalisms, in particular exploring open-ended control structures as a means of achieving enhanced flexibility and efficiency in parsing.

In subsequent papers (Wirén 1988b, Wirén 1989), he has been studying the problem of incremental parsing in a unification-based setting. Incremental parsing roughly means that an utterance is analysed bit by bit (typically from left to right), rather than in one go when it has come to an end. Incrementality is interesting both from an efficiency point of view and from the point of view of realizing artificial systems that behave naturally and "intelligently" in being able to process partial utterances.

Within natural-language processing, the word "incremental" has typically been pursued as *piecemeal* analysis of an utterance going left to right. Wirén has however suggested the adoption of a stronger notion of incrementality, prevalent in research on interactive programming environments, requiring that *arbitrary changes* should be efficiently handled. It turns out that chart parsing is eminently suited for this kind of incremental parsing since the chart basically constitutes the kind of data structure that is needed in order to keep track of previous analyses (thereby avoiding reanalysis of those parts of the previous utterace that were not affected by a change). Wirén (1989) puts forward a detailed description of how to augment a chart parser to handle incremental updates in this way.

A prototype implementation, called LIPS, embodying the ideas put forward in the paper, has been completed. It allows the user to freely edit sentences (insert, delete, and replace; characters as well as words) in a standard text editor (a Xerox TEdit). The changes are analysed incrementally and on-line by the parser which, at each moment, tries to keep the analyses (the charts) consistent with 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.

Continued work in particular aims at extending the framework to handling certain cross-sentential phenomena such as anaphoric dependencies.

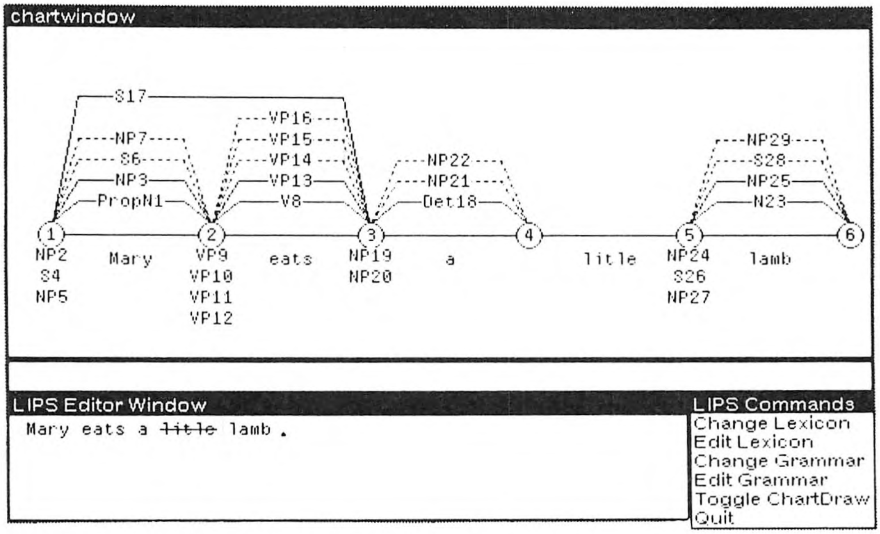


Figure 1: *LIPS: on-line, incremental parsing.*

8.3.1.2 Extensions to Unification-Based Grammar Formalisms

So called unification-based grammar formalisms in their purest form use term or graph unification as the sole information-combining operation (for example, PATR). 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 nonmonotonic multiple inheritance for the purpose of avoiding redundancy in large lexical and grammatical definitions, global well-formedness constraints, etc., introduce nonmonotonicity into the formalism.

In joint work, Lena Andersson and Mats Wirén explore various possible and proposed extensions to unification-based formalisms, also including their application to the Swedish language. The goals of the project are to survey 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 roughly with the power of a Boolean lattice. Continued work aims at integrating (some of) these extensions in a parsing and grammar-development system as well as studying how to integrate default reasoning and nonmonotonic multiple inheritance.

8.3.2 Knowledge representation for natural language understanding

The knowledge required to understand a text or an utterance in a dialogue is varied and complex. It does not only comprise linguistic knowledge in a narrow sense (morphology, syntax, semantics) but also knowledge of the world (especially that domain which is being talked about) and knowledge about discourse structure, e.g. how an object's saliency in the discourse affects the language used to refer to it, or about what kinds of response are possible after a certain type of initiative from another dialogue participant. It is a difficult issue, not only to track down and give appropriate representation of this knowledge, but even more so to integrate it and make optimal use of it in a working system.

Our approach to this problem is to rely on a uniform representation format for object descriptions (both linguistic objects and domain objects), to use simple and general rule formalisms such as those mentioned in the previous section and to employ unification and other types of constraint-propagation as the main processing formulas.

8.3.2.1 FALIN — an integrated dialog system.

In a dialog system called FALIN, we used a chart-parser and an LFG-like grammar in conjunction with an object-oriented knowledge representation tool to investigate various ways of integrating linguistic and non-linguistic knowledge in the interpretation of context-dependent expressions such as pronouns, definite descriptions and elliptical sentences.

FALIN is able to cope with different kinds of speech-acts: commands for the creation, manipulation and deletion of objects as well as questions and assertions about objects and their properties. It has been implemented as a simple drawing-system for the reason that this allows the user a visual access to the universe of discourse, i.e. it is easy to see whether the system interprets an input correctly. A typical interaction with FALIN looks as follows:

User: Draw a circle in the upper, left corner.
System: OK.
User: Draw a black circle under it.
System: OK.
User: Move it to the right.
System: OK.
User: Name it A1.
System: OK.
User: Draw a rectangle above A1.
System: OK.
User: What is the area of the rectangle?
System: The answer is: 4400.

The system is modularized in the sense that you can deal with morphology, vocabulary, syntax and semantics separataly and (with some exceptions) at the optimal level of generality. The system keeps a strict division between morphological, syntactic and semantic objects while connections between objects of different levels can be established for classes of objects as well as for individual tokens (lexical items, object types, etc.).

The system is also modularized in the sense that it consists of various components that can be used in other systems. Both the parser and the dictionary system have been used in other systems as well. An overview of the system is given in figure 2.

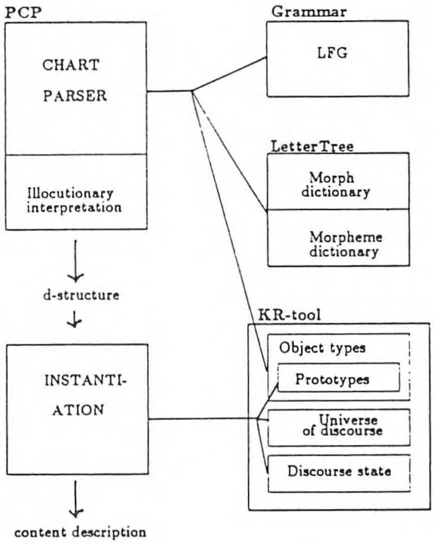


Figure 2: *Overview of FALIN.*

FALIN is also an integrated system, but not quite to the extent which was aimed for. The parser follows one strategy only and the interaction between syntactic and semantic processing is inflexible and sometimes yields inaccurate results. There is hardly any interaction between semantic processing and referent determination.

An essential idea in FALIN is that utterances themselves are objects which are categorized and analyzed in the interpretation process. The interpretation of an utterance results in a description which can be associated with the utterance and stored. Two important levels of this decription are functional structure and content structure. These two structures have the same format but contain different kinds of information. The functional structure describes the input utterance and its constituents as syntactic objects employing attributes for grammatical relations and morphosyntactic features. The content structure describes the utterance as a message with attributes representing (i) types and properties of the objects referred to in the utterance, (ii) attributes representing modes of reference, i.e. whether a set of referent properties should be used to identify a given referent or create a new one, and (iii) attributes identifying particular discourse roles of referents, such as being the speaker, or the topic, of an utterance. The ideas behind the system are discussed in Ahrenberg (1988a; 1988b) and a detailed description of it can be found in Ahrenberg (1989).

8.3.2.2 Temporal information in natural language

In Magnus Merkel's work on temporal aspects of natural language the emphasis has been put on temporal expressions (Merkel 1988). In his licentiate thesis the grammatical framework developed in the FALIN system is used for a grammar that recognizes and determines semantic functions of Swedish temporal expressions in texts. (The grammar is based on results from a study of over 1200 temporal occurences of temporal expressions in newspaper articles.)

Many temporal expressions are indexical and depend on the discourse context for their interpretation. Two contextual parameters, speech time (ST) and Temporal Focus (TF) (cf. Webber) are exploited in the interpretation of indexical temporal expressions. ST holds the utterance situation (or time of speech) and is necessary for the instantiation of indexicals such as *today*. TF contains the most likely temporal candidate (an event or explicit time) of the discourse for indexical expressions such as *the next day*. In Merkel (1989) these parameters are used in an algorithm to temporally structure a sequence of sentences describing situations in a narrative text. The proposed algorithm takes events as primary individuals and exploits a knowledge base of prototypical event descriptions together with grammatical knowledge (e.g. tense specifications) and discourse principles.

8.3.2.3 Discourse representation

Discourse representation has a double interest to us. On the one hand we are interested in the computational modeling of discourse as explained in the two previous sections. On the other hand we have an interest in cognitive models of discourse.

The notion of *discourse representation* has come more and more into focus in cognitive science research on text understanding. There exist many approaches to the problem of discourse representation, and a number of theoretical notions such as frame, script, context space, focus space, discourse model have been put forward.

Nils Dahlbäck, in his thesis work, starts out from Johnson-Laird's theory of *mental models* and evaluates it as a theory of discourse representation both theoretically and empirically. In Dahlbäck (1989) it is claimed that previous experimental studies by Johnson-Laird and co-workers fall into two classes. One class uses texts with strong visuo-spatial content, another more natural texts, or at least text with less strong spatial content. The results of the first group strongly supports the notion that the discourse representation used by the subjects has the analoge qualities claimed by Johnson-Laird, but there is some evidence that this can be due to the subjects use of mental imagery in the task. The results of the second group, while strongly supporting the notion that the subjects use a discourse representation richer than a linguistic representation of the text, but which puts less constraints on its format.

The empirical studies comprise two experiments, both belonging to the first class mentioned above. The first experiment investigated the role of background information in interpreting such texts, and showed that the possibility of using background knowledge obliterated the ameliorating effects of referential discontinuity described by Johnson-Laird. The second experiment studied the effects of differnt cognitive strategies in these reading tasks. The results show among other things that subjects using imagery replicate the response patterns of the previous studies, whereas subjects using a more 'verbal' strategy show a different pattern, which gives some empirical support for the analysis above concerning the relation between mental imagery and mental models as discourse representatioms used in these tasks.

8.3.2.4 Representations and the human mind

The theory of mental models is also the starting point for an analysis of the concepts of 'symbol' and 'knowledge' as used in AI and cognitve psychology. This work is used in an analysis of the critique of the so-called representational theory of the mind. In Dahlbäck (1989) it is argued that the word 'symbol' is used in two different senses, and that the confusions arising from this is one of the causes for the controversy on the symbolic nature of the mind and of the possible limitations of AI-systems (Newell, Dreyfus, Winograd). It is also claimed in this paper that AI systems should be regarded as extensions of the human working memory rather than as machine equivalents of general human intelligence.

8.3.3 The study of dialogues between human users and NLIs

A natural language dialogue between a computer and a human user by means of a natural language interface differs in important respects from human dialogues, spoken as well as written. To some extent these differences are known, but there is a need for more empirical studies aiming at uncovering the similarities and differences between these types of dialogues.

Studies of this kind 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. (e.g. 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 our 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%) are the most common anphoric 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 uses only a small set of standardized expressions.

Future work in this area will focus on differences between dialogues with people and with computers, and on the development of an implementable discourse representation for our dialogues.

For the analysis of the dialogues we have designed an interactive tagging system called DagTag (Ahrenberg & Jönsson, 1988). DagTag uses descriptors (attribute-value pairs) as tag units and is more general than most existing tagging systems in its ability to deal with constituents above (and between) 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 sequences, 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 dialogues can be searched by means of *a model dag*, a dag combining arbitrary descriptors. The search will return all units whose tags subsume the model dag.

A new version of DagTag is currently being developed. (Jönsson & Ahrenberg, 1989). This version is actually something more than just a tagging system. Beside being better at supporting interactive, manual tagging it also gives the user the opportunity to employ a parser for semi-automatic tagging. The parser, a version of Wirén's incremental parser, suggests parses that the user inspects. He may then change the analysis in the ordinary interactive fashion before it is stored. During this work he also has access to the rules and dictionary entries of the grammar and may update them. An advantage of the incremental parser in this mode of work is that partial analyses may be saved, while the grammar is updated. After completion of the updates the parser can resume where it was.

In essence this means that system has tools to support rapid grammar acquisition. In addition, it will provide some in-built learning heuristics to generalize from the rules acquired. In this way the linguistic characteristics of a set of dialogues may be given a formal description straight from the data. If the data provides a representative sample of the "sub-language" appropriate for an application, we should in fact be able to generate the linguistic knowledge-bases for the application in this way. Work in this direction is already planned as described below.

8.4 Towards general-purpose NLI's?

To be of general use as an interface system a dialog system must meet a number of different 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 application interfaces.

The studies described above of man-machine dialog in natural language have been aimed at the detection of significant differences in language and language use between that kind of dialog and dialogs between human beings. And although we have indications of the existence of specific traits in man-machine NL-dialog that could be exploited in the design of such systems, we also find a lot of variation depending on the type of application as well as on users. This has suggested to us that NLI-construction requires a modular approach and a set of tools and techniques for choosing and putting together the modules in a way that fits the requirements of a given application and a given set of users. The modules would consist of dictionaries, grammar rule sets, conceptual hierarchies, discourse representations, and so on, that are compatible with each other and interpretable by a given set of parsers, generators and translators.

8.4.1 Design and customization of natural language interfaces

Configuration of a natural language interface would then be a matter of identifying the user needs and applying the tool-box and library to the task. This work will have several phases. One phase is the analysis of the semantics for the new background system so that appropriate background system routines are accessed. Another is the customization of the linguistic components; lexicon, syntax and discourse component. If an NLI should be useful in a real application it will require a reasonable large lexicon and an appropriate grammar. One way to achieve a large lexicon is by using a large general lexicon and grammar which is updated with domain dependent words, concepts and rules. Doing such an update is a time consuming task involving many guesses concerning what domain specific words and concepts that will be used. It is for instance probable that the language used in an expert system differs from a relational data base system.

By analyzing a certain application through a Wizard of Oz simulation, the number of guesses can be reduced and a more reliable lexicon and grammar developed. For this purpose we need a Natural Language Interface Management System (NLIMS). Such a system will start with an existing lexicon and grammar, a number of different semantic components (for relational databases, expert systems etc), a number of different interaction and discourse models ranging from simple Q/A-models to more sophisticated consultant models and a control structure. The customization using a NLIMS begins by selecting a semantic component and interaction model. One also select domain specific words and concepts and possibly also a number of linguistic constructs that are domain specific and specified by the customer. Further, the objects, frames, that connect the words and expressions from lexicon to semantic objects in the application are specified.

The NLI is then refined through a series of simulations and tests using prototypes; by applying the acquisition functions of Dagtag, the lexicon and grammar is automatically updated with new words, concepts and linguistic constructs.

A large generally applicable lexicon and grammar gives a slow language analysis. By analyzing the simulations and study which words from the lexicon that are most used and what type of utterances that are most common it is possible to separate these out and put them in an active lexicon and grammar. This also means that the number of possible interpretations are reduced.

8.5 A brief look at the future

The research areas of NLPLAB (listed in the introduction) have remained virtually the same for the three years of its existence. While these areas will continue to be central to us, we also intend to widen our competence in the years to come: from parsing and interpretation to generation; from dialog systems to other applications, such as machine-translation and, consequently, to work with other languages than Swedish. We expect this development to take place in a context of increased regional, national and international cooperation.

A project concerned with text representation, which encompasses expansions in all of the three directions mentioned above, is started during 1989. It is funded by CENIIT and is shortly described in chapter 13.

NLPLAB personnel: Magnus Merkel, Lena Andersson, Mats Wirén, Arne Jönsson, Lars Ahrenberg, Nils Dahlbäck and Lisbeth Linge.

8.6 List of publications

The following are the publications referenced in the text above. For a full set of recent NLPLAB publications, please see the appendix at the end of the annual report.

Ahrenberg, L. (1988a). An Object-Oriented Dialogue System for Swedish, Nordic Journal of Linguistics Vol 11, No 1-2: 3-16.

Ahrenberg, L. (1988b). On the Integration of Linguistic Knowledge and World Knowledge in Natural Language Understanding. Research Report LiTH-IDA-R-89-23, May 1989. To appear in proceedings of the Nordic Conference of Text Comprehension in Man and Machine, Sigtuna October 27-28, 1988.

Ahrenberg, L. (1989). A constraint-based model for natural-language understanding and a pilot implementation. Research Report LiTH-IDA-R-89-22.

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

Dahlbäck, Nils (1989). A Symbol Is Not A Symbol. In *Proc. IJCAI'89*

Dahlbäck, Nils (in press). Mental Models And Text Coherence. In *Proc. First Nordic Conference On Text Linguistics*

Dahlbäck, Nils & Arne Jönsson (1989): Empirical Studies of Discourse Representations for Natural Language Interfaces. In *Proceedings of the Fourth Conference of the European Chapter of the ACL*, Manchester, UMIST, 10-12 April 1989, pp. 291-298.

Jönsson, Arne & Lars Ahrenberg (1989): 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 the 9th ICCH Conference*, Toronto.

Jönsson, Arne & Nils Dahlbäck (1988): Talking to a Computer is not like Talking to your Best Friend. In *Proceedings of SCAI '88*, pp. 49-61.

Merkel, Magnus (1988). A Novel Analysis of Temporal Frame-adverbial Phrases. In *Proceedings of COLING '88*, Budapest, pp.426-430.

Merkel, Magnus (1989). Temporal Structure in Discourse. To appear in *Proceedings from the eleventh Scandinavian conference on linguistics*, Torshavn.

Wirén, Mats (1987). A Comparison of Rule-Invocation Strategies in Context-Free Chart Parsing. *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, Copenhagen, Denmark: 226—233. Also research report LiTH-IDA-R-87-13, Department of Computer and Information Science, Linköping University, Linköping, Sweden.

Wirén, Mats (1988 a). A Control-Strategy-Independent Parser for PATR. *Proceedings of the First Scandinavian Conference on Artificial Intelligence*, Tromsö, Norway: 161—172. Also research report LiTH-IDA-R-88-10, Department of Computer and Information Science, Linköping University, Linköping, Sweden.

Wirén, Mats (1988 b). 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 (1988 c). *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.

Wirén, Mats (1989). Interactive Incremental Chart Parsing. *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, Manchester, England: 241—248. Also research report LiTH-IDA-R-89-24, Department of Computer and Information Science, Linköping University, Linköping, Sweden.

9.

PELAB
The Programming Environments Laboratory

Peter Fritzson
Bengt Lennartsson

During the report period the research in PELAB has been focused on two areas:

- development of an architecture for a programming environment supporting design, implementaton, and maintenance of software for distributed systems, *the PEPSy project*, and
- development and evaluation of tools and methods needed for a software specialist developing application specific environments, where the domain expert can create and maintain end-user systems, *the Multi-Level Software Architecture project.*

The research is organized in *kernel projects*, one for each of the areas above. Within the kernel projects there are individual thesis projects. The idea behind this organization is that the kernel project shall give a social and intellectual environment for the graduate students. Each kernel project has its own visionary objective, a direction to go rather than a state possible to reach. For the thesis projects, however, there are specific distinct goals. Each thesis project should be a significant step in the direction indicated by the goal of its related kernel project.

Our view is that our role is to develop and investigate new methods, tools, and programming environment architectures. We are focusing on support systems for the software specialists. We should work on proof-of-concept rather than on development of full scale programming support environments of production quality. We are very happy, however, to assist in transferring our results and experiences to industrial use whenever we can detect an interest.

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

9.1 PELAB Personnel, guests, and visitors 1988/89

The members of PELAB share their time between undergraduate education and research. The research part is 20 to 80 per cent of full time. It varies from person to person, and also from one year to the other.

Bengt Lennartsson, PhD 1974, lab. leader (until June 1989)
Gunilla Lingenhult, secretary

Supervisors:

Johan Fagerström, PhD 1988 (from Sept. 1988)
Peter Fritzson, PhD 1984 (lab. leader from July 1989)

Employed graduate students:

Rober Bilos, licentiate 1987
Johan Fagerström. PhD 1988 (until Aug. 1988)
Bengt Karlstrand, BSc (until Feb. 1988)
Mariam Kamkar, licentiate 1987
Yngve Larsson, MSE (until March 1989, licentiate May 1989)
Sven Moen, MSE (during the spring 1989)
Nahid Shahmehri, licentiate 1987
Lars Strömberg, MSE
Ola Strömfors, licentiate 1986

Guest researchers:

Artur Krepski, PhD, Polish Academy of Sciences. Sept. 1988.
Petr Kroha, PhD, Technical University, Prague. Feb.-June 1989.

Selected visitors from abroad:

Ray Ford. Unversity of Kansas, Lawrence. Sept. 88.
Robert Glass. State College, PA. May 1989.
Allen Goldberg. Kestrel Institute. June 88.
Claus Lewerentz. Aachen. June 88.
Walter Tichy. University of Karlsruhe. June 88.

Associated persons:

Ulf Cederling, University of Växjö
Azadeh Ghaemi, Programsystem AB
Yngve Larsson, Softlab AB (from March 1989)
Sven Moen, graduate student at Brown University (from fall 1989)
Tommy Olsson, lecturer IDA
Mikael Pettersson, undergraduate student
Kjell Post, graduate student at UCSC
Tom Rindborg, Softlab AB
Öystein Santi, undergraduate student
Dick Schefström, TeleLogic AB
Dan Strömberg, Swedish Defence Research Institute
Jerker Wilander, Softlab AB

PELAB personnel, front row: Johan Ringström and Peter Fritzson. Back row: Johan Fagerström, Gunilla Lingenhult, Mikael Pettersson, Mariam Kamkar and Nahid Shameri.

9.2 The PEPSy Project

Johan Fagerström, Yngve Larsson, Lars Strömberg

The general goal of the PEPSy project is to develop tools and methods for the design and implementation of software for distributed systems. Designers and programmers not only face many of the traditional problems of software design and manufacturing. They must also be able to handle new complications introduced by the nature of distribution. Added difficulties include non-determinism, complex causal relationships between events, and complex timing of events. A distributed system is often also expected to provide extra functionality compared to a corresponding centralized system. For instance, if one machine is out of order in a system, most users will only expect degraded response-time, not being left without computing power. It is therefore important that designers and programmers are provided with powerful tools for their tasks.

The PEPSy project introduces a paradigm and a number of appropriate tools for design and test of distributed applications. By paradigm we understand a conceptual framework for structuring ideas and concepts when solving problems. The PEPSy paradigm will support the designer of a distributed system. The premises on which this research is based are:

1. There is a natural paradigm on which design of distributed systems can be based.
2. Structures introduced by the paradigm can be used for analysis, test, and maintenance of distributed systems.
3. Programming environment tools can be improved since more knowledge of the system can be built into them.
4. The paradigm can be used to enforce a limited and structured mechanism for incremental changes in the system.
5. Documentation of system components and their relationships can be done in a consistent and intuitive way.

The main point of the project is that the structures introduced by the paradigm must be kept and exploited both during design and testing. One is not helped by a programming method that introduces a structure just to break it down again. Most models for concurrent systems, for instance [Hoare-85] and [Milner-80], are mainly used for specification and formal analysis. They do not support later phases, such as debugging. One noticeable exception is the HPC-model [LeBlanc-85] which was developed to describe distributed operating systems.

For our purposes, a distributed system consists a network of communicating processes. The processes and the inter- process communication links form a logical network. The logical network is mapped onto a physical network which consists of a number of computers connected by, for instance, a local area network such as an Ethernet. Critical applications for this kind of systems can be found in areas such as process control, telecommunications, defense systems, and office automation. Understanding and controlling complex distributed systems will be of utmost importance both in the short and long perspective. It is therefore important that development and maintenance of such systems are done in a controlled fashion aided by modern software techniques.

Distributed systems introduce a number of new dimensions to system design. For instance, designers, and sometimes users, must take time delays and limited bandwidth into account. Distributed systems also introduce a number of new types of possible bugs, and debugging is further complicated by non-reproducability of events, non-determinism, and complex timing of events. If robust, reliable, and correct programs are to be constructed, we must handle this extra complexity compared to sequential systems.

9.2.1 Project Description

To understand the activity of design and implementation of distributed systems it is necessary to study specification methods, system architecture, modularity concepts, programming languages, etc. The PEPSy project therefore naturally includes studies and discussions together with other groups within the department. We have taken a top-down approach to design and implementation of distributed systems. Initially, we have therefore developed a structural model which makes structures in distributed systems explicit. Design within the model starts with semi-formal specifications as in [Liskov-86].

The structural model includes mechanisms for defining well-structured interfaces between components and abstraction tools used to encapsulate internal behavior of objects. The model also supports a method that allows structured and limited changes in the designed system. This is extremely important since unstructured changes in large software systems inevitably lead to deterioration of the system [Cox-86]. In more detail the model includes processes that communicate via logical channels. Sets of processes can be turned into a "black box" by the use of a so called control unit. A black box can then be treated by the programming environment as a single component. The resulting design will be a hierarchicly organized system. The programming environment is aware of this organization and can thus exploit system structures to provide the programmer with a view of the system which is consistent with the structure of the design. The paradigm and a prototype implementation (in Smalltalk) was presented in more detail in Johan Fagerström's Ph.D.thesis (September 1988).

Our research has been focused on the following areas during the report period:

1. *Programming languages for distributed systems*

   Traditional languages such as Pascal and Lisp can be extended with primitives supporting concurrency. Another approach is to construct new languages designed for distributed systems such as Argus, Cell, or Conic. Object-oriented languages such as Smalltalk-80 also provide a natural metaphor for distributed computations. Following an in-depth study, this area has been continuously followed during the entire project life. One sub-project of PEPSy has studied support for configuration analysis and control in modern Algol-based languages such as Ada and Mesa. The language Conic was used to implement a distributed version of the Smalltalk prototype mentioned above as a part of Yngve Larsson's licentiate thesis.

2. *Paradigms for distributed systems*

   A structural model based on encapsulation as an abstraction mechanism and communication for process composition has been designed and implemented in both Smalltalk and Conic (on a set of SUN workstations).

3. *Tools supporting debugging*

   A number of prototype tools based on our research and experiences with existing parallel systems have been implemented. For example, a trace tool together with intelligent inspection routines, based on knowledge of data and control dependencies and system timing of events, support the programmer in detecting subtle bugs introduced by varying time delays.

4. *Management of static and dynamic objects*

   Naming and addressing of objects are traditionally difficult in distributed systems. The programmer should be allowed to refer to objects by name without knowing their exact location within the system. At the same time he must be allowed to specify the object's whereabouts for performance reasons. Tools supporting browsing of static and dynamic information are essential for both program development, testing, and system maintenance. We support dynamic reconfiguration of the system based on tools developed in Yngve Larsson's licentiate thesis (May 1989).

5. *User interaction*

   A consistent and supportive user interface is of utmost importance due to large system sizes, complex states and dynamic behavior in distributed systems. Therefore, extra efforts must be put into developing a uniform programming tool interface, based on modern bit-mapped displays and pointing devices. Few such systems exist, the ones that do are tailored for particular types of architectures. For instance, [Snyder-84] describes a graphically oriented programming environment for an ensemble architecture. This is an area where more research is needed.

9.2.2 References

[Cox-86] Brad J. Cox: *Object Oriented Programming - An Evolutionary Approach.* Addison - Wesley Publishing Company, 1986.

[Hoare-85] C.A.R. Hoare: *Communicating Sequential Processes.* Prentice Hall International Series in Computer Science, 1985.

[LeBlanc-85] Thomas J. LeBlanc and Stuart A. Friedberg: *Hierarchical Process Composition in Distributed Operating System.* Computer Science Department, The University of Rochester.

[Liskov-86] Barbara Liskov and William Weihl: Specification of distributed programs. *Distributed Computing.* Vol.1, No.2, 1986, pp.102 - 118.

[Milner-80] Robin Milner: *A Calculus of Communicating Systems.* Lecture Notes in Computer Science 92, Springer - Verlag. 1980.

[Snyder-84] Lawrence Snyder: Parallel Programming and the Poker Programming Environment. *IEEE Computer.* Vol.17. No.7. July, 1984. pp.27-36.

9.3 PEPSy-DICE: An integrated incremental development environment for distributed applications.

Peter Fritzson et. al.

The PEPSy system supports program development and maintenance of a distributed application by providing incremental configuration of processes, PEPSy modules and communication ports. It also has the potential of providing debugging support by tracing messages between ports. However, it gives no debug support within processes, which are regarded as black boxes from the system's point of view.

A natural extension is to investigate a system architecture which also provides debugging and incremental development support within processes. As incremental compilation and debugging support within a single process is provided by the DICE system, *Distributed Incremental Compiling Environment*, a previous PELAB project, it would possible to achieve this goal by combining results from PEPSy and DICE. This can also be regarded as a generalization of the DICE system to manage multiple target processes.

Aiming at this goal, several master thesis projects have been run, supervised by Peter Fritzson. An early effort (1985-87) was to port the DICE system to the Sun Workstation, by semi-automatically translating the DICE system to Pascal from a subset of Interlisp on the DEC-20. This was eventually successful, but required a substantial effort in restructuring the system, supplying missing type information, and implementing general list, string and array handling packages that had been available in Interlisp. The choice of Pascal as an implementation language was motivated by the desire to have a system that could support its own development.

The second task was to port the DICE host-target debugging primitives to Berkeley Unix 4.2 on the Sun workstation. This was done as a master thesis project by Mikael Andersson during 1987/88. Communication was based on TCP/IP instead of DecNet as in the previous version. On each machine, a server process can manipulate register and memory contents of a target process by using the Unix Ptrace primitive. In addition, the input/output emulation primitives were generalized by utilizing Unix pseudo teletypes to catch terminal I/O from the target process.

An important part of the porting effort was the retargeting of the DICE Incremental Code Generator to MC68020. Only PDP-11 or DEC-20 code could be generated previously. The major part of this work was done during the fall of 1988 by Oystein Santi, in cooperation with Peter Fritzson, and is reported in LiTH-IDA-R-13. Some follow up and completion work on the code generator port is being done by Mikael Petterson during 1989.

An ongoing masters project (1989), by Krister Joas, is to generalize the DICE debugging primitives to support multiple target processes, and to integrate the PEPSy configuration editor into the system. The configuration editor has been implemented by Yngve Larsson, and is described in his licentiate thesis.

9.4 The MLSA project

Peter Fritzson, Rober Bilos, Mariam Kamkar, Nahid Shahmehri

The overall goal of this project is to provide development and maintenance support for systems which use two or more language levels, e.g. the Very High Level language level, and the ordinary high level language level. We are interested in several aspects of such languages, both design of the language itself, compilation and transformation methods, and debugging. Below we briefly describe a few subprojects within this umbrella project.

9.4.1 A Specification language for the semantics of Incremental Symbol Processing.

PhD thesis project for Rober Bilos

One goal is to design a VHLL specification language for use in the generation of incremental compilers, i.e. when generating the part of the compiler which performs visibility analysis and handles declarations. This specification language should preferably be more concise, higher-level, and easier to read, than current approaches such as attribute grammars, which have performance problems in this kind of application. The very high level specifications should be converted to executable code by means of program transformations similar to those used in the compilation of VHLL languages such as Refine [Goldberg-83], down to ordinary high level languages such as Lisp. The second goal of this project is to define a set of transformations.

9.4.2 Algorithmic Debugging for Imperative Languages.

PhD thesis project for Nahid Shahmehri

Algorithmic Debugging is a technique for semi-automatic bug location introduced for Prolog 1982, [Shapiro-83]. This method in its current form works only for programs without side-effects, i.e. essentially functional programs. So far it has, to our knowledge, been used only for Prolog. It should be emphasized that algorithmic debugging works only for languages with some kind of procedure/function abstraction.

Our idea is to extend the method of algorithmic debugging to be able to cope with imperative programs with side-effects, by using program transformations and results from data flow analysis. The imperative program with side-effects is transformed into a functional program without side-effects, for which algorithmic debugging is applicable. These transformations are guided by data-flow analysis, i.e. def-use analysis. Thus, our version of algorithmic debugging consists of a program transformation phase followed by a conventional algorithmic debugging phase.

Our preliminary analysis of the proposed method indicates that less transformations might be needed than we first thought, i.e. that loops need not be eliminated. This would simplify the implementation within a realistic programming environment.

During the actual bug location 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. We are using the improved method.

9.4.3 Program slicing and Affect Chaining analysis for array elements.

PhD thesis project for Mariam Kamkar

In this project we would like to combine recent results regarding array index dependence analysis with program slicing and affect chaining analysis techniques. The result would be dataflow analysis algorithms with greater precision than the current ones, which can be applied to optimization or can be used to provide better debugging facilities.

9.4.4 References

[Drabent-88] W. Drabent,S. Nadjm-Tehrani, J. Maluszynski: *Algorithmic Debugging with Assertions.* LiTH-IDA-R-88-04.

[Goldberg-83] A. Goldberg, G. Kotik: *Knowledge-Based Programming - An Overview of Data Structure Selection and Control Structure Refinement.* Kestrel Report KES U 83.7. Kestrel Institute, 1801 Page Mill Road, Palo Alto, CA 94304. November, 1983.

[Shapiro-83] E.Y. Shapiro: *Algorithmic Program Debugging.* MIT Press, 1983.

9.5 Other activities

During the report period Ola Strömfors has continued to work half-time for Programsystem AB on the development of the Ada Environment from TeleSoft/TeleLogic.

Bengt Lennartsson has been conducting a project *System Design in Software* for The Trade Association of the Swedish Mechanical and Electrical Engineering Industry (Mekanförbundet). This work inspired him to start a new activity, *Industrial Software Technology*, within the department from July 1989. Peter Fritzson has been appointed as new lab leader from PELAB from then on.

Researchers from PELAB has participated in the programs for continuing education run by the department. A number of seminars directed to people from industry have been arranged. In January 1989 PELAB hosted a Nordic Workshop for Programming Environment Research with participation from leading academic groups in the Nordic countries, and also some qualified participants from industry.

9.5.1 An object oriented database for incremental symbol processing.

There is a problem in representing and processing declarations within programs incrementally and efficiently, especially for very large programs. Ideally, one would like to have a single database for all declarative information, which could be used both for compilation, debugging and query processing. This database, and associated operations, should be general enough to cope with most current and future programming languages. In his report "Incremental Symbol Processing", Peter Fritzson presents a fairly general model for incremental visibility and declaration processing.

This is connected to Rober Bilos' PhD project, where he designs a specification language and transformations down to executable symbol processing modules. Such a module will use the incremental symbol processing operations mentioned above.

During the spring of 1989, a guest researcher, Petr Kroha, has cooperated with Peter Fritzson in investigating how such a database could be implemented in terms of more or less standard database operations. The initial analysis indicates that precomputing and caching the transitive closure of some relations involving declarative regions, would greatly speedup the lookup operation. Petr Kroha has also extended the Single Instruction Machine idea and studied a compiler with scheduling for such a machine.

9.6 Future projects

An increasing emphasis will be put on the subject area of program transformations, both as a means of compiling VHLL specifications, for compiler generation, and for compilation to parallel architectures. Also, several of our current and past research activities are concerned with debugging support within programming environments, both in conjunction with incremental compilation, as an application of data flow analysis, algorithmic debugging, and debugging within distributed systems. We expect to pursue this research topic further, both regarding debugging and trace support for distributed systems, and debugging of programs and transformations written in very high level languages.

10. RKLLAB
The Laboratory for Representation of Knowledge in Logic

Erik Sandewall
Professor of computer science

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.

10.1 Researchers and Projects

10.1.1 Activities

The activities of RKLLAB during 1988 and the first half of 1989 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
4. constraint programming systems

The work in RKLLAB is mainly supported by STU, The Swedish Board for Technical Development and by the IT4 research programme.

Professional knowledge and information management systems: the LINCKS project.

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 by combining logic and differential calculus
3. modelling of road traffic environments

The LINCKS project was previously part of RKLLAB, but split off to become a separate entity from July 1, 1989. Since this volume should reflect the present departmental organization, the LINCKS research is described in the next chapter.

Work on plan-guided systems is done in the framework of the Linköping A.I. Center (LAIC) together with FOA and industrial partners, and as part of the IT4 research program. The division between the research on non-standard logics and on plan-guided systems is basically one between more fundamental and more applied work.

10.1.2 Laboratory members

The following researchers have participated in RKLLAB's activities during 1988-89 (except the members of the LINCKS project which are listed in the next chapter):

Laboratory leader:
Erik Sandewall

Lab members having or completing a Ph.D. or licentiate degree during the period:
Christer Bäckström
Douglas Busch, Ph.D.
Dimiter Driankov, Ph.D.
Johan Hultman
Jalal Maleki
Michael Reinfrank

Other graduate students and masters thesis students ("exjobbare"):
Patrick Doherty
Peter Haneklou
Christer Hansson
Magnus Morin
Anders Nyberg
Tommy Persson

Shu Hua
Lennart Staflin
Mikael Svensson
Ulf Söderman

Engineering support:
Leif Finmo

The following degrees were awarded to RKLLAB members during 1988-89:

Dimiter Driankov completed his Ph.D. The title of the thesis was: "Towards a Many-Valued Logic of Quantified Belief".

Johan Hultman completed his licentiate degree. The title of the thesis was: "A Software System for Defining and Controlling Actions in a Mechanical System"

A total of 23 papers by RKLLAB members were published, or accepted for publication, in international journals or conferences during the same 18 month period, as listed at the end of this chapter. (This does not include the degrees and the publications of the LINCKS group, which are detailed in the next chapter).

10.2 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, regardless of whether you actually expect to eventually arrive at that destination. 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. If things were done right we would expect to have a 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 should 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 such a theory of change must be 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, given the state of the art in 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 for more details.

10.3 Non-monotonic logic

Erik Sandewall, Douglas Busch, Patrick Doherty, Lennart Staflin

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 using 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 three-valued interpretations, where proposition symbols (etc.) are assigned either of the truth-values "true", "false", or "unknown". An *aggregate* consists of a set of such interpretations and a preference ordering (a partial order) on them. The value of a formula is then defined in each aggregate, and the evaluation rule for the D operator uses the preference relation. Erik Sandewall and Patrick Doherty have defined the model theory for the logic, for the simple propositional case. Patrick and Lennart have implemented semantic entailment-checkers.

Multiple inheritance with exceptions is a favorite testing ground for non monotonic logic. The basic issues in multiple inheritance are very simple in principle, but turn out to be surprisingly complex when you scratch the surface. Also this problem has been studied a lot and much is known about it. Patrick has developed a systematic translation from inheritance graphs to our default logic; the translation is modular in the sense that each arc in the graph can be translated individually. Thus there is no need to use extra "specificity literals". The translation has been checked for a large number of examples, using the above mentioned entailment-checkers.

The present work in this group focuses on building a systematic framework for non-monotonic logics that are based on a preference order on models, including the different varieties of circumscription as well as the logic described above.

10.4 Reason maintenance

Michael Reinfrank

Reason maintenance systems (JTMS, ATMS, etc.) are today the best developed candidate implementations of non-monotonic logics. However most work on reason maintenance systems has been implementation-oriented, and the relationships between non-monotonic logics (particularly, usable non-monotonic logics) and reason maintenance systems have remained unclear.

Michael Reinfrank has addressed this problem area in the following way. First, a reason maintenance system is characterized formally as a set of justifications, each being a rule of the form

$$\langle a|b \rightarrow c \rangle$$

where a is a monotonic antecedent, and b is a non-monotonic antecedent for the conclusion c. A justification is valid in a set S of propositions iff $a \in S$ and $b \notin S$. The familiar JTMS concepts such as in/out labellings, grounded proofs, etc. can then be defined in a straight-forward way.

As the next step, it is observed that a justification is essentially a self-referential formula in AEL normal form,

$$La \wedge \neg Lb \rightarrow c$$

in Konolige's autoepistemic logic AEL. This makes it possible to define labellings and other JTMS concepts in logical terms.

The approach has in addition been pursued for the characterization of ATMS's. Also since AEL is essentially equivalent to default logic, by these results a TMS can be regarded as computing a condensed representation of stable, maximal model sets.

Michael Reinfrank worked here until mid-1988, and is now with Siemens AG in Munich. He continues his participation in RKLLAB activities by means of short visits. The work described above has been presented in papers and in a Ph.D. thesis which will be submitted at the end of 1989.

10.5 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) 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 described above, and in temporal reasoning (see below).

10.6 Temporal logic, reasoning about actions

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 LINCKS group) during earlier years, and has continued during 1988 through Christer Bäckström's work. 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).

Christer Bäckström has extended the basic approach in a number of ways. In particular, he has added a fourth characterizing partial state, called a keep condition, which also applies to the whole duration. The rule for the action is that if the precondition holds when the action starts, and if the prevail condition holds for the duration of the action, then the keep condition also applies for the duration of the action, and the post-condition holds when the action ends. He has also generalized the structure of the partial states, and studied the use of this approach in some application areas.

The "explicit temporal logic" approach has been developed by Erik Sandewall and Christer Hansson. This approach is similar to the one that Shoham uses in his thesis and his book (Reasoning about change), but with another preference relation on models than chronological minimization which Shoham uses. This approach allows us to represent processes over time, which contain actions which occur sequentially or concurrently, and which have direct or indirect effects on properties in the world. Temporal prediction, temporal postdiction, explanation and planning can be obtained as special cases of non-monotonic deduction.

10.7 Algorithms for temporal reasoning

Erik Sandewall, Christer Hansson

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 will be submitted as a licentiate thesis towards the end of 1989.

10.8 Reasoning about physical change

Erik Sandewall, Tommy Persson, Lennart Staflin, Ulf Söderman

A complete logical theory of change must deal with both "physical" (spontaneouos, 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 Shoham-style 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 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.

On-going work by the members of the group considers the modelling of several types of simple physical systems using this approach. 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.

10.9 Plan recognition in aeroplanes

Dimiter Driankov, Peter Haneklou

The behavior of an aircraft can be naturally characterized in terms of the approach described in the previous section. The aircraft performs successive maneuvers, each requiring a certain period of time. Each maneuver is limited by the laws of physics and the capacity of the aeroplane. Particularly for military fighter aircraft, the limiting case is often also the actual maneuver, since one needs to push the aircraft to its extreme capability. The choice of a succession of maneuvers is guided by an ultimate goal or intention.

An important problem is then, given observations of an aircraft to determine which plan(s) the aircraft may possibly or likely be carrying out. The observations should characterize the first maneuvers in a multi-step plan, and one wants to identify the remainder of the plan as early and as reliably as possible.

Dimiter Driankov and Peter Haneklou study this problem, within the framework of our cooperation with FOA 3 (a Defense research agency in Linköping).

10.10 Modelling traffic behavior

Erik Sandewall, Shu Hua, Anders Nyberg

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 the 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.

10.11 Software architecture for autonomous agents

Johan Hultman, Magnus Morin, Anders Nyberg, Mikael Svensson

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, we do not ask "how can those theoretical methods be implemented", but rather: "what should the software architecture be like, in order to satisfy the needs of all the participating technologies, including *also* that the logic-based A.I. contributions can be supported there?"

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.

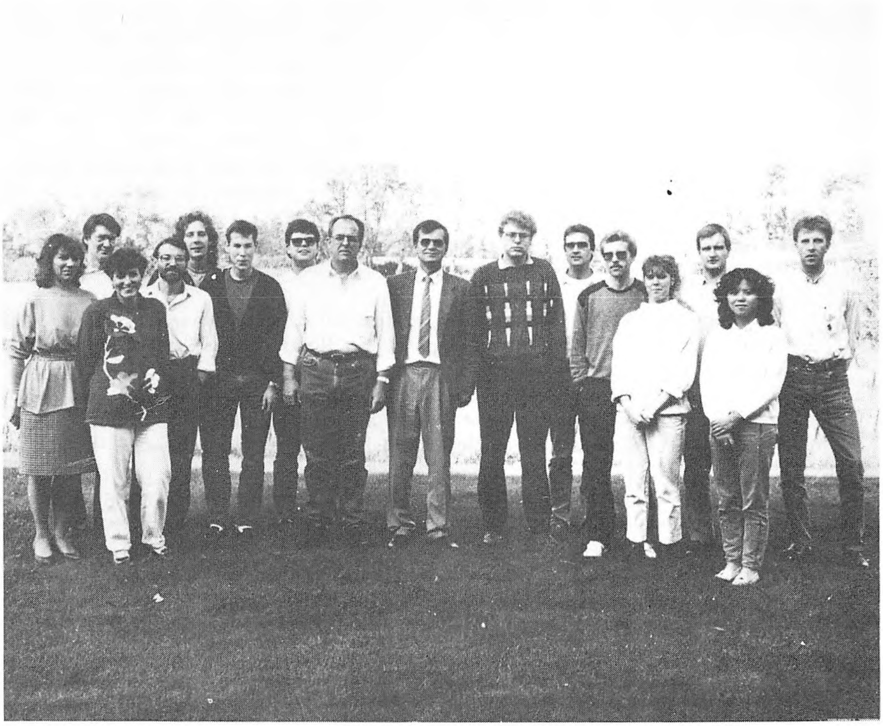
The architecture is therefore reminiscent of the one used by Rosenschein et al (formerly SRI International, now Teleos). It is being implemented in a target system based on a Motorola 68000 processor, using the PSOS operating system and the C programming language for its lowest layers. A SUN workstation, 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.

10.12 International activities

As already described, RKLLAB participates in the Eureka project Prometheus. In particular, Erik Sandewall is European co-coordinator for ProArt, the A.I. branch of Prometheus. We organized a European-level ProArt meeting in Kolmården (near Norrköping) in September, 1988.

We have also taken part in the COST 13 collaborative project number 21, on "Advanced issues in knowledge representation", together with universities in Brussels, Pisa and Rome. In the context of COST 13, we organized a "Second international workshop on non-monotonic reasoning" in Munich on 13-15 June, 1988. The workshop was co-sponsored by COST 13, AAAI, and Siemens AG, and Michael Reinfrank was in charge of local arrangements. The proceedings from the workshop have been published by Springer Verlag.

Professor Dan Stefanescu, of the University of Massachusetts in Boston, visited RKLLAB during several weeks in May, 1988. Dr. David Etherington, AT&T Bell labs, visited RKLLAB and the LINCKS group for a week in May, 1989.



RKLLAB personnel: Anne-Marie Jacobson, Leif Finmo, Lillemor Wallgren, Dimiter Driankov, Per Österling, Magnus Morin, Christer Hansson, Douglas Busch, Erik Sandewall, Tommy Persson, Patrick Doherty, Lennart Staflin, Lena Wigh, Christer Bäckström, Shu Hua and Ulf Söderman

10.13 Publications

RKLLAB members were authors of the following external publications (papers published in books, journals, or international conference proceedings) during 1988 or the first half of 1989. (A more complete list is given in appendix E.)

International publications 1988-89:

1. **Christer Bäckström:** A Representation of Coordinated Actions. *Proc. of the 1st Scandinavian Conf. on Artificial Intelligence*, Tromsö, Norway, March 9-11, 1988.
2. **Christer Bäckström:** A Representation of Coordinated Actions Characterized by Interval Valued Conditions. In *Proc. of ISMIS'88*, Torino, Italy, October 12-15, 1988.

3. **Christer Bäckström:** Action Structures with Implicit Coordination: In *Proc. AIMSA'88,* Varna, Bulgaria, September 20-23, 1988.
4. **Patrick Doherty:** A Semantics for Inheritance Hierachies with Exceptions using a Logic of Preferential Entailment. In *Proc. SCAI'89,* Tampere, 1989.
5. **Patrick Doherty:** A Correspondence between Inheritance Hierarchies and a Logic of Preferential Entailment. In *Ras, editor, Methodologies for Intelligent Systems,* 1989
6. **Dimiter Driankov:** Inference with consistent probabilities in expert systems. In *Proc. Int. Joint Conf. on A.I,* pp 899-901 vol 2, Milano, August,1987. Also in *Int. Journal of Intelligent Systems*, 1988.
7. **Dimiter Driankov:** Non-truthfunctional Aspects of Belief/Disbelief Pairs. In *Proc of the 1st IFSA-EURO Workshop on Approximate Reasoning,* April 5-9, 1988.
8. **Dimiter 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.
9. **Dimiter 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.
10. **Dimiter Driankov:** Towards a Many-Valued Logic of Quantified Belief: The information lattice. Presented at the *3rd IFSA Congress,* Seattle, Washington, USA, August 6-11, 1989.
11. **Dimiter 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.
12. **Dimiter Driankov:** Non-truthfunctional Quantified Beliefs. In *Proc. of the 3rd World Congress on Fuzzy Systems.* Seattle, August 6-11, 1989.
13. **Shu Hua:** An Expert System Mechanical Engineering Design System. In *Proc. of the FEMCAD-88 Conf.* Paris, 1988.
14. **Johan Hultman, Anders Nyberg, Mikael Svensson:** A Software architecture for autonomuons systems. In Proc. of the *6th Int. Symposium on unmanned, untethered submensible Technology.* Ellicott City, Maryland, June 12-14, 1989.
15. **Jalal Maleki:** Towards a system for temporal reasoning. In Proc. of the *Scand Conf. on A.I,* Tampere, Finland. Also LAIC-IDA-89-TR5.
16. **Michael Reinfrank, Oskar Dressler, Gerd 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.
17. **Erik 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.
18. **Erik 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.
19. **Erik Sandewall:** Future Developments in Artificial Intelligence. In *Proc. of European Conf. on Artificial Intelligence (ECAI),* Munich, August, 1988. (Invited paper)
20. **Erik Sandewall:** Artificial Intelligence Techniques for Complex Information Processing in Vehicles. In *Proc. XXXVI Convegno Internazionale delle Comunicazioni,* Genoa, pp. 431-437, 1988.
21. **Erik Sandewall:** Combining Logic and Differential Equations for Describing Real-World Systems. In *Proc. of the Int. Conf. on Knowledge Representation,* Toronto, Canada, 1989.
22. **Erik 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.
23. **Erik Sandewall:** A Decision Procedure for a Theory of Actions and Plan. In Z. Ras (ed.), *Methodologies for Intelligent Systems,* IV, 1989.

11.

The LINCKS Group
Intelligent Information Systems

Lin Padgham

11.1 Overview

The area of interest for the LINCKS project is Intelligent Information Systems. The LINCKS group have previously built an object-oriented database with some basic support for history maintenance and parallel editing. The aim 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 and intelligent support for a variety of user tasks. There is a strong emphasis on the interaction between theory and practice with a resulting emphasis both on relatively formal theoretical work and the use of that theory in implementing real systems.

11.2 Members

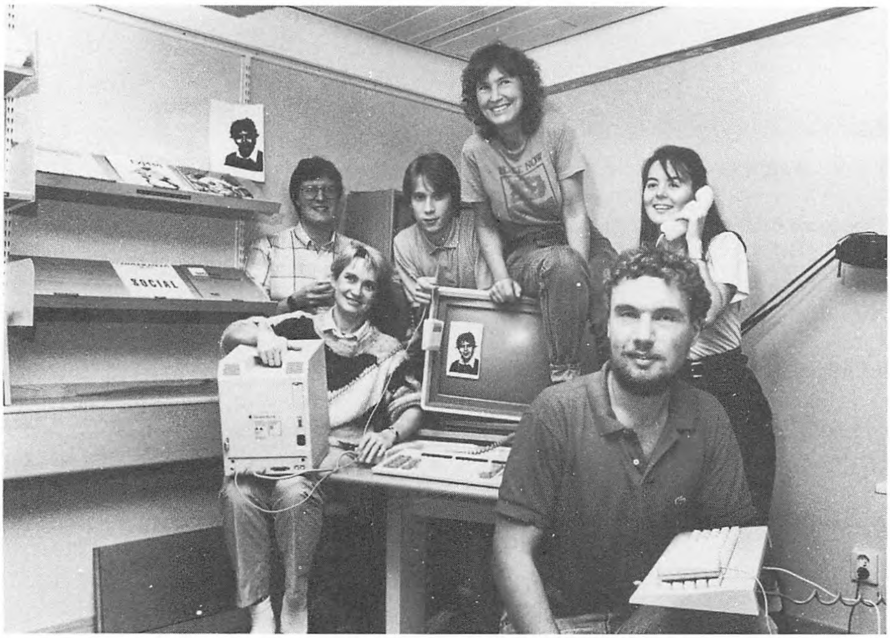
The following people have been members of the LINCKS group during all or part of 1988/89:

- Lin Padgham (Project leader, PhD June 1989)
- Toomas Timpka (PhD June 1989. Toomas is also project leader for the 'Computer Support in Consultation' (MDA) project which works in close co-operation with the LINCKS project).
- Ralph Rönnquist (licentiate degree 1987)

The work in the LINCKS group is mainly supported by STU, The Swedish Board for Technical Development, with some additional funding also from CENIIT

- Peter Åberg (licentiate degree 1989)
- Michael Jansson (senior undergraduate student, starting as graduate student July 1989)
- Ewa Rauch (research engineer)
- Nilgün Mat (starting as a guest researcher, August 1989)
- Timothy Teitenberg (starting as a guest researcher, August 1989)

Per Hedblom and Stefan Wallin, members of the Computers in Consultation project (MDA), and Tingting Zhang, a member of ASLAB, have also been working in close co-operation with LINCKS during much of 1988/89.



LINCKS personnel: Timothy Teitenberg, Eva Rauch, Mikael Jansson, Lin Padgham, Toomas Timpka and Nilgün Mat.

11.3 Activities

There are a number of closely interrelated and sometimes overlapping research activities which have been pursued within LINCKS during 1988/89. We describe each of these very briefly:

11.3.1 Non-monotonic Reasoning with Inheritance

We have further developed a model of typing which was introduced in an earlier LINCKS report (1986) to give a formal theory of default reasoning about objects. Types are described in terms of two sets of characteristics - necessary characteristics and typical characteristics. A default operator then allows assumption of the typical characteristics as long as they are not counterindicated in some way.

A representation and algorithms suitable for implementation have also been developed for gathering information according to this theory.

Lin Padgham has completed a PhD thesis with the title Non-Monotonic Inheritance for an Object-Oriented Database.

Tinting Zhang from ASLAB has been working with LINCKS using the LINCKS model of types to explore medical diagnosis and classification. She is building a prototype diagnostic aid system which we expect to be able to report further on in the next annual report.

11.3.2 Dynamic Management of Presentation

The LINCKS database is an object-oriented repository for many different kinds of information, which it is expected that users will use in widely varying contexts. The appropriate presentation of database objects may well vary depending on factors such as the context the user is working in, who the user is or what sort of display the user is sitting in front of (e.g. text terminal or display with graphic capabilities). One of the focuses of LINCKS has been to find mechanisms for allowing flexibility in the way information is presented, whilst not requiring each application to write a separate and perhaps poorly integrated user interface.

During the current year Peter Åberg has completed a licentiate thesis which describes a design for a presentation manager, which allows information regarding the user interface (both presentation and interaction) to be stored in the database and retrieved and used as appropriate. The system described has much in common with a User Interface Management System (UIMS), but also allows a greater flexibility and complexity than is available in current generation UIMS's. The major difference compared to a UIMS is that the LINCKS presentation manager also manages collection of the data which is to be displayed in the various windows, as well as instantiation of windows, buttons etc.

The LINCKS group are currently beginning implementation of the presentation manager, integrated with the editor and hypertext capabilities described below. This implementation is expected to be completed during the last half of 1989, and should give a stable new layer of the LINCKS system.

11.3.3 Editing

When editing in a database of structured objects, one wishes to be able to display a complex object, made up of sub-objects as a single editable entity, at the same time as one maintains the underlying structure of sub-objects. This is not possible with standard word-processing programs.

As a final undergraduate project, Michael Jansson has, under the guidance of the LINCKS group, developed an editor, based on Micro-Emacs, which is suitable as a text editor for text objects which are made up of sub-objects - e.g. reports made up of sections which are in turn made up of paragraphs; or forms which are made up of labels and fields to be filled in. Following the completion of his final project Michael has worked with other members of the LINCKS group on design of the next layer of LINCKS, incorporating the object-oriented editor.

There are plans to eventually extend the editor to edit also graphical objects, but this work has not yet begun.

11.3.4 Hypertext

The LINCKS database was designed and implemented as a 'hyperobject repository' where a user (or application) could easily follow links from one object to related objects. This facility has now been extended so that links can originate not only from the smallest sub-object, but also from an individual word or phrase within a (text) object. This work was done as part of a final undergraduate project, by Per Hedblom and Stefan Wallin.

A prototype interface to the hypertext facilities was developed on an Apple Macintosh, also as part of the same final project. The hypertext facilities are being incorporated, together with the editor, into the next layer of LINCKS.

11.3.5 Reasoning about change

Ralph Rönnquist has done some preliminary work in exploring description of dynamic systems. By dynamic systems we mean systems where things change without someone within the system taking an action to cause the change. Such systems are of course the rule rather than the exception, but current models for describing and reasoning about change are inadequate for such systems. Examples of changes that happen without user action in an information system could be arrival of mail, or information becoming outdated due to the passing of time.

Ralph is also currently exploring formal descriptions of change processes within the lincks system, including user-initiated changes, such as editing. This work appears promising and is expected to lead to a PhD thesis in the next 1-2 years.

11.3.6 Applications

During 1987 LINCKS group established contacts with medical users in order to have a concrete application area for exploring the efficacy of particular research approaches within the rather broad area of Intelligent Information Systems. Toomas Timpka, a medical doctor who has done considerable research in computerised support for general practitioners, has during 1988 joined the LINCKS group part time. This has allowed us to combine our experience to help guide further research efforts. Toomas is also leader for the project 'Computer Support for Consultation' which is an inter-disciplinary project to explore the use and form of computer support for a team of health care workers in a community health centre. The software systems being built for evaluation in this project are being developed within and using LINCKS.

Per Hedblom and Stefan Wallin, under the guidance of Toomas Timpka, have as a final project developed a prototype hypertext medical information system, using as a basis a textbook in general medicine. This prototype system is built on top of the existing LINCKS system. The prototype is being further developed as part of the 'Computer Support for Consultation' project.

Tingting Zhang from ASLAB has been working with Lin Padgham to build a prototype diagnostic aid system for urinary tract infections, using the LINCKS typing model. If technically successful this will also be integrated into and evaluated within the 'Computer Support for Consultation' project.

We have explored applications in the area of mechanical engineering, such as material selection and integration of a cad-system. Some contacts have been made but no significant work has yet been done in this area.

11.4 Contacts and Co-operation With Other Groups

The LINCKS group have during 1988/89 developed a number of contacts with other groups and individuals, both locally and internationally. Below are some of the people/groups with which we have developed some significant co-operation.

11.4.1 International Contacts for Exchange of Ideas

During 1988/89 we have communicated by mail and exchanged papers with Dr Robert Neches from I.S.I. California, USA. Dr Neches does research in the area of user interfaces to complex systems. During 1989/90 Peter Åberg from LINCKS will be spending a year visiting and studying with Dr Neches.

Dr Randy Miller is internationally renowned for his work with Internist medical diagnosis expert system. Dr Miller visited our group during fall 1988, and was interested in our plans for using the typing model as an aid in medical diagnosis. Dr Miller has given us access to parts of the Internist database and program which will be extremely valuable in evaluating and testing our diagnosis software.

Dr Robert Strandh is working at the University of Bordeaux, France. He has, amongst other things done some work on an object-oriented operating system (OOZ). Three members of the LINCKS group visited Dr Strandh for a week in spring 1989 and had a useful exchange of ideas regarding various concepts important to object-oriented operating systems. This co-operation is expected to continue with further visits for discussion in the future.

11.4.2 Local Research Co-operation

We have worked very closely with the Computer Support for Consultation project ever since its inception. LINCKS addresses a number of issues considered important for this project (e.g. flexibility, history maintenance). At the same time the CSCP project gives an opportunity to test the ideas developed within LINCKS in a real-world setting, obtaining feedback from real users.

During 1988 we established contact with the mechanical engineering department. We explored the possibilities for using LINCKS to further develop a material selection system developed within the mechanical engineering department. We did some preliminary work restructuring the data within their relational database into a form appropriate for LINCKS and suitable for envisaged use of the data.

During 1988/89 we have various forms of co-operation and contact with ASLAB. Two ASLAB members have worked with us on papers and/or projects. Tingting Zhang has been working on the previously described medical diagnosis system; Peter Eklund worked with Lin Padgham and Ralph Rönnquist on the paper 'A Skeptical Reasoner using Matrix Representations'. A number of LINCKS members attended ASLAB's yearly meeting, and we have regular informal discussions regarding common research interests.

11.4.3 Research Groups Using LINCKS

Professor Bryan Mayo of Århus University, Denmark, has requested and received the current version of the LINCKS database system. He intends to use the system for teaching purposes, and for student projects. We expect to have communication with Professor Mayo regarding feedback, discussion of ideas, and also exchange of software in both directions.

Professor Mannila of the University of Helsinki, Finland is working on support for document handling, using an object-oriented database which is implemented on top of a relational database. He is interested in using the LINCKS database to compare both for speed and ease of use. Professor Mannila has received a copy of the LINCKS system, and we expect to have further contact regarding installation and eventual feedback.

11.4.4 Industrial Co-operation

ProgramSystem AB is a local spin-off company with which we established contact during 1987. They have used the LINCKS database for a number of in-house tools and prototypes. We have maintained contact during 1988/89 and are currently exploring and discussing co-operation in two new areas - software related to Program System's UIMS and LINCKS presentation manager, and a prototype project for Televerket, Swedens telecommunications company.

We have also had discussions with the Mechanical Engineering Union regarding a possible co-operative project in advanced information support, together with the Department of Mechanical Engineering.

11.4.5 COST13

Lin Padgham has participated in the workshop on 'Inheritance in Knowledge Representation and Programming Languages' organised as a part of the COST 13 collaborative project number 21, on 'Advanced issues in knowledge representation'.

11.4.6 Conferences and Workshops

Members of the LINCKS group have attended the following international conferences and workshops:

- CHI '88, Computer Human Interaction, May 1988
- AAAI '88, St Paul, Minnesota, USA, August 1988
- ISMIS '88 International Symposium on Methodologies for Intelligent Systems, Torino, Italy, October 1988.
- Workshop on Inheritance in Knowledge Representation and Programming Languages, Pisa, Italy, February 1989.
- Workshop on Inheritance Reasoning and Specificity, St. Louis, Minneapolis, USA, April 1989.

11.5 Reports and Papers

The following theses, reports and papers have been written during the current year.

11.5.1 Theses

1. Lin Padgham, Non-Monotonic Inheritance for an Object-Oriented Database, 1989, ISBN 91-7870-485-5 (PhD thesis).
2. Peter Åberg, Design of a Multiple View Presentation and Interaction Manager, LiU-Tek-Lic 1989:17. (licentiate thesis).
3. Michael Jansson, Object-Oriented Editor - undergraduate thesis.

11.5.2 Published Papers

PAD88 Padgham, L. A Model and Representation for Type Information and its Use in Reasoning with Defaults. *Proceedings of AAAI '88* St Paul, Minnesota, August 20-26, 1988, vol 2, pp.

PAD89 Padgham, L. Negative Reasoning Using Inheritance, *Proceedings of IJCAI '89*, Aug 20-25, 1989, Detroit, Michigan, USA.

PAD88 Padgham, L. NODE: a database for use by intelligent systems. *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, Torino, Italy, October 17-19, 1988.

PAD89 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.

RÖN89 Ralph Rönnquist, Reasoning about Control of Dynamic Systems, accepted for publication in *Proceedings of the International Symposium on Methodologies for Intelligent Systems*, Charlotte, N.C., USA October 12-14, 1989.

TIM89 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.

TIM89 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.

ÅBE89 Arja Vainio-Larsson, Rebecca Orring, Peter Åberg, Metoder för utvärdering av människa-maskin gränssnitt: En litteraturstudie. LiTH-IDA-R-88-20.

ÅBE89 Peter Åberg, Information Views in LINCKS. *Proceedings: 1st Annual IDA Conference on Computer Information Science.*

11.5.3 Distributed Unpublished Papers

It is expected that a number of these papers will be published during the coming year.

PAD89 Lin Padgham, Ralph Rönnquist and Peter Eklund, A Skeptical Reasoner Using Matrix Representations

PAD89 Lin Padgham, Theory for Type Reasoning

ÅBE89 Arja Vainio-Larsson, Peter Åberg, Utvärdering av människa-maskin gränssnitt (Slutrapport).

12.

The MDA Group People, Computers and Work

Toomas Timpka

12.1 Introduction

Computer-based information systems have traditionally been designed for use by one person. Today, however, it is quite common among many categories of professionals, here called *consultants*, whose work essentially consists of contacts and communication with *clients*, who seek information, advice or help, to use one or more computer-based information systems in carrying out their work in the presence of clients.

In designing present-day systems, little or no heed has been taken of the real work situation of consultants; a dialogue between two persons, where an information system on one hand can provoke stress and irritation (due to unwieldiness, rigidity or opacity), or on the other can disturb and divert attention away from the client and the cause for his visit. Furthermore, concern has seldom been shown for organizational context. In e.g. health care, the consultant is usually a member of a team.

The project "Datorstö(r)d vid konsultation?"* 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. However, conclusions and results are expected to be directly relevant for many other types of consultation situations within the today rapidly growing service sector.

---

* *"Computer (*) in consultation", where the (r) makes the difference between the Swedish words for support and disturbance, either of which can be substituted for the asterisk, giving the sense, if not the feeling, of the play on words.*

The project is supported mainly from the research programme "People, computers and working life", jointly sponsored by STU, The Swedish Board for Technical Development and AMF, The Swedish Work Health Fund.

Within the project, activities at the Department of Computer and Information Science and the Department of Community Medicine have been coordinated, for the purpose of investigating diverse consultation situations in health care from the points of view of psychology, work organization, computer science and technology, and further, to develop and evaluate experimental information systems specifically aimed at supporting 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.

Work within the project is supported by a grant from the MDA program**, which is financed on an equal basis by the Swedish Work Environment Fund and the National Board for Technical Development.

Project work is carried out in cooperation with the staff of the primary health care center in Mjölby, situated 40 km west of Linköping. Close contacts are also maintained with the groups of Professors Werner Schneider and Gösta Tibblin, both at the University of Uppsala. Project supervisor is Professor Per Bjurulf from the Department of Community Medicine. The group at IDA consists of (August 1989):

- Toomas Timpka (director),
- Hans Holmgren (coordinator),
- Göran Goldkuhl,
- Lin Padgham,
- Per Hedblom,
- Lars Reshagen, and
- Leili Lind.

The County Council of Östergötland has been represented by Tom af Klercker as well as the board of the Mjölby Health Care Center, which is integrated by representatives of the direction and the labor unions. During 1988-89 work within the project has also been carried out by Peter Åberg and Stefan Wallin from IDA, as well as by Elisabeth Arborelilus, Ann-Charlotte Nilsson, Anders Södergren, Birgitta Franzén and Anna-Lena Hautaniemi from the Department of Community Medicine.

** *MDA is an acronym of the Swedish words for people, computer technology and work life.*

MDA personnel: Toomas Timpka, Lena Wigh, Leili Lind and Per Hedblom

12.2 Project activities 1988-89

Project work during the fall of 1988 (project start in october) and the spring of 1989 has been organized in three main directions: empirical studies of consultation situations at the Health Care Center in Mjölby, studies of and participation in the changes of work organization at the Center, as well as development of prototypes of computer-based information systems adapted for the consultation situation. The project is also discussed from the point of view of the health care providers at Mjölby Health Care Center, i.e. the County Council.

12.2.1 Empirical studies of the consultation situation.

A preliminary study of a video-recording method developed by members of our group, coordinated with a project concerned with the doctor-patient relationship, consisting of a method description and theoretical background, as well as recording and analysis of medical consultations in primary care has been concluded and documented in several papers [Timpka 1989], [Timpka and Arborelius 1989], [Arborelius and Timpka 1989]. Two of these papers are presented at the 6:th world Conference for Medical Information Processing

(MedInfo '89).

Data collection

Video recordings

A total of 24 medical consultations have been recorded on video tape at the health care center in Mjölby. The recordings include the complete consultations, from the moment the patient calls and is answered by the nurse at the reception desk, over the joint decision as to if and when to come and see a physician, the actual visit, including possible tests carried out by the laboratory and/or nursing staff, as well as the meeting with the physician, through to payment of the consultation fee.

Each of these recordings have been shown in their entirety to the respective patients and staff members, who have had the opportunity to comment on the recorded material. The comments have been recorded on audio tape and the moment in time in which each comment was made has been noted. These comments are now in the process of being written down by a member of the administrative staff.

Telephone consultation

In addition to the above-mentioned video recorded sessions, 22 recordings of telephone consultations have been carried out. These are cases where the patient is not scheduled for an appointment, but has the opportunity to discuss his or her problem with a nurse over the telephone. These recordings have also been shown to the respective staff members, who have had the opportunity to comment on the material in the same way as described above. The patients involved have been interviewed.

Interviews

All members of the staff of the health care center in Mjölby who come into direct contact with patients have been interviewed about their working conditions, with particular emphasis on where they turn when any problem develops. All material is being coded and recorded in a machine-readable format for future statistical analysis.

Theoretical work

In parallel with the more applied work we have developed methods of analysis which are applicable to this type of recordings. These methods have been tested on similar types of recordings, and have been modified to better suit the present purpose.

12.2.2 Systems development

The Mjölby Health Care Center has long been struggling with problems concerning organization and the functioning of day-to-day work. One of the main problems has consisted of the serious difficulty of recruiting regular district physicians.

The health care center was constructed during the era when large scale organizations were considered to be ideal, and in 1988 the County Council of Östergötland started an organizational development project, the aim of which is to determine whether the large health care center (staff numbers approximately 110) can be broken down into smaller units. Participants in the development project include representatives of the different categories of staff at the health care center. The project leader comes from the central staff of the County Council. The aim of the systems development group of the present project during its first six months (October 1988 to March 1989) has been to follow as well as participate in the County Council project at the Mjölby Health Care Center.

The County Council project is concerned with the creation of new forms of work organization. Its nature is more general than that of the rsearch project, and has not touched on issues of computer support or consultation situations per se. There is a clear overlap between the County Council project and our own research oriented development of aspects of consultation situations.

A new general design of work organization can be seen as a framework for the development of consultation situations. A comprehension of the change process at the health care center as built up from an inside view has also been considered essential. This is also one motive for participating in the process. Our candidate for a particular change method (SIM) has been adopted for the County Council project, and since this method now is common to the two projects, the possibilities for fruitful interaction between the two change processes now seem to increase. The change processes are integrated in the sense that they involve partially overlapping aspects of work and organization.

From an initial stance of being rather passive observers in the County Council project, we have gradually become more participative observers. Göran Goldkuhl has, during the period December to March, increased his participation in the project, which has been financed by the County Council on a consultative basis. The project has involved a relatively deep penetration of different problems, an analysis of important objectives of primary care and how these are grounded in the minds of the staff of the health care center, and the working over of a number of proposals for changes. As has been mentioned, the SIM method for change analysis has been used in carrying out this work.

In addition to the County Council project we have, regarding empirical work, also carried out a rather broad documentation study of the present work processes. The health care work has been documented through interviews with representatives of different categories of staff. In this study as well, the SIM method has been used.

12.2.3 Development of computer support

The work within the computer science area of the project has been carried out in cooperation with the LINCKS group and has been concentrated on two areas: design of the user interface, with a particular emphasis on software that supports interactive definition of user interfaces, and implementation of a hypertext system, where NODE, an object-oriented database developed by the LINCKS group, provides the development base.

A design model of the handling of presentation and interaction in the LINCKS system has been completed* [Åberg 1989]. The model is based on concepts and principles within the area User Interface Management Systems (UIMS). In the model, all information pertaining to presentation of data to the user is separated from other components. Methods of presentation and interaction can also be dynamically selected, at any time during a session, which makes it possible to present the same information in different ways to different users. Work on implementing the model, LOOCS, is underway within the LINCKS group.

The implementation of the hypertext system in LINCKS is structured into three program layers (Timpka, Padgham, Hedblom et al. 1989). The outermost layer is the user interface which is implemented on the Apple Macintosh with a supplementary Radius 18-inch screen. It is implemented in ObjectPascal, using the MacApp object-oriented software framework. The second layer manages the functionality specific to particular implementations (use of keyword attributes, structure and reference links, etc.). The third layer manages the database functionality, including such things as connecting *source points* in the text to the actual outgoing links, and filtering the image text so that markers for link source points are not seen.

The content of the hypertext system consists of a primary care knowledge base, *Gösta's book*, compiled by the faculty at the Department of Family Medicine at Uppsala University. The main topic is support of diagnosis, and the target group is all physicians and nurses working in primary health care. The texts are originally authored in hypertext form (non-sequentially, with a wealth of cross-references).

12.2.4 The County Council

Background

The Mjölby Health Care Center has for many years suffered from a difficulty to recruit competent general practitioners. The sheer size of the health care

---

* *The design model has been completed within the MDA-supported project.*

center, including an uptake area of 22 000 people, and with eight positions for general practitioners, only 2,2 of which have occupants at the moment, may tend to discourage potential applicants. Another quite possibly negative factor is emergency ward attendance hours of 8 am to 10 pm daily, and an uptake area of 40 000. The size of the organization, with 64 positions occupied by approximately 110 persons would seem to be somewhat too large for effective and transparent management. All of this has led to difficulties to attain our goals. In order to come to terms with these problems the primary care direction of the County Council of Östergötland started a change and development project in the fall of 1988. The board of the health care center took charge of the direction of the project, and, after hearing the suggestions of the representatives of the different categories of staff, as well as those of the of the unions, appointed a working group consisting of members of all the different categories of staff at the health care center. The head of the development department of the County Council was appointed as project director. To this working group were associated representatives of the MDA-supported research project (Hans Holmgren and Göran Goldkuhl). They have participated to a varying degree as observers and agents. The working group has chosen a change methodology (SIM) developed by Göran Goldkuhl and associates.

The situation in late spring 1989

During an explorative phase, from september 1988 to february 1989, the working group has investigated and come to an agreement about the problems and errors that are associated with the old organization. A number of graphs have been produced that synthesize the results in a graphical manner. On the basis of these graphs, possible solutions have been discussed, starting in march. The main problem areas are:

The direction:	Accessibility and transparency
Organization:	From large scale to small scale
Work activites:	The future of the emergency ward

At a late stage this spring, the working group was maintaining discussions about the shape of the future health care center. The primary question was: How is it possible, aiming for the goals of primary care (accessibility, proximity, quality, continuity and wholeness of perspective), to shape the future organization with an emphasis on the patient's demands and wishes?

References

* 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. Proc. of MedInfo '89, Elsevier, Amsterdam 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, Arborelius E. Study of the practitioner's knowledge need and use during health care consultations. Part 1: Theory and method. Proc. of MedInfo '89, Elsevier, Amsterdam 1989.

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

* Åberg P. Design of a multiple view presentation and support manager. (Licenciate Thesis.) Department of Computer Science. Linköping University, 1989.

13.
CENIIT
The Center for Industrial Information Technology

In early 1986, our University's School of Engineering decided to focus on *Industrial information technology* (IIT) as a primary area for new research efforts. The 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 it 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.

Government approval of the plan came in 1987, with funding being available from July 1988. During the academic year 1987/88 the effort in industrial information technology was planned by a group with representatives from our department (Erik Sandewall) and the departments of Mechanical Engineering, Electrical Engineering, and Physics. Dr. Tore Gullstrand from Saab-Scania also participated as a representative for industry.

The actual work in the new *Center for Industrial Information Technology, CENIIT*, started in mid-1988. It 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 fopr product development and production planning*, for instance in connection with computer-aided design and process planning.

The work in CENIIT projects is supported by the School of Engineering at Linköping University

In addition, the following research areas are in preparation:

- *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 the committee consisted of:

Erik Sandewall, Department of computer and information science (chair)
Tore Gullstrand, Saab-Scania AB
Lennart Ljung, Department of electrical engineering
Leif Nilsson, ABB STAL AB
Benny Odenteg, BT Systems AB
Hasse Odenö, Department of mechanical engineering
Christer Svensson, Department of Physics

The following projects are currently undertaken as our department's contribution to CENIIT:

13.1 Geometrical algorithms

The purpose of this activity is to develop competence in the area *efficient algorithms for three-dimensional geometrical problems and the planning of movements*, in an active communication with applied projects in other departments. The work involve sequential as well as parallel algorithms.

An example of a research topic covered is planning of movements for a linkage (robot arm) through a 2-dimensional maze, with respect to the time complexity of the traversal algorithm. Another project is concerned with data reduction in terrain modeling, involving the determination of a continous surface that approximates a given set of measurements.

This research is closely linked with other projects in ACTLAB and is further described in chapter 2 of this report.

Area leader: Per-Olof Fjällström (from 1989), Rolf Karlsson (until 1988).

13.2 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, inclduing object-oriented approaches, conceptual modelling of database contents, and integration with special types of representations, such as graphics, text and knowledge bases. The relationship to database technology for administrative applications, relational database and information retrieval systems are also of interest.

Initially research has been focused on object-oriented approches 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 the LINCKS group are established. Applications are mainly pursued in the are of mechanical engineering, especially in the areas of solid mechanics, engineering materials and production engineering.

Current activities and results so far are described in greater detail in the CAELAB chapter (chapter 4) of this report.

Area coordinator: Sture Hägglund

13.3 User Interface Management Systems

This area covers certain aspects of human-computer interaction, with a special emphasis on tools for design and management of user-computer dialogues. UIMS-related projects so far are mainly carried out as part of on-going, separately funded research (ASLAB and LINCKS). We expect issues of human-computer interaction to be of vital importance in several CENIIT project, and plan to expand with additional qualified researchers in this area. Recent visitors specializing on UIMS, include Ralph D. Hill from ECRC in Munich (now at Belcore, New Jersey) and Elaine Frankowski of Honeywell. Industrial connections in the area are established, for instance through the KTP (knowledge transfer) cooperation with TeleLogic Programsystem AB, which develops one of the first and most advanced commercial UIMS products, and through study projects with industrial liason groups (sponsored by 'Mekanförbundet').

Area coordinator: Sture Hägglund

13.4 Industrial Software Technology

The objective in the project Industrial Software Technology (Industriell programvaruteknik) 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 are difficult to study and to model in an academic environment, the approach is to start with some empirical investigations in the software industry.

This project has a basic funding from CENIIT, with the funding period starting in July 1989. Hence there are yet no results from this activity. The activity will be conducted by Bengt Lennartsson, and there will initially be at least two more persons involved part-time.

We are aware of the existing literature from empirical studies of industrial software development. However, we have the ambition to contribute significantly to the area despite the small volume of our effort. Firstly, what has been published has mainly been based on military projects within USA. Our impression is that the situation in Swedish software industry is slightly different both in terms of education of the software designers and organization of their work. Secondly, we will in our first steps concentrate on qualitative observations rather than on estimates of productivity and quality in numbers. In an earlier study, funded by Mekanförbundet, The Trade Association of the Swedish Mechanical and Electrical Engineering Industry, we identified some useful experiences from Swedish software industry, and we will study some of these in more detail.

The large scale software development in Sweden can be found in the Ericsson and in the ASEA Brown Boveri groups, and we hope establish a cooperation where we are able to identify, analyze, formulate, and make available their experiences, which will be auvery useful input into our undergraduat courses as well as an inspiration to our ordinary more tool and method oriented research projects.

Area leader: Bengt Lennartsson

13.5 Computer support for automation

This project started during the spring 1989 when Dr. Anders Törne was recruited from ABB Corporate Research in Västerås. The aim is to follow up previous cooperation in applied projects with more fundamental studies of computer support for engineering, in particular manufacturing processes. The research takes an interdisciplinary approach and emphsizes cooperation with the mechanical engineering department.

From July, 1, 1989 the project forms the core of the new CAELAB (*computer Assistance in Engineering*). A more detailed description of the research area and project goals is given in chapter 4 of this report.

Area leader: Anders Törne

13.6 Conceptual text representation for generation and translation

The purpose of this project is (i) to develop a representation of textual structure and function 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.

The project intends to work with real but restricted forms of texts. 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. 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. The first text type to be considered is most likely going to be machine part descriptions as they appear in product manuals.

The software to be developed can be embedded in application systems in various ways. A high-level representation of a text can form the basis of multi-lingual generation, given grammars and dictionaries for the languages of concern. Within the current project the languages of primary concern will be Swedish and English. It can also be used for translation purposes 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 a text.

Area leader: Lars Ahrenberg

13.7 Plan-guided systems for autonomous vehicles and computer-integrated manufacturing

This area is for the time being covered by existing projects in IDA's laboratories, primarily within LAIC (Linköping AI Consortium) involving the cooperation programme with Foa (The Defense Research Institute) and the IT4 activities on road trafic systems (Prometheus). For further details, refer to section 1.5.2 and to chapter 10 on RKLLAB.

Additional CENIIT funding for this research is not claimed at present.

Area coordinator: Erik Sandewall.

14.

Educational Programmes

Figure 14.1 below indicates the levels of degrees in the Institutes of Technology (i.e. schools of engineering) in the Swedish university system. The figures indicate the nominal numbers of years for the studies in each step.

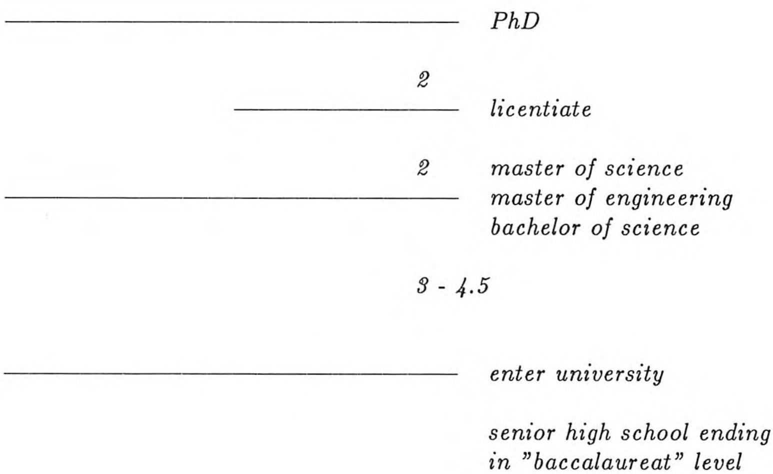


Fig 14.1. *Levels of degrees*

The graduate study programme provides the studies *from* the level of master of engineering, *to* the licentiate and/or PhD degrees. 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. The courses given by our department for the undergraduate education, up to the master's degree level, are listed in appendix C.

14.1 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 *basic courses*, each of which is given approximately every 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. Admission to graduate studies is nominally free for students with the appropriate qualifications, but it is not realistic nor recommended to start graduate studies without being admitted as a member of one of the research groups.

The programme leads to one of the following degrees:

1. *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.
2. *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.

Recent courses and seminars are listed in appendix B.

14.2 Organization of undergraduate education

Undergraduate education accounts for roughly 35% of the total budget in the department, with an additional 5% for continuing education and technology transfer activities. (Close to 15 MSEK together.)

The undergraduate education at IDA is organized as follows:

The Committee for Undergraduate Education (IDUN - IDA's undervisningsnämnd), headed by Anders Haraldsson, is responsible for the contents of courses given by the department and the planning of teachers for the courses. There are representatives from the student unions in the board.

The department is responsible for the subject areas computer science (datalogi), telecommunication and computer systems (telesystem/datorsystem) and administrative data processing (administrativ databehandling).

There is a responsible director of undergraduate studies (studierektor) for each subject area. The computer science area is further divided into subareas. The directors are

Anders Haraldsson, computer science
Lise-Lotte Raunio, administrative data processing

Mikael Patel, telecommunication and computer systems, system programming
Arne Jönsson, artificial intelligence
Mats Wiren, natural language processing
Rolf Karlsson, theoretical computer science (until 1989)

The directors and many of the courses belong closely to a research laboratory and the division of areas of responsibilities reflects to a great extent our laboratory organization.

Appendix C lists the courses given during the academic year 1988/89, teaching personal and computer facilities for undergraduate education.

14.2.1 Review of teaching organization

This section briefly reviews the undergraduate education system for the benefit of the international reader, using terms from the U.S. educational system for comparison. More details on courses etc are then given in appendix C.

Students are admitted to the university after having completed senior high school with a matriculation exam, usually the year when the student is 19 years old. The student is admitted to a specific "study line", which defines what he or she will be majoring in. The school of engineering has the following study lines:

Mechanical Engineering
Electrical Engineering and Applied Physics
Engineering and Economics
Computer Engineering
Computer Science

The first four of these study lines are nominally for 4 1/2 years, in practice often more. They lead to the degree which in Swedish is termed "civilingenjör", but which is used for all branches of engineering, not only for civil engineering in the English language sense. It is comparable to a Master's degree, with the qualification that the concluding thesis project is of moderate size - about three months' work - and is usually performed in industry and not as a research assignment. The courses in the study line are almost exclusively of a technical character; the student is assumed to have acquired the necessary knowledge of "Western Civ", foreign languages, etc. before matriculation.

For the Computer Science study line, which is a knowledge engineering type education, the nominal study time is 4 years instead of 4 1/2. Study lines in the School of Arts and Sciences are often for 3 years, and end with a Bachelor of Science degree.

In the School of Engineering, the study lines are not directly tied to departments. There is a matrix organization, where each study line buys courses from (at least potentially) all the departments. In particular, IDA sells courses to all five study lines. When we refer to courses in the "undergraduate" eduction in this volume, we really mean the courses in these study lines leading to a Master's or a Bachelor's degree.

All the students in 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 students who go to graduate school must have completed the 'undergraduate' degree with (in principle) 60 points of computer science courses. One full-time academic year is 40 points so one point is roughly one work-week. The graduate study proceeds through two successive levels, the "teknologie licentiat" degree nominally after two years, and the "teknologie doktor" (Ph.D. in engineering) nominally after two additional years. Both levels require a combination of coursework and a thesis. The "tek.lic." can therefore be seen as an advanced Master's degree with a substantial, research oriented thesis.

There are two reasons for having the licentiate degree in the system. For those students wishing to go into industry, it is a good break point in the education. Industry wants people to be as young as possible when they come, and the licentiate has already had a participation in research which is a sufficient background for many industrial jobs. Secondly, for students who contemplate whether to enter a research education at all, the Ph.D. seems a very long way off, and the licentiate is a more immediate and tangible goal. Many of the licentiates continue towards the Ph.D. but appreciate having had the breakpoint.

Students that come from the 4-year Computer science study line or from the 3-year Systems Analysis study line (Computer science and business administration) go to "filosofie licentiat" and "filosofie doktor" degrees. Three years on those lines gives 60 points in computer science, so the 2+2 years research education is counted from the end of the third year. Thus most of the courses in the fourth year of the Computers science study line are valid for graduate studies also. Students are encouraged to complete the study line, independently of any graduate studies. Stipends for graduate studies will normally not be available until after that fourth year of study.

14.3 Undergraduate curricula in the school of engineering

The Linköping University has since 1975 had a strong position in undergraduate curricula and teaching in computer science. Linköping is today the only Swedish university which offers the three main 3-4.5 years undergraduate study programs in the area of computer science and systems analysis. An increasing part is also other educational activities such as a continuing education programme in computer science for industry.

As the first institute of technology in Sweden we started 1975 the D study line (Computer Science and Technology - "Datateknik-linjen") as a four-year (now converting 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 1982 at all other Swedish institutes of technology.

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. This programme is at Linköping 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 what 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 is also passing through changes. The new computer science base with discrete mathematics, logics and to start programming 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 possibilities for specialization during the last years. In particular a new set of courses prepare students for software engineeing 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.

The mechanical engineering programme has been extended with a new specialization that combines mechanical and computer engineering. We believe that especially research in artificial intelligence will be significant within that specialization.

14.4 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 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 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.

14.5 Undergraduate studies in administrative data processing

Including management information systems analysis and information systems analysis and design.

The group for administrative data processing (the ADP group) is responsible for the courses given by IDA in the undergraduate *Systems analysis* study program in the School of Arts and Science, Linköping University.

The program for systems analysis ranges over three years of fulltime 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 program 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 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, matematics and statistics. The second year ends with about five months of on the job training.

During the last year the students can choose one of the following three specializations:

- Methods for data analysis (data analysis), aimed at statistical methodology and statistical analysis methods. This specialization includes documentation and presentation of projects where storage and retrieval of data are crucial.
- Development of computer programs and program systems (program

development) aimed at program development, methodology and technology. This specialization contains courses about operating systems, compilers, interpreters etc.

- Development of information systems (systemeering), aimed at methodology for design and evaluation of information systems. The program includes in-depth studies of budgeting and accounting and their relation to project management and systems budgeting.

All three specialisations end with a term-paper reporting the development and implementation of an individual project.

14.5.1 Research connection

Administrative data processing belongs to the School of Arts and Science, where research is concentrated in 'themes' and located in a separate department. The ADP group thus does not have a research organization of its own, although several of the group members are affiliated with the research laboratories in the department or enganged in independent research (Göran Goldkuhl).

The subject area covered by the group deals mainly with social aspects of design and use of software for administrative applications in private companies and public services. Essential problems are the transition from natural to formal languages and vice versa together with prerequisites for, constraints on, and effects of computerized support for activities where teamwork, personal judgement and experience traditionally have been, and are expected to be, of great importance. This topic comprises systems development and tools for analysis of information requirements and tools for prototyping, the drawing up of technical requirements specifications and other kinds of user-oriented documentation and evaluation of effects caused by the use of computerized systems. It does also contain - from a general point of view - social methodology for describing administrative professional activities, for implementation, maintenance and evaluation of user-oriented computerized support.

The undergraduate study programme for Systems Analysis takes the main part of the group's teaching efforts. Beyond that we give separate single-subject courses to the level of postgraduate studies as well as courses in other study programmes.

Appendix A

Department organization

The Department of Computer and Information Science (IDA) at Linköping University covers three teaching subjects (computer science, 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. 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.

The lab leader is responsible for supervision and guidance of the work in his 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.

Important and general issues regarding research or undergraduate studies are treated by the research committee or the committee for undergraduate education respectively. The research committee, headed by Erik Sandewall and with Lillemor Wallgren as secretary, handles research activities and graduate education. This committee suggests the annual budget for each lab, based on grant situation, and can also modify the lab structure by merging, splitting, creating, or deleting labs and appointing lab leaders. Admission of doctorate 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.

The Committee for Undergraduate Education, headed by Anders Haraldsson with Carina Björkman as secretary, is responsible for the organization of undergraduate courses and continuing education for industry. Most of the

Department of Computer and Information Science

Organization:

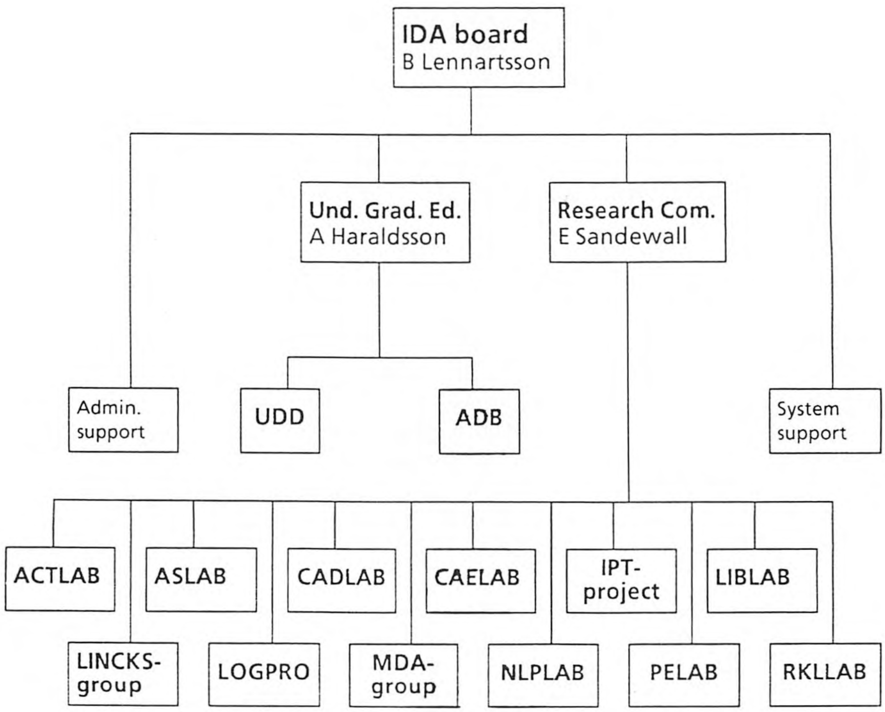


Figure A.1. *Organization of the department.*

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. The executive responsibility for undergraduate studies are taken by the *directors of studies*, with Anders Haraldsson responsible for the study programs within the School of Engineering and Lise-Lotte Raunio for those in the School of Arts and Science.

Formally all significant administrative decisions, such as the annual budget are taken by the Department Board. The board is chaired by Bengt Lennartsson, with Inger Emanuelson as secretary. 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.

Running economy and personnel issues are handled by Inger Emanuelson, who is also the leader for the group providing administrative 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 equipment of course can be granted that right for a specific period of time.

The department budget for the fiscal year 1988/89 balances at 34.7 MSEK. (One MSEK is at present approximately 0.15 USD.) The resources for undergraduate education supplied by the university amount to 13.4 MSEK, and corresponding resources for research and graduate education are 5.8 MSEK. The research activities are thus heavily dependent on external sources, where the Swedish Board for Technical Development, STU, is the main contributor (88/89: 7.6 MSEK). Additional funds are provided by the Swedish Coumcil for Planning and Coordination of Research, FRN, (88/89 1.3 MSEK). During 1988/89 the Swedish National Information Technology Program has supplied about 3.6 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 1 MSEK. Commissioned education programmes for industry are budgeted at about 2 MSEK 1987/88. Costs for office space and investment in equipment are not included in the above figures.

Department leadership:

Bengt Lennartsson, department chairman

Erik Sandewall, research committee chairman
Anders Haraldsson, undergraduate education committee chairman

Administrative office:

Inger Emanuelson, administrative manager
Carina Björkman, general educational secretary
Lillemor Wallgren, general research secretary

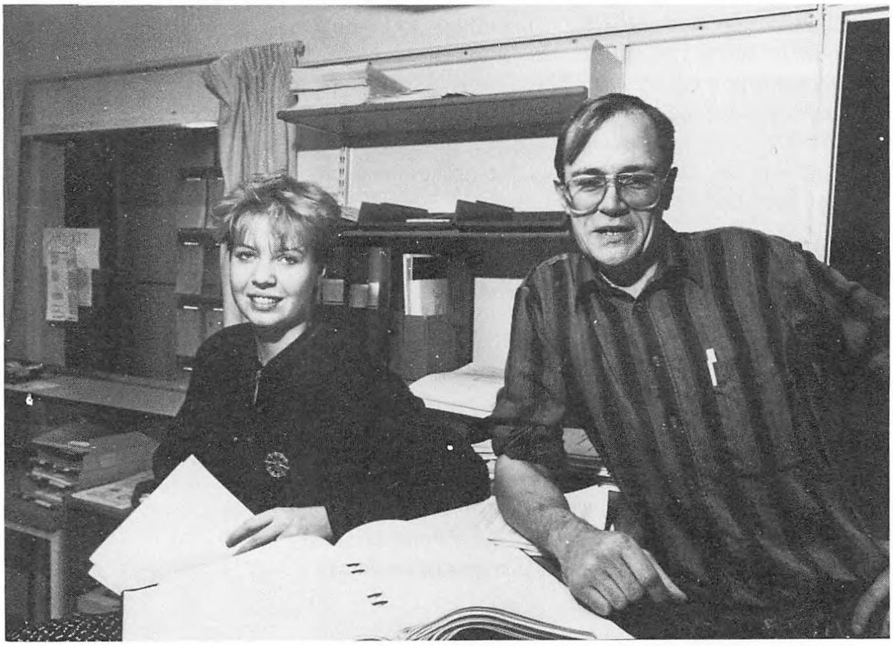
Madeleine Allert, office assistant
Britt-Marie Ahlenbäck, secretary

Anne-Marie Jacobson, secretary
Barbara Ekman, secretary
Carita Lilja, secretary
Lisbeth Linge, secretary
Gunilla Lingenhult, secretary
Nils Nilsson, porter
Bodil Mattsson Kihlström, secretary
Siv Söderlund, secretary
Lena Wigh, office assistant

Technical services:

Anders Aleryd, managing engineer
Mats S Andersson, senior research engineer

Ulf Dahlén, research engineer
Leif Finmo, research engineer
Dimitrios Fotiadis, research engineer
Arne Fäldt, senior research engineer
Bernt Nilsson, research engineer
Björn Nilsson, senior research engineer
Rolf Nilsson, lecturer
Peter J. Nilsson, research engineer
Göran Sedwall, engineer
Ola Strömfors, lecturer
Katarina Sunnerud, research engineer



Madeleine Allert and Nils Nilsson

Appendix B

Graduate Study Program.

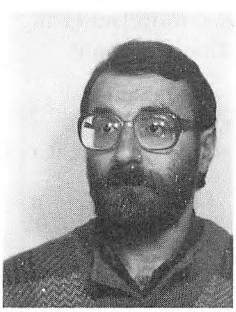
Faculty presently engaged in graduate study program.

Ahrenberg, Lars, PhD, Uppsala 1987. Assistant professor (*högskolelektor*), computational linguistics. Group leader, NLPLAB. Previous affiliation Uppsala and Göteborg.

Natural language processing, computational linguistics, user interfaces.

Douglas Busch, PhD, Rockefeller 1973. Assistant professor (*högskolelektor*) of logic and theoretical computer science. Previous affiliation Mcquarie University, Sydney, Australia.

Application of theories from formal logic to problems in theoretical computer science and artificial intelligence; algebraic specification theory, intuitionistic type theory non-monotonic logic; philosophical questions in artificial intelligence.



Dimiter Driankov, PhD, Linköping 1989. Assistant professor *(högskolelektor)*, logic and AI.

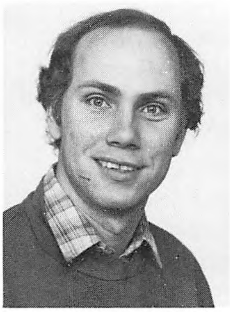
Reasoning under uncertainty, many-valued logics, knowledge-based plan-recognition, decision support systems.

Johan Fagerström, PhD, Linköping 1988. Assistant professor (*högskolelektor*), computer science. Director of graduate studies.

Distributed systems, parallel systems, operating systems.

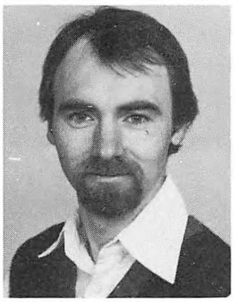
Per-Olof Fjällström, PhD, Stockholm 1985. Assistant professor (*högskolelektor*), theoretical computer science. Previous affiliation KTH and IBM. Group leader, ACTLAB.

Computational geometry, analysis of algorithms, data structures.



Peter Fritzson, PhD, Linköping 1984. Assistant professor (*högskolelektor*), computer science. Thesis supervision in PELAB.

Tool generation, incremental compilation technology, programming environments, program transformations.



Göran Goldkuhl, PhD, Stockholm 1980. Associate professor (*docent, högskolelektor*), administrative data processing. Group leader in ADP 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. Director of undergraduate studies in computer science. Previous affiliation Uppsala.

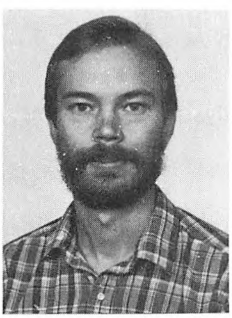
Programming languages and systems, programming methodology, program manipulation, partial evaluation.

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.

Sture Hägglund, PhD, Linköping 1980. Professor of knowledge-based systems. Group leader, ASLAB. Previous affiliation Uppsala.

Expert systems and artificial intelligence applications, database technology, human-computer interaction.

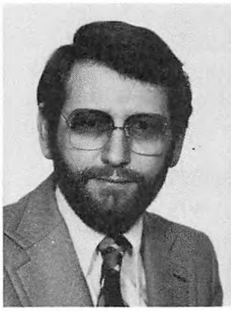


Rolf Karlsson, PhD, Waterloo 1984. Assistant professor (*högskolelektor*) until 1988, theoretical computer science. Now at Lund University.

Data structures, algorithm analysis, computational complexity, computational geometry.

Krzysztof Kuchcinski, PhD, Gdansk 1984. Acting professor of computer systems. Group leader, CADLAB. Previous affiliation Technical University of Gdansk, Poland.

Computer architecture, VLSI, computer-aided design, system testing.

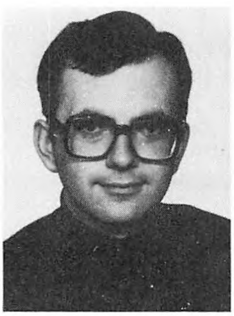


Harold W. Lawson Jr., PhD, Stockholm 1983. Professor of telecommunication and computer systems until 1988. Several previous affiliations, also in industry. On leave for Swedish International University 1987-88. Now independent consultant in Stockholm.

Computer architecture, VLSI, Computer-aided design, methodology of computer-related education and training.

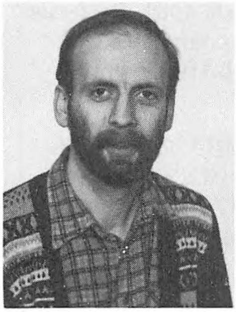
Bengt Lennartsson, PhD, Göteborg 1974. Associate professor (*högskolelektor*), software engineering. Previous affiliation Luleå. Group leader, PELAB until 1988. Department chairman.

Programming environments, real-time applications, distributed systems.



Christos Levcopoulos, PhD, Linköping 1987. Associate researcher (*forskarassistent*), theoretical computer science until 1988. Now at Lund University.

Computational geometry, analysis of algorithms, data structures.



Andrzej Lingas, PhD, Linköping 1983. Associate professor (*docent, högskolelektor*), theoretical computer science until 1988. Previous affiliation Warszawa and MIT. Group leader, ACTLAB. Now professor of computer science at Lund university.

Complexity theory, analysis of algorithms, geometric complexity, graph algorithms, logic programming, VLSI theory.

Jan Maluszynski, PhD, Warszawa 1973. Professor of programming theory. Several previous affiliations. Visiting professor at the Department of Computer science, University of Utah, Salk Lake City 1988-89. Group leader, LOGPRO.

Logic programming, formal language theory, amalgamation of programming paradigms.

Lin Padgham, PhD Linköping 1989. Assistant professor *(högskolelektor)*, computer science. Previous affiliation University of Oregon, USA, and industry. Group leader, LINCKS.

Intelligent information systems, default reasoning, object-oriented systems, representational issues, inheritance reasoning.

Zebo Peng, PhD, Linköping 1987. Assistant professor (*högskolelektor*), computer architecture.

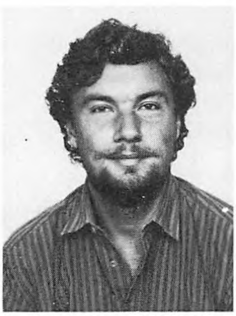
Automated synthesis of digital systems, formal description of hardware, VLSI, computer-aided design, computer architecture.

Erik Sandewall, PhD, Uppsala 1969. Professor of computer science. Research committee chairman. Group leader, RKLLAB. Several previous affiliations.

Representation of knowledge with logic, autonomous agents, knowledge-based planning.

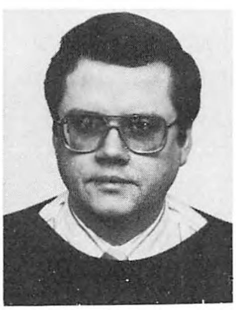
Erik Tengvald, PhD, Linköping 1984. Assistant professor (*högskolelektor*), computer science until 1988. Now at Carlstedt Elektronik AB, Göteborg.

Artificial intelligence, knowledge representation, planning and problem solving, expert systems.



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. Assistant 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 process control and supervision. Robot programming.

Graduate Study Course Program 1987-88

Basic and Occasional Graduate Courses:

Lower Bound Techniques (Rolf Karlsson)

Analysis and Complexity of Parallel Algorithms (Andrzej Lingas)

Algorithm Analysis and Complexity Theory (Andrzej Lingas)

Computational Geometry (Christos Levcopoulos, Andrzej Lingas)

Office Information Systems - OIS (Roland Hjerppe)

Advanced Computer Architectures (Krzysztof Kuchcinski, Mikael Patel)

Negation in Logic Programming (Maurizio Martelli, Jan Maluszynski)

Logic Survey (Douglas Busch)

Advanced Course in Compiler Construction, especially Incremental Compilation (Peter Fritzson and invited lectures)

Reason Maintenance Systems (Michael Reinfrank)

Qualitative Reasoning (Peter Struss)

Introduction to Petri Nets (Krzysztof Kuchcinski, Jan Maluszynski)

Human-Computer Interaction (Lars Ahrenberg, Sture Hägglund, Arja Vainio- Larsson)

Research-Related Courses and Seminars:

Förändringsanalys (Göran Goldkuhl)

Systemvärdering - seminarieserie (Göran Goldkuhl)

Systemutvecklingsmetoder - en jämförande analys (Göran Goldkuhl, Birger Rapp, IPE)

Kunskapsomgivningar på parallella maskiner (Erik Tengvald)

Conceptual Structures and Knowledge Base Management Systems (Sture Hägglund)

HYPERCATalog-projektet (Roland Hjerppe)

Logikprogrammering - seminarieserie (Jan Maluszynski)

Planstyrda system (Erik Sandewall)

Software Reliability (Bo Bergman, IKP)

Graduate Study Course Program 1988-89

Basic and Occasional Graduate Courses:

Computer Algebra (Rolf Wasen, Rolf Karlsson)

A Survey of Parallel Algrithms for Shared-Memory Machines (Jyrki Katajainen, Christos Levcopoulos, Andrzej Lingas)

Computational Geometry (Jyrki Katajainen, Christos Levcopoulos, Andrzej Lingas)

Data Structures and Graph Algorithms (Andrzej Proskurowski)

Design Methodologies for VLSI Computer Architecture (Zebo Peng, Krzyztof Kuchcinski)

Denotational semantics (Wlodek Drabent)

Category Theory (Andrzej Tarlecki)

Natural Language Understanding (Lars Ahrenberg)

Epistemic Logics (Douglas Busch)

Non-Monotonic Logics (Michael Reinfrank)

Logic II (Douglas Busch)

Modal Logic (Douglas Busch)

Advanced Course in Artificial Intelligence (Dimiter Driankov, Erik Sandewall and Patrick Doherty)

Research-Related Courses and Seminars:

Informationsbehovsanalys (Göran Goldkuhl)

Knowledge Acquisition for Expert Systems (Sture Hägglund)

Engineering Data Bases (Sture Hägglund)

Automated High-Level Synthesis of VLSI Digital Systems - Seminaries (Krzysztof Kuchcinski)

Partialevaluering - Seminaries (Anders Haraldsson)

Graduate Study Course Program 1989-90

(As planned 1989.)

Basic and Occasional Graduate Courses:

Distribuerade Operativsystem (Johan Fagerström)

Temporal Logics (Douglas Busch)

Programmeringsmiljöer (Bengt Lennartsson)

Distribuerade Databaser (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 (Peter Eklund, John H. Holland, et al.)

Research-Related Courses and Seminars:

Knowledge Engineering Methodology - studiecirkel (Sture Hägglund)

Intelligent Tutoring Systems - seminarieserie (Sture Hägglund)

Grammatikteori (Lars Ahrenberg)

A Selection of Seminars 1988/89

General seminars spring 1988

19/1 Neil Jones, University of Copenhagen. Interpretation and its application to logic programs.

5/2 Dan Strömberg, IDA. Licentiatföreläsning. Transfer and distribution of Application Programs.

23-25/2 Kent Petersson, Chalmers Tekniska Högskolan. Martin-Löf Type Theory. Type Theory as a programming language.

25/2 John Gilberg, Cornell University. Parallel Graph Algorithms in Sparse Matrix Computation.

4/3 Kristian Sandahl, IDA. Licentiatföreläsning. Case Studies in Knowledge Acquisition, Migration and User Acceptance of Expert System.

8/3 Sven-Olai Hoyland, University of Bergen. Bin-Packing in 1.5 dimensions.

16/3 Yvonne Waern, Psykologiska Inst. Stockholm. Datorsystem och mänskligt tänkande.

3/4 Pierre Deransart, INRIA, France. The formal Specification of Prolog Standard.

12/4 Doc. Ingemar Nordin, Tema H. Vetenskap och Teknologi.

3/5 Pierre Deransart, INRIA France. The formal Specification of Prolog Standard.

10/5 Stefan Arnborg, NADA, Stockholm. Generalisering av Ohms lagar för parallell- och seriekoppling.

25/5 Thomas Lozano-Perez, MIT Artificial Intelligence Laboratory. HANDEY: A Task-Level Robot System.

26/5 Walter Tichy, University of Karlsruhe. What can Software Engineers Learn From Artificial Intelligence? Tolls for Software Configuration Management. EDGE: A Knowledge-based Editor for Directed Graphs.

30/5 Allen Goldberg, Kestrel Institute i Palo Alto. An Overview of Activities at Kestrel Institute and KIDS: Kestrel's Interactive Development System.

31/5 Allen Goldberg, Kestrel Institute i Palo Alto. Optimizing Iterative Expressions.

6/6 Mats Wiren, IDA. Licentiatföreläsning. On Control Strategies and Incrementality in Unification-Based Chart Parsing.

9/6 Christer Bäckström, IDA. Licentiatföreläsning. Reasoning about Intedependent Actions.

9/6 Rolf Karlsson, IDA. Docentföreläsning. Geometrisk komplexitet på ett gitter.

9/6 Claus Lewerentz, Aachen Technical University in West Germany. The IPSEN (Incremental Programming Support ENvironment) applied to Programming in the Large.

General seminars fall 1988

14/7 Jörg Sack, Carleton University. How to throw an object through a

	window or Toward an automatic disassembling or a 3-dimensional machine assembly.
24/8	Subir Kumar Gosh, TATA, Institute of Fundamental, Bombay. An output sensitive algorithm for computing visibility graphs.
5/9	Paul J. Moran, Department of Computer Science, Trinity College, Dublin. The Unification of different Aspects of Computer Science or the Language is the Environment.
5/9	Yoav Shoham, Stanford University. Icke-monoton logik och temporal reasoning.
12/9	Ray Ford, Department of Computer Science, Iowa, USA. Real Time software.
15/9	Johan Fagerström, IDA. Disputation. A Paradigm and system for Design of Distributed Systems.
18/9	Dr. Artur Krepski, Institute of Computer Science, Polish Academy of Science. CIA is an environment to conduct experiment in code inprovement (optimization).
21/9	Johan Hultman, IDA. Licentiatföreläsning. A Software System for Defining and Controlling Actions in a mechanical System.
30/9	Mike Weintraub, Columbus, Ohio. An Explanation Based Approach to Corrective Learning.
30/9	Per-Olof Fjällström, Stockholm. Computer-Aided Geometric Design.
4/11	Dimiter Driankov, IDA. Disputation. Towards a Many-Valued Logic of Quantified Belief.
15/11	Jim Howe, University of Edinburgh. Artificial Intelligence gets into First Gear.
28/11	Erland Jungert, FOA. Docenturföreläsning.
25/11	Tim Hansen, IDA. Licentiatföreläsning. Diagnosing Faults using Knowledge about Malfunctioning Behavior.
6/11	Sture Hägglund, IDA, CENIIT arbetsseminarium. Tekniska databaser och kunskapsbaserade system.
13/12	Dr. John Hallam, University of Edinburgh. Robotics at Edinburgh University, Featurebased navigation.
15/12	Lee Naish, Melbourne University. Nu-prolog debugging environment.

General seminars spring 1989

17/1	David L Parnas, Queen's University, Canada. A Mathematical Basis for Computer Assisted Software Engineering.
31/1	Dr Don Sannella, University of Edinburgh. Formal development of ML programs from specifications.
2/2	Dany Suk's group, Chalmers Tekniska Högskola. Dataflow-like hardware accelerator for functional languages.
7/2	Sven Holmberg, Volvo PV. 3-D rasterteknik för geometrisk beskrivning och dess tänkbara användning inom bilindustri.
6/3	Anders Törne, IDA. CENIIT-seminarieum, Datorstöd för automation.
14/3	Linda Smith, IDA. Design, Development, and Evaluation of Expert Search

Intermediary Systems.

21/3	Lennart Rohlin, Sandvik Coromant. Expertsystem för planering av skärande bearbetning.
29/3	Jonas Löwgren, IDA. Licentiatseminarium. Supporting Design and Management of Expert System User Interfaces.
13/4	Ola Petersson, IDA. Licentiatseminarium. An Adaptive Sorting in Sequential and Parallel Models.
18/4	Heikki Mannila, Helsingfors Univ. Unifications, deunifications and their complexity
18/4	Åsa Rudström, Stockholms Univ. Case Based Reasoning and Dynamic Memory.
25/5	Prof. R. Ubar, Univ. Tallin. Alternative Graphs and Digital Device Testing.
29/5	Robert Glass, USA. Cognitive Aspects of (Software) Design.
30/5	Yngve Larsson, IDA. Licentialseminarium. Dynamic Configuration in a Distributed Environment.
6/6	David Etherington, Art. Int. AT&T Bell Lab. Paradoxes of Commonsense Reasoning.
6/6	Jakob Nielsen, Univ. of Denmark. Usability Engineering at a Discount.
8/6	David Etherington, Art. Int. AT&T Bell Lab. Hierarchical Knowledge Bases and Efficient Disjunctive Reasoning.
9/6	Lin Padgham, IDA. Disputation. Non-Monotonic Inheritance for An Object-Oriented Knowledge-Base.
9/6	Eduard Hovy, Inf. Science Inst, USCL. Text generation under pragmatic constraints.
15/6	John Conery, Oregon. Object Oriented Programming with First order Logic.
20/6	Stephen Fickas, Univ. of Oregon. Critiquing Software Specifications.
20/6	Dr Saumya Debray, Univ. of Arizona. A Dataflow Analysis Scheme for Logic Programs.

Appendix C

Undergraduate Education.

1. Undergraduate teaching in the School of Engineering

The group for undergraduate teaching (the UDD-group) is responsible for courses in the two subjects *Computer Science* and *Telecommunication and Computer Systems* given in the undergraduate study programs in School of Engineering, Linköping University. These study programs, and number of students accepted annually, are:

Computer Science (C) for 30 students
Computer Science and Technology (D) for 120 students
Industrial and Management Engineering (I) for 180 students
Mechanical Engineering (M) for 120 students
Applied Physics and Electrical Engineering (Y) for 180 students

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. See part 1.3.3.

Courses. During 88/89 IDA will give a total of approx 75 different courses.In the engineering study programs IDA gives 50 courses with a total of 3300 students, 10 single-subject courses, and about 14 external courses for industry with about 500 participants. All engineering programs have at least one introductory course in computer science and programming.

In the C- and D-programs and in the variants towards computer science in the M- and Y-programs (which students can choose after the second year) there are courses in

- programming methodology
- assembly programming
- data structures
- data bases
- compiling techniques
- principles of programming languages

- concurrent programming
- operating systems
- artificial intelligence
- computer networks
- computer architecture
- computer aided design of electronics
- discrete simulation

The C-and D-programs include two software projects. One done individually during the first year and one in a group during the third. In the projects both oral presentations and written reports are required.

In the C-program a number of human-oriented courses are given:

- linguistics, introductory course
- computational linguistics
- psychology, introductory course
- psychology of communication
- interactive systems

There are also courses in theoretical computer science;

- logic, introductory course
- formal languages and automata theory
- programming theory
- logic programming

and courses in artificial intelligence:

- introduction to AI
- AI programming
- knowledge representation
- natural language processing

Computer facilities. A variety of computer systems are available to our students. Most courses use a DEC-20 computer running the TOPS-20 operating system and supporting about 60 terminals.

There are two UNIX computers (one SUN 3/280 and one Gould PN6000) for teaching purposes with totally 25 terminals, two PC laboratories with MacIntoshes and IBM compatible PC's, and one laboratory with twelve Xerox LISP machines and one with eight SUN workstations.

There are 11 terminal rooms (8-9 terminals per room) and a network for connecting terminals to the various computer systems available for educational purposes.

Staff. The teaching is done by full or half time employed lecturers, by other persons with research appointment, by graduate students having teaching assistantships, and by the students themselves as part-time course assistants.

During 88/89 the staff consists of

10 full time and 1 half time senior lecturers (associate professors)
7 full time and 3 half time lecturers (assistant professors)
6 other persons, professors and research assistants
about 40 postgraduate students with 25% - 50% teaching assistantships
c. 2 teachers from other subjects and from industry
c. 45 part-time course assistants

Personnel.

Anders Haraldsson, PhD, associate professor in computer science, director of undergraduate studies
Barbara Ekman, secretary

The following persons from IDA are teaching one or more courses:

Lars Ahrenberg, PhD
Pia Arendell, BSc
Rober Bilos, Lic
Staffan Bonnier, MSc
Douglas Busch, PhD
Nils Dahlbäck, B.A.
Patrick Doherty, BSc
Johan Fagerström, PhD
Björn Fjellborg, MSc
Anders Haraldsson, PhD
Sture Hägglund, Prof
Arne Jönsson, Lic
Mariam Kamkar, Lic
Rolf Karlsson, PhD
Krzysztof Kuchcinski, PhD
Christian Krysander, MSc
Tony Larsson, Lic
Bengt Lennartsson, PhD
Jalal Maleki, Lic
Jan Maluszynski, Prof
Magnus Merkel, B.A.
Sven Moen, MSc
Lotta Månsbacka
Ulf Nilsson, MSc
Torbjörn Näslund, MSc
Kerstin Olsson, MSc
Tommy Olsson, MSc

Mikael Patel, Lic
Zebo Peng, PhD
Ola Petersson, Lic
Ivan Rankin, BSc
Roland Rehmnert, MSc
Anders Räntilä, MSc
Erik Sandewall, Prof
Nahid Shahmehri, Lic
Ola Strömfors, Lic
Katarina Sunnerud, MSc
Erik Tengvald, PhD
Inge Wallin
Olle Willen, BSc
Mats Wiren, Lic

The Licentiate exam is between MSc and PhD.

Carina Björkman, Lena Andersson, Åke Pettersson, Kerstin Olsson, Anders Haraldsson, Arne Jönsson, Katarina Sunnerud, Tommy Olsson, Lotta Månsbacka and Christian Krysander.

Listing of Undergraduate Course Program 1988/89

Course (in Swedish)	Teacher
Databaser (C3, I4, Y4)	Christian Krysander
Databaser (M4)	Christian Krysander
Programmeringsspråk (C4, D4)	Tommy Olsson
Orientering datateknik och datorutrustning (C1, D1)	Anders Räntilä
Programmering i Ada(C4, D4, Y4, I4)	Tommy Olsson
Programmeringsmiljöer (C4, D4)	Bengt Lennartsson
Ada och programspråk (D3)	Tommy Olsson
Systemutveckling, teori och tillämpning (C4, D4)	Pia Arendell
Artificiell intelligens D (D3)	Ivan Rankin
AI-programmering (C4)	Jalal Maleki
Logik, grundkurs (C1, D4, D2)	Erik Sandewall
AI-kunskapsrepresentation (C4)	Douglas Busch
Psykologi grundkurs (C2)	Nils Dahlbäck
Databehandling av naturligt språk (C4)	Mats Wiren
Processprogrammering (C3, D3)	Johan Fagerström
Operativsystem (D4, Y4, I4)	Ola Strömfors
Konstruktion och analys av algoritmer (C4)	Rolf Karlsson
Programmering Y, grundkurs (Y2)	Anders Räntilä
Programmerung Y, fortsättningskurs (Y3, Y4)	Tommy Olsson
Kompilatorer och interpretatorer (Y4)	Nahid Shahmehri
Programmeringsteori II (C4)	Staffan Bonnier
Logikprogrammering (C3, D4)	Ulf Nilsson
Programmeringsteori (C3)	Nahid Shahmehri
Realtids- och processprogrammering (M4, Y4, I2)	Erik Tengvald
Programmering och projektarbete i Pascal (C1, D2)	Kerstin Olsson
Lagringssstrukturer (C2, D2, Md3)	Sven Moen
Programutvecklingsmetodik M (Md3)	Olle Willen
Programutvecklingsmetodik och programmeringsprojekt C (C3)	Christian Krysander
Data och programstrukturer C (C2)	Patrick Doherty
Expertsystem - metodik och verktyg (C4, D4)	Sture Hägglund
Distribuerad problemlösning (D4, C4)	Johan Fagerström
Datorer och datorutrustning (I1)	Inge Wallin
Beräkningsbarhet och komplexitet (C4)	Rolf Karlsson
Datorspråk (C3, D3, D4, I4)	Rober Bilos
Artificiell intelligens C (C3)	Arne Jönsson
Artificiell intelligens D (D4)	Jalal Maleki
Programutveckling (I1)	Lotta Månsbacka
Datastrukturer och programutvecklingsmetodik (I2)	Mariam Kamkar
Datalingvistik (C2)	Lars Ahrenberg
Datorsystem och programmering (M2)	Lotta Månsbacka
Programmering i inkrementellt system (C1)	Anders Haraldsson
Programmering i inkrementellt system (D1)	Anders Haraldsson
Lingvistik grundkurs (C1)	Magnus Merkel
Formella språk och automatateori (C2)	Torbjörn Näslund
Kommunikationspsykologi (C3)	Nils Dahlbäck
Datornät (D4, Y4, I4, Md4)	Björn Fjellborg
Datorarkitektur (D4, C4, Y4)	Krzysztof Kuchcinski
Datorstödd elektronikkonstruktion (D4, Y4, I4)	Tony Larsson
Datalogi 1 - baskurs (enstaka kurs Norrköping)	Katarina Sunnerud
Datalogi 2 - programmeringsprinciper (enstaka kurs Norrköping)	Tommy Olsson

Continuing education 1988/89

Datorsystem och programutveckling, 15p, LM Ericsson, 11 deltagare
(ingår i Dator- och teleteknik, 60p) ht 1988
Processprogrammering/operativsystem, 5p, Ellemtel, 20 deltagare, vt 1989
AI/Expertsystem, 28p, ASEA, 20 deltagare, började hösten 1987
Programutveckling, 2p, LM Ericsson, 19 deltagare, (ingår i Dator- och teleteknik, 60p) ht 1988
Programutveckling, 5p, LM Ericsson, 19 deltagare (ingår i Dator- och teleteknik,60p) vt 1989
Kurs i Ada, BT Electronics, 15 deltagare, augusti 1988
Kurs i Ada, Saab, Missiles, 8 deltagare i november 1989 och 10 deltagare i mars 1989
Processprogrammering, 2.5p, Ellemtel, 18 deltagare, våren 1988
AI/Expertsystem, 25p, Volvo, 18 deltagare, började våren 1989
Programvaruproduktion, 9p, LM Ericsson, 14 deltagare, vt 1988-ht 1988.

2. Undergraduate teaching in the School of Arts and Science

The group for administrative data processing (the ADB-group) is responsible for the courses given by IDA in the undergraduate *Systems analysis* study program in the School of Arts and Science, Linköping University.

The program for systems analysis ranges over three years of fulltime 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 program constitute a common core of basic studies for all students. Third year specializations:

- Methods for data analysis (data analysis).
- Development of computer programs and program systems (program development).
- Development of information systems (systemeering).

Personnel
Göran Goldkuhl, PhD, senior lecturer
Lise-Lotte Raunio, MSc, lecturer, director of undergraduate studies
Carina Björkman, secretary
Carita Lilja, secretary
Siv Söderlund, secretary

Pia Arendell, BSc, lecturer
Åke Engquist, undergraduate assistant
Hans-Olov Ganning, undergraduate assistant
Hans Holmgren, MScEng, lecturer
Mikael Johansson, lecturer
Anders Knutsson, undergraduate assistant
Dan Michaeli, undergraduate assistant
Rolf Nilsson, BSc, lecturer
Torbjörn Näslund, postgraduate assistant
Annie Röstlinger, BSc , lecturer
Eva-Chris Svensson, MSc, lecturer
Kjell Westerlund, lecturer
Per Övernäs, BSc, lecturer

Listing of Undergraduate Course Program 1988/89

Course (in Swedish)	Teacher
Systemvetenskapliga linjen	
- Systemvetenskaplig grundkurs	Pia Arendell
- Utveckling och förändring av informationssystem	Göran Goldkuhl
- Inledande programmering och datateknik:	
Pascal	Rolf Nilsson
Cobol	Kjell Westerlund
- Systemutvecklingsprojekt:	
Systemering	Pia Arendell
Programmering	Kjell Westerlund
- Formella metoder för systemering, teori och tillämpning:	
Systemering	Eva-Chris Svensson
Programmering	Torbjörn Näslund
- Lagringsstrukturer	Mariam Kamkar
- Datorteknik	Katarina Sunnerud
- Arbetsvetenskap	Gunilla Hagstedt
- Beslutsteoretiska aspekter på informationssystem	Eva-Chris Svensson
- Praktik	Lise-Lotte Raunio
- Vetenskapsteori	Göran Goldkuhl
- Datajuridik	Georg Hellgren
- Programutvecklingsmetodik	Olle Willen
- Utvärdering av datasystem	LiseLotte Raunio
- Interaktiva system och databaser	Hans Holmgren
- Ekon/Samhällsvetenskaplig utvärdering av informationssystem	Annie Röstlinger
- Systemprogramvara:	
Kompilatorer/Interpretatorer	Nahid Shahmehri
Programmering i inkr system	Anders Haraldsson
Operativsystem/Datornät	Björn Fjellborg
- Avslutande projektarbete:	
Systemering	Eva-Chris Svensson
Programmering	Hans Holmgren

ADB, fristående kurser, Linköping

- Programmering, grundkurs	Rolf Nilsson
- Systemering, grundkurs	Mikael Johansson
- Programmering i administrativ miljö	Kjell Westerlund
- Datorisering av administration	Lise-Lotte Raunio
- Systemutvecklingsprojekt	Pia Arendell
- Databehandlingsteknik	Katarina Sunnerud
- Datalagring	Mariam Kamkar, Hans Holmgren
- Förändringsanalys och Projektadm	Annie Röstlinger
- Experimentell systemutveckling	Pia Arendell

ADB, fristående kurser, Norrköping

- Programmering, grundkurs	Kjell Westerlund
- Systemering, grundkurs	Mikael Johansson
- Systemutvecklingsprojekt	Pia Arendell

Förvaltningslinjen

- Administrativt utvecklingsarbete	Lise-Lotte Raunio

Statistikerlinjen

- Systemutvecklingsmetodik	Rolf Nilsson
- Databasteknik	Rolf Nilsson
- 4G-språk	Rolf Nilsson

C-linjen

- Systemutveckling, teori och tillämpning	Pia Arendell

Back row: Hans Holmgren, Eva-Chris Svensson, Mikael Johansson and Siv Söderlund. Front row: Rolf Nilsson, Torbjörn Näslund, Carita Lilja, Annie Röstlinger and Lise-Lotte Raunio.

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 24 XEROX 1186 Lisp Machines with file servers and laser printers, 23 SUN-3 workstations with file servers, two DECsystem-2060 and a VAX 780 (which is shared with the Physics department).

Already in 1986, the department was able to create a network of 32 Xerox AI workstations. The major part of this equipment was made available through the Xerox Corporation / Rank Xerox University Grants Programme, where the Linköping grant was the largest awarded in Europe and represented at that time one of the largest installments of AI equipment in Europe.

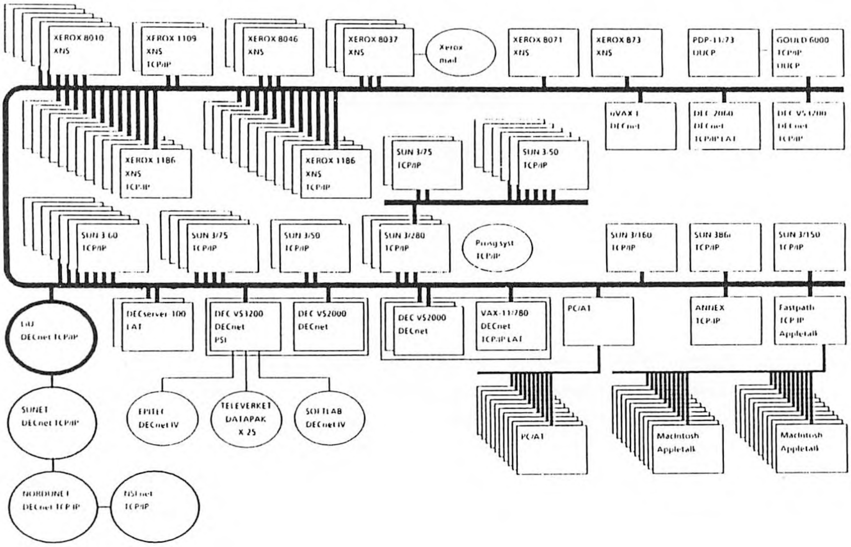
During the coming year the two DECsystem-2060 and the VAX 780 will be taken out of operation and 155 SUN Sparcstations and 7 SUN fileservers will be installed, together with some additional VaxStations etc. This will complete the transition from timesharing to networked workstations as the preferred computing paradigm in the department.

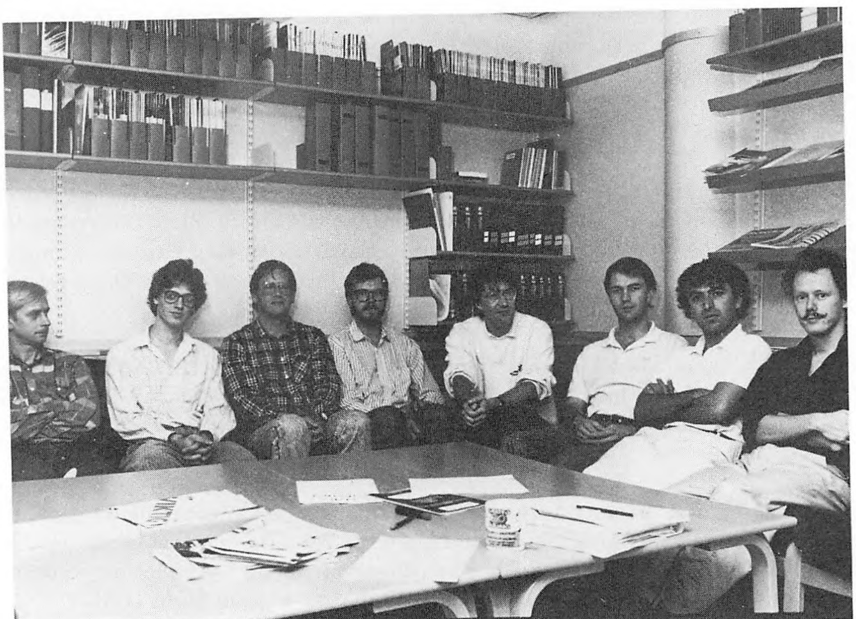
In addition there are lots of smaller computers (MicroVax, PDP-11s, XEROX Viewpoint systems, Macintoshes and other PC:s of various kinds.) There is also special purpose equipment, especially for text processing or for specific research projects.

For research on architecture-related problems, the department also has acquired a number of transputers and a shared NCUBE parallel computer. We also share an IBM 9370 with the department for production economy (economic information systems), made available by a grant from IBM.

The schematic picture on the next page shows the local network and the accessible computer systems.

Network at IDA LiU 890712





Technical services staff: Rolf Nilsson, Göran Sedvall, Björn Nilsson, Leif Finmo, Bernt Nilsson, Dimitrios Fotiadis and Peter J. Nilsson.

Appendix E
Publications

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.

(*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).

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.**

EXTERNAL PUBLICATIONS SINCE 1986

(*Papers published in books, journals or international conference proceedings.*)

1. **Lars Ahrenberg:** Lexikalisk-Funktionell Grammatik på svenska. In *Papers from the 5th Scandinavian Conf. of Computational Linguistics,* University of Helsinki, Dept of General Linguistics pp. 1-12, 1986.
2. **Lars Ahrenberg:** Parsing into Discourse Object Description. In *Proc. of the 3rd Conf. of the European Chapter of the Association for Computational Linguistics* in Copenhagen, April 1-3, 1987.
3. **Lars Ahrenberg:** The Formalization of Field Grammars. In Proc. of the *7th Int. Conf. of Nordic and General Linguistics.* Thorshavn, 7-11 August, 1989..le;**Lars Ahrenberg, Arne Jönsson:** An Interactive System for Tagging Dialogues. *XIV ALLC Conf.* in Göteborg, June 1-5, 1987.
4. **Lars Ahrenberg:** Functional Constraints in Knowledge-Based Natural Language Understanding. In the 12th International Conf. on *Computational Linguistics* in Budapest, August 22-26, 1988.
5. **Lars Ahrenberg:** On the Intergration 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.
6. **Elisabeth Arborelius, Toomas 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, Amstedam, 1989.
7. **Elisabeth Arborelius, Toomas 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,* in press, 1989.
8. **Rober Bilos:** A Token-Based Syntax Sensitive Editor. In *Proc of the Workshop on Programming Environments - Programming Paradigms.* Roskilde, 1986.
9. **Rober Bilos, Peter 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.
10. **Lisbeth 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.
11. **Lisbeth Björklund, Birgitta Olander:** The Mystery Cat - The Modern Magician. Accepted for *7. Nordiske IoD,* Denmark, August 28 - 30, 1989.
12. **Lisbeth Björklund, Birgitta Olander, Linda Smith:** The Personal HYPERCATalog. Accepted for *1989 Annual Meeting of the American Society for Information Science,* Washington DC, October, 1989.

13. **Hans Block:**, SPORT-SORT - Sorting Algorithms and Sport Tournaments. In *Proc. of the 25th Annual Allerton Conf. on Communication, Control, and Computing,* University of Illinois at Urbana-Champaign, 1987.
14. **Staffan Bonnier, Jan 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. Also in *Proc. of PLILP'88,* LNCS 348, Springer-Verlag, 1989.
15. **Christer Bäckström:** A Representation of Coordinated Actions. *Proc. of the 1st Scandinavian Conf. on Artificial Intelligence,* Tromsö, Norway, March 9-11, 1988.
16. **Christer 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, 1989.
17. **Christer Bäckström:** A Representation of Coordinated Actions Characterized by Interval Valued Conditions. In *Proc. of ISMIS'88,* Torino, Italy, October 12-15, 1988.
18. **Christer Bäckström:** Action Structures with Implicit Coordination: In *Proc. AIMSA'88,* Varna, Bulgaria, September 20-23, 1988.
19. **Christer Bäckström:** A Representation of Interacting and Interferring Actions: In *Proc. SAIS-88 Workshop,* Umeå, Sweden, May 24-25, 1988.
20. **Shamsul Chowdhury:** Expert System Aid in Statistical Analysis and Interpretation of Data. In *Proc. of the Society of Reliability Engineers,* Outaniemi, 1986.
21. **Shamsul Chowdhury:** State of the art in statistical expert systems. In *Proc. of the Conf. on Expert Systems and their Applications,* Avignon, 1987.
22. **Shamsul Chowdhury, Ove Wigertz:** Microcomputer Oriented Knowledge-Based System for Health Care Improvement in the Developing World. In *Proc. of the IEEE/EMBS 9th Ann. Conf.,* Boston, 1987.
23. **Nils Dahlbäck, Arne Jönsson:** Analyzing Human-Computer Dialogues in Natural Language. Accepted to the *3rd IFAC/IEA Conf. on Man-Machine Systems Analysis, Design and Evaluation,* Oulo, Finland, June, 1988.
24. **Nils Dahlbäck, Arne Jönsson:** Talking to a Computer is not Like Talking to Your Best Friend. *Proc. of the 1st Scandinavian Conf. on Artificial Intelligence,* Tromsö, Norway, March 9-11, 1988.
25. **Nils Dahlbäck:** Mental Models and Text Cherence. In *Proc. of the Nordic Conf. on Text Comprehension in Man and Machine,* Stockholm, October, 1988.
26. **Nils Dahlbäck, Arne Jönsson:** Emirical Studies of Discourse Representations for Natural Language Interfaces. Accepted to the *4th Conf. of the European Chapter of the Association for Computational Linguistics,* Manchester, April 10-12, 1989.
27. **Nils Dahlbäck, Arne Jönsson:** Analyzing Human-Computer Dialogues in Natural Language. In Ranta (ed.) *Analysis, Design and Evaluation of Man-Machine Systems,* Pergamon Press, Oxford, 1988.
28. **James A Dean, Andrzej Lingas, Jörg Sack:** Recognizing Polygons or How to Spy, in *The Visual Computer, International Journal of Computer Graphics,* Springer Verlag. See *Proc. of the Allerton Conf. on Communication, Control and Computing,* Urbana, Illinois, 1986, for a preliminary version.
29. **Pierre Deransart, Jan Maluszynski:** Programmation logique et grammaires d'attributs, in: *Informatique-85, Symposium sovieto-francaise,* Valgus, Tallinn, 1987, 4-10.
30. **Pierre Deransart, Jan Maluszynski:** A Grammatical View of Logic Programming. In *Proc. of PLILP'88,* LNCS 348, Springer-Verlag. Orleans, France, May 16-18, 1989.
31. **Hristo Djidjev, Andrzej Lingas, Jörg 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.
32. **Patrick Doherty:** A Semantics for Inheritance Hierachies with Exceptions using a Logic of Preferential Entailment. In *Proc. SCAI'89,* Tampere, 1989.
33. **Patrick Doherty:** A Correspondence between Inheritance Hierarchies and a Logic of Preferential Entailment. In *Ras, editor, Methodologies for Intelligent Systems,* 1989

34. **Wlodzimierz Drabent:** Do Logic Programs Resemble Programs in Conventional Languages? In *Proc. of the 4th IEEE Symposium on Logic Programming,* San Francisco, Aug. 31 - Sept.4, 1987
35. **Wlodzimierz Drabent, Jan Maluszynski:** Proving Runtime Properties of Logic Programs. In *Proc of the TAPSOFT'87,* Pisa 1987, LNCS 250, 167-181.
36. **Wlodzimierz Drabent, Jan Maluszynski:** Inductive Assertion Method for Logic Programs. Accepted for publication in the special issue of *Theoretical Computer Science,* 1988.
37. **Wlodek Drabent, Simin Nadjm-Tehrani, Jan Maluszynski:** Algorithmic Debugging with Assertions. In *Proc. of META 88 Workshop on Meta Programming in Logic Programming,* Bristol, June 21-24, 1988. To appear in LNC-series, Springer-Verlag.
38. **Wlodek Drabent, Simin Nadjm-Tehrani, Jan Maluszynski:** The Use of Assertions in Algorithmic Debugging. In *Proc. of the Int. Conf. on Fifth Generation Computer Systems,* Tokyo, 1988.
39. **Wlodek Drabent, Jan Maluszynski:** Inductive Assertion Method for Logic Programs. In *Theoretical Computer Science 59,* North-Holland, pp 133-155, 1988.
40. **Dimiter Driankov:** Inference with single fuzzy conditional proposition. *Int. Journal of Fuzzy Sets and Systems* vol 24, No 1 pp. 51-63.
41. **Dimiter 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.
42. **Dimiter Driankov:** Many-valued logic for belief-intervals: The logical lattice. In *Proc. of the 2nd World Congress of the Int. Fuzzy Sets Association,* pp. 426-429, Tokyo, July 20-25, 1987.
43. **Dimiter Driankov:** Uncertainty Calculus with Verbally Defined Belief Intervals. In *Proc. Int. Joint Conf. of Intelligent Systems,* 1987.
44. **Dimiter Driankov:** A Many-Valued Logic for Belief/Disbelief Pairs: In *Proc. of the 2nd Int. Symp. on Methodologies for Intelligent Sytems,* Charlotte, NC, USA, October, 1987. (Extended abstract).
45. **Dimiter Driankov:** Inference with consistent probabilities in expert systems. In *Proc. Int. Joint Conf. on A.I,* pp 899-901 vol 2, Milano, August,1987. To appear in *Int. Journal of Intelligent Systems,* 1988.
46. **Dimiter Driankov:** Non-truthfunctional Aspects of Belief/Disbelief Pairs. To appear in *Proc of the 1st IFSA-EURO Workshop on Approximate Reasoning,* April 5-9, 1988.
47. **Dimiter 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.
48. **Dimiter 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.
49. **Dimiter Driankov:** Towards a Many-Valued Logic of Quantified Belief: The information lattice. Presented at the *3rd IFSA Congress,* Seattle, Washington, USA, August 6-11, 1989.
50. **Dimiter 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.
51. **Dimiter Driankov:** Non-truthfunctional Quantified Beliefs. In *Proc. of the 3rd World Congress on Fuzzy Systems.* Seattle, August 6-11, 1989.
52. **Peter Eklund, J Kellett:** Prospects for Conceptual Graphs in Engineering Interfaces. In *Proc. of the Industrial Engineering Applications of AI and ES,* Tullahoma, Tenessee, June, 1989.
53. **Peter Eklund:** Using Path Algebras and Conceptual Graphs to Encode Knowledge Bases. In *Proc. of the 2nd Scandinavian Conf. SCAI,* Tampere, June, 1989.
54. **Peter Eklund:** Prospects for Conceptual Graphs in Knowledge Acquisition Interfaces. In *Proc. of the 3rd European Knowledge Aquisition Workshop,* Paris,

June 3-7, 1989.

55. **Henrik Eriksson, Kristian Sandahl:** Knowledge-based Planning of Experiments in a Biochemical Domain - Membrane-Protein Purification. In *Proc. of the 2nd Scandinavian Conf. on AI,* 1989.
56. **Johan Fagerström:** Experiences with Occam: A Simulator for Asynchronous Processes. In *Proc of the 19th Hawaii Int. Conf. on System Sciences, Hawaii,* Jan, 1986, pp. 95-102.
57. **Johan Fagerström:** Tradeoffs in an Architecture based on Asynchronous Processes. In *Proc of the 2nd Nordic Symposium on VLSI in Computers and Communications,* 1986.
58. **Johan Fagerström, Mikael R.K. Patel:** High-level Simulation of Systolic Architectures. In *Proc of the Int. Workshop on Systolic Arrays,* Oxford, 2-4 July, 1986.
59. **Johan Fagerström, Yngve Larsson and Lars Strömberg:** Debugging Techniques for Distributed Environments. In *Proc. of the Workshop on Compiler and Incremental Compilation* in Bautzen, East Germany, October 11-18, 1986 and the *Proc. of the Workshop on Programming Paradigms and Programming Environments* in Roskilde, Denmark, October 22-24, 1986.
60. **Johan Fagerström, Lars Strömberg:** A Paradigm and System for Design and Test of Distributed Applications. In *Proc. of IEEE Compcon Spring 1988.* San Fransisco, CA, February, 1988.
61. **Johan Fagerström:** Design and test of Distributed Applications. *10th Int. Conf. on Software Engineering,* Raffles City, Singapore, April 11-15, 1988.
62. **Johan Fagerström:** Enabling Structured Debugging of Distributed Sytems. In the *Proc. of the Workshop on Parallel and Distributed Debugging.* Madison, WI, May, 1988.
63. **Johan Fagerström, Yngve Larsson:** Two Contributions on Debugging Distributed Systems. Also in *Proc. of ACM SIGPLAN/SIGOPS, Workshop on Parallel and Distributed Debugging,* Madison, Wisconsin, May 5-6, 1988.
64. **Björn 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.
65. **Björn Fjellborg, Krzysztof Kuchcinski, Mikael Patel, Zebo 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.
66. **Per-Olof Fjällström, Jyrki Katajainen, Christos Levcopoulos, Ola Petersson:** A Sublogarithmic Parallel Algorithm for Finding the Convex Hull of Sorted Points, submitted for publication.
67. **Hartmut Freitag, Michael Reinfrank:** An Efficient Interpreter for a Rule-Based Non-Monotonic Deduction System. In *Proc. of the German Workshop on AI, GWAI-87,* Springer Verlag, 1987.
68. **Peter Fritzson:** Systems and Tools for Exploratory Programming. Overview and Examples. In *Proc. of the Workshop on Programming Environments - Programming Paradigms,* Roskiide University Centre, Denmark, October 22-24, 1986.
69. **Peter Fritzson:** A Common Intermediate Representation for C, Pascal, Modula-2 and Fortran-77. In *Proc. of the Workshop on Compiler Compilers and Incremental Compilation,* Bautzen, DDR, October 12-17, 1986.
70. **Peter 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.
71. **Sture Hägglund, Christer Hansson, Tomas Sokolnicki:** Knowledge-Based Training of Case Management Routines and Emergency Procedures. In *Proc. of the 3rd Int. Conf. on Expert Systems,* London, 1987.
72. **Sture Hägglund:** Emerging Systems for Computer-Based Knowledge Processing in Office Work. In *Office: Technology and People, 4,* pp 135-145, 1989.
73. **Sture Hägglund:** The Linköping Approach to Technology Transfer in Knowledge Engineering. To appear in *The Knowledge Engineering Review,* vol 2, No 3, Cambridge University Press, 1988.

74. **Sture 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.
75. **Sture 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.*
76. **Sture Hägglund, Ivan Rankin:** Investigating the Usability of Expert Critiquing in Knowledge-Based Consultation Systems. In *Proc. of the ECAI Conf.* Muenchen, July, 1988.
77. **Tim 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.
78. **P Hedblom, S Wallin, G Tibblin, Toomas Timpka:** Hypertext med ny databasteknik hjälper allmänläkaren. Sammanfattningar. Svenska Läkarsällskapets Riksstämma 1988, Hygia, 97:225, 1988.
79. **Shu Hua:** An Expert System Mechanical Engineering Design System. In *Proc. of the FEMCAD-88 Conf.* Paris, 1988.
80. **Roland Hjerppe:** Project HYPERCATalog: Visions and preliminary conceptions of an extended and enhanced catalog. Published in *Intelligent Information Systems for the Information Society,* B C Brookes. Ed. In *Proc. of the IRFIS 6th Conf.* (International Research Forum in Information Science), Frascati, Italy, 15-18 Sept, 1985. Elsevier Science Publishers B.V. (North-Holland), 1986, pp. 211-232.
81. **Roland Hjerppe:** Electronic Publishing: Writing Machines and Machine Writings. The impact of computers on text. Published in *Annual Review of Information Science and Technology,* vol. 21, 1986, M Williams. Ed. Knowledge Industry Publications Inc, pp. 123-166.
82. **Roland Hjerppe:** Knowledge Organizing, Collection Derived, and User Established Structures. Published in Online Public Access to Library Files: *2nd National Conf.* J Kinsella Ed. Elsevier International Bulletins, Oxford 1986, pp. 101-110.
83. **Roland Hjerppe:** LINS - LIBLAB's Name Handling System. A knowledge-based system for authority control of personal names according to AACR2, Ch.22, Headings for persons. June, 1987, p.8. I: C. Bossmeyer, ED. The Library of the Future. In *Proc. of the ELAG (European Library Automation Group),* 11th Library Systems Seminar, Frankfurt, April 1-3, 1987. Deutsche Bibliothek, Frankfurt am Main. 1987. ISBN:3-922051-19-7. pp.67-80.
84. **Roland Hjerppe:** Computer Networks as a Publication Medium? Implications for Libraries. August, 1987, p 4. Presented at *53rd IFLA General Conf.* ,Brighton, England, August 16-21, 1987.
85. **Roland Hjerppe:** IFLA and Professional Communication. Presented at the *IFLA 53rd General Conf.* ,Brighton, England, August 16-21, 1987.
86. **Johan Hultman:** COPPS A Software System for Defining and Controlling Actions in a Mechanical System. In *Proc. of the IEEE Workshop on Languages for Automation*, Wien, September, 1987.
87. **Johan Hultman, Anders Nyberg, Mikael Svensson:** A Software architecture for autonomuons systems. In Proc. of the *6th Int. Symposium on unmanned, untethered submensible Technology.* Ellicott City, Maryland, June 12-14, 1989.
88. **Arne Jönsson, Nils Dahlbäck:** Talking to a Computer is not like Talking to Your Best Friend. In *Proc. of the SCAI,* Tromsö, Norway, 1988.
89. **Arne Jönsson, Lars Ahrenberg:** Extensions of a Descriptor-based Tagging System into a Tool for the Generation of Unification-Based Grammars.To appear in the *16th Int. ALLC Conf. and the 9th ICCH Conf.* Canada, June 6-10, 1989.
90. **Mariam Kamkar, Nahid Shahmehri, Peter 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, 1988.
91. **Mariam Kamkar, Nahid Shahmehri:** Runtime Dependent Program Flow Analysis. In *Proc. of the Workshop on Programming Environments - Programming Paradigms,* at Roskilde University Centre, Denmark, October 22-24, 1986.

92. **Rolf G. Karlsson:** Point Location in Discrete Computational Geometry. In *Proc. of the 6th Brazilian Congress on Computing,* 1986, pp. 561-569.
93. **Rolf G. Karlsson:** Greedy matching on a grid. In *BIT,* 1988.
94. **Rolf G. Karlsson, Mark Overmars:** Normalized Divide-and-Conquer: A Scaling Technique for Solving Multi-dimensional Problems. In *Information Processing Letters,* 26, pp. 307-312, 1988.
95. **Rolf G Karlsson, Mark Overmars:** Scanline Algorithms on a Grid. In *BIT,* 1988.
96. **Jyrki Katajainen, Christos Levcopoulos, Ola Petersson:** Local Insertion Sort Revisited. In *Proc. of the Int. Symposium on Optimal Algorithms,* Varna, Burlgaria, LNCS, Springer-Verlag, 1989.
97. **Jan Komorowski, Jan Maluszynski:** Logic Programming and Rapid Prototyping. Report TR-01-86, Harward University, Aiken Computation Laboratory. In *Science of Computer Programming,* 9, 1987, pp. 179-205.
98. **Krzysztof Kuchcinski and Zebo Peng:** Microprogramming Implementation of Timed Petri Nets. *In Proc. of the 2nd Nordic Symp. on VLSI in Computers and Communications,* Linköping, Sweden, June, 1986.
99. **Krzysztof Kuchcinski, Zebo Peng:** Parallelism Extraction from Sequential Programs for VLSI Applications. Presented at *Euromicro 87,* Southsea-Portsmouth, 1987.
100. **Krzysztof Kuchcinski, Zebo Peng:** Microprogramming implementation of timed Petri nets. In *North-Holland INTEGRATION, the VLSI Journal,* No 5, 1987.
101. **Krzysztof Kuchcinski, B. Wiszniewski:** Path Analysis of Distributed Programs. In *Proc. of the 16th ACM Computer Science Conf.*, Atlanta, Georgia, February, 23-25 1988.
102. **Krzysztof Kuchcinski, Zebo Peng:** Parallelism Extraction from Sequential Programs for VLSI Applications. In *Euromicro Journal, Microprocessing and Microprogramming, Vol.23, Nos 1-5,* pp 87-92, 1988.
103. **Krzysztof Kuchcinski, Zebo 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.
104. **P Kuczora, Peter Eklund:** A Conceptual Graph Implementation for Knowledge Engineering. In *Proc. of the Human and Organisational Issues in Expert Systems* (edited by Anna Hart and Dianne Berry), *Clarondon Press,* April, 1989.
105. **Tony Larsson:** Semantics of a Hardware Specification Language and Related Transformation Rules. *In Proc. of the 2nd Nordic Symp. on VLSI in Computers and Communications,* Linköping, Sweden, June, 1986.
106. **Tony Larsson:** Semantics of a Hardware Specification Language. *Microprocessing and Microprogramming,* vol.18, No 1-5, pp. 637-643, 1986.
107. **Tony Larsson:** Specification and Verification of VLSI Sytems Actional Behaviour - *8th International Symposium on Computer Hardware, Description Languages and their Applications,* Amsterdam, April 27-29, 1987.
108. **Tony Larsson:** Semantics of a Hardware Specification Language and Related Transformation Rules. In *North-Holland INTEGRATION, the VLSI Journal,* No 5, 1987.
109. **Tony Larsson:** Hardware Verification based on Algebraic Manipulation and Partial Evaluation. Also in *Proc of IFIP WG 10.2 International Working Conference on The Fusion of Hardware Design and Verification,* Scotland, July 3-6, 1988.
110. **Tony Larsson, Zebo 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.
111. **Yngve 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.
112. **Bengt Lennartsson:** Programming Environments and Paradigms - Some Reflections. In *Proc. of the Workshop on Programming Environments - Programming Paradigms,* Roskilde, Denmark, October 1986.
113. **Bengt Lennartsson:** Multi-Level Software Architectures - Industrial Experiences and Research Directions. This report is a revised version of a paper in *Advance*

Working Papers of The Second Int. Workshop on Computer-Aided Software Engineering. Cambridge, Massachusetts, USA, July 12-15, 1988.

114. **Jacek Leszczylowski, Staffan Bonnier, Jan Malsuzynski:** Logic Programming with External Procedures: Introducing S-Unification. A shorter version is to appear in *Information Processing Letters,* North Holland.
115. **Christos Levcopoulos:** Fast Heuristics for Minimum Length Rectangular Partitions of Polygons. In *Proc of the 2nd ACM Symposium in Computational Geometry,* Yorktown Heights, June, 1986.
116. **Christos Levcopoulos:** An Omega (square root(n)) Lower Bound for the Non-Optimality of the Greedy Triangulation. In *Information Processing Letters,* No 25 (1987), pp. 247-251.
117. **Christos Levcopoulos, Andrzej Lingas:** On the Approximation Behavior of the Greedy Triangulation for Convex Polygons. In *Algorithmica,* No 2, 1987, pp. 175-193.
118. **Christos Levcopoulos:** Improved Bounds for Covering General Polygons with Rectangles. In *Proc of the 7th Conf. on Foundations of Software Technology and Theoretical Computer Science,* Pune, INDIA, December 17-19, 1987. *Lecture Notes 287, Springer Verlag.*
119. **Christos Levcopoulos, Andrzej Lingas och Jörg Sack:** Nearly Optimal Heuristics for Binary Search Trees with Geometric Generalizations. In *Proc of the ICALP'87* Karlsruhe, West Germany, July, 1987. *Springer Verlag Lecture Notes 267.*
120. **Christos Levcopoulos, Andrzej Lingas, J. Sack:** Heuristics for Optimal Search Trees with Geometric Generalizations. Accepted for publication in *Theoretical Computer Science,* November, 1988. (A preliminary version in Proc. *of 14th Colloquium on Automata, Languages and Programming, Karlsruhe, Lecture Notes in Computer Science 267, Springer - Verlag).*
121. **Christos Levcopoulos, Mark H Overmars:** A Balanced Search Tree with 0(1) Worst-Case Update Time. To appear in *Acta Informatica 26,* 1988.
122. **Christos Levcopoulos, Ola 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.
123. **Christos Levcopoulos, Ola Petersson:** Heapsort-Adapted for Presorted Files. In *Proc. of 1989 workshop on Data Structures and Algorithms,* Ottawa, August 17-19, 1989, LNC, Springer-Verlag.
124. **Christos Levcopoulos, Jyrki Katajainen and Andrzej Lingas:** An Optimal Expected-Time Parallel Algorithm for Voronoi Diagrams. To appear in the *1st Scandinavian Workshop on Algorithm Theory,* Halmstad, July, 1988.
125. **Christos Levcopoulos, Andrzej Lingas:** There are planar graphs almost as good as the complete graphs and as short as minimum spanning trees. Accepted to the *Int. Symposium on Optimal Algorithms* Varna, Bulgarien, May 29 - June 2, 1989.
126. **Andrzej Lingas:** Subgraph Isomorphism for Biconnected Outerplanar Graphs in Cubic Time. In *Proc. of the 3rd Symposium on Theoretical Aspects of Computer Science,* January, 1986, Orsay, France. *Lecture Notes in Computer Science, vol 210, Springer Verlag.* To appear in *Theoretical Computer Science.*
127. **Andrzej Lingas:** On Approximation Behavior and Implementation of the Greedy Triangulation for Convex Planar Point Sets. In *Proc of the 2nd ACM Symposium in Computational Geometry,* Yorktown Heights, June, 1986.
128. **Andrzej Lingas, Andrzej Proskurowski:** Fast Parallel Algorithms for the Subgraph Homeomorphism and the Subgraph Isomorphism Problems for Classes of Planar Graphs. In *Proc. of the 7th Conf. on Foundations of Software Technology and Theoretical Computer Science,* Pune, INDIA, December 17-19, 1987, *Springer Verlag Lecture Notes 287.*
129. **Andrzej Lingas, Christos Levcopoulos, Jörg Sack:** Algorithms for Minimum Length Partitions of Polygons of Polygons. In *BIT,* Vol 27, No 4, pp. 474-479.
130. **Andrzej Lingas:** A New Heuristic for Minimum Weight Triangulation. In *SIAM Journal of Discrete and Algebraic Methods.* Vol 8, No 4, pp. 646-653. (This is an improved version of LiTH-IDA-R-84-15).
131. **Andrzej Lingas:** A Space Efficient Algorithm for Greedy Triangulation.In *Proc. of*

the 13th IFIP Conf. on System Modelling and Optimization, Tokyo, 1987, Springer Verlag Lecture Notes.

132. **Andrzej Lingas, Maciej Syslo:** A Polynomial-time algorithm for subgraph isomorphism of two-connected series-parallel graphs. To appear in *Proc. of the ICALP,* Tampere, July, 1988. Lecture Notes in Computer Science, Springer Verlag.
133. **Andrzej Lingas, M. Karpinski:** Subtree Isomorphism is NC reducible to Bipartite Perfect Matching. Accepted for publication in *Information Processing Letters,* May, 1988.
134. **Andrzej Lingas:** Subgraph Isomorphism for Two-connected Series-parallel Graphs. Accepted to *the ICALP'88, Conf.* 1988.
135. **Andrzej 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, August, 1988.
136. **Andrzej 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.
137. **Jonas Löwgren:** History, State and Future of User Interface Management Systems. In *SIGCHI Bulletin, vol 20, No 1,* 1988.
138. **Jonas Löwgren:** An Architecture for Expert System User Interface Design and Management. Accepted to the *Symposium On User Interface Software and Technologies.* Williamsburg, Virginia, November 13-15, 1989.
139. **Jan Maluszynski, Torbjörn Näslund:** Fail Substitutions for Negation as Failure. In *Proc. of North American Conf. on Logic Programming,* Cleveland, Ohio, October 16-20, 1989.
140. **Jalal Maleki:** VIVID. The Kernel of a Knowledge Representation Environment Based on the Constraints Paradigm of Computation. In *Proc. of the 20th Hawaii Int. Conf. on System Sciences*, Kailua-Kona, 1987.
141. **Jalal Maleki:** Towards a system for temporal reasoning. In Proc. of the *Scand Conf. on A.I,* Tampere, Finland. Also LAIC-IDA-89-TR5.
142. **Magnus Merkel:** The Interpretation of Swedish Temporal Frame-Adverbials.In *Proc. of the 10th Scandinavian Conf. on Linguistics*, Bergen, June 11-13, 1987.
143. **Magnus Merkel: Temporal Structure in Discourse. In Proc. of the 7th Int. Conf. of Nordic and General Linguistics.** Thorshavn, August 7-11, 1989.
144. **Magnus Merkel:** A Novel Analysis of Temporal Frame Adverbials. In *Proc. of the 12th International Conf. on Computational Linguistics* in Budapest, August 22-26, 1988.
145. **Sven Moen:** Drawing Dynamic Trees. Accepted for publication in *IEEE Software.*
146. **Torbjörn Näslund:** An Experimental Implementation of a Compiler for Two-Level Grammars. In: Z.W Ras and M. Zemankova (eds.). In *Proc. of the 2nd Int. Symposium, Methodologies for Intelligent Systems*, North-Holland, 1987, pp. 424-431.
147. **Ulf Nilsson:** AID: An Alternative Implementation of DCGs. *New Generation Computing,* vol 4, No 4, pp. 383-399, 1986.
148. **Ulf 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.
149. **Henrik Nordin:** Using Typical Cases for Knowledge-Based Consultation and Teaching. In *Proc of the 3rd Annual Conf. on Applications of Expert Systems,* Orlando, Fla., 1986.
150. **Lin Padgham:** A Description of LINCKS: Linköpings Intelligent Knowledge Communication System. In *Proc. of the IFIP Working Conf. on Methods and Tools for Office Systems*, Pisa, Italy, October 22-24, 1986.
151. **Lin Padgham, Ralph Rönnquist:** From a Technical to a Humane Environment: A Software System Supporting Co-operative Work. In *Proc. of the GDI International Conf. on USER INTERFACES*, Rüschlikon, Switzerland, October 20-21, 1986.
152. **Lin Padgham, Ralph Rönnquist:** An Imperative Object Oriented System. In *Proc. of the 20th Hawaii International Conf. on System Sciences*, 1987, vol 1, p 516.
153. **Lin Padgham:** A Model and Representation for Type Information and its Use in Reasoning with Defaults, to appear in *Proc. AAAI-88*, St. Paul, USA, 1988.

154. **Lin Padgham:** NODE: A Database for Use by Intelligent Systems. Also in *Proc. of ISMIS '88,* Torino, Italy, October 12-15, 1988.
155. **Lin Padgham:** A Model and Representation for Type Information and Its Use in Reasoning with Defaults. Also in *Proc. of AAAI'88, American Association for Artificial Intelligence*, August, 1988.
156. **Lin 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.
157. **Mikael Patel, Arne Jönsson:** An Interactive Flowcharting Technique for Communicating and Realizing Algorithms. In *Proc of the 19th Annual Hawaii Int. Conf. on System Sciences, HICSS-19,* 1986.
158. **Mikael Patel:** A Threaded Interpretive Language Supporting Programming in the Large. In *Proc. of the 6th Rochester Forth Conf,* University of Rochester, Rochester, New York, June 11-14, 1986.
159. **Mikael R.K. Patel:** What should a Programming Environment for Threaded Interpretive Languages provide? Also in *Proc. of the 8th Rochester 4th Conf. on Programming Environments*, Rochester, New York, USA, June, 1988.
160. **Mikael 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.
161. **Mikael R.K Patel:** Selecting Design Representation for High level Synthesis of Hardware Systems. In *Proc. of the 4th Int. Workshop on High-Level Syntheis,* Kennebunkport, Maine, USA, October 15-18, 1989
162. **Zebo Peng and K Kuchcinski:** Synthesis of Control Structures from Petri Net Descriptions, *Microprocessing and Microprogramming,* Vol.18, No 1-5, 1986, pp. 335-340.
163. **Zebo Peng:** Synthesis of VLSI Systems with the CAMAD Design Aid. In *Proc. of the 23rd ACM/IEEE Design Automation Conf,* Las Vegas, June, 1986.
164. **Zebo Peng:** Integration of VLSI Design Tools by a Unified Design Representation. Published as a part of the *Proc of the 2nd Nordic Symposium on VLSI in Computers and Communications,* June 2-4, 1986.
165. **Zebo Peng:** A Formal Approach to the Synthesis of VLSI Systems from their Behavioral Descriptions. In *Proc of the 19th Annual Hawaii Int. Symp. on System Sciences, Hawaii,* January 1986, pp. 160-167.
166. **Zebo Peng:** Construction of Asynchronous Concurrent Systems from their Behavioral Specifications. In *Proc. of the 10th World Computer Congress IFIP-86,* Dublin, Ireland, September 1986, pp. 859-864.
167. **Zebo Peng:** Semantics of a Parallel Computation Model and its Applications in Digital Hardware Design. Also in *Proc. of the 1988 Int. Conf. on Parallel Processing,* Pennsylvania State University, August 15-19, 1988.
168. **Zebo Peng:** Let's Design Asynchronous VLSI Systems. Also published in *Microprocessing and Microprogramming, the Euromicro Journal, Vol. 24, 1988.*
169. **Zebo Peng:** A Horizontal Optimization Algorithm for Data Path/Control Synthesis. Also in *Proc. of the 1988 IEEE Int. Symposium on Circuits and Systems,* Helsinki Univ. of Technology, Finland, June 7-9, 1988.
170. **Zebo Peng, Krzysztof 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
171. **Zebo Peng, Krzysztof Kuchcinski, Bryan 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.
172. **Ivan Rankin:** SMORF - An Implementation of Hellberg's Morphology System. In Papers from the *5th Scandinavian Conf. of Computational Linguistics,* University of Helsinki, Dept of General Linguistics, pp. 161-172, 1986.
173. **Ivan Rankin:** Generating Multi-Paragraph Text in an Expert Critique System. At *the Nordic Conf. on Text Comprehension in Man and Machine,* Stockholm, October,

1988.

174. **Ivan Rankin:** Deep Generation of a Critique. To appear in *the Second European Workshop on Natural Language Generation,* Edinburgh, April, 1989.
175. **Ivan Rankin, Sture Hägglund, Yvonne Waern:** Generating User-Centered Explanations in a Critiquing Context. In *Proc. of the AAAI-88 Workshop on Explanation.* S:t Paul, Minnesota, August, 1988.
176. **Michael Reinfrank, Hartmut Freitag:** An Integrated Non-Monotonic Deduction and Reason Maintenance System. In Herbert Stoyan (ed.) *Proc. of the Workshop on Truth Maintenance*, Berlin, 1986. Springer Verlag.
177. **Michael Reinfrank:** Reason Maintenance Systems. In Herbert Stoyan (ed.) *Proc. of the Workshop on Truth Maintenance*, Berlin, 1986. Springer Verlag.
178. **Michael Reinfrank:** Multiple Extensions, where's the Problem? In F. Brown (ed.) *The Frame Problem in Artificial Intelligence. Proc. of the 1987 Workshop.*
179. **Michael Reinfrank, Oskar Dressler, Gerd 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.
180. **L Reshagen, J-O Hilden, B Lindström, Kristian 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.
181. **Ralph Rönnquist:** The Information Lattice of Networks Used for Knowledge Representation. In *Proc. of the Int. Symp. on Methodologies for Intelligent Systems,* Charlotte, NC, October, 1987.
182. **Ralph Rönnquist:** Reasoning about Control of Dynamic Systems. Accepted to the *ISMIS'89.*
183. **Kevin Ryan:** The Value of Mixed Metaphors in Computer Education. In *Proc. of the Nat. Computer Education Conf.*, San Diego, 1986.
184. **Kevin Ryan, J A Redmond, and Others:** Surveying Software Tools for a Method Driven Environment. *IFIP,* Dublin, September, 1986.
185. **Kristian Sandahl:** The Migration of Expert Systems into Production Environments. In *Proc. of the Nord-Info Seminar on Knowledge Engineering,* Köpenhamn, 1986.
186. **Erik Sandewall, Ralph Rönnquist:** A Representation of Action Structures. In *Proc. of the 5th National Conf. on Artificial Intelligence, AAAI-86*, Philadelphia, 1986.
187. **Erik Sandewall:** Non-Monotonic Inference Rules for Inheritance with Exception. In *Proc. of the IEEE, Special Issue on Knowledge Representation,* 1986.
188. **Erik Sandewall:** Specification Environments for Information Management Systems. Panel position paper in *Proc. of the IFIP Congress,* 1986.
189. **Erik Sandewall:** The Pipelining Transformation on Plans for Manufacturing Cells with Robots. In *Proc. of the Int. Joint Conf. on A.I.*, Milano, August, 1987.
190. **Erik 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.
191. **Erik Sandewall:** An Approach to Non-Monotonic Entailment. Also 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.
192. **Erik Sandewall:** Future Developments in Artificial Intelligence. In *Proc. of European Conf. on Artificial Intelligence (ECAI),* Munich, August, 1988. (Invited paper)
193. **Erik Sandewall:** Artificial Intelligence Techniques for Complex Information Processing in Vehicles. In *Proc. XXXVI Convegno Internazionale delle Comunicazioni,* Genoa, pp. 431-437, 1988.
194. **Erik Sandewall:** Combining Logic and Differential Equations for Describing Real-World Systems. In *Proc. of the Int. Conf. on Knowledge Representation,* Toronto, Canada, 1989.
195. **Erik 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.

196. **Erik Sandewall:** A Decision Procedure for a Theory of Actions and Plan. In Z. Ras et al, editor, *Methodologies for Intelligent Systems,* IV, 1989.
197. **Tomas Sokolnicki:** Integrating Tutoring in Knowledge-Based Systems. In *Proc. of the SCAI 89, Workshop on AI Solutions in Education,*Tampere, June, 1989.
198. **Ola Strömfors:** Editing Large Programs Using a Structure-Oriented Text Editor. In *Proc. of the Int. Workshop on Advanced Programming Environments.* Trondheim, Norway, June, 1986.
199. **Ola Strömfors:** A Structure Editor as a Template for Programming Environment Functions. In *Proc. of the Workshop on Programming Environments - Programming Paradigms,* at Roskilde University Centre, Denmark, October 22-24, 1986.
200. **Erik Tengvald:** Advances in Intelligent Robotics Systems. In *Proc. of the SPIE Cambridge Symposium on Optical and Optoelectronic Engineering,* Cambridge, USA, November, 1988.
201. **Erik Tengvald, Erland 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.
202. **Toomas Timpka, Per Bjurulf:** The Semantics of Diagnosis and Management of Genitourinary Infections: A cross-specialty study. To appear in *Family Practice* Vol 6:1989.
203. **Toomas Timpka, Elisabeth 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.
204. **Toomas Timpka, Marie Ekström, Per Bjurulf:** Information Needs and Information Seeking Behaviour in Primary Health Care. To appear in Scandinavian Journal of Primary Health Care, 1989.
205. **Toomas Timpka:** Introducing Hypertext in Primary Health Care: A Study on the Teasibility of Decision Support for Practitioner. In *Computer Methods and Programs in Biomedicine,* 1989;29:1-13.
206. **Toomas Timpka, Roland Hjerppe, J. Zimmer, M. Ekström:** Introducing Hypertext in Primary Care: Supporting the Doctor-Patient Relationship. In *Proc. of RIAO-88,* Cambridge, 1988.
207. **Toomas Timpka, Lin 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.
208. **Toomas Timpka, Per Bjurulf, T. Buur:** Audit of Decision-Making Regarding Female Genitourinary Infections in Out-Patient Practice. In *Scandinavian Journal of Intectious Disease.* In press, 1989.
209. **Toomas Timpka, Elisabeth Arborelius:** The GP's Dilemmas: A Study of Knowledge Need and Use During Health Care Consultations. In *Proc. of the Methods of Information in Medicine,* January, 1990.
210. **Yvonne Waern, Sture Hägglund, Ivan Rankin:** Expert Critiquing Systems - psychological and computational aspects of generating comments. In *Preprints of ECCE Conf.* Cambridge, England, September, 1988.
211. **Arja Vainio-Larsson:** Metaphors as Communicators of Conceptual Ideas. In *the 9th Scandinavian Research Seminar on Use and Development of Information Systems,* Båstad, August 19-22, 1986.
212. **Mats Wirén:** A Comparison of Different Rule-Invocation Strategies in Context-Free Chart Parsing. In *Proc. of the 3rd Conf. of the European Chapter of the Association for Computational Linguistics,* in Copenhagen, Denmark, April 1-3, 1987.
213. **Mats Wirén:** A Control-Strategy-Independent Implementation of PATR. *Proc. of the 1st Scandinavian Conf. on Artificial Intelligence,* Tromsö, Norway, March 9-11, 1988.
214. **Mats Wirén:** Interactive Incremental Chart Parsing. Accepted to the *4th Conf. of the European Chapter of the Association for Computational Linguistics,* Manchester, April 10-12, 1989.

DEPARTMENTAL REPORTS 1988-89

LiTH-IDA-R-88-01	**Jacek Leszczylowski, Staffan Bonnier, Jan Maluszynski:** Logic Programming with External Procedures: Introducing S-Unification.
LiTH-IDA-R-88-02	**Ulf Nilsson:** Inferring Restricted AND-Parallelism in Logic Programs using Abstract Interpretation.
LiTH-IDA-R-88-03	**Staffan Bonnier, Jan Maluszynski:** Towards a Clean Amalgamation of Logic Programs with External Procedures.
LiTH-IDA-R-88-04	**Wlodzimierz Drabent, Simin Nadjm-Tehrani, Jan Maluszynski:** Algorithmic Debugging with Assertions.
LiTH-IDA-R-88-05	**Christer Bäckström:** Keeping and Forcing: How to Represent Cooperating Actions.
LiTH-IDA-R-88-06	**Christer Bäckström:** A Representation of Coordinated Actions Characterized by Interval Valued Conditions.
LiTH-IDA-R-88-07	**Erik Sandewall et al:** IDA Annual Report 1987.
LiTH-IDA-R-88-08	**Erik Sandewall:** Formal Semantics for Reasoning about Change with Ramified Causal Minimization.
LiTH-IDA-R-88-09	**Peter Fritzson:** Incremental Symbol processing.
LiTH-IDA-R-88-10	**Mats Wirén:** A Control-Strategy-Independent Parser for PATR.
LiTH-IDA-R-88-11	**Mats Wirén:** An Incremental Chart Parser for PATR.
LiTH-IDA-R-88-12	**Erik Sandewall, Johan Hultman, Erik Tengvald:** Software Architecture and Programming Paradigms for Robotics Applications.
LiTH-IDA-R-88-13	**Erik Tengvald:** Ett kartorienterat planeringssystem för autonoma farkoster, en design diskussion.
LiTH-IDA-R-88-14	**Jonas Löwgren:** History, state and Future of User Interface Management Systems.
LiTH-IDA-R-88-15	**Lin Padgham, Ralph Rönnquist:** LINCKS: An Imperative Object Oriented System.
LiTH-IDA-R-88-16	**Lin Padgham:** A Model and Representation for Type Information and Its Use in Reasoning with Defaults.
LiTH-IDA-R-88-17	**Bengt Lennartsson:** Multi-Level Software Architectures - Industrial Experiences and Research Directions.
LiTH-IDA-R-88-18	**Nils Dahlbäck:** Mental Models and Text Understanding - a Commented Review.
LiTH-IDA-R-88-19	**Lin Padgham, Ralph Rönnquist:** From a Technical to a Humane Environment: A Software System Supporting Cooperative Work.
LiTH-IDA-R-88-20	**Arja Vainio-Larsson, Rebecca Orring, Peter Åberg:** Metoder för utvärdering av människa-maskin gränssnitt: En litteraturstudie.
LiTH-IDA-R-88-21	**Johan Fagerström, Lars Strömberg:** A Paradigm for Distributed System Design and Test.
LiTH-IDA-R-88-22	**Johan Fagerström:** Design and Test of Distributed Applications.
LiTH-IDA-R-88-23	**Johan Fagerström, Yngve Larsson:** Two Contributions on Debugging Distributed Systems.
LiTH-IDA-R-88-24	**Johan Hultman, Anders Nyberg:** Realizing Action Plans and Response Rules in a System Tool for an Autonomous Vehicle.
LiTH-IDA-R-88-25	**Jonas Löwgren:** Metaphor and Information Systems: From Syracuse to Computer Use. (DISCUSSION DEBATE).
LiTH-IDA-R-88-26	**Lin Padgham:** NODE: A Database for Use by Intelligent Systems.
LiTH-IDA-R-88-27	**Erik Sandewall:** Non-Monotonic Entailment for Reasoning about Time and Action. Part I: Sequential Actions.
LiTH-IDA-R-88-28	**Erik Sandewall:** Non-Monotonic Entailment for Reasoning about Time and Action. Part II: Concurrent Actions.
LiTH-IDA-R-88-29	**Erik Sandewall:** Non-Monotonic Entailment for Reasoning about Time and Action. Part III: Decision Procedure.
LiTH-IDA-R-88-30	**Erik Sandewall:** Future Developments in Artificial Intelligence.
LiTH-IDA-R-88-31	**Erik Sandewall:** The Semantics of Non-Monotonic Entailment Defined Using Partial Interpretations.
LiTH-IDA-R-88-32	**Erik Sandewall:** An Approach to Non-Monotonic Entailment.

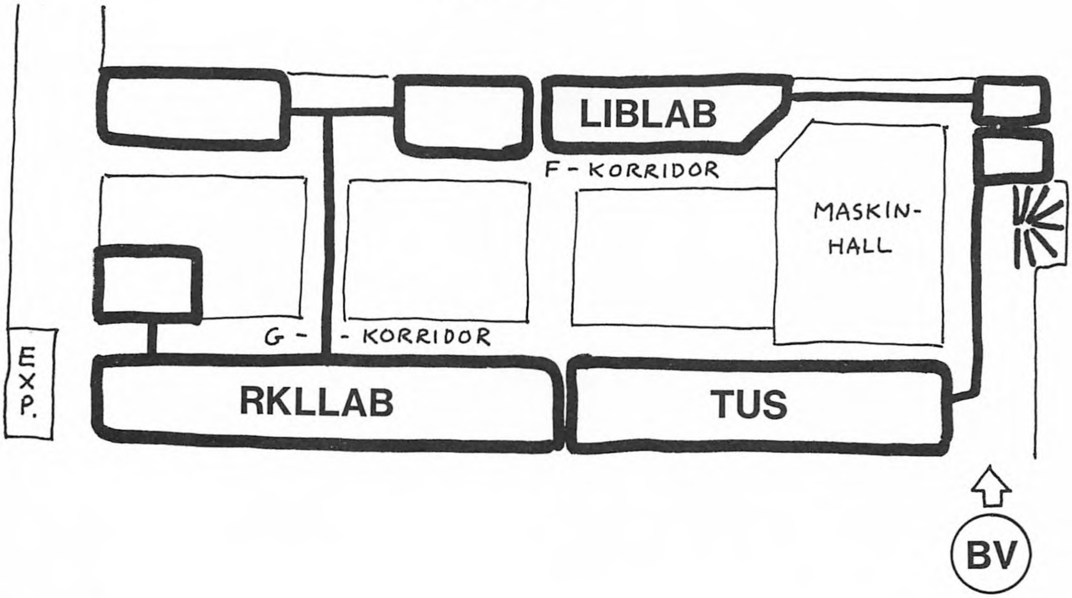
LiTH-IDA-R-88-34	**Arne Jönsson, Nils Dahlbäck:** Talking to a computer is not like talking to your best friend.
LiTH-IDA-R-88-35	**Zebo Peng:** A Horizontal Optimization Algorithm for Data Path/Control Synthesis.
LiTH-IDA-R-88-36	**Zebo Peng:** Semantics of a Parallel Computation Model and its Applications in Digital Hardware Design.
LiTH-IDA-R-88-37	**Tony Larsson, Zebo Peng:** A Relational Approach to VLSI Design Databases.
LiTH-IDA-R-88-38	**Zebo Peng:** Let's Design Asynchronous VLSI Systems.
LiTH-IDA-R-88-39	**Zebo Peng, Krzysztof Kuchcinski, Bryan Lyles:** CAMAD: A Unified Data Path/Control Synthesis Environment.
LiTH-IDA-R-88-40	**Mikael R.K Patel:** The State of the Art of High Level Synthesis.
LiTH-IDA-R-88-41	**Tony Larsson:** Hardware Verification based on Algebraic Manipulation and Partial Evaluation.
LiTH-IDA-R-88-42	**Mikael R.K Patel:** What should a Programming Environment for Threaded Interpretive Languages Provide?
LiTH-IDA-R-88-43	**Göran Rydqvist, Mikael R.K Patel, Leif Larsson:** Types in Space Towards Flexible High-speed Object Oriented Data Management Tools of the Future.
LiTH-IDA-R-88-44	**Ulf Nilsson:** Towards a Framework for the Abstract Interpretation of Logic Programs.
LiTH-IDA-R-88-45	**Ulf Nilsson:** Systematic Construction of Domains for Abstract Interpretation Frameworks.
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.
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.
LiTH-IDA-R-89-06	**Andrzej Lingas:** Greedy Triangulation can be Efficiently Implemented in the Average Case.
LiTH-IDA-R-89-07	**Andrzej Lingas:** Subgraph Isomorphism for Connected Graphs of Bounded Valence and Bounded Separator is in NC.
LiTH-IDA-R-89-08	**Christos Levcopoulos, Jyrki Katajainen, Andrzej Lingas:** An Optimal Expected-Time Parallel Algorithm for Voronoi Diagrams.
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.
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.
LiTH-IDA-R-89-13	**Oystein Santi:** Retargeting of an Incremental Code Generator to MC68020.
LiTH-IDA-R-89-14	**Olof Johansson:** A Perspective on Engineering Database Research.
LiTH-IDA-R-89-15	**Arja Vainio-Larsson:** Hypertext/hypermedia från ide till praktisk tillämpning.
LiTH-IDA-R-89-16	**Christos Levcopoulos, Ola Petersson:** An Optimal Parallel Algorithm for Sorting Presorted Files.
LiTH-IDA-R-89-17	**Christos Levcopoulos, Ola Petersson:** A Note on Adaptiave Parallel Sorting.
LiTH-IDA-R-89-18	**Rober Bilos:** A Survey of Visibility Control Models.

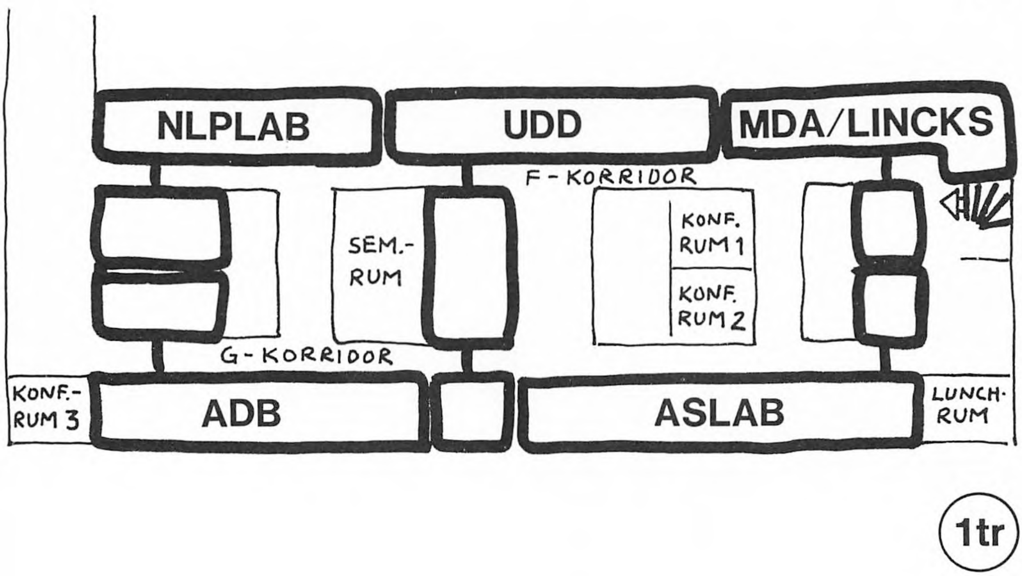
LiTH-IDA-R-89-19	**Göran Rydqvist:** Representing the Natural World - Some Implications of Prototype Theory in the Context of ARtificial Intelligence.
LiTH-IDA-R-89-20	**Jyrki Katajainen, Christos Levcopoulos, Ola Petersson:** Local Insertion Sort Revisited.
LiTH-IDA-R-89-21	**Christos Levcopoulos, Ola Petersson:** Heapsort - Adapted for Presorted Files.
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.
LiTH-IDA-R-89-24	**Mats Wirén:** Interactive Incremental Chart Parsing.
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 Objet-Oriented Database Approach to the Symbol Processing in an Inremental 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.

LIBLAB RESEARCH REPORTS

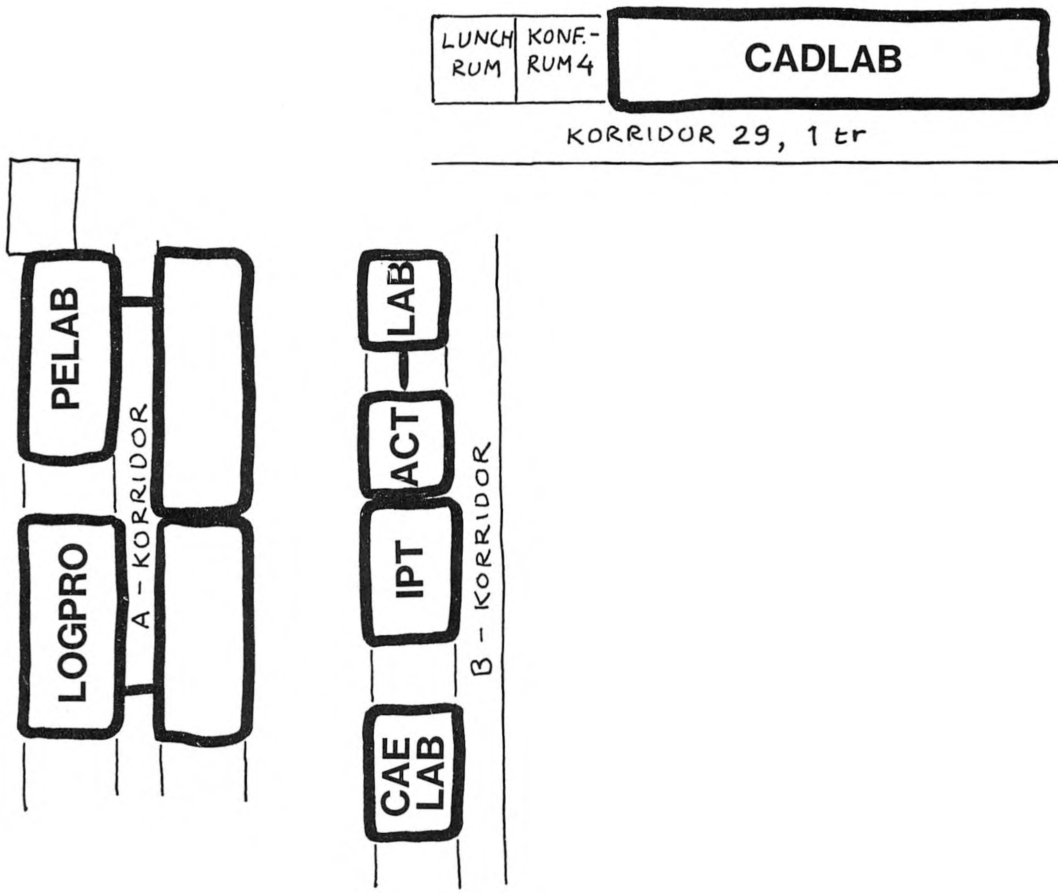
LiU-LIBLAB-R-88:1	**Arja Vainio-Larsson:** Hypermedia and Human-Computer Interaction.
LiU-LIBLAB-R-88:2	**Lisbeth Björklund, Roland Hjerppe, Manny Jägerfeld, Birgitta Olander, Arja Vainio-Larsson:** Forskningsprogram för LIBLAB. Also in english version.
LiU-LIBLAB-R-89:1	**Arja Vainio-Larsson:** Hur enkla uppgifter blir till komplexa problem: Om människa-dator interaktion.
LiU-LIBLAB-R-89:2	**Arja Vainio-Larsson:** Hypermedia, a vision for Human-Computer Interaction.

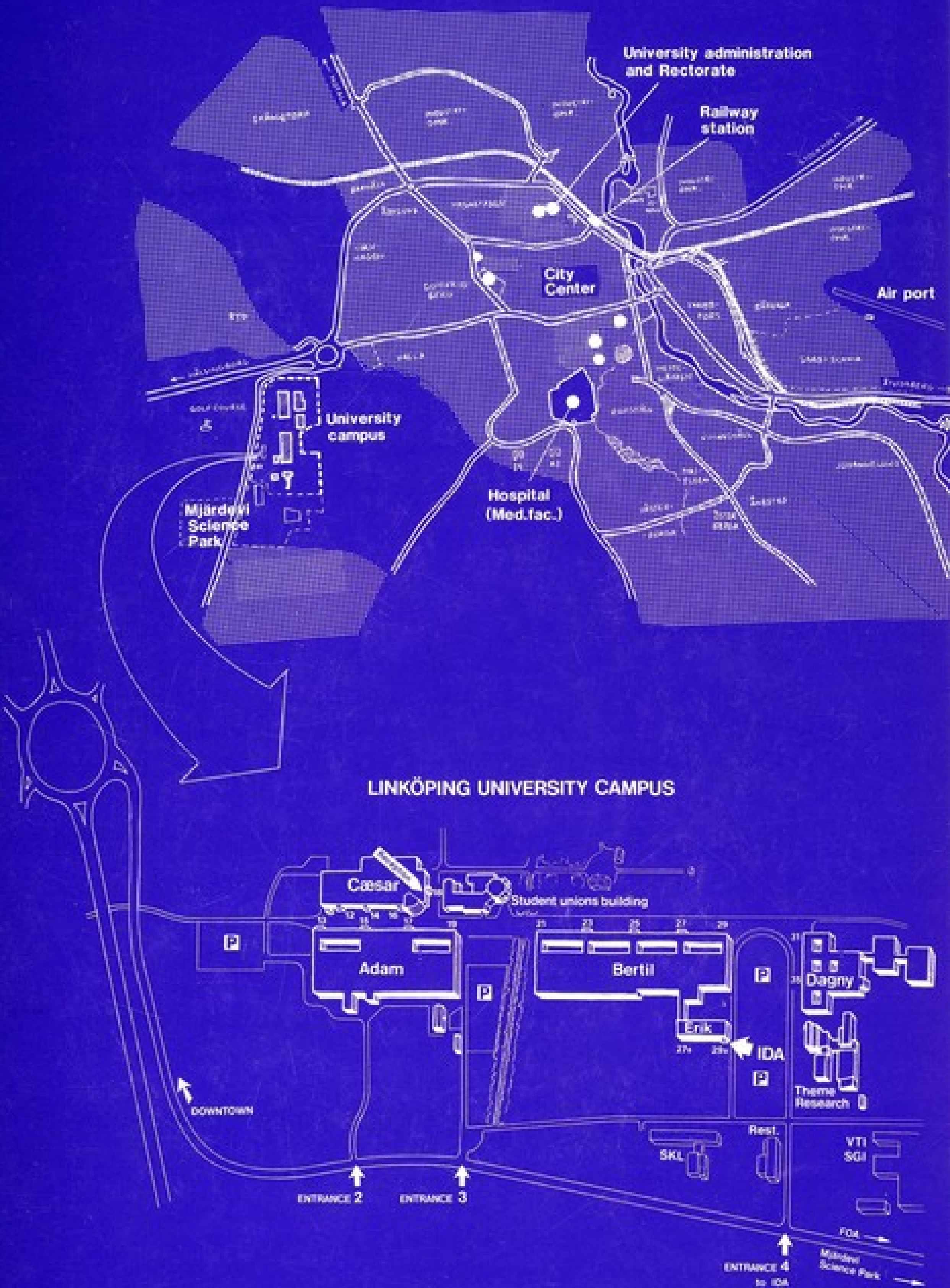
Hus E


LIBLAB
F - KORRIDOR
MASKIN-
HALL
G - - KORRIDOR
EXP.
RKLLAB
TUS
BV


NLPLAB
UDD
MDA/LINCKS
F - KORRIDOR
SEM.-
RUM
KONF.
RUM 1
KONF.
RUM 2
G - KORRIDOR
KONF.-
RUM 3
ADB
ASLAB
LUNCH-
RUM
1tr

Hus B


LUNCH RUM
KONF.- RUM 4
CADLAB
KORRIDOR 29, 1 tr
PELAB
LOGPRO
A - KORRIDOR
LAB
ACT
IPT
B - KORRIDOR
CAE LAB



University administration and Rectorate
Railway station
City Center
Air port
University campus
Mjärdevi Science Park
Hospital (Med.fac.)
LINKÖPING UNIVERSITY CAMPUS
Cæsar
Student unions building
Adam
Bertil
Dagny
Erik
IDA
Theme Research
Rest.
SKL
VTI SGI
DOWNTOWN
ENTRANCE 2
ENTRANCE 3
ENTRANCE 4 to IDA
FOA
Mjärdevi Science Park